Cloud Computing: Migrating to the Cloud, Amazon Web Services, and Google Cloud Platform
Abstract

Cloud computing provides great benefits and challenges for small, medium, and large organisations. Whether it is a financial, technology, or engineering sector any company may find a useful cloud component for the organisation's needs. Though with its benefits come many challenges, experts believe that cloud computing advantages outweigh the disadvantages. With more research in the area of cloud computing, the challenges will be dealt with.

Amazon and Google are the prominent cloud services providers with different components and services that are evolving. However, there are many other cloud vendors and this makes it difficult for potential cloud users to migrate or choose a suitable vendor from the numerous available ones.

This thesis, using design science approach describes the implementation of cloud computing by different cloud providers, i.e., examines and demonstrates how Amazon and Google implemented and structured their cloud infrastructure. Also, it provides the criteria for accessing the suitability of a business or organisation for choosing a cloud provider. In addition, a web application hosted on both AWS and GCP platforms is developed in order to demonstrate the workability of the framework and guideline for selecting a cloud provider. The application is called KlaudCelet. KlaudCelet is a recommender system that is developed based on previous research on cloud deployment frameworks and proposed model (CPSM; Cloud Provider Selection Model). The recommendation by the system is reliable because it is based on research results. KlaudCelet recommends a deployment model, a cloud service provider, and a decision either to migrate or not. CPSM was not used in KlaudCelet but the model by Keung & Kwok and Misra & Mondal were used.

The guideline developed was used to create a recommender framework. The need for solving this problem was shown by conducting a review of previous research and a small-scale survey among IT experts. The survey found that IT experts are optimistic about cloud computing adoption despite its challenges such as security, privacy and leaning curve. The decision to migrate to the cloud must take into consideration the organisation's state and culture, cloud deployment model, and choosing a suitable cloud service provider.

Keywords
Cloud computing, Amazon Web Services, Google Cloud Platform, cloud provider selection

Supervisor
Doctor Dorina Rajanen
# Table of Contents

Abstract .......................................................................................................................... 2  
Table of Contents ........................................................................................................... 3  
Acronyms and Abbreviations ....................................................................................... 5  
1. Introduction ............................................................................................................... 7  
2. Research Background and Research Questions ..................................................... 10  
   2.1 Related Research .................................................................................................. 10  
      2.1.1 Migration to the Cloud .................................................................................. 10  
      2.1.2 Economics of CC Migration ........................................................................ 10  
      2.1.3 Privacy and Security .................................................................................... 11  
      2.1.4 Cloud Service Providers ............................................................................. 11  
      2.1.5 Frameworks that Describe CC .................................................................... 12  
   2.2 Research Problem and Research Aim ................................................................ 13  
   2.3 Research Questions ............................................................................................. 13  
3. Research Methods ................................................................................................... 15  
   3.1 Design Science Research Methodology ................................................................ 15  
   3.2 Design Science in Information Systems ............................................................. 16  
   3.3 Implemented Research Methodology Process .................................................... 17  
   3.4 Questionnaire Design ........................................................................................ 19  
   3.5 Overview of Thesis Research Process ................................................................ 19  
4. Amazon and Google Cloud Services ....................................................................... 21  
   4.1 Amazon Web Services (AWS) ............................................................................ 21  
      4.1.1 Structure ...................................................................................................... 22  
      4.1.2 Components ............................................................................................... 24  
   4.2 Google Cloud Platform (GCP) ............................................................................ 25  
      4.2.1 Structure ...................................................................................................... 25  
      4.2.2 Components ............................................................................................... 28  
   4.3 Comparison between AWS & GCP ..................................................................... 30  
      4.3.1 Structure and technical capabilities ............................................................. 30  
      4.3.2 Performance ................................................................................................ 30  
      4.3.3 Business models and pricing ....................................................................... 30  
      4.3.4 User interface and learning costs ................................................................. 31  
5. Cloud Computing Survey ....................................................................................... 32  
   5.1 Survey Data ......................................................................................................... 32  
   5.2 Survey Administration ....................................................................................... 33  
   5.3 Survey Participants ............................................................................................. 33  
   5.4 Discussion of Survey Findings ......................................................................... 34  
6. Migration to the Cloud ............................................................................................ 35  
   6.1 Cloud Migration Criteria and Guidelines ........................................................... 35  
      6.1.1 General Guidelines for Migrating to AWS and GCP .................................. 36  
      6.1.2 Security Guidelines for Migrating to AWS and GCP .............................. 37  
   6.2 Cloud Provider Selection Model (CPSM) Framework ........................................ 38  
      6.2.1 Years of Operation and Experience ............................................................ 38  
      6.2.2 Application support .................................................................................... 39  
      6.2.3 Amount of available services ...................................................................... 39  
      6.2.4 Geographical presence .............................................................................. 39  
      6.2.5 Pricing model ............................................................................................ 39  

