Adama Science & Technology University

SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTING

DEPARTMENT OF COMPUTING

A MASTER'S THESIS

ON

CLOUD COMPUTING FRAMEWORK OF RESOURCE SHARING FOR HEALTHCARE INSTITUTION IN ETHIOPIA

BY:
SIDRAK HABTEMARIAM TEREDDA

JANUARY, 2018

Adama Science & Technology University

SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTING

DEPARTMENT OF COMPUTING

A MASTER'S THESIS

ON

CLOUD COMPUTING FRAMEWORK OF RESOURCE SHARING FOR HEALTHCARE INSTITUTION IN ETHIOPIA

THIS THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE DEGREE

OF

MASTER OF SCIENCE IN SOFTWARE ENGINEERING

BY:

SIDRAK HABTEMARIAM TEREDDA

JANUARY, 2018

DECLARATION

This thesis contains information obtained from authentic and highly regarded sources. To present reliable data and information reasonable efforts have been made, but for the validity of all materials or the consequences of their use the researchers cannot assume responsibility. The researchers have attempted to cite all material reproduced in this thesis and apologize to copyright holders if permission has not been obtained. If any copyright material has not been acknowledged, please let us know so we may rectify in any future works.

Sidrak Habtemariam Teredda ______________________

Signature

Dr. Ravindra Babu (PhD) ______________________

Signature

Adama Science & Technology University

SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTING

DEPARTMENT OF COMPUTING

BY:
SIDRAK HABTEMARIAM TEREDDA

Name and Signature of Members of the Examination Board

Dr. Ravindra Babu (PhD) Advisor	______________________ Signature
______________________ Internal Examiner's Name	______________________ Signature
______________________ External Examiner's Name	______________________ Signature
______________________ Chairperson's Name	______________________ Signature

Acknowledgement

All that we know is a sum total of what we have learned from all who have taught us, both formally and informally. I am forever indebted to the countless outstanding men and women who, by their commitment and dedication to becoming the best they could be, have inspired me to do the same.

First and above of all, I would like to thank the almighty God for giving me this wonderful opportunity and helping me in every way of my success. And by the help of several people this thesis has reached to this position.

I would like to thank my advisor Dr. Ravindra Babu for his continuous and constructive reviews of this work. I also would like to thank Dr. Anandraj for his advice and all dedicated help to the completion of this thesis.

Finally, I would like to thank my family and friends who support me in different ways. Thank you, all my classmates, for your knowledge sharing experiences. I would like to thank everyone at the ASTU, School of Electrical Engineering and Computing, Department of Computing, who has encouraged and assisted me throughout the work of this thesis.

Abstract

Information technology in the delivery of healthcare is to enhance the availability and reliability of improved healthcare services. Cloud computing has been identified generally as the next big deal in computing infrastructure and it offers some benefits by allowing to use infrastructures like networks, storages, and servers, platforms like operating systems and middleware services and software such as application programs. Healthcare professionals do not always have all the information they require and at the time when it is most needed to make rapid patient-care decisions. Also, healthcare services are still being provided using manual tools and technologies. The adoption of current technologies such as cloud computing provides an avenue to overcome these challenges. Resource sharing is a mode of operation which consists of various collaborative activities between resource centers and other concerned bodies. So currently, there are many promising improvements such as collaborative methods to electronic repositories, electronic interactions, establishing content consortiums and other practical experiences for resource sharing purpose. The research finding shows that cloud computing is the better information communication technology utilization mechanism for healthcare centers and service delivery requirements, for it enables wise and strategic use of technology which significantly reduces cost.

A rigorous analysis of the latest research on Cloud Computing as an alternative to IT provision, management and resource sharing is done. It also took into account the best practices for Cloud Computing usage within different hospitals by interviewing with selected practitioners, physicians and healthcare IT professionals. This study proposes community cloud platform framework for sharing resources in the healthcare institutions. The framework helps the health workers about the health system technologies in depth. Also, analysis and comparison of some of cloud computing freely available platforms and tools were done by putting different features as comparison criteria. The prototype was implemented using OpenShift online Redhat's platform. The prototype is very interactive and user friendly which have resource sharing and accessing portal. Hence, the system is used as a common alliance and electronic means of resource sharing, knowledge sharing, collaboration and learning center in the delivery of healthcare institutions.

Key Words: ICT, Healthcare Information System, Cloud Computing, Electronic Health, Resource Sharing

Contents

Acknowledgement i

Abstract ii

List of Figures vi

List of Tables viii

List of Acronyms ix

CHAPTER ONE 1

INTRODUCTION 1

1.1 Background of the Study 1
1.2 Statement of the Problem 3
1.3 Purpose of the Study 4
1.4 Objective of the Research 4
1.4.1 General objective 4
1.4.2 Specific objective 5
1.5 Research Questions 5
1.6 Overview of the Methodology 5
1.7 Significance of the study 6
1.8 Scope and Limitation of the Study 7
1.9 Organization of the Thesis 8

CHAPTER TWO 10

LITEERATURE REVIEW AND CONCUPTURAL DISCUSSION 10

2.1 Cloud Computing 10
2.1.1 Cloud Computing Service Delivery and Deployment Models 11
2.1.2 Cloud Computing Enablers 14
2.1.3 Cloud Application Architecture and Design Considerations 16
2.1.4 Cloud Computing Related Terms and Technologies 17
2.2 Cloud Computing and Healthcare 18
2.2.1 Healthcare IT 18
2.2.2 The current role of technology in healthcare 19
2.2.3 Challenges in the Present healthcare 19

2.2.4 Cloud computing in the health sector ... 21
2.2.5 Benefits of Health Care Service in the Cloud ... 22
2.2.6 Risks of health care services in the cloud ... 24
2.3 Related Works ... 25
CHAPTER THREE ... 28
METHODOLOY ... 28
3.1 Research Design ... 28
3.2 Process Model ... 29
3.3 Data Source ... 30
3.4 Data Collection Techniques ... 31
3.5 Issues of Reliability and Validity ... 32
3.6 Sampling Techniques ... 32
CHAPTER FOUR ... 33
CLOUD SERVICE AND PROVIDER SELECTION ... 33
4.1 Introduction ... 33
4.2 Cloud Service Providers ... 34
4.2.1 EMC ... 34
4.2.2 Google ... 35
4.2.3 Amazon Web Services ... 38
4.2.4 Microsoft ... 39
4.2.5 IBM ... 40
4.2.6 VMware ... 41
4.2.7 Manjrasoft ... 43
4.3 Open Source Support for Cloud ... 46
4.3.1 Open Source Tools for IaaS ... 48
4.3.2 Open Source Tools for PaaS ... 53
4.3.3 Open Source Tools for SaaS ... 55
4.4 OpenShift Online ... 60
4.4.1 Overview ... 60
4.4.2 Gears and Cartridges ... 61
4.4.3 Developer Interaction ... 63
CHAPTER FIVE ... 65

CLOUD BASED SERVIECES FOR HEALTHCARE INSTITUTIONS IN ETHIOPIA ... 65
5.1 Building the Framework ... 65
5.2 Design Goals and Constraints ... 66
5.3 General Cloud Framework of Resource sharing for Healthcare Institutions ... 67
5.4 The Developed Architectural Solution for Healthcare Institution ... 68
5.4.1 User Interface Layer ... 70
5.4.2 Software as a Service (SaaS) Layer ... 70
5.4.3 Platform as a Service (PaaS) Layer ... 70
5.4.4 Infrastructure as a Service (IaaS) Layer ... 71
5.5.5 Security and Management Mechanisms ... 71
CHAPTER SIX ... 73
DEVELOPMENT OF THE PROTOTYPE ... 73
6.1 The Architecture of the System ... 73
6.2 Use Cases ... 75
6.3 Logical View ... 81
6.4 Process View ... 83
6.5 Data Model ... 85
6.6 Implementation Overview ... 86
6.7 System Evaluation ... 88
6.7.1 Unit Testing ... 88
6.7.2 Integration Testing ... 89
6.7.3 Usability Testing ... 90
6.8 Finding and Discussion ... 91
CHAPTER SEVEN ... 93
CONCLUSION AND FUTURE WORK ... 93
7.1 Conclusion ... 93
7.2 Future Work ... 94
References ... 95
Appendix ... 102

List of Figures

Figure 2.1: NIST definition of cloud computing ……………….…………………………..…. 12

Figure 3.1: Research Methodology ………………………………………………………. 28

Figure 4.1: Cloud services by EMC ……………………………………………….......... 35

Figure 4.2: Google App Engine …………………………………………………..…. 37

Figure 4.3: AWS ……………………………………………………………………..,,. 38

Figure 4.4: Overview of the Aneka platform ……………………………………….…. 44

Figure 4.5: Infrastructure virtualization in OpenNebula ……………………….……..... 49

Figure 4.6: Eucalyptus Cloud architecture ……………………………….……….…..... 50

Figure 4.7: OpenStach conceptual architecture ……………………………………...,,. 52

Figure 4.8: The three OpenShift offerings ……………………………………………..... 54

Figure 4.9: Architecture of VCL …………………………………………….……….…..... 56

Figure 4.10: Apache VCL conceptual overview ………………………………….……… 57

Figure 5.1: General cloud framework of resource sharing for healthcare institutions ………..... 68

Figure 5.2: Conceptual cloud computing framework of resource sharing for healthcare institutions …………………………………………………………………………….... 69

Figure 5.3: Overview of development view of the E-Health system ………………………...… 71

Figure 6.1: Kruchten '4+1' view model ………………………………………………......... 74

Figure 6.2: Use Cases …………………………………………………………………... 75

Figure 6.3: Sequence diagram of resource providers uploading resources …………………...... 82

Figure 6.4: Sequence diagram-users resource access …………………………………….. 83

Figure 6.5: Activity diagram for registration and login ……………………………………... 84

Figure 6.6: Activity diagram to access the overall system services …………………….…. 85

Figure 6.7: Data Model ………………………………………………………………………… 86

Figure 6.8: The developed healthcare application on the cloud ………………………………... 87

Figure 6.9: cloud portal desktop interface ………………………………………………………… 88

List of Tables

Table 4.1: Tools and Services Offered by Companies ……………………………………….… 44

Table 4.2: Details of Cloud Service Providers ………………………………………………...... 45

Table 4.3: The features and benefits of OpenShift Online ……………………………………… 60

Table 4.5: OpenShift Online system component ……………………………………………….. 62

Table 4.6: OpenShift Online application component …………………………………...……. 62

Table 6.1: Sample unit test case ………………………………………………………………… 89

Table 6.2: Sample Integration Test case ………………………………………………………… 90

Table 6.3: Usability test case ……………………………………………………………………... 90

List of Acronyms

API: Application Programming Interface

ASP: Application Service Provision

AWS: Amazon Web Service

BPaaS: Business Process as a Service

CLI: Command Line Interface

CSP: Cloud Service Provider

EMR: Electronic Medical Record

EHR: Electronic Health Record

FAQ: Frequently Asked Questions

GUI: Graphical User Interface

HDI: Healthcare Delivery Institution

HIT: Healthcare Information Technology

HTML: Hypertext Markup Language

HTTP: Hypertext Transfer Protocol

IaaS: Infrastructure as a Service

ICT: Information and Communication Technology

IDE: Integrated Development Environment

IT: Information Technology

JSON: Java Script Object Notation

NIST: National Institute of Standards and Technology

PaaS: Platform as a Service

REST: Representational State Transfer

SaaS: Software as a Service

SDK: Software Development Kit

SQL: Structured Query Language

VM: Virtual Machine

XML: Extensible Markup Language

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

The utilization of the Internet and Information Communication Technology (ICT) to deliver electronic health resources is considered mainstream in the twenty first century, yet in developing countries somehow seen as a luxury. This has far reaching effects on healthcare centers in these countries, which often include a lack of basic ICT infrastructure and limited or no support for the resource sharing in the use of digital online information sources. It is increasingly accepted that in the future most information sources and desktop applications currently used will be mainly accessed through the Internet, now increasingly referred to as ‘the cloud’ [1]. Even though cloud computing in healthcare is of growing interest only few successful implementations yet exist and many papers just use the term “cloud” synonymously for “using virtual machines” or “web-based” with no described benefit of the cloud paradigm. This means that at healthcare institutions level ICTs should be adopted as a matter of urgency to enable resource sharing service to access this new direction in Internet technology and application delivery. Information management in healthcare institutions particularly in rural areas is a complex task [3]. High quality healthcare depends on extensive and carefully planned information processing. The current development of cloud computing in the healthcare context will have a big impact on the healthcare sector. It is evolving as a key computing platform for sharing resources that include infrastructures, software, applications, and business processes [3]. Virtualization is a core technology for enabling cloud resource sharing.

Health information will no longer have to physically carry the documents and data around; instead it will be able to access the information in the ‘cloud’ anywhere, from any connected device. This blending of traditional system with online applications and tools via the Internet is thus a fundamental concept of cloud computing. According to [2], the 21st century HIT has created the ability to electronically stores, maintains, and moves data across the world in a matter of seconds and has the potential to provide healthcare with enormous increasing productivity and quality of services. It provides a way of using ICT to share resources and to deliver quality healthcare services. Thus, the integration of ICT with healthcare will help any country to share resources and to deliver quality health care services to all citizens. It provides efficient and effective patient information handling and sharing of resources at the right time among medical professionals for clinical decision making and initiative.

The term cloud computing refers to the delivery of scalable information technology (IT) resources over the internet, as opposed to hosting and operating those resources locally, such as on a care delivery organization or healthcare center network. Those resources can include applications and services, as well as the infrastructure on which they operate. By deploying IT infrastructure and services over the network, an organization can purchase these resources on an as-needed basis and avoid the capital costs of software and hardware. Many researchers from computer science and related fields have studied its application in the healthcare system for saving life by automatically sending emergency health information to the ambulance [4], for remotely monitoring the patient; for using telemedicine technologies like EHR and others that are deployed and installed in one hospital [2]. On the other hand, web service technologies and concepts are at the heart of cloud computing to make its application more accessible, flexible and easy to use for the health industries to share resources and to deliver high quality health services equally for all citizens in the country. With cloud computing, IT capacity can be adjusted quickly and easily to accommodate changes in demand. While remotely hosted, managed services have long been a part of the IT landscape, a heightened interest in cloud computing is being fueled by ubiquitous networks, maturing standards, the rise of hardware and software virtualization, and the push to make IT costs variable and transparent.

Cloud computing delivers infrastructure, platform, and software as services, which are made available as subscription-based services in a pay-as-you-go model to customers. These services in industry are respectively referred to as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).

Cloud computing allows to use applications without installing them on their computers and also allows access to saved files from any computer with an Internet connection. Our study desired to develop cloud based resource sharing framework for the healthcare institutions to create better, effective, interactive and collaborative resource sharing center for the health workers in order they develop ability, problem solving skill, knowledge and experience sharing to deliver quality health services.

In generally, by using web service and cloud computing technologies, the whole resources have been stored in one place in the cloud and available for authorized users ubiquitously through the use of broad network access. The resources are provided and controlled by one HDI and the other healthcare institution's health workers get access to these resources from this HDI which reduces the cost, saves time, accessed in easy and smart way for the health institutions because they would not provide/prepare the resources by themselves. Hence this thesis elaborates the application of cloud computing technology to share resources in the healthcare area; develops cloud based framework of resource sharing for healthcare institutions in Ethiopia that supports sharing of resources; saves resource preparation cost and time; makes the resources more deliverable, accessible and effective, increases the use of cloud-health application.

1.2 Statement of the Problem

The use of ICT has fundamentally changed the practices and procedures of nearly all forms of endeavor within business and governance. Within healthcare center, ICT has begun to have a presence but the impact has not been as extensive as in other fields. The use of ICT in resource sharing lends itself to more simply access medical information. There have been a number of factors impeding the wholesale uptake of ICT for healthcare system. These have included such factors as a lack of funding to support the purchase of the technology, a lack of training among established medical practitioners, a lack of motivation and need to adopt ICT in healthcare centers. Although the capacity building tools and resources are available in different organizations [12], the problem is, there is no means to access and use them because some organizations restrict dissemination of their tools and other organizations have not published these tools. Hence, a knowledge exchange network that creates partnership and collaboration among medical practitioners is necessary.

Currently different industry leading companies are investing more on cloud computing. Some of the top cloud computing companies are: Amazon Web Service (AWS) provide Infrastructure-as-a-service (IaaS) offerings in the cloud for organizations requiring computing power, storage, and other services. Google (SaaS, PaaS), Google App Engine is Google's platform-as-a-service (Paas) offers building and hosting web applications on the Google Infrastructure. Currently Python and Java are the supported programming languages. Microsoft Azure Service Platform (PaaS) Azure Service Platform is Microsoft's PaaS offering an operating system called Windows Azure that serves as a runtime for the applications and provides a set of services such as: NET Services, SQL Services, and Live Services [5].

For a particular ICT service, a sufficient level of aggregation for efficiency cannot be achieved within one Healthcare center but, rather, must be achieved at a higher level of aggregation, beyond a single healthcare center. Efficiencies may be realized in aggregating personnel, expertise, licensing, business continuity, and other benefits far beyond simply joining computer hardware.

Hence the main concern of this study will be finding solutions for the above aforementioned factors so that healthcare institutions can get the advantages of cloud computing in efficient and affordable manner.

1.3 Purpose of the Study

The basic purpose of Cloud Computing Framework of Resource sharing for Healthcare Institution in Ethiopia is to create healthcare learning environment and collaborations for health Practitioners at remote and district areas and specialists in urban hospitals to improve the potential of the health workers by providing consecutive sharing of resources and experience. This will ultimately result in improvement of health practitioner's potential to deliver quality health service and help to reach rural areas. The design also used to create opportunities for international agencies to share free resources hence they can be able to join the system to support developing countries and also it is used to interface with the substantial number of Ethiopian and non-Ethiopian health professionals living outside the country to share their experience and knowledge.

1.4 Objective of the Research

1.4.1 General objective

The general objective of this study was to develop cloud computing framework of resource sharing architecture that enables healthcare workers and other knowledge users to access and share resources from one place to develop knowledge and ability in their profession.

1.4.2 Specific objective

- Discuss and describe the overall concept of the term “cloud computing” and examine the current ICT service delivery strategy and efficiency.
- Explore the potential benefits of Cloud Computing to healthcare center.
- Examine the necessity of Cloud Computing for healthcare center in Ethiopia.
- To analyze and compare the cost effectiveness of using cloud computing technology in Ethiopian healthcare center.
- Designing a prototype cloud computing framework model that can be placed in a one place and can provide cloud service for higher healthcare institutions in Ethiopia.
- To identify and recommend future research directions for further investigation on the benefits of Cloud Computing for Ethiopian healthcare center.

1.5 Research Questions

The research is intended to get answers for the following research questions.

- Do we need to change the way we are currently deploying and using ICT in Ethiopian higher healthcare institutions?
- Could cloud computing be an answer for minimizing the previously mentioned problems?
- Is it possible to design effective framework of cloud based resource sharing for healthcare institutions in Ethiopia?
- What areas of cloud based healthcare system require further research?

1.6 Overview of the Methodology

The overall works of this thesis, different methods and techniques was employed. Data collection and analysis techniques, programming and tools techniques, evaluation and testing mechanisms, comprehensive document analysis as well as prototyping as a methodology were used. Primary data from government and private healthcare delivery institutions were collected directly and secondary data through documentary analysis were collected. For the purpose of collecting data, interview technique was used using questionnaire. The questionnaires were distributed to hospitals in Addis Ababa, Adama and Debre Zeyit. Because of budget constraints, we could not distribute the questionnaire outside Addis Ababa, Adama and Debre Zeyit.

In advance to the above methods, different tools were used to accomplish this thesis. Some of the tools are window 10 operating system Microsoft office 2016 for preparing the document; Adobe Photoshop to create images and UML tools for designing the different views of the framework. As noted above, UML tools were used due to the following reasons: using standardized notation without sacrificing specialized model data, increased domain and design model reuse, increased customer involvement/understanding of problems translation to product solution. The framework is built using OpenShift Online.

1.7 Significance of the study

The study will allow the Ethiopian healthcare institutions to consider alternative ways of deploying ICT infrastructures for an efficient and effective resource sharing service delivery. The study also will give new insights to the Ethiopian Federal Ministry of Health and concerned government sectors on how to invest ICT budgets on healthcare institutions. Additionally, the study can be used as a baseline for further studies of this newly emerged ICT utilization strategy to be considered in different governmental and non-governmental organizations of Ethiopia. Other beneficiaries include: Healthcare institutions in general, researchers, capacity building providers, fundraising organizations, ICT organizations etc. The main beneficiaries and specific benefits of the result is listed below

Benefits for Healthcare institutions:

- The healthcare institutions will receive different resources provided by one general hospital as a service from cloud service providers.
- It will used for establishing multi-sector partnership by contacting other organization through the system without going to the organization directly whether the organization is private or governmental and local or foreign.
- It will used as initiatives to develop and implement IT functions and to provide supportive supervision to the staff throughout the organizations.

- It makes the resource sharing cost effective, interactive and deliverable for the healthcare institutions and it leads to sustainable ability to solve problems.

Benefits for Health workers:

- Enables mobilization of training resources. The health practitioners in the healthcare institutions will access the training materials anytime anywhere using their electronic devices.
- Used to create healthcare learning environment in which the health practitioners in the world shares their experience and knowledge for others in the health area and for the community.
- It provides multiple mode of training with the help of current technology.

Benefits for Fundraising organizations:

- For the funding organization and voluntaries, it will used to show the health outcomes of this study they fund by directly visiting the system output or report measured by different indicators.

Benefits for Researchers:

- Researchers from medical science and software engineering fields can use the result of this thesis as an input to their study. It will also help them to design a better architecture based on new technologies together with other medical technologies.

Benefits for ICT Organization and for the public:

- ICT organizations will get market opportunity by providing communication services to those institutions that would implement cloud based resource sharing system.
- The society would get access to proper medical care and quality health service if the design could be implemented successfully.

1.8 Scope and Limitation of the Study

The intention of the study is to examine the current ICT service delivery strategies to support healthcare institutions in Ethiopia, to consider cloud computing in the field of healthcare service specifically in Ethiopia and propose a cloud computing framework of resource sharing that could be used as a baseline for implementation of Cloud Computing in Ethiopian healthcare institutions.

The full implementation of the framework requires more time, funds or money and additional human resource in order to be workable in the real world. And the framework would not intend to support different language. So, this study would focus on the development of the framework that will support only English language. This is because of it needs translators if the online articles and comments are written by different language which is big project by its own.

1.9 Organization of the Thesis

This thesis discusses in detail Cloud Computing Framework of Resource sharing for Healthcare Institution in Ethiopia. It can also be used as a handbook for cloud. It contains 6 chapters that follow a standard format for easy and useful guide: Introduction, and details related to the chapter title with several subsections, supported by a suitable number of diagrams, tables, figures, etc.

To start with, Chapter 1 aims to give a brief introduction about Cloud Computing Framework of Resource sharing for Healthcare Institution in Ethiopia available paradigms of computing. This provides the required basic knowledge about the thesis to start with cloud technology.

Chapter 2 gives review of literatures on cloud computing and cloud based resource sharing; and the details of various related works that are related to our work.

Chapter 3 presents the methodology of the work in detail. What design is better for Ethiopian healthcare institutions, why the specific design is selected and basically the methodologies and sample cloud services that can be provided with the proposed cloud computing framework is also discussed

Chapter 4 discusses selection of the platforms and tools. Also discusses concepts related with cloud computing design consideration is covered.

Chapter 5 contains information about the proposed Cloud Computing Framework of Resource sharing for Healthcare Institution in Ethiopia. It contains the proposed cloud computing framework architecture in detail. It also contains the implementation proposal for the proposed Cloud Computing framework.

Chapter 6 discusses we provide the general overview of the prototype implementation of the proposed architecture that is the front-end web application interface development on OpenShift online cloud platform.

The final chapter, Chapter 7, discusses conclusions about the research and recommendations for future research direction.

CHAPTER TWO

LITEERATURE REVIEW AND CONCUPTURAL DISCUSSION

2.1 Cloud Computing

Cloud computing has evolved through a number of phases which include grid and utility computing, application service provision (ASP) and software as a service (SaaS). But the overarching concept of delivering computing resources is rooted in the sixties and has developed along a number of lines, with Web 2.0 being the most recent evolution. So, cloud computing is long term dream of grid computing (It is a network of computing or processor machines managed with a kind of software such as middleware, in order to access and use the resources remotely.). It has started around 1970s when the concept of Virtual Machine (VM) created, but it becomes well known with the arrival of Salesforce.com in 1999, which pioneered the concepts of delivering enterprise applications via a simple website. The next development was Amazon Web Services in 2002, which provided a suite of cloud-based services. Then in 2006, Amazon launched its Elastic Compute cloud (EC2) as a commercial web service that allows small companies and individuals to rent computers on which to run their own computer applications. Another big milestone came in 2009, as Web 2.0 hit its stride, and Google and others started to offer browser-based enterprise applications, though services such as Google Apps [20].

Over all, cloud computing is just another step in the history of communication, it is all about as a service. Cloud computing is a buzzword as written by some authors [21, 22]. But some experts and scholars refuse the idea of buzzword because services are not new brand, there have been network services since the invention of the internet [23, 24]. Also cloud computing is defined as anytime, anywhere, with any device, accessing any service from the perspective of customers and different professionals defines cloud computing as IT as a Service (ITaaS). Generally, in different scholars, books and cloud experts' manuals, the concept of cloud computing has been discussed and defined in a number of ways.