Acronyms and Abbreviations

AMI: Amazon Machine Image
APS: Application Platform Services
API: Application Program Interface
AWS: Amazon Web Services
CC: Cloud Computing
CDN: Contents Delivery Network
CSP: Cloud Service Provider
DNS: Domain Name System
DRA: Durable Reduced Availability
EC2: Elastic Compute Cloud
ELB: Elastic Load Balancing
GAE: Google App Engine
GCE: Google Compute Engine
GCP: Google Cloud Computing
HTTP: Hypertext Transfer Protocol
IT: Information Technology
ICT: Information and Communication Technology
LAMP: Linux Apache MySQL and PHP
MAMP: Macintosh Apache MySQL and PHP
NoSQL: Non Structured Query Language
OLAP: Online Analytical Processing
PHP: PHP Hypertext Preprocessor
REST: Representational State Transfer
RDS: Relational Database System
SNS: Simple Notification Service
SQS: Simple Queue Service
S3: Simple Storage Service
SDK: Software Development Kit
SOAP: Simple Object Access Protocol
VPN: Virtual Private Network
WAMP: Windows Apache MySQL and PHP
XAMPP: Cross platform Apache MySQL PHP and Perl
1. Introduction

Data storage is an important key factor in computing and a backbone for many organizations’ business activities. It is one of the IT infrastructure tools, which has created challenges and complexity for many technology companies due to its constant changes and advancement built on existing tools. Over a decade, the trend evolves from a desktop-centred architecture to client-server applications. Moreover, recently, the increasing adoption of web services and service oriented architecture infrastructures has been the new trend (Varia & Mathew, 2014; Lin & Chen, 2012). In addition, adequate utilization of companies’ resources such as servers, data warehouse, and hardware tools has been a growing problem in IT industries. That is, making proper utilization and predictions has been problematic (Barr, 2010). For instance, an IT firm predicts that its company will make use of 100GB of data storage in the next 2 years, but after the 2 years the company ended up using higher or lower part of the predicted data storage; this is a typical situation in a company and it is not easily foreseen due to the uncertainty in many businesses. Cloud computing evolves to solve these problems of resources utilization, data storage, and many more computing needs. (Barr, 2010)

Cloud computing (CC) is a technology that has been in existence over a decade (Lin & Chen, 2012). It could be thought of as being a “coherent, large-scale, publicly accessible collection of compute, storage and networking resources” (Barr, 2010), that are allocated through HTTP requests (i.e. Web services calls). According to the U.S National Institute of Standards and Technology (NIST), cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (such as networks, servers, storage, applications and services) that can be rapidly provisioned with minimal management effort or service provider interaction (Divya & Jeyalatha, 2012; Mell & Grance, 2009). NIST classified CC into four deployment models; private, public, hybrid, and community. Private clouds are owned by enterprises, public clouds are huge infrastructure that is sold to the public, community cloud is a shared infrastructure that is being used by specific community and hybrid cloud is made up of two or more cloud deployment models.

Cloud computing has three different service models: software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS) (Mell & Grance, 2009). In addition to these three models, the International Telecommunication Union recognized a fourth service model, namely network as a service (NaaS) (Nakhimovsky & Myers 2004). SaaS involves software applications that are installed and operated by cloud vendors (Google, Amazon) and used by clients across different locations. Clients just use the system without worrying about maintenance, while providers manage the system. Examples of SaaS include Google apps, and Apple iCloud. On the other hand, NaaS provides network and transport capabilities, mainly Virtual Private Network and bandwidth. Also, IaaS is the most basic model where a computer system itself serves as the service and provides hosted virtual machine that allows single machine to act as multiple computing environments and behave like individual computers. IaaS involves the use of computing resources without needing a physical computer presence. Further, IaaS can be organized and packed as a PaaS (for example, XAMPP, WAMP, MAMP and LAMP PHP application servers). One example of IaaS is virtual private server (VPS). The last, but not least among the cloud service models is PaaS. PaaS is an adequate framework used for hosting clients’ developed applications (Chang, Walters & Wills,
The content of the NIST agreed definition and models have been the basis for many researches in cloud computing and contributed to the increase in wider adoption of cloud computing.