Perhaps the most generally accepted definition has been put forth by the US Government's National Institute of Standards and Technology (NIST). Accordingly, "Cloud computing is a model for enabling network users‟ on-demand access to a shared pool of configurable computing resources that can be rapidly provisioned and released to the client without direct service provider interaction. This model is composed of five essential characteristics, three service models and four deployment models." [76] The five essential characteristics that cloud computing system must offer are:

i. On-demand self-service: A client can provision computer resources without the need for interaction with cloud service provider personnel.
ii. Broad network access: Access to resources in the cloud is available over the network using standard methods in a manner that provides platform-independent access to clients of all types. This includes a mixture of heterogeneous operating systems, and thick and thin platforms such as laptops, mobile phones, and PDA.
iii. Resource pooling: A cloud service provider creates resources that are pooled together in a system that supports multi-tenant usage. Physical and virtual systems are dynamically allocated or reallocated as needed.
iv. Rapid elasticity: Resources can be rapidly and elastically provisioned. From the standpoint of the client, cloud computing resources should look limitless and can be purchased at any time and in any quantity.
v. Measured service: The use of cloud system resources is measured, audited, and reported to the customer based on a metered system.

2.1.1 Cloud Computing Service Delivery and Deployment Models

In addition to the five key characteristics, there are seven commonly known models in cloud computing based on its service that are provided to the user and the way of deploying such services (figure 2.1). Three things [25] that provide consumer capabilities are often mentioned in the context of cloud computing. That means Cloud Service Providers (CSPs) offer their services according to three fundamental models. They are Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). Each service is built on top of the other.

SaaS is frequently used model in which an application runs directly on the cloud and the user accesses it via the Internet. It refers to on-demand software which is built on top of cloud's PaaS. It is a web based cloud service model that makes the software available entirely through a thin client interface like web-browser. As a result, the users of SaaS software don't care about where the software is hosted, which operating system it uses and by what programing language it developed. In this model, the user does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

Even if it has restrictions on the client's application development freedom, the SaaS model is most popular and holds the majority of the market share compared to other two models, due to ease of adaptability, good return on investment and reduced time to market.

With **PaaS**, services are built on top of the cloud's Infrastructure-as-a-Service (IaaS) and consumers can create and deploy their own applications onto the cloud infrastructure. So that, it provides virtual environments for developing, deploying, monitoring, and managing applications online without the cost of buying and managing the corresponding software tools or hardware. The users have the freedom to develop their own applications using the language of their choice without reliance on a web-based console. The caveat with this model is the interoperability of the application between different development environments.

IaaS provides the underlying infrastructure which enables consumers to access processing, storage, networks and other fundamental computing resources in order to deploy and run arbitrary software, including operating systems and applications. But IaaS doesn't provide applications to the consumers; it simply offers hardware to them in order to put whatever they want on it.

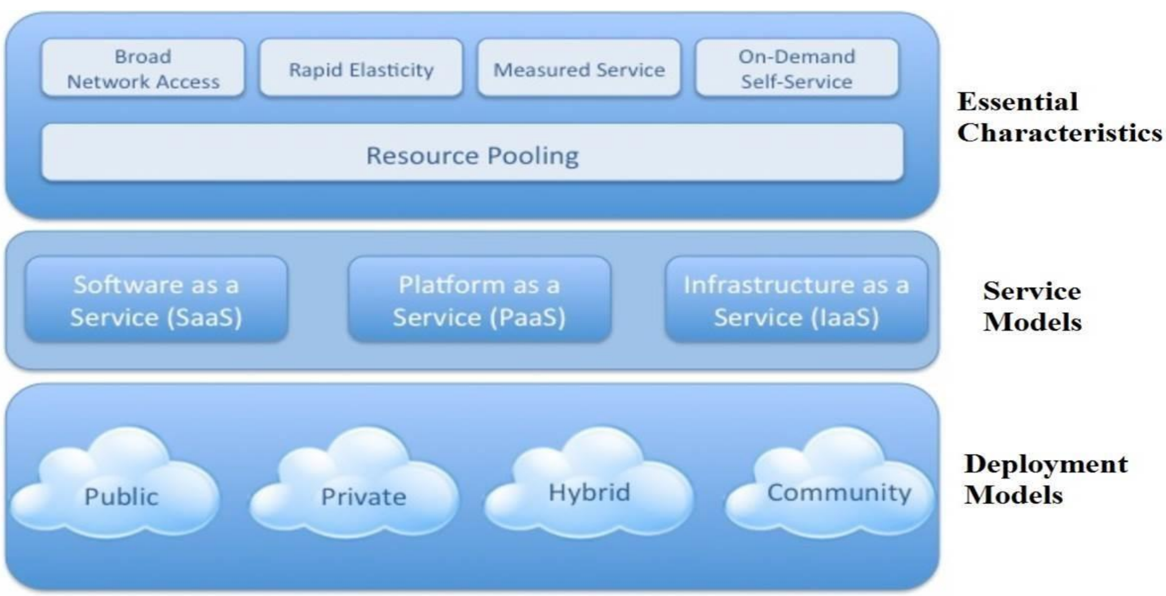


Figure 2.1: NIST definition of cloud computing [25]

The four ways (models) of deploying cloud computing are: Private Cloud, Public Cloud, Community Cloud, and Hybrid Cloud, which define the implementation of the cloud infrastructure within an organization.

Private cloud: It is the most expensive model among peers. The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It means the cloud serves only its designated organization and is not shared among others. The users are given privilege access using highly secure keys and often use external clients like VPN to log into the cloud. This also provides the client with complete access to the cloud infrastructure thereby enabling them to customize the platform to their specific needs. It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.

Community cloud: It is the most cost-effective cloud model. The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises. The cost is amortized over all these different organizations.

Public cloud: It is the most popular and serves a large number of users. The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider. In this model, the customer shares the cloud infrastructure with other customers. The operating and maintenance costs of the cloud are the responsibility of the cloud service provider. The cloud-hosting provider charges a fee, using a license model of pay per user. This benefits the organization monetarily as it eliminates the need for spending on operational IT costs. This model suits small businesses, which require load balancing and management, hosting SaaS applications, using interim infrastructure for development and testing of their applications. Most service providers are transparent with the physical locations of the datacenters. They provide the user with various levels of security packages to the public cloud users, based on subscription models.

Hybrid cloud: It is a combination of two or more of the above-mentioned models. Means that, the cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds). This enables organizations to partition their data based on the sensitivity of the information and host them on various clouds, striking a perfect balance between privacy and cost effectiveness.

2.1.2 Cloud Computing Enablers

Cloud computing is increasing in its use and importance in health and other sectors [27]. It sparked a significant amount of interest amongst IT industry and research [28]. So it becomes the promising solution in almost all sectors including the health sector in order they use ICT to provide quality services. Having this, it is better to know the key technological breakthroughs for realizing the promise of cloud computing which is called cloud enablers. As defined in [29],

Cloud enabler refers to the technologies and manufacturers that serve as the backbone for all cloud computing products and services. A broad term incorporating technology vendors and solutions, a cloud enabler allows an organization to build, deploy, integrate and deliver cloud computing solutions. They are primarily IT firms that develop hardware, software, storage, networking and other related product serving as a cloud environment component.

Four enablers of cloud computing technology have been explained in [30], these are virtualization, multi-tenancy, web services and cost-effective hardware which contribute to the development of effective and efficient cloud computing. Similarly, Yogesh K. and Dwivedi N. [31], presents six technology enablers for realizing the promise of Cloud Computing. These are Low-cost access and computing devices (LCACDs), Parallel programming, communication networks, Open Source Software (OSS), Cloud access to high performance computing and green computing

Virtualization technology is a key enabler and the foundation of cloud computing which is a method of running multiple independent virtual systems on a less physical resource, makes one computer act as many, sharing the resources of those computer hosts across multiple environments. It enables the division of the physical resources of the machine based on the application requirement and allows for dynamic reconfiguration of resources as required. The main benefit of using virtualization is the portability associated with the creation of n-numbers of virtual machines between different servers. This improves efficiency in the use of existing hardware as well as creating potential for rapid scaling up of resources on the fly and it makes the cloud function more effectively.

Multi-tenancy is the way in which access is given to several clients, who simultaneously use the same resource without intruding on each other's space or data stores. It is the key common attribute of both public and private clouds, and it applies to all three layers of a cloud such as IaaS, PaaS and SaaS. Often, multiple clients use a single application on the server and artificial software authentication protocols are set up to ensure privacy. This leads to optimal use of physical computational resources. Depending on the deployment model of the cloud, there are varying levels in the implementation of multi-tenancy concepts.

Web services cloud computing uses web services for connection, which is a software system that enables communication between two machines over a network. It provides a standardized platform to clients, allowing them to access applications in the server through a network. It uses XML for data tagging, SOAP for data transfer, WSDL for service description and UDDI for available service. By default, this system does not have any GUI features. However, most cloud service providers integrate this system to a web page with a GUI based application to offer specific functionalities.

Low-cost access and computing devices (LCACDs) are other enablers which are expected to play a pivotal role in the adoption of cloud computing among the masses, especially in the developing counties. Also Parallel programming enhances user experience by increasing effective use of multi-core processor. It is known that cloud applications are fully internet based. So high speed communication networks are required and the authors described as an essential cloud enabler.

Another key enabler of cloud computing is Open Source Softwares (OSS). It provides users full access to source code, which can be modified to suit specific user requirements and reduces the cost pertaining to SaaS. Cloud access to high performance computing is used to increase adaptability by E-researchers in order to use E-infrastructures for their scientific enquiry which is a long time dreams of grid computing. The last but not the list cloud computing enabler is green computing to make efficient use of electricity in cloud computing infrastructures that is deployed in a very large data centers.

2.1.3 Cloud Application Architecture and Design Considerations

Understanding of which application architectures are fit/best appropriate to operate in the cloud as well as the kinds of applications and data that run well in cloud environments, data backup needs and system workloads are the basic concepts to build application to run in a cloud environment. As described by Bernard Golden, there are three architectural choices for cloud environments: Traditional architecture; asynchronous application architecture; and Synchronous application architecture. As the author discusses, because of not all cloud environments are equal, we need to research and design our application appropriately for the particular environment offered by our cloud provider. In addition, we need to determine which applications and data sources used by the application will run in the cloud. Also, we need data backup to ensure recoverability in case of crashes [32].

Traditional architecture can be used for cloud application only if there is stable demand rather than tolerate huge variations in system load. A traditional architecture calls for one or more Web server systems interacting through a middle-tier software framework, ultimately interacting with a database. Infrastructure as a Service cloud providers; such as Amazon Web Services (AWS) as well as GoGrid and Rackspace accommodate this architecture.

In synchronous application architecture, end-user interaction is the primary factor in which large numbers of users may hit the system in a short duration and potentially it creates poor performance. To avoid this problem providing enough Web servers to handle total traffic; providing enough middleware to manage demand and providing a data tier that scales is required and must be included in the architecture.

Asynchronous application architecture does not support end-user interaction but focused on information processing like an extract from a database or a set of files. So this application architecture is used if there is no need of user interaction.

After selecting appropriate application architecture and platform, the applications will be designed. When designing applications for the cloud, we have often found it useful to know the most important part of the design of any application. This section describes design considerations [33] to overcome challenges to rapidly deliver reliable, manageable, feasible, scalable and economical cloud solutions for varying business requirements.

The feasibility which is the ability to deliver and maintain the system, within budgetary and time constraints like SLA must include under consideration. Availability describes the ability of the solution to operate in a manner useful to the consumer in spite of transient and enduring faults in the application and underlying operating system, network and hardware dependencies. So we need to consider uptime guarantee, Replication and failover, disaster recovery, performance and security. For gaining high availability, multiple zones, Graceful Degradation, global data replication, Distributed data management platforms and Asynchronous Messaging must be under consideration in cloud based applications.

Scalability is focus on any requirement to add additional capacity to the application and related services to handle increases in load and demand. So, we need to consider capacity, platform/data and the load to avoid contention issues and bottlenecks. There are two kinds of applications that need the scalability of cloud environments: user-facing or synchronous (i.e., a user interacts with the system and waits for an answer) and asynchronous (i.e., data is input, processed, and eventually concludes being worked up, with no one sitting around waiting for the results). Also, manageability which describes ability to understand the health and performance of the live system and manage site operations is one of the design considerations. So, we need to consider monitoring and deployment concepts to make the application more manageable.

2.1.4 Cloud Computing Related Terms and Technologies

A familiarity with cloud computing terminology will help anyone to follow the industry's developments and to have better understanding about the new language that has emerged in the area of cloud computing. There are hundreds of terms related to cloud computing. However, in this study presents some of the terms [34] and technologies [35] related to cloud computing.

Cloud computing typically has characteristics of Grid computing, Virtualization, Utility Computing and Autonomic computing [35].

Grid Computing: involves a network of computers that are utilized together to gain large supercomputing type computing resources. Using this network of computers large and complex computing operations can be performed.

Virtualization: it is the act of creating a virtual rather than a physical version of a computing environment, including computer hardware, operating system, storage devices, and so forth. It introduces a layer between Hardware and operating system. It also forms the foundation of cloud technology and helps in improving server utilization.

Utility computing: is a provisioning model in which services are available as needed, and users are charged for specific usage, in a manner similar to municipal utilities such as electricity or water. It helps in reducing initial investment. As the computing requirements for an individual or an organization changes, the billing changes accordingly, without incurring any additional cost. If the usage has reduced, then billing will also reduce accordingly.

Autonomic computing: is an initiative started by IBM in 2001. Autonomic means "selfmanaging" computers. In Autonomic computing, computers can automatically correct themselves without human intervention. Hence the goal of autonomic computing is for computing infrastructure to self-correct itself in unforeseen situations.

2.2 Cloud Computing and Healthcare

Related literatures (books, articles and the Internet) are reviewed in order to have clear understanding on this research works. Different techniques and tools which are relevant for the current research will be analyzed, modified and used from previous works.

A typical healthcare ecosystem consists of the providers namely doctors, physicians, specialists (those who work out of hospitals, clinics, nursing homes, etc.), payers (health insurance companies), pharmaceutical companies, imaging centers, IT solutions and services firms, and the patients [15]. In the context of technology usage or information management, the healthcare ecosystem handles a major key process, the healthcare provisioning process.

This process manages massive information loads and as such relies heavily on computing resources

2.2.1 Healthcare IT

IT systems for healthcare can bring lots of advantages in many ways, since this sector depends very much on information. In terms of computer usage, hospitals are even seemed to be outdone by the public administration: collection of data is mostly done on paper and is seldom fed into a computer system. The little data that does reach a computer system usually stays in isolated systems, such as a database for lab analysis values. However, especially in the healthcare domain, a closer integration of systems and a use of computer-aided data processing could be very helpful. Like almost no other domain, quality of healthcare depends on the availability of data. When a clinical decision has to be made, all the required information has to be available. [32]

Integration of Information and Communication Technology (ICT) enables faster feedback, remote monitoring and analysis and above all ensure mobility of individuals across countries. Neither these benefits come for free, nor can they be achieved without proper knowledge of the pitfalls and complexities specific to the domain of healthcare. This subsection tries to show health care challenges and available information technologies that help to address the challenges.

2.2.2 The current role of technology in healthcare

Information and Communication Technology (ICT) has performed a major role in digitizing and communicating patient information, leading to rapid patient diagnosis which further leads to faster time-to-treatment and superior overall health services [15]. Stakeholders in the healthcare industry have benefited by ICT applications in terms of efficiency and quality.

Communicating digitized patient information is typically done through a system like 'Telecare'. Volumes of patient data are transformed into information for decision-support through HIT applications and systems that are crucial for providing successful telecare services. Typically, healthcare providers make use of systems like Hospital Management Information System (HMIS), Picture Archiving and Communication Systems (PACS) , Electronic Medical / Health Records (EMR /EHR) system or Personal Healthcare Records (PHR) system to facilitate clinical workflows in order to provide telecare services.

2.2.3 Challenges in the Present healthcare

The digitization of clinical data, particularly in the form of EMR / EHR or PHR, and the automation of back office operations, will generate lots of data. As a result managing this data is not an easy task. As such, it becomes mandatory for them to put a robust network of IT systems in place [15]. Maintaining such a robust network of systems in-house increases the overall operational costs. The current ecosystem faces several challenges that demand technological advancement for sustaining itself in the future.

- **Ageing population driving the need for advanced, cost-effective technology**

Patient care for the elderly is necessitating advanced technologies and expensive healthcare services. This is increasing patient care provisioning costs for healthcare service providers. With the generation of large amounts of health data or information, huge IT storage infrastructure and backup solutions would be required. The use of technologies such as mobile phones, PDAs, laptops, iPads, etc. to reach out to patients in remote locations may not be a cost-effective solution. The back-end technology infrastructure needed to provide such services is expensive to source and maintain

- **Inadequate government spending**

  Government spending on healthcare has been largely inadequate in developing countries. This has led to the poor development of the public healthcare infrastructure. As appropriate funding is unavailable, governments and healthcare providers need to focus on sourcing / developing and deploying cost effective solutions that include technologies that could aid in providing healthcare to the masses.

- **Shift in disease burden from acute to chronic diseases challenging the limits of the traditional IT environment**

  There is a significant shift in the disease burden from acute to chronic diseases in developing countries. According to a World Bank report, the incidence rate of cancer, diabetes, obesity and heart disease in developing countries are nearing those of developed nations, and these chronic 13 diseases would be the leading cause of death in developing countries by 2015. With the spread of the disease burden globally, there is expected to be an increasing need of resources, both hum an and material, to address this need. This is likely to further add to the additional information management infrastructure requirement that can raise healthcare provisioning costs. Deploying advanced technology efficiently and cost effectively would be essential to address the growing demand for healthcare facilities in the remotest of areas. Innovation in the technology deployment is critical to meet this demand.

- **Workforce shortages and migration of skilled labor affecting specialist availability**

  Healthcare providers in developing countries are also struggling to maintain a well-trained and committed workforce. There is a huge mismatch in terms of the disease burden and resources required to handle them in these countries. Moreover, lower salaries, lack of better career prospects for healthcare professionals in developing countries is driving brain drain and creating a demand-supply gap. As this gap widens, accessing superior medical expertise would become expensive, further increasing healthcare costs.

- **Changing regulatory framework forcing digitization of health records to handle information efficiently**

  At the core of Healthcare Information Technology (HIT) applications is the management of patient records. Digitization of these or converting them to Electronic Health Records (EHRs) offers several important benefits to all stakeholders. Realizing the potential of cost containment by deploying EHRs, several countries have slowly started implementing HIT policies, making EHRs mandatory.

  Among the various geographies, Europe currently is a leader in health IT and EHR adoption followed by Asia-Pacific and the Americas. Countries like Australia, Canada, Denmark, the UK, New Zealand, the Netherlands and more recently the US have implemented EHRs, where the governments have mandated its adoption. In Canada there is a public-private partnership in the implementation of HIT and EHRs.

  Though the startup costs are high, EHRs can provide long-term solutions in delivering cost-efficient healthcare services for developed nations. In developing countries like India, there is no particular law or regulation mandating adoption of EHRs and the healthcare industry is unorganized to a large extent with minimum collaboration. In such cases, large-scale adoption of HIT and EHRs may not be a feasible option. Rather, they could first focus on localized usage of EHRs within their own network of hospitals and try improving other areas such as procurement, supply chain management and resource management.

2.2.4 Cloud computing in the health sector

The healthcare industry is evolving while grappling with several socio-economic and technological challenges along with the need to drive down HIT costs. A solution to this problem could be sought by moving to the Cloud [15].

Managing massive clinical data or EHRs requires heavy capital expenditure for sourcing and maintaining the IT infrastructure needed to store, transfer, retrieve, modify or print data and reports. At a state or a national level this could mean the storage and management of thousands of terabytes of data. All of this can be done more efficiently and at minimum costs if the data is moved into the Cloud.

The Cloud is a paradigm shift in HIT which enables stakeholders to focus more on their core competencies. In the case of the healthcare industry, it would involve the provisioning of healthcare products and services to patients by physicians, clinics, pharmacies, public health organizations, and payers. Providing a range of services across the entire spectrum of care would require the ability to scale application workloads, collaborate and share information and at the same time ensure that patient information is authentic and secured, and available anytime, anywhere. It would also mean the changing of business models, automating processes, streamlining workflows, and consolidating IT assets.

2.2.5 Benefits of Health Care Service in the Cloud

The Cloud is an IT deployment model that today's HIT departments could consider an investment towards developing a fully optimized health record and service management system.

Doctors or physicians could use medical appliances in their clinics to capture patient information and transfer it through a web-based SaaS application hosted in the Cloud (public or private) managed by the Cloud service provider. Applications would encrypt and deduplicate patient information prior to transmitting the same into the Cloud. Patients and doctors could access the records securely by exchanging secure electronic keys or third-party security devices similar to the way secure credit card transactions occur over the internet.

In a private Cloud scenario, hospitals could transmit and store data in a secure fashion between their affiliated physicians or other member hospitals through Single Sign On (SSO) access mechanism for easy and rapid access to information. In this way, hospitals can reduce their IT infrastructure burden and overall management cost.

In a public Cloud scenario, outpatient diagnostic centers can share results through the public cloud in a similar fashion like the private Cloud. Delivering services via SaaS applications in a public Cloud is also prevalent in physician offices. Similarly, drug manufacturers can make use of computing resources via cloud and reduce costs drastically using IaaS model. Every stakeholder in the healthcare ecosystem stands to benefit from the Cloud.

- Pharmaceuticals/Drug manufacturers- Drug manufacturing utilizes heavy IT infrastructure for its Research and Development (R&D) needs. The IaaS model could provide a drug manufacturer with On-Demand computing resources to perform drug research analysis, eliminating the need to retain computing capabilities and related IT expertise in-house.

- Hospitals- Using in-house or third-party SaaS applications that are housed in the Cloud, patients an provide access to their health history and information so that hospitals can streamline the admissions, care and discharge processes. Hospitals can connect to their own web portals and access patient data stored in the Cloud.
- Physicians- With the Cloud, people can provide their health history and information to their physicians anywhere, anytime, including data uploaded from health and fitness devices, to help physicians make more informed decisions.
- Pharmacies- People can administer or manage their prescriptions and associated information such as dosage, amount and frequency, and provide this information to their healthcare provider.
- Laboratories and imaging centers- Patient's diagnostic results can be transferred via suitable Apps onto Cloud-based platforms, e.g. Google Health or Microsoft Healthvault. This eliminates the need for in-house storage and helps retain historic information in a systematic manner. Healthcare providers can access these results with the patient's permission, to help them make more informed health decisions.
- Application Providers- Health and wellness companies can design and deliver health and wellness solutions compatible with Cloud platforms to offer rich user experiences and ease of managing users' sensitive personal health information.
- Device manufacturers- Health and fitness devices can be designed to work with Cloud platforms and Apps, so users can upload device data and share it with their doctors and families.
- Payers- Health payers can offer their members with innovative tools compatible with their chosen cloud platform to provide value added services by giving members' access to more health information, and thereby increasing the effectiveness of their care management programs which can help reduce claims costs. Payers can enable plan members to add health data to their records in the Cloud and provide it to their healthcare providers.

In case of the adoption of the Cloud, the responsibility of managing the underlying IT infrastructure to provide the aforementioned benefits lies with the Cloud services provider.

2.2.6 Risks of health care services in the cloud

Though the Cloud in healthcare provides several benefits for all the stakeholders of the industry, it has its own set of challenges. Some of these include security, privacy protection, disaster recovery, regulatory, governance, and the reliability of the Cloud.

Patient information security and privacy protection

The primary reasons the healthcare sector would offer resistance to making a move to the Cloud would be patient information security and privacy protection. Patient information across countries has been under the purview of legal frameworks e.g. the data privacy requirements legislated through HIPAA (Health Insurance Portability and Accountability Act) privacy rules in the US. HIPAA provide federal protection for personal health information. Similarly, the European Union has several directives pertaining to data protection. In many countries, the patient's Protected Health Information (PHI) cannot be moved out of the country of origin. Finally, the patients themselves would be concerned about the security of their personal data. The Cloud for healthcare would need to have a very strong data protection and privacy system in place to gain large scale acceptance / adoption in the marketplace. Therefore, the public Cloud environment may not be suitable for many healthcare applications. The Cloud's vulnerability to security breaches is similar to any traditional infrastructure environment. The adverse impact is, to a large extent, due to sensitivity of the information involved. A preferred solution to tackle the issue would be the use of a private Cloud coupled with secure access protocols and systems. Several Cloud service providers have come up with data and privacy protection equipment and tools; however, the industry is still skeptical of their effectiveness.

Interoperability and standardization

The Cloud offers the ability to shift all data / information onto systems and storage platforms provided / managed by Cloud service providers. At times, it may so happen that health records of patient stored, in the Cloud by his personal physician, might be needed to be accessed by the hospital that the patient was referred to. In such cases, the hospital should be able to access a patient’s health records stored on a different Cloud. At the same time, it is also expected that here would be a large number of software and applications that would exist in a Cloud. In order to ensure efficiency and optimum utilization of resources, it would be ideal if there is a high degree of interoperability and standardization between healthcare solutions for the Cloud.

Unifying the Cloud is an ideal state, but quite a distant goal to achieve, considering that there are several standards existing today. At the same time, there are also a number of Cloud standardization groups:

- Distributed Management Task Force (DMTF)
- Cloud Computing Interoperability Forum (CCIF)
- Open Grid Forum's 'Open Cloud Computing Interface Working Group' who is developing standards for managing the cloud.

Almost all industry majors like AMD, Broadcom, CA Technologies, Cisco, Citrix, Dell, EMC, Fujitsu, HP, Hitachi Limited, IBM, Intel, Microsoft, Novell, Oracle, Rackspace, Red Hat, and VMware are members of such groups and could accept one or the other standard. But it would be difficult to convince everyone in the business to accept a single interoperable / standardized system.

2.3 Related Works

At present, both industries and academics have done a lot of significant works in the area of cloud computing based healthcare resource and service delivery. Sharing of information, creating collaboration and supporting communication among medical practitioners is the most core idea of different articles that have been done by relating cloud computing and healthcare resource and service delivery issues. So that accessing patient healthcare information everywhere, with any device and every time as needed is the most important concept [2, 73, 75, 9, 36] to provide quality health and to save patients life. Several studies [73, 38, 42] include the idea that cloud computing might be used to support sharing of patient information and medical related soft wares; collaboration among stakeholders in the health area and enable patients in managing their health status.

Siddiqui describe the concept of a Telecare Medical Information System (TMIS) which deals with delivery of remote healthcare solution to the patient in their home and includes different medical services for patients and medical professionals such as real-time diagnosis and remote monitoring of physiological signals. In this work, Patients and medical practitioners use their smart phone as their complete identity over the cloud. So, they should connect to the TMIS by using their smartphone. And their smartphone needs to be equipped with authentication possibilities to ensure data privacy and data security. For this purpose, the authors propose a three-factor authentication (3FA) based on a dynamic cloud computing environment to enable remote user authentication [37]. Similarly, Van Gorp and Comuzzi discuss cloud-based personal health record system called MyPHRMachines which is an academic prototype, where a cloud is used to deploy health-related data and the application software to view and analyze it in a personal health record system. After uploading their medical data to MyPHRMachines, patients can access them again from remote virtual machines that contain the right software to visualize and analyze them without any need for conversion. As the authors describes, the patients can also build their records and can share their remote virtual machine session with selected caregivers by using this MyPHRMachine [38]. Furthermore, Takeuchi [39] present a prototypic cloud-based system to store personal health and lifestyle data using mobile devices. In a cloud infrastructure, they claim to have implemented data-mining technologies to extract individually important information such as lifestyle patterns. Although other persons like dietitians should have the possibility to add comments into the system, it is not explained how data access in the cloud will be managed.

Yao [2] presents Cloud-based hospital information system as a service for grassroots healthcare institutions, the authors created a community cloud based medical service delivery framework (CMSDF). The author's purpose is to reduce technical and economic barriers (like high upfront investment costs and the perceived lack of return on investment (ROI) and challenges of implementation, maintenance, and renewal) to deliver quality healthcare service by enabling sharing of software and tools from large general hospital to its associated smaller health institutions including community health service centers or rural clinics. In the prototype CMSDF, a cloud-based Virtual Desktop Infrastructure is owned and managed by a large hospital which is able to share its medical software as SaaS with the smallest healthcare institutions. According to the author for the 34 cooperative sanatoriums that participated, 89.9% of investment and maintenance cost were saved because the smaller facilities had not to buy and host expensive software on their own.

According to Ratnam and Ramayah [40], A structural equation modeling approach for the adoption of cloud computing to enhance the Malaysian healthcare sector, cloud computing could be the method of choice to reduce the escalating costs of data storing and sharing. Even if the authors did not describe this cloud system in detail, they have mentioned that a cloud platform using Microsoft Windows Azure was used as prototype architecture. In contrast, Hsieh and Hsu have presented a very comprehensive and detailed description of a 12-lead ECG telemedicine service based on cloud computing [41]. They have clearly described how the processing, visualization, management and eLearning services are deployed within the commercial Microsoft Azure cloud platform. They have further presented the reasons for adopting the Azure platform and the financial background of the implementation, based on the Azure pricing model with monthly costs directly related to CPU hours and GB storage used.

Moreover, Hidden [42] described the development of e-Science Central (a platform as a service which itself was built on an infrastructure as a service environment) which allows scientists to store, analyze and share data in the cloud. Their article clearly illustrates not only the set of cloud services provided, which cover data storage services, but also service execution, workflow enactment and security. Finally, as one of three case studies, they present a medical pilot investigation (the MOVEeCloud project) where medical specialists assess the physical activity of patients based on data uploaded to the e-Science Central cloud by wearable accelerometers.

To summarize, any of the researchers didn’t touch the problem or unavailability of healthcare knowledge exchange network to enable sharing of healthcare information, tools and approaches to create collaboration and to develop knowledge, ability, experience, technology adaption and problem-solving skill. Therefore, we have tried to discuss these related works done by other researchers which include telemedicine related works, patient’s self-management works, medical imaging and clinical information system works as a broad domain. By considering untouched gaps discussed above, we propose to design cloud based framework of resource sharing for Ethiopian healthcare institutions.

CHAPTER THREE

METHODOLOY

In order to achieve the general and specific objectives of the study and to solve the problem by answering the research questions, different methods are used. After deciding the area of interest, the first step was performing literature review to get previous works about cloud computing framework on the area of different healthcare institutions.

The next step was conducting interview with patients, practitioners and healthcare providers, a stakeholder in healthcare including doctors, nurses, physicians and ICT experts from different healthcare organizations including public and private hospitals and clinics to collect data for better understanding of the current ICT utilization strategy, effectiveness and efficiency of service delivery, and their recommendations for better service delivery are covered by the interview; and to know their expectations on the cloud based resource sharing system. The last step was selection of tools to design the proposed framework and to implement its prototype for the purpose of proof of concept and system evaluation purpose.

Based on these steps, a design of cloud based capacity building resource sharing architecture and its prototype was developed to address the problem (Figure 3.1).

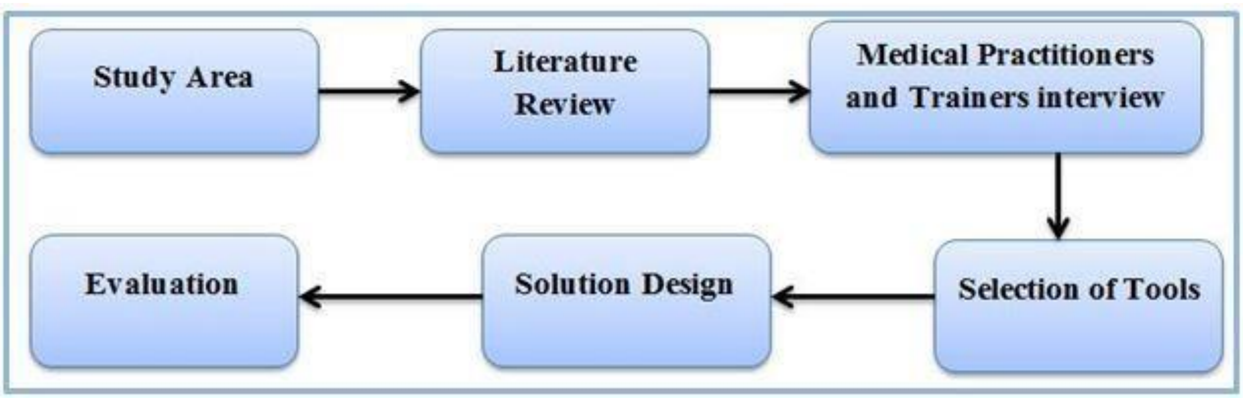


Figure 3.1: Research Methodology

3.1 Research Design

For any research studies, there are four commonly used approaches which are quantitative, qualitative, design oriented and mixed method [43]. The quantitative research approach uses standardized measures, numerical values, have large sample size and analyze data using statistical program. It is used when one starts with a hypothesis and tests for validation and invalidation of that hypothesis. It is applicable to phenomena that can be expressed in terms of quantity or number [44]. On the other hand, qualitative research approach is focused on describing situations, phenomenon, problem or event; tends to be more in-depth and have smaller sample size [45]. A mixed methods research design is a procedure for collecting, analyzing, and mixing both quantitative and qualitative research and methods in a single study to understand a research problem [46]. It is mostly used when one type of research is not enough to address the research problem and when practically multiple view points; biased and unbiased; subjective and objective ideas occurred [55]. Design-Oriented Research creates an artifact by artificial object and its basic objective is producing new knowledge during this process. It is mostly focus on knowledge flows and process steps that is problem awareness, suggestion, development, evaluation and conclusion; eventually aims to get result or output [47]. In this study, a combination of qualitative and design oriented research method were used for data manipulation and to evaluate the functionality of the prototype. Due to qualitative approach focuses on involving quality and helps in having insight in to problems, more tasks in the study were qualitative.

To show the usefulness of the proposed cloud based resource sharing framework for healthcare organizations and their workers, design-oriented research methodology was used in this study. Design-oriented research method is basically a problem-solving model [48].

3.2 Process Model

In software Engineering, process model is a description of the sequence of activities carried out in the software project and the relative order of these activities. There are several process models which include prototyping, unified process, incremental and agile. We can rarely combine process models or use together in software development process [51]. In this study, prototyping and Unified process models were used together to accomplish the overall objectives. Prototyping is about creating some parts of the system being developed. It is not a standalone and complete development methodology rather it is a model to test specific features in the situation of complete methodology [52] like unified process.

Hence, Prototyping is a process model that offers a solution to the problem and it is the practice of building an early version of a system which shows particular features of the actual system. There are two types of prototyping which are throwaway and evolutionary prototyping. In throwaway prototyping the different parts of the system are developed and then will be ignored after demonstrating the basic functions that the system will carry out. Thus the final deliverable system will differ from the prototype. But evolutionary prototyping technique implements the prototype of the system that demonstrates its basic features and finally it will be part of the final deliverable system.

The unified process (UP) or rational unified process (RUP) is the most recent software development process model which provides overall plan and strategy to be used as part of it. It has two purposes, the first one is meeting user's need by the use of techniques like use cases, iteration and emphasis on the architecture of the software. The second purpose is accommodating risks by making use of iterative technique. UP consists of four phases: inception, elaboration, construction and transition. Inception phase consists of a feasibility study to establish whether the proposed system is worthwhile. In elaboration phase, the creation of general architecture of the system and requirements lists is completed. In construction phase, the development of the actual system, in this case the prototype of the actual system is performed. The transition phase involves making the system in to use by installing and running this system. Generally, the unified process model importantly uses use cases, iteration and software architecture techniques; and the prototyping model is concerned with the development of incomplete version of the system. Therefore, these are the reasons to select these models for this study.

3.3 Data Source

The data which is used in this thesis as an input is acquired from the following sources: -

- Stakeholders from Government and Private Hospitals which includes medical practitioners, trainers and ICT professionals working in the HDIs. In this way different government and private HDIs were selected.
- Comprehensive literature review and document analysis from the internet is also another way of data sources for this work. So, internet is one of the sources of data as explained in the methodology overview section of chapter one.

- The other source of data in this work was observation. During this study observation were used as a means to collect the required data specially to find health practitioners problem related to lack of resources.

3.4 Data Collection Techniques

This thesis was carried out in three different cities of the country Addis Ababa, Adama and Debrezeit. The interviewed professionals were both from ICT professions working in the healthcare system of the HDIs and the medical practitioners of the HDIs. Also, the interviewees were from both private and public HDIs. This enables us to collect the required data correctly as well as it makes the data collection procedure easier.

This data collection technique involves two phases: (1) in-depth interviews with 50 health workers and ICT professionals from Bishoftu Hospital, Adama General Hospital, Alert Center, Korea Hospital, etc concerning their ideas about capacity building program, its indispensableness, current situations, barriers and solutions to provide it consecutively and about cloud computing as well as their interest if it is cloud based; (2) observation based on work place study about health worker's technology adaption, experience and problem solving skills and their medical knowledge. Although the need of consecutive capacity building program for health workers is clear, efforts were made to get the correct information about the health worker's ability and experience in their work place from different patients. This is done through different techniques but getting the right information was extremely difficult as observed.

The interview mainly focused on importance of resource sharing for the health practitioners, how currently provided to them and what efforts were done to provide it consecutively to them. Also, how the health practitioners get access to the health resources and materials, in terms of technology used and collaboration and knowledge environment reserved were emphasized.

The main purpose of this work was, looking forward to use the current technology for sharing and accessing capacity building resources in healthcare institutions to solve the problem of lack of healthcare learning, knowledge sharing and networking and collaboration center. Cloud computing based platform architecture was proposed to solve the problem through sharing of all capacity building resources to all healthcare workers at all level. Cloud computing has a solution for storing and processing large amount of capacity building resources, for sharing and accessing these resources with cost effective, flexible, scalable and multi-tenant way. This will help the healthcare workers to develop ability, experience and knowledge, problem solving skill and technology adaption in their profession by accessing the shared resources using their mobile phone and other existing ICT infrastructures ubiquitously.

3.5 Issues of Reliability and Validity

The overall works of this thesis, different methods and techniques was employed. Data collection and analysis techniques, programming and tools techniques, evaluation and testing mechanisms, comprehensive document analysis as well as prototyping as a methodology were used. Primary data from government and private healthcare delivery institutions were collected directly and secondary data through documentary analysis were collected. For the purpose of collecting data, interview technique was used using questionnaire. The questionnaires were distributed to hospitals in Addis Ababa, Adama and Debrezeit. Because of budget constraints, we could not distribute the questionnaire outside Addis Ababa, Adama and Debrezeit.

3.6 Sampling Techniques

There are different sampling techniques in qualitative research to sample the research participants. This includes purposive sampling, snowball sampling, quota sampling, typical case sampling, convenience sampling and others [53]. In this study, selective or purposive sampling technique was used to collect the qualitative data. So, data collection was started by identifying the participants using purposive sampling technique. Purposive or selective also known as judgment sampling technique is mostly used technique in qualitative research due to the qualities the informant possesses. Healthcare practitioners from different healthcare delivery institutions were chosen directly through personal contact. The healthcare practitioners and other concerned bodies were selected based on appropriateness for the study; the knowledge they have and technology used, [55] because this technique uses appropriateness principles in place of random sampling.

CHAPTER FOUR

CLOUD SERVICE AND PROVIDER SELECTION

4.1 Introduction

Cloud computing is one of the most popular buzzwords used these days. It is the upcoming technology provisioning resources to the consumers in the form of different services like software, infrastructure, platform, and security. Services are made available to users on demand via the Internet from a cloud computing provider's server as opposed to being provided from a company's own on-premise servers. Cloud services are designed to provide easy, scalable access to applications, resources, and services and are fully managed by a cloud service provider. A cloud service can dynamically scale to meet the needs of its users, and because the service provider supplies the hardware and software necessary for the service, there is no need for a company to provision or deploy its own resources or allocate information technology (IT) staff to manage the service. Examples of cloud services include online data storage and backup solutions, web-based e-mail services, hosted office suites and document collaboration services, database processing, and managed technical support services.

Cloud services can be broadly classified into three types: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). With growing technologies, many more services are emerging in this field, such as Security as a Service (SeaaS), Knowledge as a Service, and Data Analytics as a Service.

To identify the potential cloud services that can be provided to healthcare institutions, the researchers started by analyzing the requirements for healthcare institutions by means of interview and looking at different institutions with best cloud service practices. The analysis process was based on interviewing ICT directors from selected institutions, reviewing best cloud service practices of healthcare institutions around the world.

This section provides an overview on the different cloud platform service providers and more in detail about Redhat's OpenShift online platform for this work. For the purpose of developing prototype of the proposed framework, we have started to find the best cloud platform as a service provider. As a result, we have got a lot of available tools by conducting advanced search and we have selected only the five tools for comparison and analysis to use the best one for our intension.

The relevance selection based on availability (open source or commercial) was conducted to remove obviously commercial tools because only freely available tools were needed for this study as an academic purpose.

4.2 Cloud Service Providers

Many companies have come forward to adapt the cloud environment and ensure that the users as well as the companies benefit from this. Amazon, Microsoft, Google, Yahoo, EMC, Salesforce, Oracle, IBM, and many more companies provide various tools and services in order to give cloud support for their customers.

4.2.1 EMC

EMC is one of the leading global enterprises that require dynamic scalability and infrastructure agility to meet changing applications as well as business needs. EMC chose cloud computing as the ideal solution to reduce the complexity and optimize the infrastructure. Offering Information Technology as a Service (ITaaS) reduces the energy consumption through resource sharing.

EMC IT

Virtualization is the main concept behind the success of EMC IT. By virtualizing the infrastructure, allocation of the resources on demand is possible. This also helps to increase efficiency and resource utilization.

EMC IT provides its business process units with IaaS, PaaS, and SaaS. Figure 4.1 gives an overview of the services offered by EMC, which are explained in the following:

1. IaaS offers EMC business units the ability to provision infrastructure components such as network, storage, computing, and operating systems individually or as integrated services.

2. PaaS provides the secure application and information frameworks on top of application server, web server, database, unstructured content management, and security components as a service to business units from which to develop solutions. EMC IT offers database platforms (Oracle Database as a Service, SQL Server as a Service, Greenplum as a Service) and application platforms

(application development, Enterprise Content Management as a Service, Information Cycle Management as a Service, Security PaaS, Integration as a Service) for the purpose of development.

3. SaaS provides applications and tools in a service model for business enablement. EMC IT brought together several existing business solutions under the unified architecture named as Business Intelligence as a Service. It also offers Enterprise Resource Planning (ERP) and Customer Relationship Management (CRM) as a Service.

4. User Interface as a Service (UIaaS) provisions user and interface experience, rather than provisioning the actual device used.

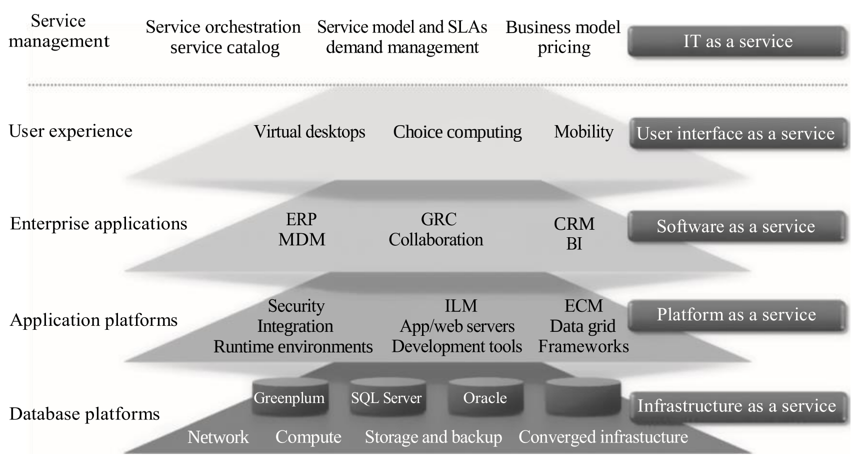


Figure 4.1: Cloud services by EMC. (Adapted from EMC IT's journey to the private cloud, applications and the cloud experience, White Paper-EMC.)

4.2.2 Google

Google is one among the leading cloud providers that offer secure storage of user's data. It provides cloud platform, app engine, cloud print, cloud connect, and many more features that are scalable, reliable, as well as secure. Google offers many of these services for free or at a minimum cost making it user friendly.

Cloud Platform

Google Cloud Platform enables developers to build, test, and deploy applications on Google's highly scalable and reliable infrastructure. Google has one of the largest and most advanced networks across the globe. Software infrastructures such as MapReduce, BigTable, and Dremel are the innovations for industrial development.

Google Cloud Platform includes virtual machines, block storage, NoSQL datastore, and big data analytics. It provides a range of storage services that allow easy maintenance and quick access of user's data. The cloud platform offers a fully managed platform as well as flexible virtual machines allowing the user to choose as per the requirements. Google also provides easy integration of user's application within the cloud platform.

Applications hosted on the cloud platform can automatically scale up to handle the most demanding workloads and scale down when traffic subsides. The cloud platform is designed to scale like Google's own products, even when there is a huge traffic spike. Managed services such as App Engine or Cloud Datastore provide autoscaling that enables application to grow with the users. The user has to pay only for what he or she uses.

Cloud Storage

Google Cloud Storage is a RESTful online file storage web service for storing and accessing one's data on Google's infrastructure. Representational state transfer (REST) is an architectural style consisting of a coordinated set of architectural constraints applied to components, connectors, and data elements within a distributed system. The service combines the performance and scalability of Google's cloud with advanced security and sharing capabilities. Google Cloud Storage is safe and secure. Data are protected through redundant storage at multiple physical locations.

The following are the few tools for Google Cloud Storage:

- Google Developers Console is a web application where one can perform simple storage management tasks on the Google Cloud Storage system.
- gsutil is a Python application that lets the user access Google Cloud Storage from the command line.

Google App Engine

Google App Engine lets the user run web applications on Google's infrastructure. App Engine applications are easy to build, easy to maintain, and easy to scale as traffic and data storage needs grow. With App Engine, there are no servers to maintain: Just upload the application, and it is ready to serve users.

The app can be served from the user's own domain name (such as http:// www.example.com/) using Google Apps. Otherwise, it can be served using a free name on the appspot.com domain. An application can be shared with the world or limit access to members of an organization. Figure 4.2 shows the different modules in Google App Engine. Integration of cloud computing services with support services and client capabilities is shown in the diagram.

Google App Engine supports apps written in several programming languages. With App Engine's Java runtime environment, one can build one's app using standard Java technologies, including the JVM, the Java servlets, and the Java programming language—or any other language. App Engine also features a Python runtime environment, which includes a fast Python interpreter and the Python standard library. App Engine also features a PHP runtime, with native support for Google Cloud SQL and Google Cloud Storage that works just like using a local MySQL instance and doing local file writes. Finally, App Engine

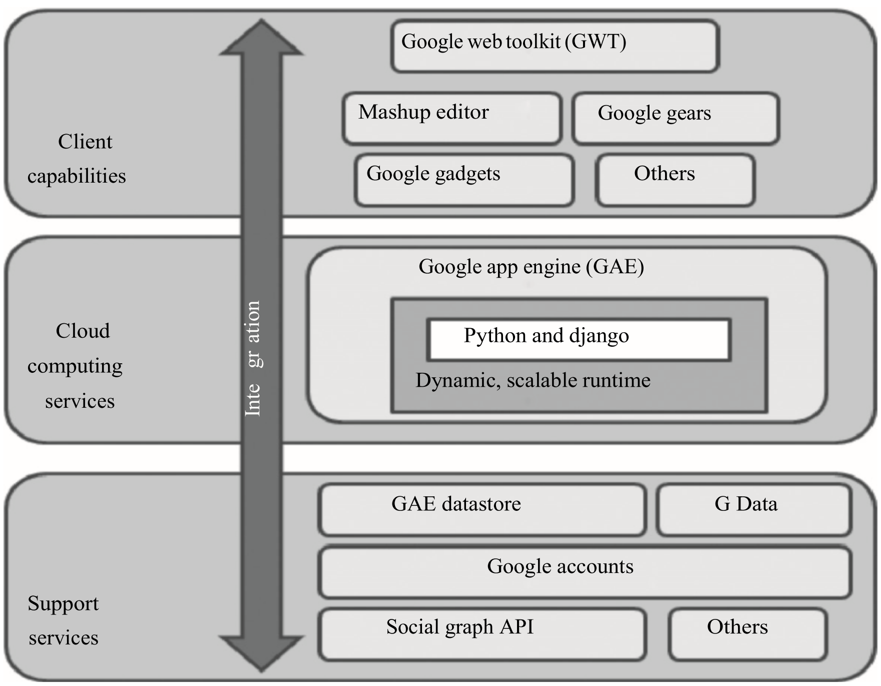


Figure 4.2: Google App Engine. (Adapted from http://rdn-consulting.com/blog/tag/azure/, accessed January 16, 2014).

provides a Go runtime environment that runs natively compiled Go code. These runtime environments are built to ensure that your application runs quickly, securely, and without interference from other apps on the system.

With App Engine also, the user has to only pay for what he or she uses. There are no setup costs and no recurring fees. The resources used by the application such as storage and bandwidth are measured in gigabyte and billed at competitive rates. One has to control the maximum amount of resources one's app can consume, so it always stays within one's budget.

App Engine costs nothing to get started. All applications can use up to 1 GB of storage and enough CPU and bandwidth to support an efficient app serving around five million page views a month, absolutely free. When billing is enabled for the application, free limits are raised, and one has to only pay for resources one uses above the free levels.

4.2.3 Amazon Web Services

Amazon Web Services (AWS) is a collection of remote computing services (also called web services) that together make up a cloud computing platform, offered over the Internet by Amazon.com. The most central and well known of these services are Amazon Elastic Compute Cloud (Amazon EC2), Amazon Simple Queue Service (Amazon SQS), and Amazon S3 as shown in Figure 4.3.

Amazon EC2 is a computing service, whereas Amazon SQS and Amazon S3 are support services. The service is advertised as providing a large

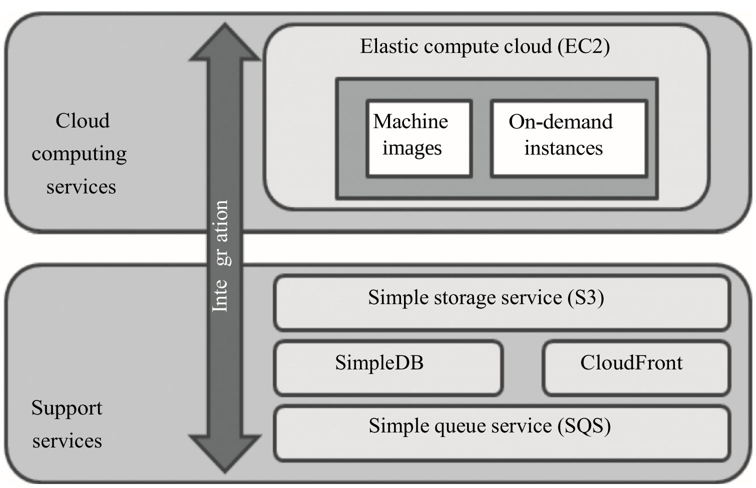


Figure 4.3: AWS. (Adapted from http://rdn-consulting.com/blog/tag/azure/, accessed January 16, 2014).

computing capacity (potentially many servers) much faster and cheaper than building a physical server farm. Amazon's data centers are located at Ashburn, Virginia, Dallas/Fort Worth, Los

Angeles, Miami, Newark, New Jersey, Palo, Alto, California, Seattle, St. Louis, Amsterdam, Dublin, Frankfurt, London, Hong Kong, Singapore, Tokyo, etc.