According to Dunn (2010) and Minoli (2010), despite the decline in the economy, CC adoption had been on the increase, predominantly in the area of green IT and amalgamation of data centers where it is being used to cut cost of operation (as cited in Chang et al., 2013). One of the advantages of CC over traditional computing is that, in adopting CC, companies pay for what they use not what is predicted to be used. Also in contrast to traditional computing, CC is flexible, and provides resilient infrastructure (Lin & Chen, 2012). Further, by using CC, the servers, data storage, and network bandwidth could be utilized appropriately and freed back to the cloud when they are not being used (Nakhimovsky & Myers 2004). Windstream (2014) stated that as more and more organizations move their business activities to the cloud, many benefits such as geographical increase, improved collaboration across many business units, better customer services and agility, and time to market and process efficiency are being reported (as cited in Garrison, Wakefield & Kim, 2015). Nowadays, cloud computing adoption is being used as a competitive advantage, allowing companies that implements CC to get more customers than their rivals (Garrison et al., 2015). Berman, Kesterson-Townes, Marshall, & Srivaths (2012) forecasted that by the end of 2015, about 90% of business and technology managers will implement one or more types of cloud computing (as cited in Garrison et al., 2015). Also, the report by Gartner (2011) predicted that by the year 2020 many organisations and government agencies will be using CC for more than half of their IT services (as cited in Garrison et al., 2015).

However, despite all the promises of cloud computing to solve various problems, many organisations are still concerned about the challenges posed by CC. With currently very few number of companies using CC, cloud computing is still in its beginning stage due to some challenges such as security, privacy, trust, location and cost-effectiveness (Misra & Mondal, 2010). Trusting another company with valuable information stored somewhere without having control of is risky and could cause unfavourable effect to organizations. What would happen if the security were accidentally compromised? (Jog & Madiajagan, 2012). Bamiah et al. (2012) and Hamouda (2012) discuss how the fear of privacy issues had contributed to the slow adoption of cloud computing paradigm in healthcare and banking sectors. Furthermore, another issue that perturbs IT practitioners and companies that wish to adopt CC is that everything is stored in a single point, which could lead to unavailable of resources if there is a natural disaster or network breakdown. In my opinion, the issue of availability of resources had been minimized based on the architecture being used by cloud vendors (Varia & Mathew, 2014). Moreover, according to (Lin & Chen, 2012), some technical challenges will be solved as time goes on with innovation in the IT fields.

Despite the above CC challenges to the IT community, as a software developer I believe that the benefits of CC contribute to software engineering by providing the software developers the right tools and instruments to be more efficient and productive. However, according to many authors, cloud computing is still an unfamiliar concept to many IT professionals the cost of learning CC and acquiring adequate skills is a major concern for managers that wish to migrate to the cloud (Lin & Chen, 2012). Misra & Mondal (2010) also pointed out that the lack of proper understanding and usage of CC has contributed to its low adoption. Thus, because the relative advantages of CC are not immediately apparent to companies, Lin & Chen (2012) pointed out that further research in this area should focus on the usage of cloud infrastructure to create advantage. These aspects form the rationale of this thesis, namely the business and academic needs to increase the
knowledge and understanding of CC characteristics, advantages, and migration possibilities.

Thus, the objective of this study is to investigate and understand the guideline for using two of the most adopted and advanced public cloud solutions (Amazon Web Services and Google Cloud Platform), and to examine criteria for choosing suitable cloud vendor for a project. Amazon Web Services (AWS) advances a flexible, resilient, elastic, and secure architecture, and Google Cloud Platform (GCP) provides a collection of centralized web applications and services as well as limited free resources. According to Olson (2012), selecting a suitable cloud platform solution is not an easy task and, organizations are locked in the choice of choosing which cloud provider would best suit their project or company. Thus, this study also aims to survey the perception of IT experts towards cloud computing, cloud computing adoption and migration. These objectives are translated into research questions in Chapter 2.