4.2.4 Microsoft

Cloud computing provides a new way of looking at IT at Microsoft called Microsoft IT (MSIT). Cloud computing is now the preferred and default environment for new and migrated applications at Microsoft. MSIT has developed a methodology and a set of the best practices for analyzing their current application portfolio for possible candidates to migrate to cloud computing. This analysis enables MSIT to select the ideal cloud computing–based environment for each application. MSIT has captured these best practices and documented them for other Microsoft customers who wish to migrate their organizations to cloud computing.

Windows Azure

Windows Azure Cloud Services (web and worker roles/PaaS) allow developers to easily deploy and manage application services. It delegates the management of underlying role instances and operating system to the Windows Azure platform.

The Migration Assessment Tool (MAT) for Windows Azure encapsulates all the information to be aware of before attempting the application migration to Windows Azure. Based on the response to a series of simple binary questions, the tool generates a report that outlines the amount of development effort involved to migrate the application, or the architecture considerations for a new application.

The Windows Azure Pricing Calculator analyzes an application's potential public cloud requirements against the cost of the application's existing infrastructure. This tool can help to compare current operational costs for an application, against what the operating costs would be on Windows Azure and SQL Azure.

Windows Azure Pack for Windows Server is a collection of Windows Azure technologies available to Microsoft customers at no additional cost for installation into their data center. It runs on top of Windows Server 2012 R2 and System Center 2012 R2 and, through the use of the Windows Azure technologies, it allows you to offer a rich, self-service, multitenant cloud, consistent with the public Windows Azure experience.

4.2.5 IBM

IBM is one among the players in the field of cloud computing offering various cloud services to the consumers. IBM cloud computing consists of cloud computing solutions for enterprises as offered by the global IT company IBM. All offerings are designed for business use, marketed under the name IBM SmartCloud. IBM cloud includes IaaS, SaaS, and PaaS offered through public, private, and hybrid cloud delivery models, in addition to the components that make up those clouds.

IBM offers an entry point to cloud computing whether a client is designing their own virtual private cloud, deploying cloud service, or consuming cloud workload applications. The IBM cloud framework begins with the physical hardware of the cloud. IBM offers three hardware platforms for cloud computing, which offer built-in support for virtualization. The next layer of the IBM framework is virtualization. IBM offers IBM WebSphere application infrastructure solutions that support programming models and open standards for virtualization.

The management layer of the IBM cloud framework includes IBM Tivoli middleware. Management tools provide capabilities to regulate images with automated provisioning and deprovisioning, monitor operations, and meter usage while tracking costs and allocating billing. The last layer of the framework provides integrated workload tools. Workloads for cloud computing are services or instances of code that can be executed to meet specific business needs. IBM offers tools for cloud-based collaboration, development and test, application development, analytics, business-to-business integration, and security.

IBM SmartCloud

IBM SmartCloud is a branded ecosystem of cloud computing products and solutions from IBM. It includes IaaS, SaaS, and PaaS offered through public, private, and hybrid cloud delivery models. IBM places these offerings under three umbrellas: SmartCloud Foundation, SmartCloud Services, and SmartCloud Solutions. Figure 4.4 briefly explains the architecture of IBM SmartCloud.

SmartCloud Foundation consists of the infrastructure, hardware, provisioning, management, integration, and security that serve as the underpinnings of a private or hybrid cloud. Built using those foundational components, PaaS, IaaS, and backup services make up SmartCloud Services.

Running on this cloud platform and infrastructure, SmartCloud Solutions consist of a number of collaboration, analytics, and marketing SaaS applications.

Along with IaaS, PaaS, and SaaS, IBM also offers Business Process as a Service (BPaaS). Infrastructure cloud services provide the consumer the provision of processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. In platform cloud services, a consumer can deploy consumer-created or consumer-acquired applications onto the cloud infrastructure created using programming languages and tools supported by the provider. Application cloud services allow consumers to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a web browser (e.g., web-based e-mail). Business process cloud services are any business process (horizontal or vertical) delivered through the cloud service model (multitenant, self-service provisioning, elastic scaling, and usage metering or pricing) via the Internet with access via web-centric interfaces and exploiting web-oriented cloud architecture. The BPaaS provider is responsible for the related business functions.

4.2.6 VMware

VMware, a leader in virtualization technology, has come up with enterprise cloud computing solutions. Having been a dominating player in the virtualization domain, VMware is currently providing a range of products for the development of private and public clouds and for leveraging the services offered by both as a hybrid cloud, such as VMware vCloud Director, VMware vCloud Datacenter Services, VMware vSphere, and VMware vShield to name a few.

Private clouds enable the better usage and management of internal IT infrastructure than the traditional methods. Greater operational efficiency, secure, fault-tolerant, well-managed computing environments can be modeled and operated. VMware's private cloud offering provides greater standardization, rapid provisioning, and self-service for all applications and unparalleled cost savings by consolidating their physical infrastructures. VMware's modular technology enables the user to select from a variety of hardware, software, and certified service providers to result in efficient cloud computing. Thus, the family of products offered by VMware promotes compatibility and retains the choice of freedom for the users to obtain desired services.

Private clouds can be created by using the VMware vSphere and VMware vCloud Director. VMware vSphere is a robust virtualization platform used to transform IT infrastructures into virtual storage, compute, and network resources and provide them as a service within the organization. VMware vSphere provides services at both the infrastructure and application levels. At the infrastructure level, it provides options to perform efficient operation and management of the compute, storage, and network resources. At the application level, service-level controls are provided for the applications running on the underlying infrastructures, leading to available, secure, and scalable applications.

The VMware vCloud Director, coupled with VMware vSphere, is a software solution that enables enterprises to build secure, multitenant private clouds by pooling infrastructure resources into virtual datacenters and exposing them to users through web-based portals and programmatic interfaces as fully automated, catalog-based services. VMware vCloud Director abstracts the virtual computing environment from the underlying resources and provides a multitenant architecture that features isolated virtual resources, independent LDAP authentication, specific policy controls, and unique catalogs. VMware vShield technologies are used to provide security to these environments by using services like perimeter protection, port-level firewall, NAT and DHCP services, site-to-site VPN, network isolation, and web load balancing. The VMware vCloud Director allows users to catalog infrastructure and application services of the desired configurations and deploy and consume them as needed. Interactions with the virtual data centers or the catalogs are through a user-friendly web portal or the vCloud API. The vCloud API is an open, REST-based API that provides scripted access, complying with the open virtualization format (OVF). The API can be used along with VMware vCenter Orchestrator to automate and orchestrate operational processes like routine tasks, activities, and workflows.

Public and hybrid cloud solutions are provided by VMware by partnering with other companies, certified as service providers. VMware vCloud Datacenter Services and VMware vCloud Express offer efficient solutions for utilizing IaaS either as a public cloud or a hybrid cloud. vCloud Datacenter Services provides a scalable environment, where internal resources are augmented with the external resources. vCloud Datacenter Services are built on the same technology and foundations as VMware vCloud Director and VMware vSphere to enable interoperability between cloud environments. Thus, the user is free to burst his private cloud into public cloud of his preferred service provider. vCloud Express is an IaaS offering delivered by leading VMware service provider partners. It is a cobranded service that provides reliable, on-demand, pay-as-you-go infrastructure. The VMware vCloud Express providers are Virtacore vCloud Express, Hosting.com, Melbourne IT, and Terremark's vCloud Express. Instance types, load balancing, storage options, and pricing vary between service providers.

4.2.7 Manjrasoft

Manjrasoft is one of the nonmajor providers of cloud services. But it has come up with a platform called Aneka that provides a set of services that help the development of applications in an easier way. Manjrasoft develops marketoriented cloud computing platforms that allow one to build, accelerate, and manage the applications ultimately saving one's time and money, leading to enhanced business productivity and profit.

Aneka Platform

Aneka provides a set of services that make enterprise cloud construction and development of applications as easy as possible without sacrificing flexibility, scalability, reliability, and extensibility. Figure 4.5 gives an overview of the Aneka platform. The key features supported by Aneka are as follows:

1. A configurable and flexible execution platform (container) enabling pluggable services and security implementations. Multiple authentication/ authorization mechanisms such as role-based security and Windows domain–based authentication are considered for this purpose.
2. Multiple persistence options including Relational Database Management System (RDBMS), Structured Query Language (SQL) Express, MySQL, and flat files.
3. Software development kit (SDK) supporting multiple programming models including object-oriented thread model, task model for legacy applications, and MapReduce model for data-intensive applications.
4. Custom tools such as Design Explorer for parameter sweep studies.

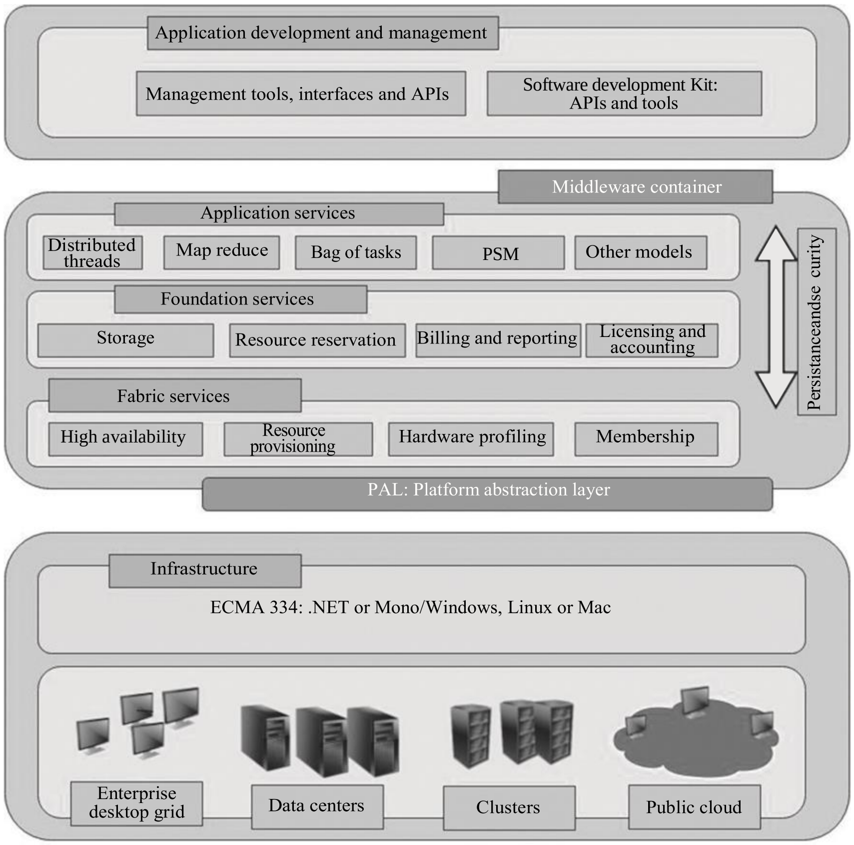


Figure 4.4: Overview of the Aneka platform. (Adapted from http://www.manjrasoft.com/aneka_architecture.html, accessed January 8, 2013.)

5. Easy to use management tool for SLA and Quality of Service (QoS) negotiation and dynamic resource allocation.
6. Supports deployment of applications on private or public clouds in addition to their seamless integration.

TABLE 4.1 Tools and Services Offered by Companies

Company Name	Tools/Services
EMC	Captiva Cloud toolkit
Google	Google App Engine, Google Docs, Google Cloud Connect Google Cloud Print
Amazon	Amazon EC2, Amazon S3, Amazon SQS

Microsoft	Microsoft Assessment and Planning Toolkit, Windows Azure Sharepoint
IBM	IBM Smart Cloud
Salesforce	Sales Cloud, Service Cloud
SAP LABS	SAP HANA Cloud
VMware	vCloud
Manjrasoft	Aneka Platform
Red Hat	OpenShift Enterprise, OpenShift Origin, OpenShift Online
Gigaspaces	Cloudify

TABLE 4.2

Details of Cloud Service Providers

Provider Name	Service Model	Deployment Model	Server Operating System
Amazon Web Services	IaaS	Public	Widows, Linux
Google App Engine	PaaS	Public	Windows
Windows Azure	IaaS	Public	Widows, Linux
IBM Cloud	IaaS	Private, hybrid	Widows, Linux
Salesforce Platform	PaaS	Public	Widows, Linux
Rackspace	IaaS	Public, private, hybrid	Widows, Linux
SAP HANA Cloud	PaaS	Public	Linux

model and deployment model offered by that system. Further details regarding the companies can be seen in the references section. An attempt has been made to list the tools/services offered by each company in Table 4.1. Though there are a number of companies, we have chosen few companies that have progressed a lot in this area. Table 4.2 gives the information about few providers and the prices on a per hour basis along with the service

Aneka allows servers and desktop PCs to be linked together to form a very powerful computing infrastructure. This allows companies to become energy efficient and save money without investing in a number of computers to run their complex applications.

Each Aneka node consists of a configurable container that includes information and indexing, scheduling, execution, and storage services. Aneka supports multiple programming models, security, persistence, and communications protocols.

4.3 Open Source Support for Cloud

Cloud computing is one of the most popular technologies nowadays. According to the National Institute of Standards and Technology (NIST), cloud computing is an on-demand, rapid-service provisioning model over the Internet in the form of compute, network, storage, and application with minimal management effort. The characteristics of cloud computing are ondemand self-service, broad network access, resource pooling, rapid e lasticity, and measured services. In order to support cloud computing, various tools are freely available and there are proprietary softwares. Open source support for cloud computing is a major developmental step for the rapid growth of cloud technology.

Open Source in Cloud Computing: An Overview

Open source refers to a program or software in which the source code (the form of the program when a programmer writes a program in a particular programming language) is available to the general public for use and/ or modification from its original design is free of charge. Open source code is typically created as a collaborative effort in which programmers improve upon the code and share the changes within the community. Open source is free whereas proprietary software is privately owned and controlled. In the computer industry, proprietary is considered as the opposite of open. A proprietary design or technique is one that is owned by a company. It also implies that the company has not divulged specifications that would allow other companies to duplicate the product.

In cloud computing environment, open source support has led to many innovations by providing various things as services, that is, X as a Service, where X can be Software, Platform, Infrastructure, etc.

Difference between Open Source and Closed Source

Some software has a source code that cannot be modified by anyone but the person, team, or organization that created it and maintains exclusive control over it. This kind of software is frequently called proprietary software or closed source software, because its source code is the property of its original authors, who are the only ones legally allowed to copy or modify it. Microsoft Word and Adobe Photoshop are examples of proprietary software. In order to use proprietary software, computer users must agree (usually by signing a license displayed the first time they run this software) that they will not do anything with the software that the software's authors have not permitted explicitly.

Open source software is different. Its authors make its source code available to others who would like to view that code, copy it, learn from it, alter it, or share it. LibreOffice and the GNU Image Manipulation Program are examples of open source software. As they do with proprietary software, users must accept the terms of a license when they use open source software, but the legal terms of open source licenses differ dramatically from those of proprietary licenses. Open source software licenses promote collaboration and sharing because they allow others to make modifications to source code and incorporate that code into their own projects. Some open source licenses ensure that anyone who alters and then shares a program with others must also share that program's source code without charging a licensing fee for it. In other words, computer programmers can access, view, and modify open source software whenever they like as long as they let others do the same when they share their work. In fact, they could be violating the terms of some open source licenses if they do not do this.

Advantages of Having an Open Source

1. Larger developer support: Open source is helpful to develop the platform. It gives much larger support for developers and gives them a feeling of ownership as they can alter whatever they like.

2. Customizable: In a closed source scenario, developers are given only options to change what the original developer chooses, whereas open source lets the developers have full control and customize the look feel.
3. More secure: Open source is much more transparent than closed source. Anyone can look over the code. Having thousands of people reading through one's code, bugs and vulnerabilities are located much quicker and submitted for fixing. It also lets the user know if the bug has been fixed as one can check the code after each release.
4. Extended community support: As a product ages, the original developer might move on and stop developing, leaving the product to age with no new fixes or features, but if it is open, then usually the community takes over and continues working on it allowing the usable life of the product to be extended well beyond what the original developer intended.

4.3.1 Open Source Tools for IaaS

IaaS is a provision model in which an organization outsources the equipment used to support operations, including storage, hardware, servers, and networking components. The service provider owns the equipment and is responsible for housing, running, and maintaining it. The client typically pays on a per-use basis. The following subsections explain some of the open source tools available for providing IaaS in cloud computing.

OpenNebula

OpenNebula is a flexible tool that orchestrates storage, network, and virtualization technologies to enable the dynamic placement of services on distributed infrastructures. OpenNebula provides a modular architecture intended to be flexible. One of these modules is the scheduler, an interesting algorithm that places virtual machines (VMs) depending on their requirements. The client communication is also managed by modules that offer interfaces based on web services. It is perhaps the only open management platform that has invested into a tailorable VM placement algorithm. As such, it may provide a nice environment for those researchers seeking to compare and develop different resource allocation strategies. A limitation found with the OpenNebula is that, like XCP, their infrastructure assumes a classical cluster-like architecture with a front-end and without any redundant services. Figure 4.5 shows the infrastructure virtualization of OpenNebula.

OpenNebula is an open source management tool that helps virtualized data centers oversee private clouds, public clouds, and hybrid clouds. It combines existing virtualization technologies with advanced features for multitenancy, automated provisioning, and elasticity. A built-in virtual network manager maps virtual networks to physical networks. OpenNebula is vendor neutral, as well as platform agnostic and application programming interface (API) agnostic. It can use kernel-based virtual machine (KVM), Xen, or VMware hypervisor.

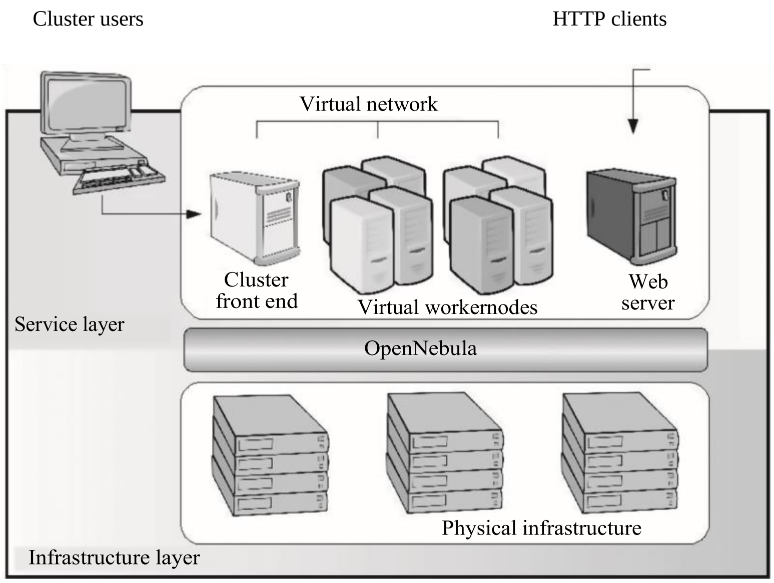


Figure 4.5: Infrastructure virtualization in OpenNebula. (Adapted from http://blog.dsa-research.org/?author=7&paged=2.)

Eucalyptus

Eucalyptus implements IaaS-style private and hybrid clouds. The platform provides a single interface that lets users access computing infrastructure resources (machines, network, and storage) available in private clouds— implemented by Eucalyptus inside an organization's existing data center— and resources available externally in public cloud services. The software is designed with a modular and extensible web services–based architecture that enables Eucalyptus to export a variety of APIs toward users via client tools. Currently, Eucalyptus implements the industry-standard Amazon Web Services (AWS) API, which allows the interoperability of Eucalyptus with existing AWS services and tools. Eucalyptus provides its own set of command line tools called Euca2ools, which can be used internally to interact with Eucalyptus private cloud installations or externally to interact with public cloud offerings, including Amazon EC2. Figure 4.6 represents the architecture of Eucalyptus Cloud. The main components in this are cluster controller, cloud controller, node controller, walrus storage controller, and storage controller:

1. Cluster controller (CC): It manages one or more node controllers and is responsible for deploying and managing instances on them. It communicates with node controller and cloud controller simultaneously. CC also manages the networking for running the instances under certain types of networking modes available in Eucalyptus.
2. Cloud controller (CLC): It is the front end for the entire ecosystem. CLC provides an Amazon EC2/S3 compliant web services interface

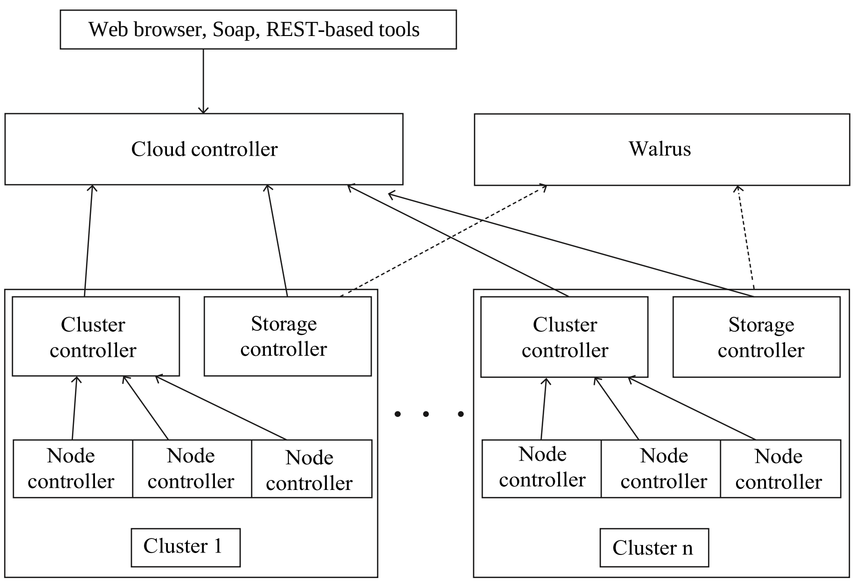


Figure 4.6: Eucalyptus Cloud architecture. (Adapted from http://mdshaonimran.wordpress.com/2011/ 11/26/eucalyptus-and-its-components/.)

to the client tools on one side and interacts with the rest of the components of the Eucalyptus infrastructure on the other side.

3. Node controller (NC): It is the basic component for nodes. It maintains the life cycle of the instances running on each nodes. It interacts with the OS, hypervisor, and the CC simultaneously.
4. Walrus storage controller (WS3): It is a simple file storage system. WS3 stores the machine images and snapshots. It also stores and serves files using S3 APIs.
5. Storage controller (SC): It allows the creation of snapshots of volumes. It provides persistent block storage over AoE or iSCSI to the instances.

While working over Xen and KVM hypervisors, Eucalyptus provides an open source solution to manage the virtual infrastructure of a cloud. The use of interfaces based on web services is one of their key characteristics, allowing native integration with Amazon services. Moreover, the hierarchical architecture is designed to reduce human intervention.

OpenStack

OpenStack, a cloud-computing project, aims to provide IaaS. It is a global collaboration of developers and cloud computing technologists producing the ubiquitous open source cloud computing platform for building public and private clouds. It delivers solutions for all types of clouds by being simple to implement, massively scalable, and feature rich. The technology consists of a series of interrelated projects delivering various components for a cloud infrastructure solution.

The goals of the OpenStack initiative are to support interoperability between cloud services and allow businesses to build Amazon-like cloud services in their own data centers. OpenStack, which is freely available under the Apache 2.0 license, is often referred to in the media as the Linux of the Cloud and is compared to Eucalyptus and the Apache CloudStack projects.

Figure 4.7 shows the architecture and project relationships of OpenStack. There are seven core components of OpenStack: compute, object storage, identity, dashboard, block storage, network, and image service.

1. Object storage allows user to store or retrieve files. The code name given for this is swift. The swift architecture is very distributed to prevent any single point of failure as well as to scale horizontally.
2. Image provides a catalog and repository for virtual disk images. These disk images are mostly commonly used in OpenStack Compute. The code name given for this is glance. Glance serves a central role to the overall IaaS picture. It accepts API requests for images (or image metadata) from end users or nova components and can store its disk files in the object storage service, swift.

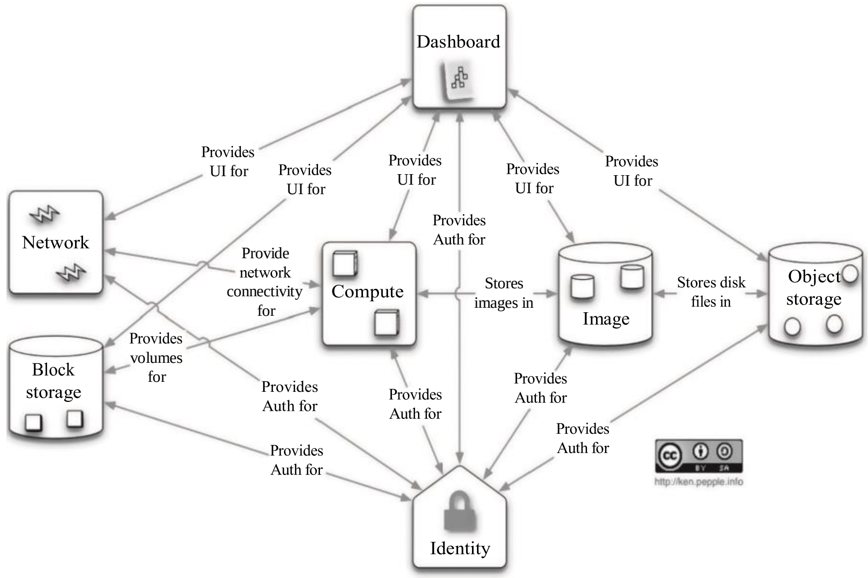


Figure 4.7: OpenStack conceptual architecture. (Adapted from http://ken.pepple.info/openstack/2012/09/ 25/openstack-folsom-architecture/.)

3. Compute provides virtual servers upon demand. The code name given for this is nova. Nova is the most complicated and distributed component of OpenStack. A large number of processes cooperate to turn end-user API requests into running VMs.
4. Dashboard provides a modular web-based user interface (UI) for all the OpenStack services. The code name given for this is horizon.
5. Identity provides authentication and authorization for all the OpenStack services. It also provides a service catalog of services within a particular OpenStack cloud. The code name given for this is keystone. Keystone provides a single point of integration for OpenStack policy, catalog, token, and authentication.
6. Network provides a network connectivity as a service between interface devices managed by other OpenStack services (most likely nova). The service works by allowing users to create their own networks and then attach interfaces to them. The code name given for this is neutron. Neutron interacts mainly with nova, where it provides networks and connectivity for its instances.
7. Block storage provides persistent block storage to guest VMs. The code name given for this is cinder. Cinder separates out the persistent block storage functionality that was previously part of OpenStack compute (in the form of nova volume) into its own service.

4.3.2 Open Source Tools for PaaS

PaaS is a category of cloud computing services that provides a computing platform for development of applications. PaaS offerings facilitate the deployment of applications without the cost and complexity of buying and managing the underlying hardware and software and provisioning hosting capabilities. It may also include facilities for application design, development, testing, and deployment. It also offers services such as web service integration, security, database integration, and storage. The following subsections give an overview of some of the open source tools that provide PaaS.

RedHat – OpenShift

OpenShift is Redhat's cloud platform which offers an application development environment for developers and provides two options for developers to interact with OpenShift such as through a web console or the command line client tools as well as developers can interact with OpenShift through an integrated development environment. So, this platform is highly customizable and offered in three forms: OpenShift Online (a public cloud-based hosting service), OpenShift Enterprise (a private PaaS that runs in your data center) and OpenShift Origin (the open source application hosting platform). In OpenShift, an application is a combination of code, configurations and application components called cartridges which include databases as well as languages and frameworks, and QuickStarts, which are applications with code and libraries already configured. So, developers work on applications and components. Multiple cartridges can run on the same server because cartridges are logically isolated from one another and it has gears and nodes to show the abstract structures that isolate components. With this, nodes hold gears that contain one or more cartridges. Even though, OpenShift have the ability of automating workflows which helps developers to scale automatically in order to handle peak workloads and to administer tasks like virtual server provisioning, configuration and scaling and supports git repositories for code management; it Works well with Git, but non-Git deployments might require additional steps [54, 56].

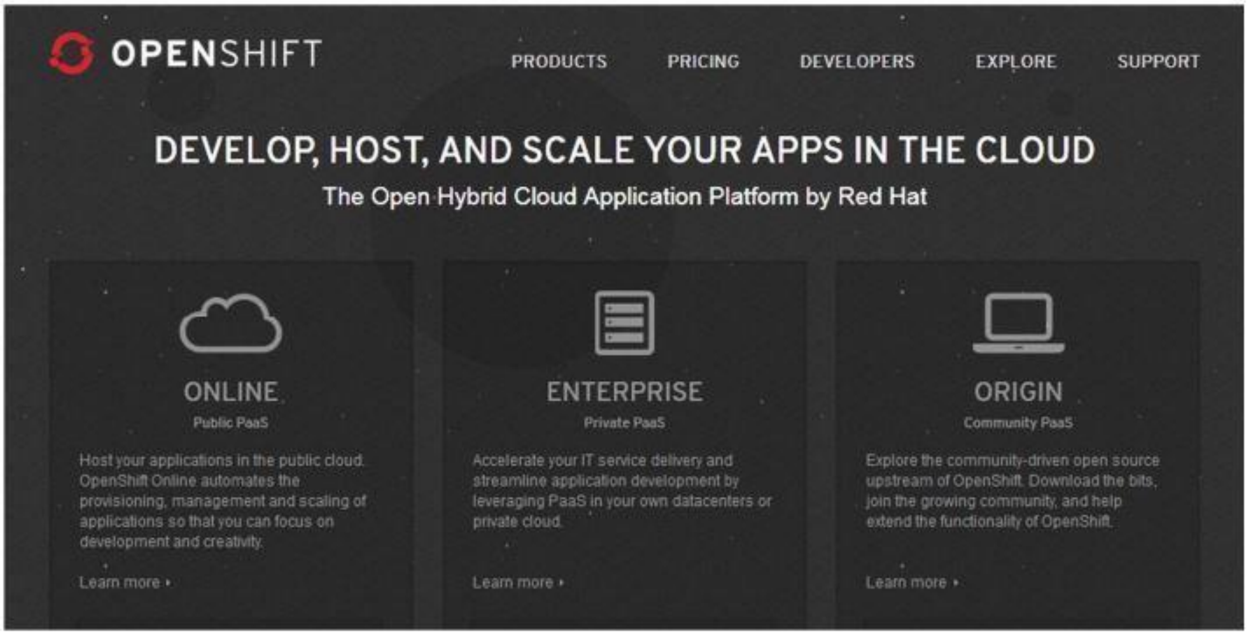


Figure 4.8: The three OpenShift offerings

Red Hat OpenShift Origin

OpenShift Origin is the open source upstream of OpenShift, the next- generation application hosting platform developed by Red Hat. This is also known as Red Hat's PaaS, OpenShift takes care of infrastructure, middleware, and management. OpenShift Origin includes support for a wide variety of language runtimes and data layers including Java EE6, Ruby, PHP, Python, Perl, MongoDB, MySQL, and PostgreSQL.

OpenShift Origin platform has two basic function units: broker and node servers. Communication between these units is through message queuing service. The broker is the single point of contact for all application management activities. It is responsible for managing user logins, dynamic shutdown (DNS), application state, and general orchestration of the applications. Customers do not contact the broker directly; instead, they use the web console, command line interface (CLI) tools, or the JBoss Tools integrated development environment (IDE) to interact with the broker over a REST-based API. The node servers host the built-in cartridges that will be made available to users and the gears where user applications will actually be stored and served. MCollective client running on each node is responsible for receiving and performing the actions requested by the broker. OpenShift Origin supports several built-in cartridges based on the most popular app development languages and databases. In order for these to work, the underlying technology must be installed on every node server in an Origin system. A gear represents the slice of the node's CPU, RAM, and base storage that is made available to each application. OpenShift Origin supports multiple gear configurations, enabling users to choose from the various gear sizes at application setup time. When an application is created, the broker instructs a node server to create a new gear to contain it. Whenever a new gear is created on a node server, CPU and RAM shares are allocated for it and a directory structure is created.

4.3.3 Open Source Tools for SaaS

SaaS is a software distribution model in which applications are hosted by a vendor or service provider and made available to customers over a network, typically the Internet. SaaS is becoming an increasingly prevalent delivery model as underlying technology that support web services and mature service-oriented architecture (SOA) and new developmental approaches, such as Ajax, become popular. The following subsections briefly explain some of the open source tools for SaaS.

Apache VCL

VCL stands for virtual computing lab. Apache VCL is an open source solution for the remote access over the Internet to dynamically provision and reserve computational resources for diverse applications, acting as SaaS solution.

VCL has a simple architecture with four main components: web portal, database, management node, and compute node. Figure 12.4 shows the architecture of VCL.

Web server represents the VCL portal and uses Linux/Apache/PHP solution. This portal provides a UI that enable the requesting and management of VCL resources.

Database server stores information about VCL reservations, access controls, and machine and environment inventory. It uses Linux/SQL solution.

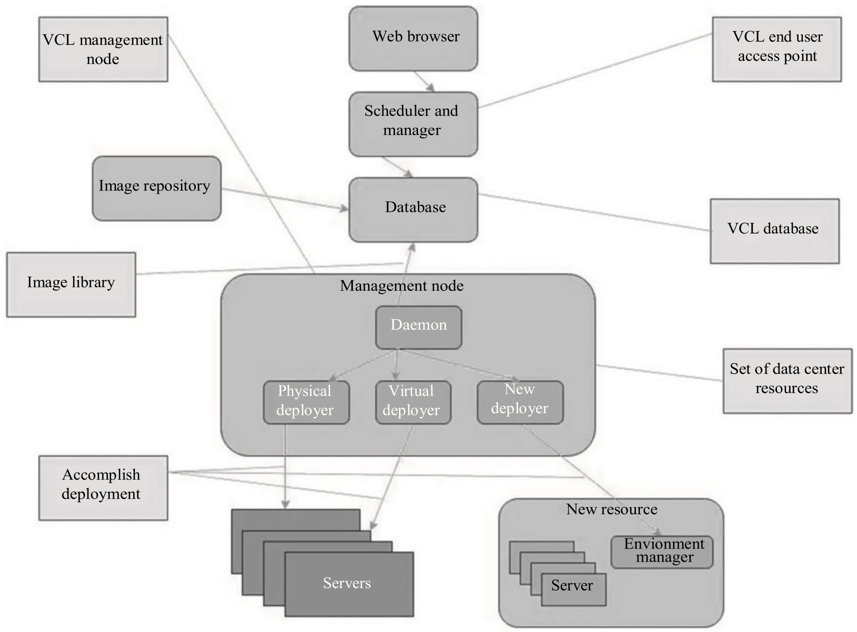


Figure 4.9: Architecture of VCL. (Adapted from http://vcl.apache.org/info/architecture.html.)

Management node is the processing engine. A management node controls a subset of VCL resources, which may be physical blade servers, traditional rack, or VMs. It uses Linux/VCLD (perl)/image library solution. VCLD is a middleware responsible to process reservations or jobs assigned by the VCL web portal. According to the type of environment requested, VCLD should assure that service (computational environment) will be available to user.

Compute nodes include physical servers, VMs, computing lab machines as well as cloud compute resource.

Conceptual overview of Apache VCL is given in Figure 12.5. Remote users connect to the VCL Scheduling Application (the web VCL portal) and request access to a desired application environment. The application environment consists of an operating system and a suite of applications. The computer types are machine room blade servers, VMware VMs, and stand-alone machines.

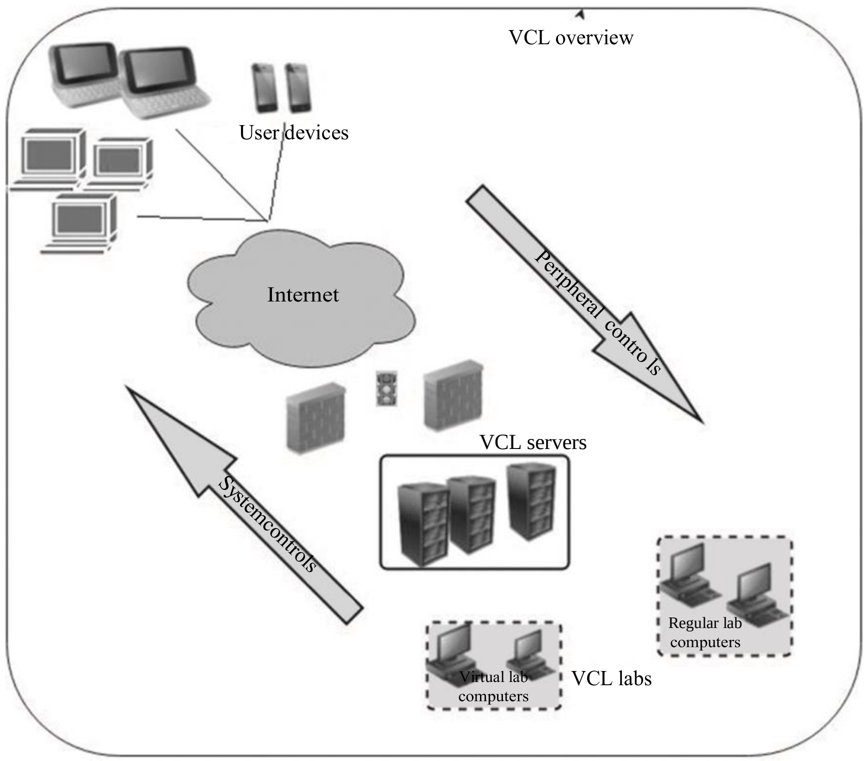


Figure 4.10: Apache VCL conceptual overview. (Adapted from https://cwiki.apache.org/confluence/ display/VCL/Apache+VCL.)

Google Drive

Google Drive is a file storage and synchronization service provided by Google that enables user cloud storage, file sharing, and collaborative editing. Files shared publicly on Google Drive can be searched with web search engines.

Google Drive lets the user store and access files anywhere—on the web, on the hard drive, or on the go. It works as follows:

- Go to Google Drive on the web at drive.google.com.
- Install Google Drive on the computer or mobile device.
- Store files in Google Drive. It is available on the device from which it is accessed.

After doing so, user will be able to access files from anywhere he or she wants to. If a file is changed on the web, by using a computer, or a mobile device, it is updated on every device where Google Drive is installed.

Google Docs

Another SaaS offering by Google is Google Docs. It is one of the many cloud computing document-sharing services. The majority of document-sharing services require user fees, whereas Google Docs is free. Its popularity among businesses is growing due to its enhanced sharing features and accessibility.

In addition, Google Docs has enjoyed a rapid rise in popularity among students and educational institutions.

Google Cloud Connect is a plug-in for Microsoft Office 2003, 2007, and 2010 on Windows that can automatically store and synchronize any Microsoft Word document, PowerPoint presentation, or Excel spreadsheet to Google Docs in Google Docs or Microsoft Office formats. The Google Doc copy is automatically updated each time the Microsoft Office document is saved. Microsoft Office documents can be edited offline and synchronized later when online. Google Cloud Sync maintains previous Microsoft Office document versions and allows multiple users to collaborate by working on the same document at the same time.

Google Spreadsheets and Google Sites also incorporate Google Apps Script to write code within documents in a similar way to Visual Basic for Applications (VBA) in Microsoft Office. The scripts can be activated either by user action or by a trigger in response to an event.

Google Forms and Google Drawings have been added to the Google Docs suite. Google Forms is a tool that allows users to collect information via a personalized survey or quiz. The information is then collected and automatically connected to a spreadsheet with the same name. The spreadsheet is populated with the survey and quiz responses.

Google Drawings allows users to collaborate creating, sharing, and editing images or drawings. Google Drawings contains a subset of the features in Google Presentation (Google Slides) but with different templates.

Dropbox

Dropbox is a file hosting service operated by Dropbox, Inc. that offers cloud storage, file synchronization, and client software. Dropbox allows users to create a special folder on each of their computers, which Dropbox then synchronizes so that it appears to be the same folder (with the same contents) regardless of which computer is used to view it. Files placed in this folder also are accessible through a website and mobile phone applications.

Dropbox provides client software for Microsoft Windows, Mac OS X, Linux, Android, iOS, BlackBerry OS, and web browsers, as well as unofficial ports to Symbian, Windows Phone, and MeeGo.

Both the Dropbox server and desktop client software are primarily written in Python. The desktop client uses GUI toolkits such as wxWidgets and Cocoa. Other notable Python libraries include Twisted, ctypes, and pywin32. Dropbox ships and depends on the librsync binary-delta library (which is written in C).

The Dropbox client enables users to drop any file into a designated folder that is then synchronized with Dropbox's Internet service and to any other of the user's computers and devices with the Dropbox client. Users may also upload files manually through a web browser.

Dropbox client supports synchronization and sharing along with personal storage. It supports revision history, so files deleted from the Dropbox folder may be recovered from any of the synced computers. Dropbox supports multiuser version control, enabling several users to edit and repost files without overwriting versions. The version history is by default kept for 30 days, with an unlimited version called Pack-Rat available for purchase.

Dropbox also provides a technology called LAN sync, which allows computers on a local area network to securely download files locally from each other instead of always hitting the central servers.

Comparison and Analysis

Based on the above comparison and analysis, we have used Redhat‟s OpenShift for our study to implement the prototype of the proposed architecture. In OpenShift, Basic systems administration tasks such as virtual server provisioning, configuration, and scaling are automated. In addition, OpenShift allows a wide variety of choice of environments (UI, command line, IDE), flexibility and accessibility of different programming languages including PHP will benefit us greatly. As a result, this is the best platform for us to spend more of our time on code and less time on setting up development environments such as configuring operating systems and installing libraries and packages. As we have discussed in section 4.3.2 of this chapter, OpenShift is offered in three forms or versions and our focus is on OpenShift online development platform which is public platform and the version used by startup program freely. It creates a file system layout and provides disk space, CPU resources, memory, network connectivity, and an Apache or JBoss server to create, deploy, and manage applications in the cloud. Hence, we can easily and quickly develop our capacity building training system prototype, we can push the code to build and run on OpenShift through the git (software version control solution), then OpenShift online orchestrates where application containers will run and manages our application to ensure its availability for end users, that means it creates universal and unique URL for our application in order it to be accessible globally.

4.4 OpenShift Online

4.4.1 Overview

OpenShift Online by Red Hat is a Platform as a Service (PaaS) that provides developers and IT organizations with an auto-scaling, cloud application platform for deploying new applications on secure, scalable resources with minimal configuration and management overhead. Also, it supports a wide selection of programming languages and frameworks [61].

Table 4.3: The features and benefits of OpenShift Online [62]

Features	Benefits
Self-Service and OnDemand Application Stacks	By enabling Developers with the ability to quickly and easily deploy application stacks, the OpenShift platform can increase productivity and foster innovation in application design and delivery.
Standardized Developer Workflows	With OpenShift, the Application Development organization can standardize the Developer workflow and create repeatable processes for application delivery in order to streamline the entire process.

Polyglot – Choice of PL and frameworks	Developers can choose between Java, Ruby, Python, PHP, Perl and other languages on the OpenShift platform allows them to choose the right tool for the job, and make a different choice for each project as needed.
Built-in Database Services	Choice of database technologies available in OpenShift automatically connected into the application stacks as needed, relational and modern NoSQL data stores.
Extensible cartridge	Developers can add other language, database, or middleware components that they need via the customizable OpenShift Cartridge system.
Multiple interaction modes	Provides multiple interaction models with the PaaS enabling developers to use a rich set of command-line tools, a powerful multi-device web console, or an Eclipsebased Integrated Development Environment.
One Click Deploy	Deploying to the OpenShift platform is as easy a clicking a button or entering a "Git push" command.
Automatic Application Scaling	The OpenShift platform enables cloud elasticity by providing automatic horizontal application scaling as application load increases.
Source Code Version Management	The OpenShift platform includes the Git distributed version control and source code management system.

4.4.2 Gears and Cartridges

OpenShift Online have two basic components called Gears and Cartridges [61]. Table 4.4 describes the basic system components of OpenShift Online. A gear is a server container with a set of resources that allows developers to run their applications and it run on OpenShift in the cloud. Currently, there are three gear sizes available with OpenShift Online: small, medium, and large. Each size provides 1 GB of disk space. The large gear has 2 GB of RAM, the medium gear has 1 GB of RAM, and the small and small gears have 512 MB of RAM. By default, there are three small gears available with a total of 1.5GB of RAM and 3GB of disk space. OpenShift Online can assign these three gears to a single application and its cartridges.

Cartridges are the plug-ins that houses the framework or components that can be used to create and run an application and used to support various programming languages, databases, monitoring services, and management. One or more cartridges run on each gear, and the same cartridge can run on many gears for clustering or scaling. There are three kinds of cartridges: web/standalone cartridges, Add-on/embedded cartridges and downloadable cartridges.

Web/standalone cartridges are the languages, frameworks or application servers that are set up to serve the web content, such as JBoss, Tomcat, Php, Python, or Node.js. Having one of these cartridges is sufficient to run an application. Add-on/embedded cartridges provides extra functionality and capabilities to application, such as a database like MySQL, MongoDB and PostgreSQL and Monitoring and Management like PhpMyAdmin, HAProxy, Cron, Switch Yard, RockMongo and Jenkins Client, but cannot be used on its own to create an application. Downloadable cartridges are available for new and existing applications along with the supported standard OpenShift Online cartridges. These are custom cartridges created by users, or available from the OpenShift community. These cartridges are downloaded and installed using the URL to the manifest of the hosted downloadable cartridge.

Table 4.4: OpenShift Online system components [61]

System Component	Description
Gears	Resource-constrained containers for application code where cartridges run. Gears determine the amount of RAM and disk space available to a cartridge.
Cartridges	Cartridges provide the functionality to run applications. Numerous cartridges are currently available to support languages such as Perl, PHP, and Ruby, as well as many database cartridges, such as PostgreSQL and MySQL.

Table 4.5: OpenShift Online application components [61]

Component	**Description**

Domain	The domain provides a unique group identifier for all the applications of a specific user. The domain is not directly related to DNS; instead, it is appended to the application name to form a final application URL of the form http://*App_Name-domain.example.com*
Application Name	The name of the application is selected by a user. The final URL to access the application is of the form http://*App_Name-domain.example.com*
Alias	DNS names can be provided for the application by registering an *alias* with OpenShift Online and pointing the DNS entry to the OpenShift Online servers.
Git repository	A Git repository is used to modify application code locally. After the code is applied, the **git push** command is required to deploy the revised code.

4.4.3 Developer Interaction

As outlined in section 4.2.1 of this chapter, there are two ways to interact with OpenShift: the Management Console or the Client Tools [63]. The OpenShift Online Management Console is a graphical interface accessed with a web browser and is best suited for:

- Setting up
- Administering and managing accounts
- Launching new applications
- Managing and monitoring applications.

While, the OpenShift Online CLI tools/rhc client tools are used to manage a cloud environment using a command line interface, and provide features that are not currently available in the management Console. The client tools are best suited for:

- Coding
- Debugging
- Advanced application management.

To work with OpenShift online client tool, we must install the rhc client tool in our machine and because we have developed this thesis work using windows operating systems; we have described the steps of installing OpenShift Online client tools on Windows operating systems. There are three general steps, these are: step1: installing Ruby programming language step2: installing Git software and step3: installing the client tool. The first two steps are pre-request software to install the rhc client tool. So before going to the client tool installation, we must install the two required software Ruby and Git. Git is a software configuration management (SCM) that keeps track of changes to software source code. With git, we can manage the OpenShift application source repository by running three simply commands: git add, git commit and git push.

CHAPTER FIVE

CLOUD BASED SERVIECES FOR HEALTHCARE INSTITUTIONS IN ETHIOPIA

5.1 Building the Framework

This chapter describes the framework of resource sharing for Ethiopian healthcare institutions. Due to different factors, it is true that ICT usage in Ethiopian Healthcare Institutions in its current situation is very primitive, as discussed in previous sections. In order to design this framework to provide sharing of resources as a service, we have taken inspiration or concepts from IBM [64], NIST cloud reference architecture [65] and available literatures on software framework architecture approaches for cloud based systems. It identifies common architectural patterns for each cloud adaption pattern which describes the technology that underlie each type of cloud implementations. The reason why we have used these two-reference framework architecture is:

1. it shows clear way of how we construct the architecture across all deployment models by providing the steps we follow and the required components which saves time.
2. It meets necessary cloud requirements like elasticity, self-service and flexible sourcing.
3. It exploits comprehensive architectural principles to ensure design scalability for efficiency, security, service management and virtualization as well as to reduce errors throughout the entire development process.

Also, we have used graphical elements or symbolic representations of icons for cloud applications from the work of Christophe F. et al. [66], which presents a collection of patterns to describe cloud computing, its service models, its deployment models and its architectural patterns in general. As a result, they have mentioned three types of cloud application architectural patterns such as elasticity pattern, availability pattern and multi-tenancy patterns. They have clearly discussed their context, challenges of each type of patterns and they have put the solution for each type of pattern with proof of concept by: sketching these patterns; analyzing their results and relating with other patterns which affect them.

Furthermore, our work intends to introduce cloud based resource sharing framework for healthcare institutions with two essential parts: the front-end and back-end; that are usually connected via internet. The first part is to illustrate the different health delivery institutions and client's infrastructures like computers, laptops, mobile phones as well as the network that is used for accessing the cloud resource sharing system. Simply, the first part consists of the visible interface and applications (example web browser like chrome, IE or Mozilla Firefox) required to access resource sharing system from the cloud providers. The second part is to illustrate the healthcare program basic components and services which is the cloud part of the framework. It consists of all the required resources to access and share capacity building resources to the end users. So, it includes various servers, middleware, storages, virtual machines, programming languages, platforms, security and management mechanisms, deployment models and other pool of resources.

5.2 Design Goals and Constraints

The overall goals of cloud based capacity building resource sharing architecture design is to provide resource sharing system which meets the needs of healthcare workers related to maximize their potential. As a result, the architecture should be flexible and easy to use and learn with low cost, increased productivity, well-defined interface and user-friendliness.