To reach its objectives, this thesis describes the implementation of cloud computing by different cloud providers, i.e., examines and demonstrates how Amazon and Google implemented and structured their cloud infrastructure. Also, it provides the criteria for accessing the suitability of a business or organisation for choosing a cloud provider. In addition, in this thesis, a web application hosted on both AWS and GCP platforms is developed in order to demonstrate the applicability of the framework for selecting a cloud provider. This thesis uses design science approach as the main research methodology (Hevner et al., 2004).

The structure of this thesis is as follows; the next chapter of this research presents the research background and research questions, related research, as well as the research aims pursued in this thesis. Chapter 3 describes research methods used in this thesis. Chapter 4 presents the analysis, composition and structure of AWS and GCP. Chapter 5 explains data collection and its results and Chapter 6 presents Cloud Service Provider Selection Model and cloud migration, while Chapter 7 demonstrates the sample application hosting. Chapter 8 discusses the results of the study and its limitations and Chapter 9 summarizes the findings and concludes the thesis.
2. Research Background and Research Questions

After the advent of cloud computing, many new cloud service providers (CSPs) have been born and many existing ones are evolving. With growing and diverse needs of different businesses, it is challenging for cloud vendors to meet users’ high demands from the cloud (Olson, 2012). Also, many attempts have been made to develop a framework of CSP selection, but there is no platform that provides a way to choose a suitable cloud provider based on its services and benefits for a company’s project(s) or organisation because the available criteria cannot be generalized (Li, Yang, Kandula and Zhang, 2010). By platform I refer to a research of cloud provider services and benefits for organisations.

This chapter describes the current knowledge and research on migration to the cloud, AWS and GCP, as well as research gaps in the area and the thesis’ research questions.

2.1 Related Research

There has been different research on cloud computing due to growing interests both from the academics and the industries around the globe. It has been discussed in many conferences, computing books, and many business journals. In the past 6 - 7 years, the increasing publicity on CC has been noisy (Olson, 2012). This chapter looks at the different classifications of previous studies on CC.

2.1.1 Migration to the Cloud

According to Tran, Keung, Liu and Fekete (2011) migration is the process of porting already existing application hosted in a conventional, in-house datacenter or traditional hosting environment to a cloud platform. In this kind of migration, it is important to consider many challenges such as compatibility, programming language, storage choice, and software differences.

In addition, migration could involve moving from a cloud service provider to another, e.g. from Amazon to Google or vice versa. However, this does not pose many challenges as many cloud providers use similar underlying technologies and services. This paper focuses on migration from traditional hosting to the cloud as well as from one cloud provider to another.

2.1.2 Economics of CC Migration

Tak, Urgaonkar and Sivasubramaniam (2012) discussed issues related to the economics of CC migration by analysing the migration costs of different deployments options. According to Tak et al., (2012), workload intensity, growth rate, storage capacity, and software licensing costs have huge cumulative effect on overall costs. Also, a related cost of migrating an existing enterprise application was classified into direct such as hardware, software, database maintenance etc. and indirect costs such as networking infrastructure, tax and so on. More so, it answers the question; does it make sense for ‘my’ application to be migrated to the cloud? Unfortunately, the research does not cover start-ups or first time cloud users; it focuses on already established organisations that need to worry about
labour cost involved in migration. Also, the paper focuses on Amazon EC2 and Microsoft Windows Azure comparison.

Tran, Keung, Liu and Fekete (2011) also researched the effort needed in migrating an application hosted on traditional hosting company to the cloud. The paper explores the economics of main migration activities through an experiment. The paper identified some cost factors of migration process and proposed taxonomy of tasks for migrating an in-house application to cloud. Their findings were supported with a case study involving migrating a .NET application running on Windows Azure. Part of the conclusion of Tran et al. (2011) experiment stated that the efforts needed for deciding cloud service providers and their services is one of the major tasks required for successful migration so as to reap the benefits of the CC paradigm.

Previous studies in CC migration discussed mainly the benefits of migration, running cost and the costs of adoption, which includes training as CC economic benefits. In contrast, Tran et al. (2011) explore the cost involved during migration time. Most of these previous studies use empirical analysis with sample application deployment from small and medium enterprises to validate their findings. However, this thesis will not discuss the cost analysis of migrating to the cloud, but will address the issue of what provider could be more cost effective in terms of the payment model.