Constraints are a limiting factor and severely restrict options for making design decisions. In software architecture design, there are two types of constraints technical and non-technical or business constraints [67]. In this study, the following technical and non-technical constraints were considered. Technical constraints include: Programming language: - php programming language to develop the prototype of the architecture due to the following reason. Easy to code and debug and more templates are available freely to use as a starting resource also we are more familiar with php than other languages. Operating system or platforms supported: this study works properly on windows and Android operating systems with the support of OpenShift Online Platform. Non-technical constraints include: Schedule: - this is the time frame for final delivery of the thesis work, so we have stuck with our schedule during this work. Budget: this is the money provided to accomplish the work which is fixed in this level, so it restricts us from collecting the required data and tools extensively. Software licensing restrictions: - we are restricted to use freely available software and tools to conduct this thesis work which lacks full features. Internet-based: - all cloud based applications are working properly on dedicated internet connection; due to this, slow internet connection may affect the accessibility or availability of the system.

5.3 General Cloud Framework of Resource sharing for Healthcare Institutions

The core function of designing cloud based resource sharing framework for healthcare institution is to create global healthcare knowledge sharing network to have resource sharing center for all health workers and at all level. Hence it facilitates the access of tools, approaches and resources that are dispersed in different healthcare delivery from one place. In addition, the process of designing cloud based resource sharing framework for healthcare institution will be progressive. Hence, according to the present condition of the program, cloud computing expertise and OpenShift online, the design of cloud based resource sharing framework for healthcare institution can familiarize the following generic cloud framework shown in figure 5.1. It contains the below mentioned components.

Physical resource pool: it includes hardware resources such as network, storage equipment and virtual server. These dispersed resources are connected together through network virtualization with the intension of creating cloud centric healthcare knowledge sharing network and shared capacity building tools and materials.

Infrastructure as a service (IaaS): The healthcare delivery organizations and end users can share full hardware resources by virtualizing physical resources. Therefore, the virtualization technology increases the user need by dynamically allocating resources to the active need so it avoids the usage of fixed storage space. It is fully managed by the cloud provider and it manages resource and environment of the physical infrastructures like server, storage and networking to the virtual machine.

Platform as a service (PaaS): It contains the OpenShift platform which will control the development environment to enable creation and hosting of applications and programs with the ability to access public clouds resources and platforms.

Software as a Service (SaaS): The end users' orders application software service that are deployed by the platform service providers through the network. The service providers gather payments based on used services. And different services such as sharing of workplace training resources, research capability training resources, shared experience, shared tools and materials management and others can be provided on this level.

OpenShift Online: As we have described in chapter four, it is Redhat‟s OpenShift online offering. This platform is used to create and deploy capacity building resource sharing system.

On the capacity building provider‟s (General hospital) center, there are individual level, community/beneficiary level, organization level and system level capacity building components which will adapt different approaches and tools.

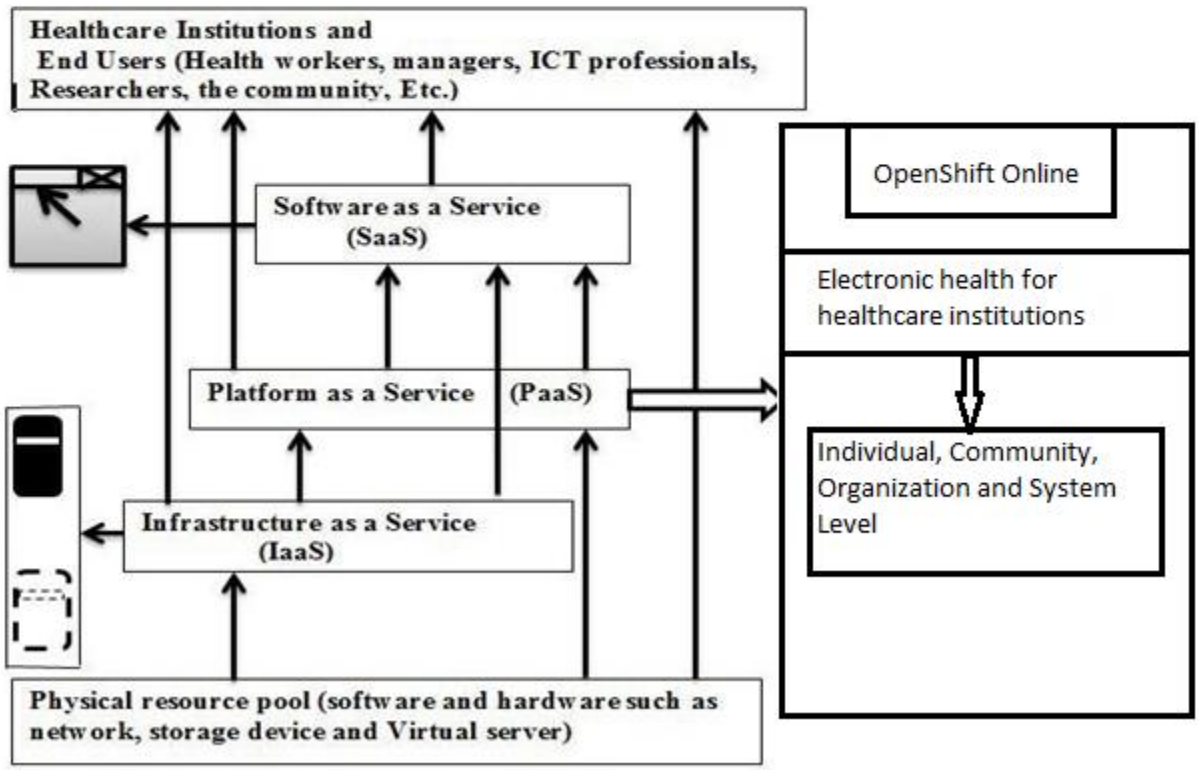


Figure 5.1: General cloud framework of resource sharing for healthcare institutions

5.4 The Developed Architectural Solution for Healthcare Institution

Based on the above stated concepts, the description of client and cloud side parts and the general framework shown in Figure 5.1, this study suggests the following architecture to implement cloud computing as a tool to provide, access and share resource sharing. Furthermore, it is designed to realize the tools sharing and to provide resource sharing services for the healthcare practitioners working in the different health sector of Ethiopia.

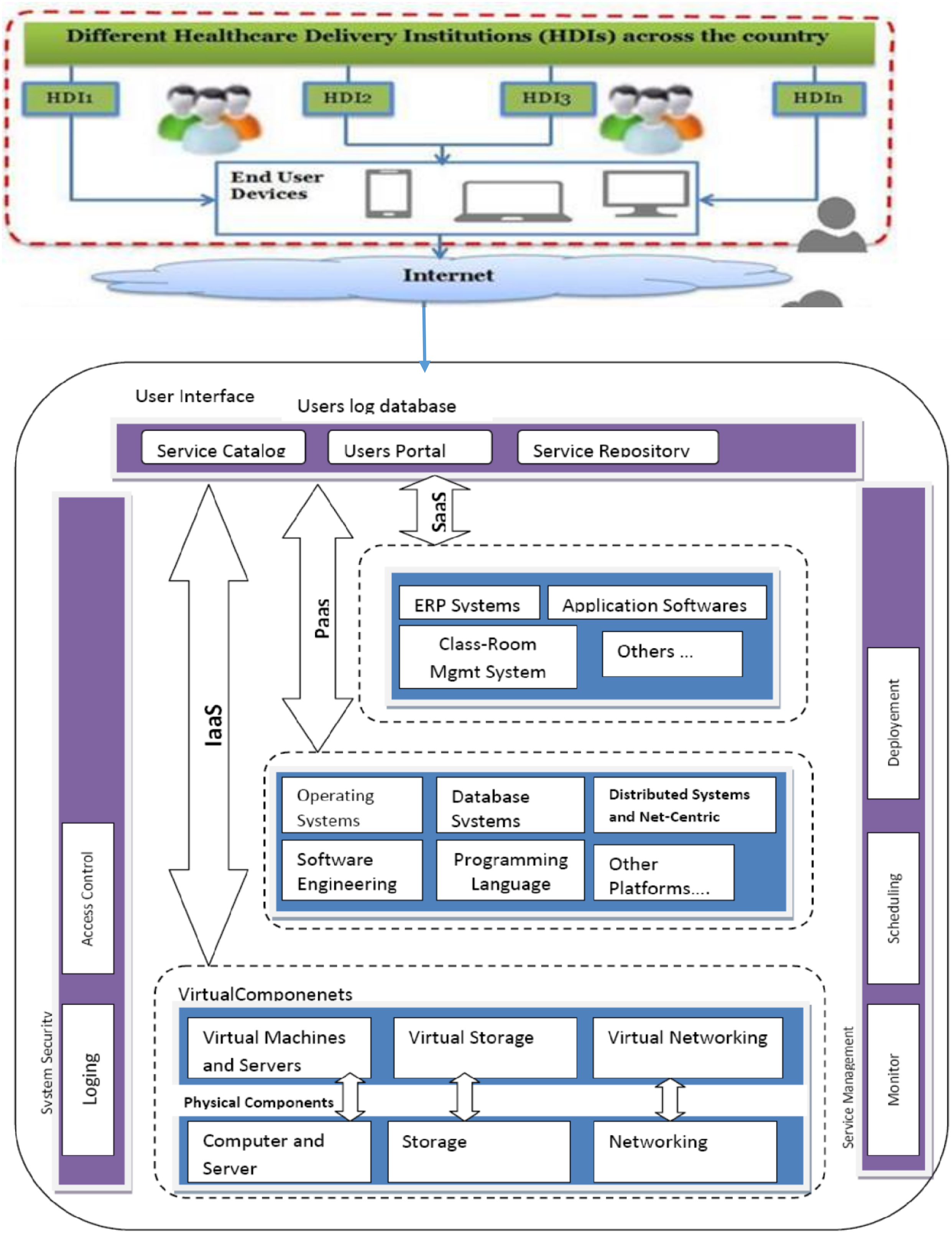


Figure 5.2: Conceptual cloud computing framework of resource sharing for healthcare institutions

The proposed framework integrates healthcare delivery institutions and cloud using dedicated internet connection as shown in fig 5.2 and it contains different layers and modules to manage the user access and service delivery. In the community cloud, the users can request any available services from the cloud provider. The details of layers are described below.

5.4.1 User Interface Layer

The user interface layer is the top layer of the cloud computing framework of resource sharing for healthcare institutions which is connected with the front-end part of the architecture via internet and act as an interface between the end users and the cloud services. It provides support for the resource sharing system to handle request from multiple types of end user devices, tracking requests to the correct end users and adjusting materials and resources according to the specific need of the users. So, it is also used for authentication and authorization mechanism. It contains two subcomponents. User Log which is used to store user information and their access level such as user name, password, user type and others. Also, it manages and supports the resource utilization by keeping notes about the resource requirement and their status; Service Repository which contains different services and their detail.

5.4.2 Software as a Service (SaaS) Layer

SaaS layer is the second layer of the architecture below the generic user interface layer which allows access to the system and any third-party application software to run on the cloud platform. Services provided in the PaaS layer can be utilized by sending request messages through Web APIs. So, healthcare institutions and users can access resource sharing system and other third-party software through web interface.

5.4.3 Platform as a Service (PaaS) Layer

PaaS layer is the third layer of the architecture which is deployed on cloud provider that can support requirement of the designed platform.

Service Layer

It provides basic services such as development environment like operating system and programming language, operational support, database management system, user management, middleware and other platforms.

5.4.4 Infrastructure as a Service (IaaS) Layer

IaaS Layer is the bottom layer of the framework in which the cloud based resource sharing system to implement. It contains the physical and virtualized infrastructure including healthcare delivery institution's ICT Infrastructures.

5.5.5 Security and Management Mechanisms

Security is the capability of a system to prevent accidental actions outside of the designed usages which is still the big issue. In cloud computing, there are a lot of security mechanisms to protect data and the whole system holding and processing it. Out of these mechanisms the following are very common and used almost in all cloud service providers in addition to their own security methods [68]. The mechanisms are Encryption, Hashing, Digital Signature, Public Key Infrastructure (PKI), Identity and Access Management (IAM) and Single Sign-On (SSO). IAM controls and tracks user identities.

These mechanisms usually provide unified APIs and can be offered as single result, custom applications, merged into different output sets or multi-function applications [69].

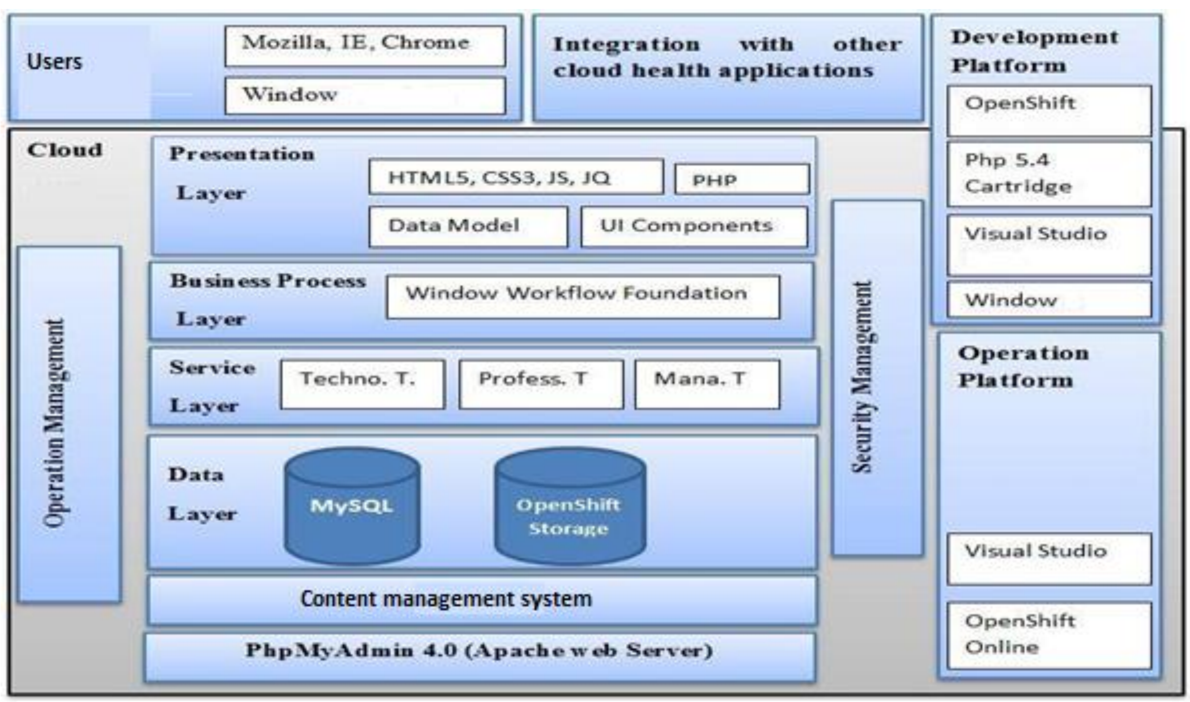


Figure 5.3: Overview of development view of the E-Health system

Figure 5.3 illustrates development view of the proposed framework. In the E-Health system, the users can access the system using different operating systems and web browsers. The UI of the system was created using HTML, CSS, JS, IQ and php as a main language, we have considered window workflow foundation for business process management. The integration of this system with other current healthcare system can be operated using JSON or other programing technologies. Different services including technological training resources, professional training resources, managerial training resources and other capacity building resources was included as a service. For the purpose of managing and storing the resources, we have used MySQL database, OpenShift storage, web content management and apache server. The development platform is OpenShift online with php5.4 cartridge; the different views of the architecture was developed using visual studio and ArgoUML.

CHAPTER SIX

DEVELOPMENT OF THE PROTOTYPE

In this chapter, we provide overview of the prototype implementation of the suggested architecture elaborated in chapter five. With this end, we are focusing on the implementation of some parts of the capacity building resource sharing system; that is the front end web application interface development on OpenShift online cloud platform, smart way of sharing technological training resources, professional and managerial resources and E-Discussion at individual, organization, system and community level, searching and accessing of these shared resources to improve medical practitioners ability, research and patient care, the accessibility of this web based interfaces from the OpenShift providers using URL of the system, workability of the linkage in between each menus, the portability or mobile and tablet device interface looking and desktop or laptop layout of the developed prototype and other important features for proof of concept . To implement the prototype of the architecture and to elaborate implementation details, the 4+1 view model of software architecture as an architecture description approach, which have presented by Kruchten [70] are used.

In the first sub portion of this chapter, we have described architecture description approach followed by use case of the system. Then we have described different view of the healthcare capacity building resource sharing system according to 4+1 view model of software architecture [70]. Finally, we have explained implementation details, testing and findings and discussions.

6.1 The Architecture of the System

Software architecture description is the collection of ways for expressing, communicating and analyzing the architecture, often arranged into various views of the architecture. Using these views to describe the architecture of software-intensive systems, Kruchten have created the 4+1 view model of software architecture [70]. The views are logical, development, process and deployment and the scenario or use case that indicates the plus one dimension which are briefly explained as follows.

Logical View: This view represents the functionality of the system which is provided to the users as a service. It can be presented by sequence diagrams, class diagrams and collaboration diagrams.

Development View: This view denotes the details and organization of modules and subsystems or library that makes the system and concerned with software management. It illustrates the system from programmer point of view which is represented by component diagrams and package diagrams.

Process View: This view considers some non-functional requirements to describe the dynamic behavior of the system by elaborating activities of the system using UML activity diagrams.

Deployment View: This view represents the software to hardware mapping which is the physical topology of the system and represented by UML deployment diagrams.

Use Cases: This dimension is denoted using UML use case diagrams and represents sequence of interaction between system's features and their actors. We have chosen specific parts of the capacity building training architecture to provide implementation details of different services of the system and interaction between applications and services that are provisioned using the system.

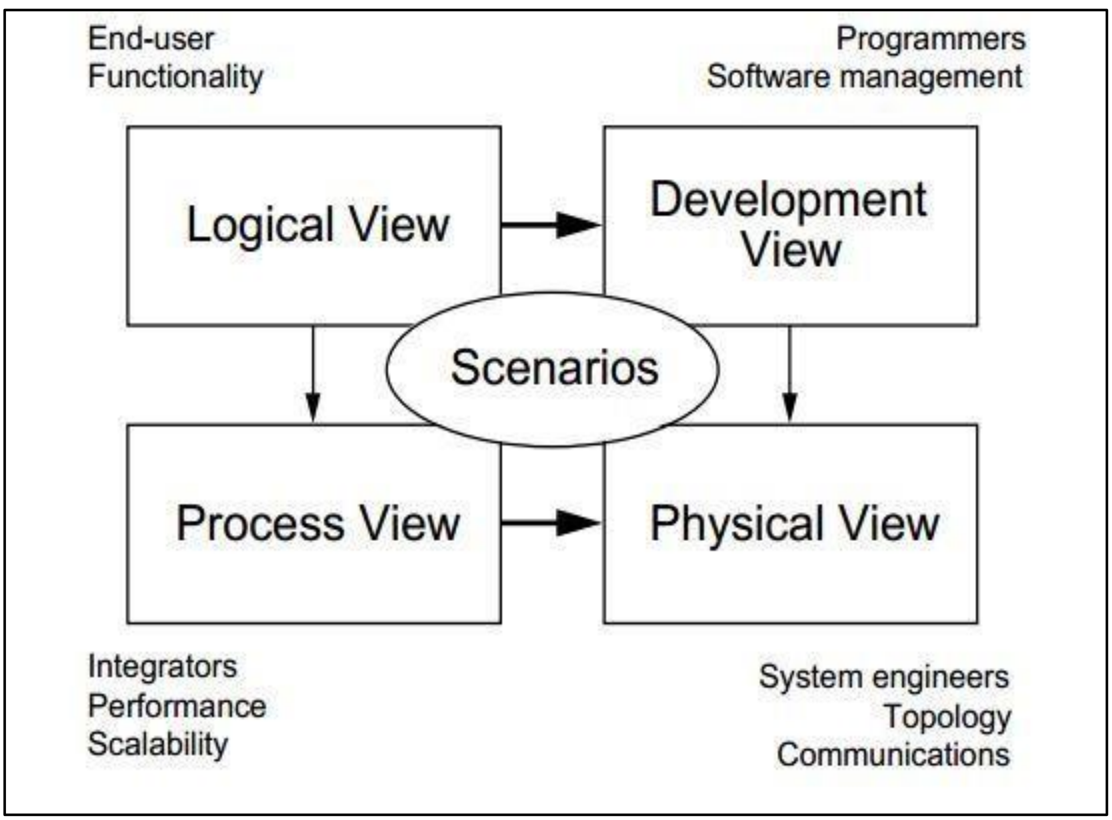
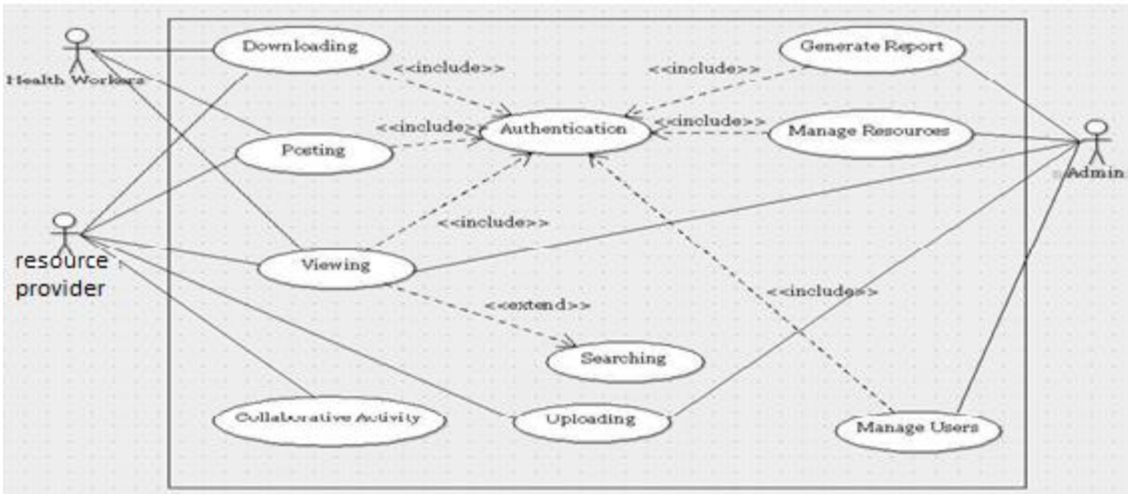


Figure 6.1: Kruchten "4+1" View Model [70]

6.2 Use Cases

Selective use cases of the healthcare resource sharing system are shown in Figure 6.1.

The following figure 6.2 incorporated most important use cases of the prototype implementation.



Authentication

Use-Case ID: UC-1

Use-Case Name: Authentication

Description: Before any approaches and their associated services provided to the end users from the OpenShift online cloud service provider, these need to be authenticated by the provider. Upon requesting the system services, the system asks the users to register. After registering to the system, the user will have User Name and password which is used to access the requested services.

Pre-Condition: Open the healthcare resource sharing system web interface by writing the URL on the browser.

Post Condition: User added to the system and has user name and password.

Actor: Users

Basic course of action:

1. The user indicates, he wants login

2. The system displays login screen

3. The user enters username and password

4. The system verifies whether the user is authorized or not

5. The system displays main features

6. The use case ends

Posting and Retrieving Comments, Experiences and other Activities

Use-Case ID: UC-2

Use-Case Name: posting

Use Case Description: The user comment, experience and other activities submitted to the site should be able to post for other peers.

Pre-Condition: Login in to the system

Post Condition: The comments and other posted resources are pushed as they created.

Includes: “UC-1: Authorization” Actor: Health workers Basic course of action:

1. The user indicates, he wants posting and retrieving comments, experiences and other activities

2. The system displays the requested features

3. The user posts and retrieves the data

4. The use case ends

Downloading Resources

Use-Case ID: UC-3

Use-Case Name: downloading resources

Use Case Description: The sharing resources should support the viewing and downloading of provisioned resources to utilize the features of the system.

Pre-Condition: The available resources provided by the cloud service provider must authenticate users

Post Condition: After successful verification, viewing and downloading of the corresponding cloud resources is done.

Includes: “UC-1: Authorization”

Actor: Health workers Basic course of action:

1. The user indicates, he wants to download resources

2. The system displays the requested features

3. The user downloads the resources

4. The use case ends

Searching the Available Materials, Tools and Approaches

Use-Case ID: UC-4

Use-Case Name: Searching

Use Case Description: This use case used to find the resources without wasting more time by visiting the entire list of resources to select their required one.

Pre-Condition: User inputs the keyword (name or title) to find the available resources Post Condition: The searched resources will be displayed for the users.

Includes: “UC-1: Authorization”

Actor: users

Basic course of action:

1. The user indicates, he wants to search resources

2. The system displays the requested features

3. The user searches the resources

4. The searched resources viewed by users

5. The use case ends

Collaborative Activities

Use-Case ID: UC-5

Use-Case Name: Collaborative Activities

Use Case Description: The Resource sharing system provides collaboration among the healthcare practitioners throughout the country and with outside world.

Pre-Condition: The user creates collaboration group

There will be collaboration among the healthcare practitioners throughout the country and with outside world.

Actor: Health workers and resource providers Basic course of action:

1. The user indicates, they want to collaborate

2. The system displays the requested features

3. The user collaborates by joining collaboration group

4. The use case ends

Viewing Resources

Use-Case ID: UC-6 Use-Case Name: Viewing

Use Case Description: This use displays all the available resources for the users.

Pre-Condition: The user searches the resources.

Post Condition: The resources displayed to users.