2.1.3 Privacy and Security

Other issues widely discussed in previous research regarded the security and privacy of CC. Since the Internet is a general window for malicious users to gain access to protected data, the security challenges in CC could come from different sources such as API, virtualization, identity management, data storage and so on. Previous research examined the main difficulties contributing to slow adoption of cloud computing paradigm.

For example, Bamiah, Brohi, Chuprat, & Brohi (2012) and Hamouda (2012) discussed how the fear of privacy and security issues had contributed to the slow adoption of CC paradigm in healthcare and banking sectors. In addition, Tianfield (2012) explains that despite the huge advantages offered by CC to all business sectors, many CSPs fail to assure privacy and confidentiality of user data. Also, Sen (2013) concluded in his paper that transparency is part of the concern that is making many large companies reluctant in moving all their projects to the cloud, and rather choose testing the cloud with small and insensitive data.

However, Younis, Merabti & Kifayat (2013) discussed many technical and cultural aspects of CC that needs improvement in order to address users’ concern of CC paradigm. This thesis is not about privacy and security in CC, but will give useful guidelines of what to consider in AWS and GCP in terms of security and privacy before selecting a CSP.

2.1.4 Cloud Service Providers

Cloud service providers’ (CSPs) offer utility-oriented IT services to use around the world by delivering platform, infrastructure, and software to consumers as subscription-based services. Also, CSPs computing resources are pooled to serve multiple consumers using multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. Therefore, a consumer of CC can unilaterally provision computing capabilities (server time, network storage) without requiring human interaction with CSPs. Different researches on cloud service providers
had been done, most of which focused on a particular component such as AWS EC2, S3 and Microsoft Azure. One of them is the research by Brantner, Florescu, Graf, Kossmann and Krasker (2008) that focused on how AWS S3 can be used to power applications database.

On the other hand, Varia & Mathew (2014) and Mathew (2014) provide extensive overview of AWS including its advantages and examination of different components and services provided by AWS. Linton (2011) provided steps required for migrating a .NET application to AWS. Robinson (2008) examined how to use different AWS components to enhance business goals.

To add more to the list of previous researches and papers that focused not only on a particular component but the entire services, Severance (2009) examined the use of Google App Engine, which is just a component of GCP. He presented programming topics such as HTML, CSS, JavaScript and AJAX and how they are being used to develop application for GCP. Cohen, Hurley & Newson (2015) discussed how to use different components of GCP such as; Google Cloud Storage, Google SQL and Google Datastore.

These articles and textbooks provide limited information about AWS and GCP because they do not provide general criteria for accessing their suitability for different project(s) and do not provide a comparison and/or integrative view of the two cloud service platforms.

2.1.5 Frameworks that Describe CC

Li et al. (2010) developed a model based on performance of public cloud providers by categorising their services and identified the problem of growing number of cloud vendors which makes it difficult to choose from their array. He created generic performance metric based on the four proposed categories of services. Based on his conclusion, he suggested that specific analysis is needed rather than broader performance metric. However, I believe that it is not enough to use performance alone as a prediction model for cloud vendor selection because cloud users are mainly interested in the performance of their own project when it is migrated to the cloud. Performance of a vendor that is focused mainly on computation will be different from that of a vendor that focuses on storage.

Keung and Kwok (2012) created a model called Cloud Deployment Selection Model (CDSM). This model helps small and medium size companies to select between buying a private cloud and renting a public cloud. The model was developed based on factors identified from interviews. The framework identifies cloud versatilities as one of the major challenges contributing to low cloud adoption. Although, the model deals with one of the concerns faced by many organisations and suggests a suitable deployment model, the CDSM model cannot be used to select a suitable cloud vendor. Prospective cloud adopters need to be versatile with the services provided by each CSP not just tell them to migrate to the public cloud without knowledge of vendor services. Therefore, an in-depth versatile framework is needed to fill this gap.

Misra and Mondal (2010) developed a framework to help organisations with an existing datacenter to examine the usefulness of migrating to the cloud. The paper focuses on the company’s resources as a basis for creating a framework of four characteristics. Each characteristic then contains different sub-factors with different weighted value based on their criticality. As with other CC frameworks, Misra and Mondal’s framework only solve a portion of the challenges without delving into the characteristics of individual
CSPs.