Includes: “UC-1: Authorization”

Extends: “UC-4: Searching”

Actor: users

Basic course of action:

1. The user indicates, he wants to view resources

2. The system displays the requested features

3. The user opens the resources

4. The opened resources viewed by users

5. The use case ends

Uploading Resources/Materials, Tools and Approaches

Use-case ID: UC-7

Use Case Name: Uploading

Use Case Description: The capacity building providers prepares the required resources to share and categorizes these resources and tools for ease of accessibility. During this process the resource providers share the resources using the system and these resources should be provisioned to the remote servers.

Pre-Condition: Login to the system

The resources are available online.

Includes: "UC-1: Authorization"

Actor: users

Basic course of action:

1. The user indicates, he wants to share resources

2. The system displays the requested features

3. The user shares/uploads the resources

4. The use case ends

Generate Report

Use-case ID: UC-8

Use Case Name: generate report

Use Case Description: The administrator of the resource sharing system prepares the report based of the available resources and its accessibility. During this process, the administrator generates the reports using the system and these reports should be provisioned to the remote servers.

Pre-Condition: Login to the system

Post Condition: The reports are available online.

Includes: “UC-1: Authorization”

Actor: Admin

Basic course of action:

1. The Admin indicates, he wants to wants to generate report

2. The system displays the requested features

3. The admin generates the report

4. The use case ends

Managing Resources

Use-case ID: UC-9

Use Case Name: Managing users

Use Case Description: The administrator of the resource sharing system manages the resources using the provided interface for resource managing purpose. The administrator of the system can delete or deactivate the resources if the resources are unwanted.

Pre-Condition: Login to the system

The resources are deleted or de-activated from the system.

Includes: “UC-1: Authorization”

Actor: Admin

Basic course of action:

1. The Admin indicates, he wants to manage resources

2. The system displays the requested features

3. The admin delete or update the resources

4. The use case ends

Managing Users

Use-case ID: UC-10

Use Case Name: Managing users

Use Case Description: The administrator of the resource sharing system manages the users using the provided interface for user management purpose. The administrator of the system can delete or deactivate the users if the users post unwanted resources.

Pre-Condition: Login to the system

Post Condition: The users de-activated or deleted from the system.

Includes: “UC-1: Authorization”

Actor: Admin

Basic course of action:

1. The Admin indicates, he wants to manage users

2. The system displays the requested features

3. The admin delete, deactivate or update the users

4. The use case ends

6.3 Logical View

The logical view of our system is explained by using a sequence diagram, two diagrams showing the resource and information sharing or uploading and accessing features of the system. The first diagram, figure 6.3 shows how the resource providers and other concerned bodies uploads and publishes resources, tools and other materials. The second diagram, figure 6.4 shows how the users access the resources from the system. The communication between the user’s web browser and the system is done via HTTP, due to the ability of HTTP to pass information in fastest way [71]. Also, HTTP is the best because almost all modern browsers understand this protocol and it fulfills accessibility requirement [72].

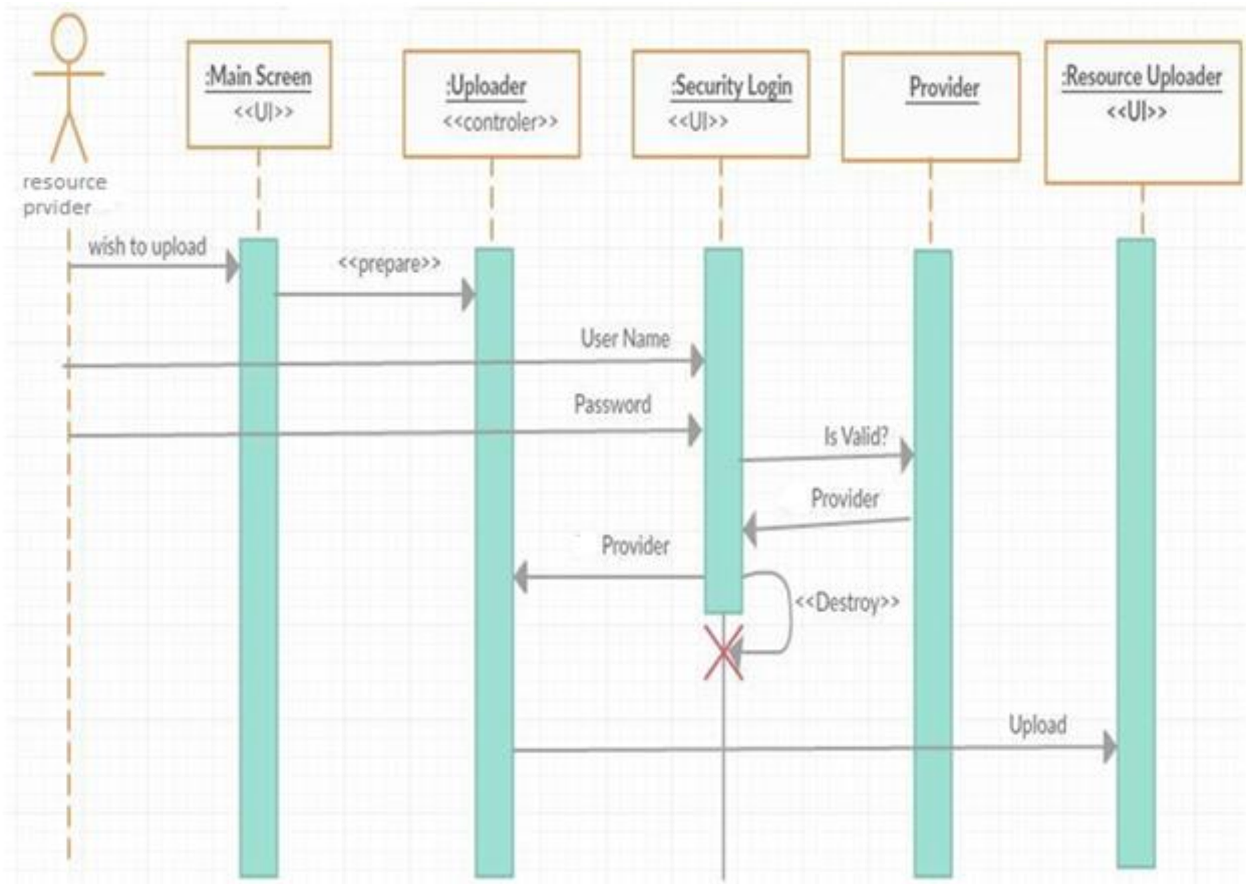


Figure 6.3: Sequence diagram of resource providers uploading resources

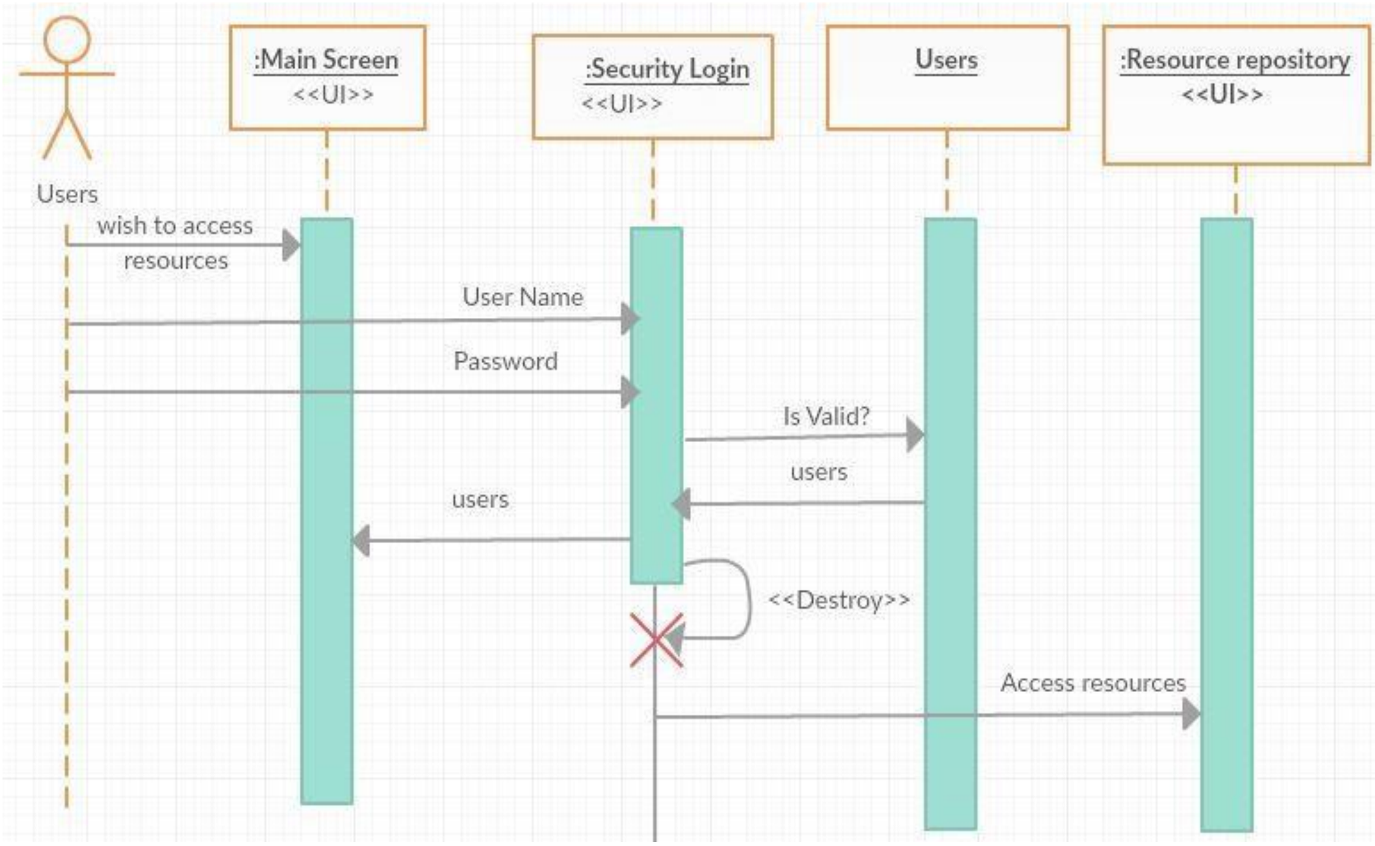


Figure 6.4: Sequence diagram-users resource access

6.4 Process View

The process view is illustrated through activity diagrams corresponding to registering, logging and accessing the system. Figure 6.5 shows the registration and login process to be part of the system users and have access privilege to the system. The activity begins by requesting the registration and login through the Registration and Login features of the system. During this process, the registration form is available to users and if they fulfill correctly, the users registered to the system. But if the user did not fulfill the form correctly, the error message is displayed for them and they fulfill the registration form again until they fulfill the form correctly and the process will be finished. In the login activity diagram, the user enters the User Name and Password after requesting the login and login will be validated. If the validation is ok, the process will end otherwise the invalid login notification will be displayed and they request login again until they make valid login.

Activity diagram for the overall system services are shown in figure 6.6. The user selects the services or features at the top of the healthcare system interface and they read the details of the selected service. If the user is not interested with the selected service, he/she will select another service. However, if the user wants the selected service, he/she starts to access the service, finish their task or activities associated with the selected feature and the process will end. On the other hand, if the user has comments related to the accessed service or has data to share for others, they prepare their comment or data, they select the browse menu, they provide their comment or data and they finish the process by publishing or uploading their comment or data to others.

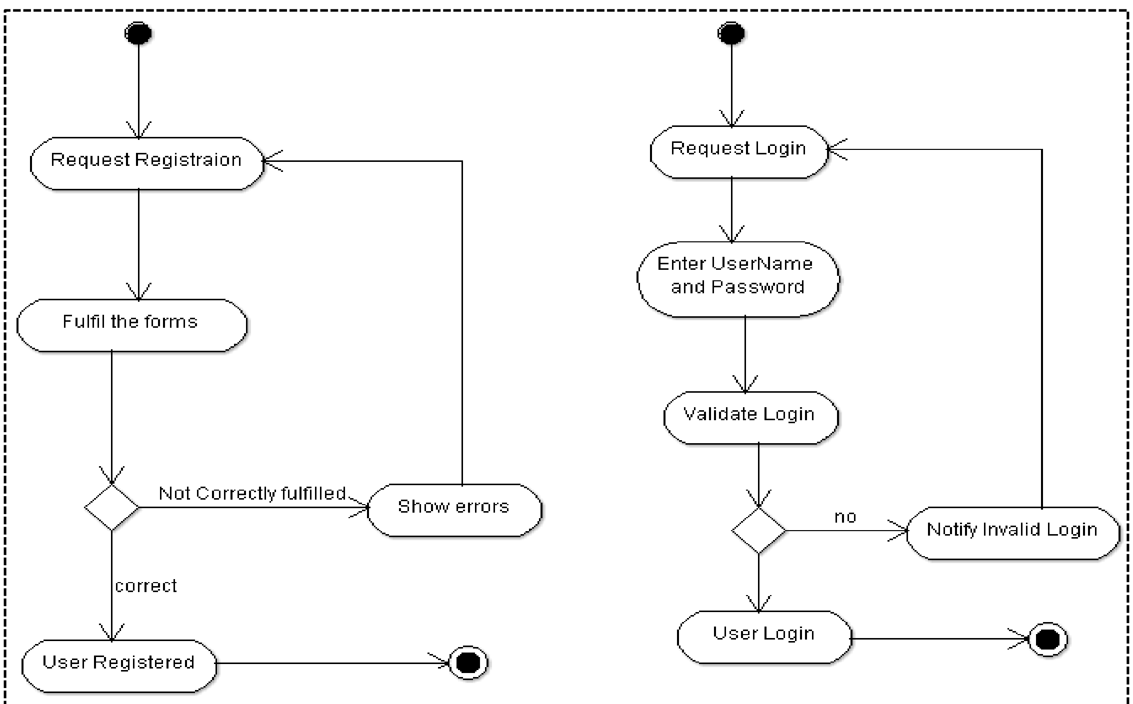


Figure 6.5: Activity diagram for registration and login

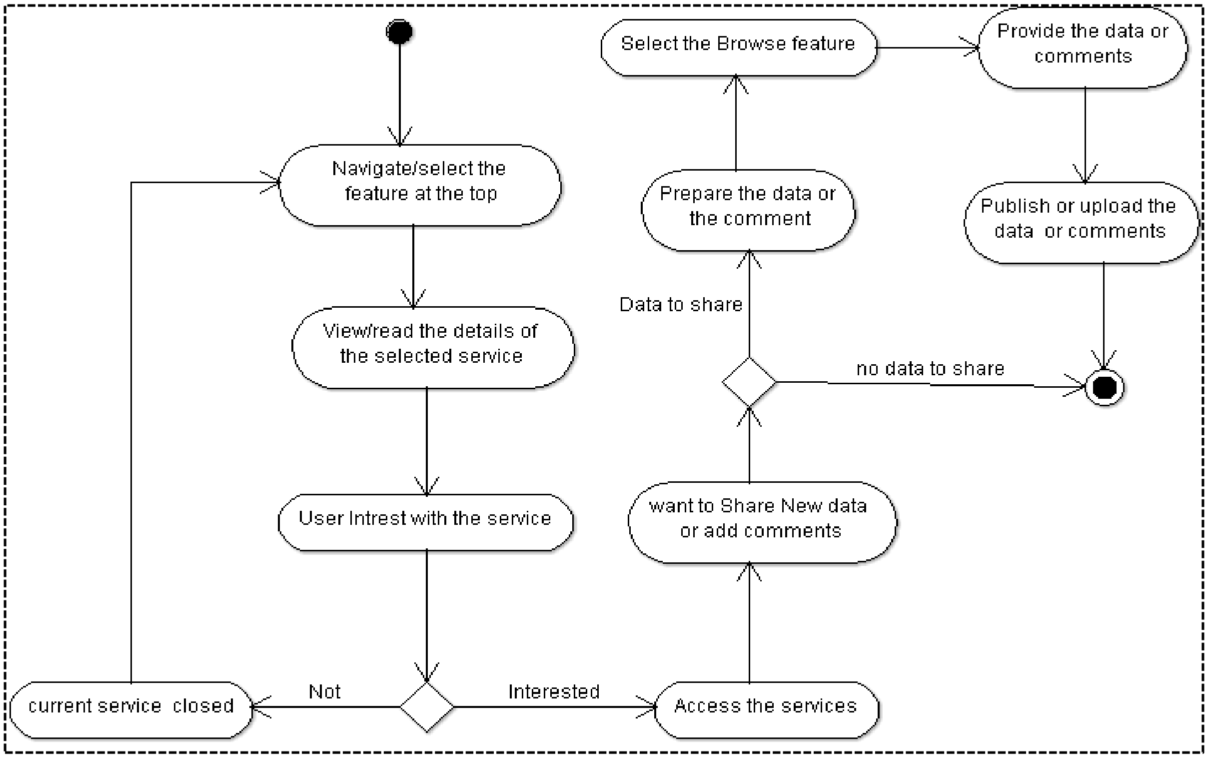


Figure 6.6: Activity diagram to access the overall system services

6.5 Data Model

Data model primarily supports the functional requirements and they define the services that the system should provide to the users. The data model used for the system is shown in figure 6.7. In order to avoid unnecessary complexities, we are not providing details of the entities inside each entity. Inter entity relationships are logical relations maintained in form of aggregation and composition of services corresponding to the entity whereas the relationship between entities is maintained at services level, i.e. relation between entities inside entity is maintained at database level in form of referential integrity constrains. The system has from one to several users. And the system has one to several repositories / databases.

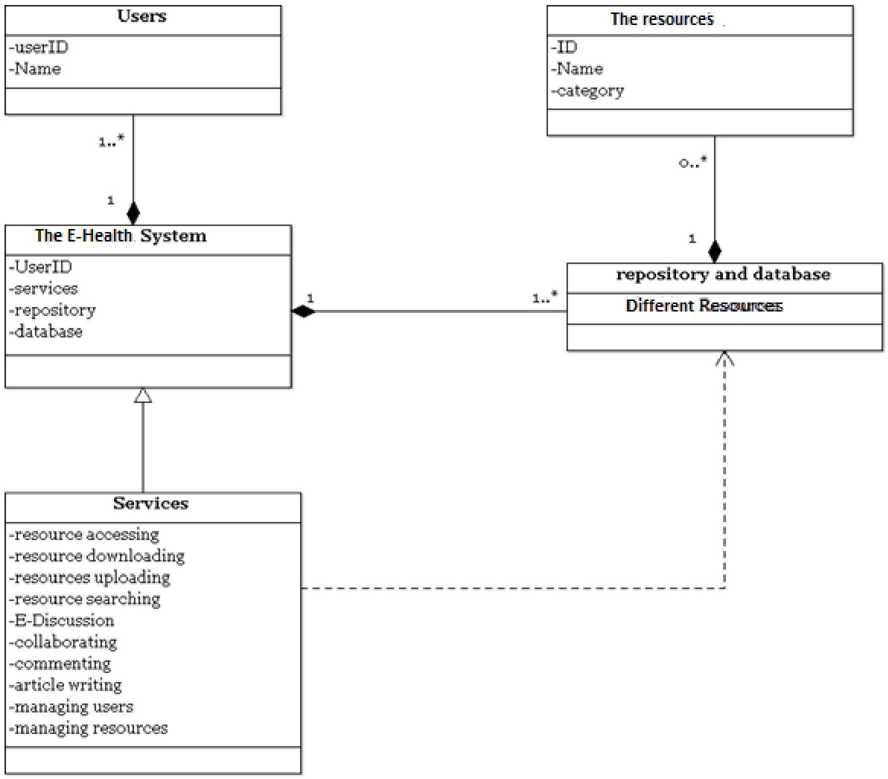


Figure 6.7: Data Model

6.6 Implementation Overview

The cloud portals are designed to be simple to navigate, easy to access the available resources and experiences, user friendly and easy for healthcare workers and other voluntaries who share their medical data and knowledge for others. The technologies that were used for development of the prototype are:

- JS, JQ, HTML and CSS for designing the web interface of the web application
- PHP language for backend side of the prototype
- OpenShift online platform for creating and testing the application
- Web and embedded cartridges to support languages and databases on the OpenShift online
- RHC client to manage the code
- Git repository to clone and deploy locally developed prototype on OpenShift online

The prototype is designed to accessed by using different devices. Figure 6.8 shows the developed application in OpenShift online web console with the associated gears and cartridges. A screenshot or web interface of the homepage for desktop or laptop view is displayed.

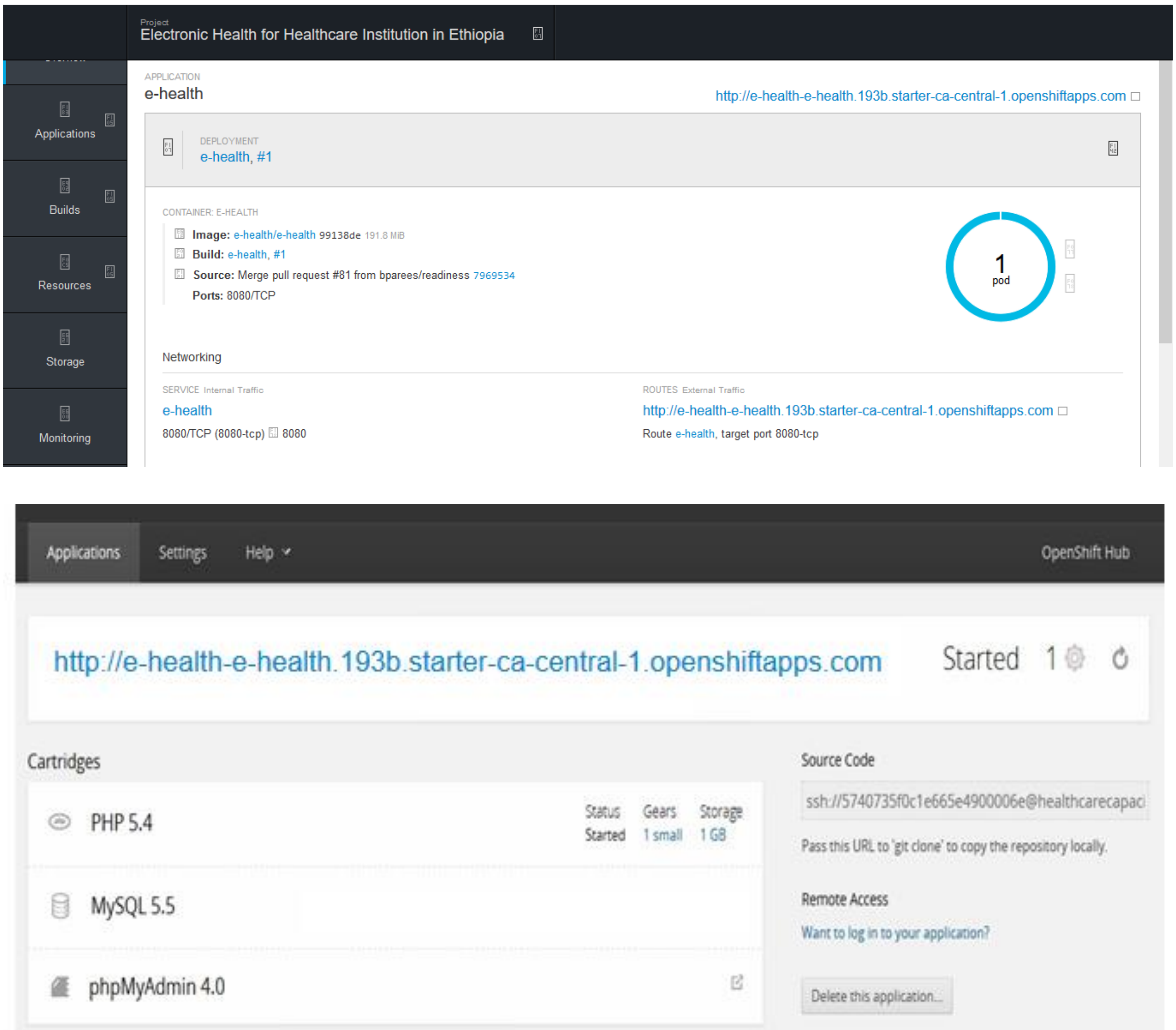


Figure 6.8: The developed healthcare application on the cloud

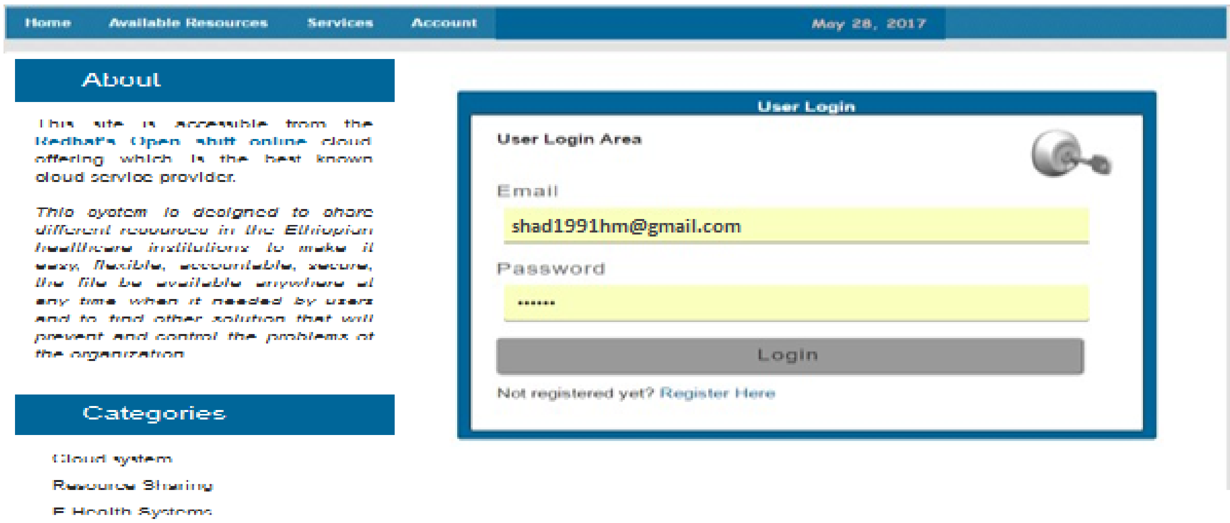


Figure 6.9: cloud portal desktop interface

6.7 System Evaluation

The system was evaluated by expertise from the area of HDIs, resource providers, cloud computing and software engineers at the early stage of the design to check its functionality and usefulness of sharing and accessing different resources. Because of this study is qualitative, evaluation criteria also based on availability, scalability, cost and performance. In terms of availability the system was accessed anywhere anytime principle using any browser and is automatically scalable with reduced cost and good performance. Any researcher can use it as a benchmark to add architectural component.