2.2 Research Problem and Research Aim

A large body of current research focuses on general security considerations before migrating or selecting a cloud service provider. However, there has not been much academic research that focuses on security, cost, and comparison of features and guidelines of services offered by individual cloud vendors. In discussing individual vendor’s services, most of the papers only address either Amazon Elastic Cloud Computing (EC2) or Google Compute Engine, as seen in the Section 2.1. Therefore, there is a need to describe the services, components, security, cost, and compare the services provided by individual cloud service providers, owing to the fact that there are numerous cloud vendors that cannot meet users’ high demands.

This thesis fills this gap in the literature by studying the features, similarities, and differences between AWS and GCP infrastructures and their components, and at the end, deduces a framework for determining suitable infrastructure for a project or company. This framework is in the form of a cloud provider selection model (CPSM). The framework is meant to be used to further develop a web application for validation. The theoretical basis of the proposed framework consists of the benchmarks provided by Li, Yang, Kandula and Zhang (2010) on comparing different public cloud providers, and the analysis of CC survey by Misra and Mondal (2010) on the suitability of CC for a company. Furthermore, research by Keung and Kwok (2012) on private and public cloud providers is used to determine either to rent a cloud or buy one. It is worth mentioning that the research by Gorelik (2013) that examined different cloud models focuses mainly on AWS, while this thesis focuses on both AWS and GCP. The selection of AWS and GCP as objects of study was because these are the most adopted and advanced cloud vendors. Chapter 4 describes the two cloud computing platforms.

This thesis aims to provide knowledge to the IT community as well as end-users and management-oriented audiences about current solutions for choosing a CSP for a general purpose, criteria for choosing CSPs, and components that are available for projects hosted on AWS and GCP. As organisations are looking for help by using technology to meet dwindling budgets (Olson, 2012), the outcome of this research will be valuable for managers and decision-makers in organisations for evaluating the propositions for using the cloud and to know which CSP best suits their company or project(s). The research approach is based on design science and is detailed in Chapter 3.

2.3 Research Questions

To provide a framework and guidelines of selecting a cloud service provider and conduct migration activities, this thesis systematizes knowledge from the literature and from a questionnaire survey to answer the following research questions:

RQ1: What are the opinions and attitudes of IT experts with regard to CC, CC adoption, and CC migration?

RQ2: What criteria are relevant for companies when migrating to a cloud service provider?

RQ3: What are the guidelines to consider when configuring application on AWS & GCP?
To answer these research questions, this thesis utilizes the design science research methodology (see Chapter 3). To demonstrate the applicability of the framework, a web application is developed and described in Chapter 7.
3. Research Methods

This chapter contains three sections. Section 3.1 describes the Design Science Research Methodology (DSRM) and its activities and procedures. Section 3.2 explains briefly the Design Science (DS) research in Information Systems (IS) field of study. Section 3.3 elaborates the research method used in conducting this study.

3.1 Design Science Research Methodology

Peffers, Tuunanen, Rothenberger and Chatterjee (2007) made it known that the absence of consensually agreed process for carrying out Design Science Research contributed to the development of a DSRM framework. The principles, practices, and procedures in the framework are to guide researchers for effective presentation and thorough conduct of the design science research. The framework was made to address three basic objectives, which are: (1) it gives researchers a nominal process model for conducting DS research; (2) it must be coherent with existing literature and DS principles; and (3) it gives researchers a conceptual model for displaying the DS research. Figure 1 presents the framework with its six successive activities comprising of; problem identification and motivation, define objectives of a solution, design and development of the artefact, demonstration of the artefact, evaluation, and communicating the final results. (Peffers et al., 2006; Peffers et al., 2007)

![Nominal Process Sequence](image)

**Figure 1.** The design science nominal process model (Peffers et al., 2007)

The first activity involves identifying the problem, and the motivation and justification of the value of the solution to the problem. It is important to have knowledge of the problem domain and the importance of its solution for the first activity. Second activity establishes the objective of the solution in relation to the problem. Researchers use qualitative or quantitative means to establish the objective of the solution based on knowledge of the problem and available solution. In the next activity, which is creation and construction (or design and development), knowledge of relevant theory to the solution is needed to determine functionality, solution architecture, and then developing the real artefact. Fourth activity could involve experimentation, case study, proof and simulation of many examples or instances of the problem. Fifth activity is concerned with artefact evaluation
and also, it answer the question of how does the artefact meets the objective of the solution? It might involve qualitative and quantitative methods for evaluation. The selection of the evaluation method depends on the nature of the problem and on the characteristics of the solution. Finally, the last activity is about communicating the findings. This involves presentation of the problem, its importance, uniqueness of the artefact and its usefulness to the researchers and other audiences. (Peffers et al., 2006; Peffers et al., 2007)

The research could start from any step and move on, though the process is structured sequentially. Figure 5 shows also four possible entry points: - problem-centred approach, objective centred solution, design and development centred and observing a practical solution. Problem-centred approach proceeds sequentially starting from problem identification and motivation in such a way that the research might starts from problem observation or suggested future research. On the other hand, objective centred solution might originate from the second activity motivated by a specific problem from the industry. Design and development approach could originate from an existing artefact derived from previous research. In addition, DSR could start by observing a current solution, which means the activity is demonstrated backwards for ensuring a thorough and valid study. (Peffers et al., 2006; Peffers et al., 2007)

3.2 Design Science in Information Systems

Effectiveness and efficiency enhancement is the main promise of information systems in organisations. People, organization’s characteristics, competences, the environment, and development methodologies determine information system achievement. IS research aims to develop and disseminate knowledge so as to (1) aid development of creative information technology applications for organisations and its management (2) to disseminate knowledge concerning to management of information technology and importantly, the use of IT for management and organizational features. (Hevner et al., 2004)

Technological and social systems complexities are both involved in IS research. Gregor and Jones (2007) state that the main aim of research in information system (IS) is to study social and technological difficulties and the phenomenon that emerge when both social and technological aspects meet. Natural science research methods are good for studying and to understand current and developing phenomena. Moreover, problems in IS required inventive and creative solutions to be formed and evaluated in a specific context (Chatterjee, 2010).

Research in IS usually relies on two paradigms, which are; behavioural and design science (March & Smith, 1995). Behavioural science is built on natural science research that wishes to develop and defend theories that forecast social and organisational existence relating to design, analyses, information systems usage and development. Its main aim is to understand and study the correlation between organisations, information systems and people so as to promote effectiveness and efficiency. On the contrary, design science research is profoundly based on engineering discipline and science of artefacts (Simon, 1996). DS is a practical research that aims to use new and creative artefacts to solve everyday problems (Simon, 1996). Further, it involves IT artefacts creation and assessment for solving problem specific to an organisation (Hevner et al., 2004).

Behavioural and design science in information system research are inseparable even though there are some differences in their nature and origin. Both are connected in order to complement each other. Thus, behavioural science aims to develop and defend theories
that predict social and organisational existence relating to design, analyses, information systems usage and development, while design science research enhances innovative potentials and identifies emergent information systems capabilities. That is, behaviourial science uses theories to ascertain the truth, while DS aims to improve its effectiveness in a larger context. (Chatterjee, 2010; Hevner et al., 2004)

Hevner et al. (2004) recommended a set of guidelines to further explain how design-science research must proceed, evaluate, and be presented. These guidelines consist of seven practices described below:

1. **Design as an Artefact**: DSR must generate a feasible artefact that could come in many forms.

2. **Problem Relevance**: DSR product must address a problem.

3. **Design Evaluation**: It is important to verify evaluate the result in order to establish it solves identified problem.

4. **Research Contributions**: design artefacts, methodologies generated in DSR must be concrete and provable.

5. **Research Rigor**: The foundation of the solution must be thorough.

6. **Design as a Search Process**: Establish previous research in the domain of the identified problem.

7. **Communication of Research**: The result must be effectively communicated to the relevant stakeholders.

### 3.3 Implemented Research Methodology Process

The DSRM is used to carry out this study. Due to the nature of this research, design and development approach is applied to ensure that the study is rigorously conducted and the findings are reliable. This study follows the sequential process starting from problem identification and motivation. Figure 6 explains the design and development-centric approach employed for this study and its subsequent section describes the activities carried out during the research.