To check the workability of the developed cloud portal, the system test was performed through unit testing, integration and usability testing.

6.7.1 Unit Testing

The resource and knowledge sharing center has been tested in perspective of resource sharing and gaining access to these resources. Both resource sharing and resource accessing were tested in OpenShift online Redhat's platform after it has been implemented using different web and cloud technologies. All the concerned bodies including the resource providers, medical practitioners, HDI‟s ICT professionals and others provide their resources and knowledge to this common alliance central cloud place and the administrator manages it in the cloud. Each cloud interface is tested independently based on the following test case

Table 6.1: Sample unit test case

UTC ID	UTC Objective	UTC description	Expected output
1	Registering users by identifying their level	Click on registration from the front page; provide information	User information sent to database
2	Get access to the resource and knowledge sharing	Click on login from the front page and provide information	Users login to the system
3	Storing data on OpenShift online	Point on Browse menu and click on add/upload resources link	Store the resources to the OpenShift online database
4	Watching videos and Downloading available resources	Point to resources menu; click on watch videos and select documents or articles	Watching videos and files stored in user's local machine

6.7.2 Integration Testing

This system was the integration of different components and tested based on unit interface, completeness of interface and by creating integrated test condition for all interface. The following integration test conditions were used

- Resource and comment storing and retrieving test, file processing test and search test were checked. Each interface is validated based on providing input and using output
- The cloud portals were checked on OpenShift online
- Resource were sent from the web portal and stored in the OpenShift online server and transactions were managed accordingly.
- Integration was done for resource accessing, sharing and management

Table 6.2: Sample Integration Test case

ITC ID	ITC objective	ITC description	Expected output
1	Check the link between resources and services	Click on resources link and then service link	To be directed to services from resources
2	Check the link between login and Browse	Enter username and password and click on login	To be directed to Browse feature

6.7.3 Usability Testing

The cloud portal should be deployed for healthcare practitioners, capacity building providers, ICT professionals working in the HDIs, researchers, medical students, any respective organization and the whole community knowledge users. Usability test was done on the perspectives of above users. This test was performed on how all users use the cloud portal and attractiveness of the interface to use. They practiced/used the designed user interface and other features of the developed prototype. Finally, they suggest different additional issues that are to be included in the system. Therefore, usability testing was done with 10 healthcare practitioners, 2 ICT professionals working in the HDIs, 6 medical students and 5 community knowledge users. To test the system, each group of the users uses their own pages that created for them. And evaluate the result in terms of usability of the system on the cloud.

Table 6.3: Usability test case

No_	Activities	Interface rate	Technical issue
1	Register and login to the system	Very good	Easy to use
2	Accessing resources and tools	Good	Very easy
3	Writing and reading articles online	Good	Easy
4	Uploading/downloading resources	Very good	Very easy

Technical system design was checked by HDI’s ICT professionals, colleagues and experts from software engineering and related fields and they provide their comment and suggestions about the developed system. They recommend that, the system should be more users friendly and easy to use to make it more effective. Hence, the feedback was positive and those comment and recommendation have been left to respective healthcare resource providers and professionals to play their roles then developing full implementation will be easy for software engineers. After all, this framework to share different resources and to create healthcare knowledge sharing network was found to be effective and important when it is fully implemented with those comments given by all stakeholders.

6.8 Finding and Discussion

Current practice of providing resources are through manual method where the materials, approaches and tools are used by the resource providers in each HDI. After that these resources and best practices are not published or electronically available for other organizations and for the healthcare practitioners. The requirement of having many trainers to train all the healthcare practitioners through the country needs high cost and more time. Furthermore, the healthcare practitioners did not access the resources as they need and could not collaborate to share experience and best practices with their peers. The proposed cloud computing framework and the implemented prototype creates common alliance to share medical resources, best practices and experiences which is not only cost effective and interactive but also easy for collaboration, sharing and accessing the resources to develop knowledge, skill and capability in their professions. In addition, the healthcare practitioners can conduct online discussion by raising specific issues and forwards their challenges and solutions.

Compared with the manual method, it is resourceful, interactive and supported by current technologies and tools but it does not mean it replaces the whole manual resource usage rather it is used as a bridge by supporting and increasing availability and collaboration with other HDIs, voluntaries and fundraising bodies. Therefore, the proposed framework increases availability and sharing of different resources, tools, approaches, best practices and experiences among the HDIs and healthcare practitioners. This reduces cost of providing and distributing different resources reduces lack of well-trained man power and lack of technologies. It increases ability of the practitioners by using the system as knowledge gaining network and self-learning environment.

The proposed framework seems to have capability of sharing and accessing resources of all types such as documents (word document, PDF files, text files, excel files), pictures, audios, videos and other types of medical information. So, it adds more flexibility in accessing resources that can be used to develop practitioner's ability and skill. Also, the prototype is customizable and managed by the cloud provider which reduces management overhead, the interactive user interface allows to easily registering and login to the system to access all available resources and to share their experiences and data to others, therefore increases collaboration of healthcare practitioners. As the users of the system with the implemented prototype shows, it can be accessed using the existing ICT infrastructures.

The tools selected to design the prototype is based on open source technologies, which allow future development with less effort, which can be affordable and manageable by the economy of the country. The prototype has been developed from the open source cloud tool (OpenShift online) selected after evaluating several available tools.

In OpenShift online, the applications are typical web applications that will run on OpenShift. At this time, OpenShift is focused on hosting web applications. A copy of the application code is checked out locally into a folder with the same name as the application. With each new application that is created with the client tools, a remote Git repository is populated with the selected cartridge, which is then cloned to the current directory on the local machine. Most of OpenShift online application life cycle has four processes: code, build, deploy and manage. The **code** involves the development of application code with the desired language and tools. The build process builds applications, OpenShift online supports various build mechanism for this purpose. The application deployment process involves making any required changes to the application code, committing those changes to the local repository, and then updating the remote repository, deploying to OpenShift online and managing.

CHAPTER SEVEN

CONCLUSION AND FUTURE WORK

7.1 Conclusion

Present economic situation will force more and more organizations at least consider adopting a cloud solution. Healthcare institutions have begun to adhere to this initiative and there are proofs that indicate significant decreasing of expenses due to the implementation of cloud solutions. The aim of this work was to identify the particularities of using Cloud Computing within healthcare institution in Ethiopia. Mainly, we have considered the risks and benefits of cloud architecture and proposed a Cloud Computing framework for healthcare institutions. This research discussed different cloud topics like cloud deployment types, service delivery models and related concepts. It also discusses the benefits and limitations of cloud computing to healthcare institutions. This research investigated the potential benefits of using Cloud Computing in resource sharing environments to overcome the current service delivery system limitations.

The research has discussed about various companies that support cloud computing by providing tools and technologies to adapt to the cloud environment. Each section briefly describes the cloud features supported in these companies. Few of the services like Google Docs and Google Cloud Print are free, whereas that of AWS, Microsoft, etc., are proprietary. Based on the specific requirements, the user has to make a trade-off between open source and closed source tools/services. we have also discussed about various tools for cloud environment provided as an open source. We have categorized the tools into different groups like open source support for IaaS, PaaS, and SaaS, researchers, and distributed computing. Under each section, we have listed and briefly explained the architecture and features of some of the tools to provide useful information to the users.

The proposed framework is just the road map for the implementation of a whole virtual cloud based teaching-learning and service delivery ecosystem. After the system is ready to use an independent study comparing the service delivery using the proposed framework with the original service delivery environment should be conducted. This framework can be tested in departmental level before fully implemented for all healthcare institutions.

High level conceptual framework architecture and different views of the architecture are shown. Redhat's OpenShift online cloud computing platform as a service technology was used to develop the prototype as well as to share and access resources. Then healthcare practitioners and other users can easily share and access resources through the portal using their computer and mobile phones to develop ability, skill and experience. Consequently, the users are capable not only to handle any challenges in their profession but also to deliver quality healthcare services for the community and to create awareness, understanding and culture of using medical and other technologies.

7.2 Future Work

HDIs to create worldwide healthcare learning environment and to implement more robust, interactive and effective system. In some cases, because of economic problem and other issues, only limited healthcare practitioners can participate in resource sharing training but the majority of the workers have not the chance of attending this training in their work place. In the future, features to support different local and international language translation can be included. Hence extending the architecture to support translation of languages to expand accessibility of the system by different speakers for sharing and accessing resources, experiences and best practices is indispensable. Part of the future work can consider integration of the framework with insurance services as well as hospital services. Even privacy and security issues can be focused. Thus, it needs to be implemented in the future to perform all in one service together with the delivery of notification about the newly added resources and notes for the member of the system using their mobile phone. Lastly, in some cases the user of the system may share unwanted or out of the systems purpose files or videos. Such files may cause security threats on user's resources, busy servers by processing these unwanted large size files and distrust by users. In future, this must be handled by specifying intermediated actions to process files before it is stored to the OpenShift server and transmitted to users.

References

[1]. Sewale Belachew Demeke (2012) A Cloud Computing Framework for Ethiopian Higher Education Institutions

[2]. Ahmed E. Youssef M. (2014) A framework for Secure Healthcare Systems Based on Big Data Analytics in Mobile Cloud Computing Environments. International Journal of Ambient Systems and Applications (IJASA).

[3]. Prasad, R., Ranjan, M., Chandra, S. (2012) Design and Implementation of a Cloud based Rural Healthcare Information System Model, UNIASCIT. 2 (1): 149-157.

[4]. Yao Q et al. (2014, July) Cloud-based hospital information system as a service for grassroots healthcare institutions. Springer J Med Syst 38(9):104.doi:10.1007/s10916-014-0104-3.

[5]. Bogatin, Google CEO's new paradigm: 'cloud computing and advertising go hand-in-hand' August 23rd, 2006. Available http://www.zdnet.com/blog/micro-markets/google-who-needs-advertising/684 and http://nanospeck.hubpages.com/hub/Best-Cloud-Service-Providers.

[6]. WHO (2006) Working Together for Health: The World Health Report.

[7]. IBM (2010) Research Collaborations with Leading Taiwanese Institutions to Deliver Wellness- Centric Healthcare via Cloud Computing, IBM Press Release.

[8]. Samuel, O.W et al. (2013, July) Enhanced Cloud based Framework for Healthcare Delivery Organizations in Developing Countries. International Journal of Computer Applications

[9]. Adu et al. (2015, May) Cloud Computing Framework for E-Health in Ghana: Adoption Issues and Strategies: Case Study of Ghana Health Service. International Journal of Computer Applications (0975 – 8887) Volume 118 – No.17

[10]. Lisanne Brown, A. L. (2001, March). Measuring Capacity Building. Carolina Population Center, Chapel Hill, North Carolina.

[11]. Donald Kaniaru, X. K., Mkandla, S., & Kavagi, L. (2002, December). Capacity Building for Sustainable Development:An overview of UNEP environmental capacity development initiatives. (D. Simpson., Ed.) UNEP Divisions, Regional Offices.

[12]. Brief, T. (2010, April). Challenges encountered in Capacity Building: A Review of Literature and Selected Tools.

[13]. (EuroHealthNet), C. C. (2010, April). Determine Working Document # 5: Capacity Building for Health Equiy: a report of capacity building actions to address health inequalities and the socio-economic determinants of health. EuroHealthNet, Czech Republic.

[14]. IFPS Technical Assistance Project (ITAP). 2012. Capacity Building of Institutions in the Health Sector: Review of Experiences in Uttar Pradesh, Uttarakhand and Jharkhand.

[15]. The Cloud Changing the Business Ecosystem, a publication of KPMG International Cooperative, 2011.

[16]. UNDP (2010, January) .Capacity Development: A UNDP Primer Capacity is Development. geneva: UNDP Capacity development group.

[17]. Josephine et al. (March, 2014) DHIS2: The Tool to Improve Health Data Demand and Use in Kenya. Journal of Health Informatics in Developing Countries

[18]. VicHealth. (2012, August).Capacity building for healthcare promotion: information sheet retrieved May 26, 2016, from: www.vichealth.vic.gov.au/Publications/Vichealth-GeneralPublications/capacity_Building_FactSheet.pdf/

[19]. Sameh El-Saharty, S. K. (2009, August). Ethiopia: Improving Health Service Delivery. Health, Nutrition and Population (HNP) Discussion Paper. World Bank. Washingiton DC: World Bank 1818 H Street, NW.

[20]. Destefani Neto, M. (2015). A Brief history of cloud computing. Retrieved May 10, 2016, from Thoughts On Cloud Website: http://www.thoughtsoncloud.com/2014/03/a-briefhistory-of-cloud-computing/

[21]. Infoway. (2010, July 6). Cloud Computing Is The Latest Buzzword. Retrieved May 3, 2016, from Infoway.us: http://www.infoway.us/article/viewarticle.php?title=Cloud-Computing-IsThe-Latest-Buzzword&view=A-1278418848FWLDg

[22]. Maguire, J. (2008, December 16). Cloud Computing: The Ever Expanding Buzzword. Retrieved May 3, 2016, from Datamation.com:

http://www.datamation.com/entdev/article.php/3791456/Cloud-Computing-The-EverExpanding-Buzzword.htm

[23]. Vogel, P. S. (2011, November 9). Cloud Computing - New Buzzword, Old Legal Issues.

Retrieved May 3, 2016, from Ecommerce Times: http://www.ecommercetimes.com/story/73714.html

[24]. Barry M. Leiner, V. G. (2015, October 15). Brief history of Internet. Retrieved May 3, 2016, from internetsociety.org: http://www.internetsociety.org/internet/what-internet/historyinternet/brief-history-internet

[25]. Marcus Y. (2015). Implementing Cloud Design Patterns for AWS: Create highly efficient design patterns for scalability, redundancy, and high availability in the AWS Cloud.

Birmingham. Packt Publishing Ltd.

[26]. George R. (2009). Cloud application architecture: building application and infrastructure in the cloud. United States of America. O‟Reilly Media, Inc.

[27]. Mahesh S .Darak. (2014). Cloud Computing & its Applications in various sectors. Asian Journal of Management Sciences 02 (03 (Special Issue)); 2014; 07-11.

[28]. Lorraine M. Kieran C; 2013. Factors Affecting The Adoption Of Cloud Computing: An

Exploratory Study", Proceedings of 21st European Conference on Information Systems (ECIS).

[29]. Techopedia. (2016). Cloud Enabler. Retrieved May 9, 2016, from techopedia website: https://www.techopedia.com/definition/26525/cloud-enabler47

[30]. Aldrich, S. (2016, March 10). Data Backup & Recovery Blog: Four Enablers of Cloud

Computing. Retrieved May 9, 2016, from evs corporation.com:

http://www.evscorporation.com/cloud-blog/four-enablers-of-cloud-computing

[31]. Yogesh K. Dwivedi Navonil Mustafee, (2010),"It's unwritten in the Cloud: the technology enablers for realising the promise of Cloud Computing", Journal of Enterprise Information Management, Vol. 23 Iss 6 pp. 673 – 679.

[32]. Golden, B. (2009, April 29). Choosing application architecture for the cloud. Retrieved May 12, 2016, from TechTarget website:

http://searchcloudcomputing.techtarget.com/news/1355058/Choosing-an-applicationarchitecture-for-the-cloud

[33]. Lydford, S. (2015, Jun 19). Cloud Design: Cloud Application Design Considerations. Retrieved May 14, 2016, from codurance website: http://codurance.com/2015/06/19/cloudapplication-design-considerations/

[34]. Sanders, J. (2015, July 22). Mini-glossary: Cloud computing terms you should know. Retrieved May 12, 2016, from Techrepublic website: http://www.techrepublic.com/blog/theenterprise-cloud/mini-glossary-cloud-computing-terms-you-should-know/

[35]. Tutorials, C. (2016). Technologies related to cloud computing. Retrieved May 12, 2016, from

Cloud tutorials website: http://thecloudtutorial.com/related.html

[36]. He C, Fan X, Li Y. (2013) toward ubiquitous healthcare services with a novel efficient cloud platform. IEEE Trans Biomed Eng.

[37]. Siddiqui Z, Abdullah AH, Khan MK, Alghamdi AS. (2014) Smart environment as a service: three factor cloud based user authentication for telecare medical information system. J Med Syst.

[38]. Van Gorp P, Comuzzi M. (2014) Lifelong personal health data and application software via virtual machines in the cloud. IEEE J Biomed Health Inform.

[39]. Takeuchi H, Mayuzumi Y, Kodama N, Sato K. (2013) Personal healthcare system using cloud computing. Stud Health Technol Inform.

[40]. Ratnam KA, Dominic PD, Ramayah T.(2014) A structural equation modeling approach for the adoption of cloud computing to enhance the Malaysian healthcare sector. J Med Syst.

[41]. Hsieh JC, Hsu MW. (2012) A cloud computing based 12-lead ECG telemedicine service.

BMC Med Inform Decis Mak.

[42]. Hiden H, Woodman S, Watson P, Cala J. (2013) Developing cloud applications using the eScience Central platform. Philos Trans A Math Phys Eng Sci.

[43]. Endris Mohammed, (2016, February) "Participatory Indigenous Knowledge Management

Architecture for Traditional Medicine", Unpublished Master"s Thesis, Adama Science and Technology University.

[44]. C. R. Kothari, (2008) Research methodology: methods and techniques: New Age

International.

[45]. NJ, Hoboken, (1998) Introduction to qualitative research methods, 3rd ed. US: John Wiley & Sons Inc.

[46]. Uma D and Pansiri, Jaloni ogulu, (2011) "Mixed methods: A research design for management doctoral dissertations," Management research review, vol. 34, no. 6, pp. 687-70

[47]. Daniel Fallman, (2007, January) "Why Research-oriented Design Isn"t Design oriented research," Knowledge, Technology & Policy, vol. 20, no. 3, pp. 193-200.

[48]. Hevner A. etal (2004) "design Science In Information Systems Research," MIS Quarterly, vol. vol. 28, no. no. 1, pp. pp. 75-105.

[49]. Mattsson, M. et al (n.d.). Software Architecture Evaluation Methods for Performance, Maintainability, Testability, and Portability.

[50]. Paul Clements, Rick Kazman Len Bass, (2003) Software architecture in practice, second edition ed. Boston, USA: Addison Wesley.

[51]. Bell, D. (2005). Software Engineering for students A Programming Approach fourth edition.

Great Britain: Henry Ling Ltd.

[52]. Wikipedia. (n.d.). Software prototyping. Retrieved July 17, 2016, from wikipedia.org: https://en.wikipedia.org/wiki/Software_prototyping.

[53]. Patton, M. Q. (2001). Qualitative evaluation and research methods (3rd ed.). Newbury Park, CA: Sage Publications.

[54]. Sullivan, D. (2014, January 31). PaaS Providers List: Comparison And Guide. Retrieved July 19, 2016, from toms it pro: http://www.tomsitpro.com/articles/paas-providers,1-1517.html

[55]. J. Creswell (2008), Educational research: Planning, conducting, and evaluating quantitative and qualitative research, 3rd ed. Upper Saddle River, NJ: Pearson Education.

[56]. Wikipedia. (n.d.). OpenShift. Retrieved July 19, 2016, from wikipedia.org: https://en.wikipedia.org/wiki/OpenShift

[57]. Wikipedia. (n.d.). Microsoft Azure. Retrieved July 20, 2016, from wikipedia.org: https://en.wikipedia.org/wiki/Microsoft_Azure

[58]. Wikipedia. (n.d.). Cloud Foundry. Retrieved July 21, 2016, from wikipedia.org: https://en.wikipedia.org/wiki/Cloud_Foundry

[59]. Wikipedia. (n.d.). Heroku. Retrieved July 22, 2016, fromwikipedia.org: https://en.wikipedia.org/wiki/Heroku

[60]. Wikipedia. (n.d.). Google App Engine. Retrieved July 24, 2016, from wikipedia.org: https://en.wikipedia.org/wiki/Google_App_Engine

[61]. RedHat. (2014). OpenShift Online 1 User Guide: Managing Applications in the Cloud with

OpenShift Online. Retrieved Jun 11, 2016, from Openshift.com: http://www.openshift.com/.../OpenShift_Online-2.0-User_Guide-en-Us.pdf

[62]. OpenShift. (2012). Benefits document OpenShift platform features. Retrieved June 17, 2016, from RedHat website: http://www.redhat.com/13633302.pdf

[63]. RedHat. (2014). OpenShift Online 1 User Guide: Managing Applications in the Cloud with OpenShift Online. Retrieved Jun11,2016,fromOpenshift.com: http://www.openshift.com/.../OpenShift_Online-2.0-Client_Tools_Installation_Guide-en-US.pdf

[64]. IBM Cloud Computing Reference Architecture (CCRA) 4.0 Overview, 2014

[65]. NIST cloud reference architecture

[66]. Fehling, C., Leymann, F., Mietzner, R., Schupeck, W. (2011) A Collection of Patterns for Cloud Types, Cloud Service Models, and Cloud-Based Application Architectures in Institute Architecture of Application Systems (IAAS) Report, Daimler A G.

[67]. Keeling, M. (2014, October 22). Reflections on Software Engineering: Dealing with Constraints in Software Architecture Design. Retrieved May 27, 2016, from neverletdown.net: http://www.neverletdown.net/2014/10/dealing-with-constraints-insoftware-architecture.html

[68]. Ali, M. S. (n.d.). Cloud Security Mechanisms. Retrieved July 12, 2016, from slide Share: http://www.slideshare.net/SajjadAliMOHAMMED/cloud-security-mechanisms

[69]. Ali, M. S. (n.d.). Cloud Management Mechanisms. Retrieved July 12, 2016, from SlideShare: http://www.slideshare.net/SajjadAliMOHAMMED/cloud-management-mechanisms.

[70]. P. B. Kruchten, (1995) "The 4+1 View Model of architecture," Software, IEEE, vol. 12, pp. 42-50.

[71]. Wiki. (n.d.). Feature: HTTPS (HTTP Secure or HTTP over SSL/TLS). Retrieved July 26, 2016, from squid-cache wiki: http://wiki.squid-cache.org/Features/HTTPS

[72]. w3.org. (n.d.). HTTP - Hypertext Transfer Protocol. Retrieved July 12, 2016, from w3.org: https://www.w3.org/Protocols/

[73]. Wikipedia. (n.d.). Software architecture. Retrieved April 11, 2016, from wikipedia.org:

https://en.m.wikipedia.org/wiki/Software_architecture

[74]. Nsw Health (2001). A Framework for Building Capacity to Improve Health.

[75]. N. Karthikeyan& R. Sukanesh. (2012, August 3) Cloud Based Emergency Health Care Information Service in India. Springer

[76]. P. Mell, and T. Grace, (2010) The National Institute of Standard and Technology (NIST), Definition of Cloud Computing, Communications ACM 2010, 53(6):50.

Appendix

Interview Questioners

1. What are the challenges in the current healthcare scenario?
2. What are the advantages and disadvantages of resource sharing?
3. Do you think sharing resources will have a positive impact on quality of care? If yes, in what way?
4. Who are the beneficiaries when resources are shared (physicians, patients, healthcare organizations or others)? Please specify each in detail with their benefit
5. What things should be done, prerequisites, or required for effective resource sharing?
6. In your opinion, what are the technologies used currently for sharing medial images?
7. What are the challenges in using these technologies?
8. Could you believe cloud computing has the potential to enhance resource sharing system? If yes in what way?
9. What do you think the most challenge of cloud computing for healthcare institutions?
10. Cloud you list services, tools, provisioning models and providers appropriate for resource sharing?
11. Which provisioning model (private, hybrid, community, public) is better for and resource sharing in healthcare institutions?

Expert Survey Questioners

1. How do you see the framework's application interfaces, where resource sharing solution will request and provide information to other systems that are typically reside at the point of patient care?
   a. Do you feel in the interfaces are enough? If not mention to be added?
   b. What are the positive and negative impacts of these interfaces on provisioning of resource sharing
   c. What are the impacts of these interfaces on the framework?
2. Do you feel that the communication standards used in the framework are appropriate in support of information exchange between systems implemented in a large variety of development environments (technical) and communication environments?
   a. Does the framework used most recognized communication standards/
   b. Have you noticed new feature in this framework in the utilizing more standards?
   c. Do you feel there is an excluded standard that must be incorporated? If yes please specify.
   d. Do those standards enable to safely and effectively share information across a disparate and heterogeneous healthcare enterprise?