**Figure 2.** The Research Methodology Process Adopted in this Thesis
Problem identification and motivation

Many questions on CC issues have been answered and some are still unanswered. Since the advent of CC paradigm, many cloud service providers have evolved and many are still evolving. According to Olson (2012), as users are running to technology for help, many CSP are not able to deliver customers’ high demands.

Therefore, to answer the problems pertaining to CSP and project migration to the cloud, many solutions have been proposed to enhance the transition and the cost-benefit ratio such as frameworks, benchmarks, and analyses (Keung & Kwok, 2012; Misra & Mondal, 2010). However, according to previous studies and research (Li et al., 2010) for many IT professionals and managers it is still a challenge to choose a suitable cloud provider based on their services and benefits for a company’s project(s) or organisation because the available criteria are not useful in terms of components, services and structure. To provide a framework and guidelines of CSP and migration activities, this thesis employs previous research findings and systematizes knowledge about AWS and GCP to answer the following questions: What are the criteria to consider before for migrating to the cloud?, and What configuration should be organisation consider when using AWS and GCP?

Objectives of the solution

As pointed out in Chapter 2, previous research has been carried out to address cloud computing issues generally and different issues on migrating an existing application to the cloud. However, previous findings are difficult to be generalized because each proposed solution is suitable to a subset of cloud computing concerns. This study builds upon existing proposed criteria to deduce or develop a new and general solution to the problem. The objective of this solution are to increase understanding towards cloud computing characteristics and benefits, and to provide a framework that informs the IT community, end users, and management-oriented audiences about guidelines and current solutions for choosing a CSP, as well as about services and components that are available for their projects when hosting on AWS and GCP by answering the three research questions in Chapter 2.

Design and development

This phase of the project involves developing a JavaScript powered web application to present the CPSM framework deduced from literature and based on the questionnaire responses by IT experts. Software development best practices are used in designing and developing the artefact. Furthermore, latest programming models, design patterns, and model view controller (MVC) architecture software development frameworks are applied during the development phase. The CPSM framework design and detailed development process are carefully documented in the thesis.

Demonstration

Due to the nature of the research questions and available resources, I demonstrate a sample application hosting on both AWS and GCP infrastructure in Chapter 7 and also present the analysis of questionnaire in Chapter 5. Though not all the components of the AWS and GCP infrastructure would be utilized in the application, but important components will be presented. The demonstration involves tasks such as managing an application workflow, saving big data on the cloud, monitoring a cloud application, storing images and so on. Implemented demonstration details are presented in Chapter 7.
In addition, the tasks involved during migration to the cloud are also presented.

**Evaluation**

After the background review, a questionnaire is designed and analysed in Chapter 5 by asking IT experts their experience, perception of cloud and CSPs. The web application is also meant to demonstrate the workability of the deduced framework, in the sense that, the web application is developed using different criteria, similar with the ones used in the proposed framework, to automate the deduced criteria. The evaluation of the web application is further examined in the Discussion chapter in the light of the questionnaire results, previous research, and proposed guidelines.

**Communication**

This thesis communicates the research problem and its importance, the proposed solution and its usefulness to the computing community, as well as future work ideas to continue this endeavour. The thesis reports the design and development of the web application as well as the link from which the application can be accessed. Moreover, the thesis reports the results of the questionnaire answered by IT professionals to stress the usefulness of the proposed artefact. This research utility, novelty, and thoroughness of its design have been presented.

### 3.4 Questionnaire Design

As shown in Chapters 1 and 2, one of the objectives of this study was to examine IT professionals’ knowledge, experience, and perception of CC (adoption and migration). The main rationale was to evaluate the importance of this research, in terms of timeliness, that is, to gather current data about the problem as it has been identified in previous research.

Questionnaire is a way of gathering data from people in a particular area with the aim of getting scholarly opinions and ideas. Questionnaire is a convenient and easy method of data collection. It reduces cost and energy, and it usually gives adequate information for analysing the goal of the research work (Alseadoon, 2014). The questionnaire used in this thesis is a structured one, which involves predefined questions.

I have chosen descriptive analysis to summarize the survey answers, as this is the best suitable method for collected data.

### 3.5 Overview of Thesis Research Process

Table 1 below describes how this thesis reflects each aspect of the design science research processes.

**Table 1. Research process overview**

<table>
<thead>
<tr>
<th>Research Process</th>
<th>Thesis Chapter</th>
<th>Research question</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem identification and motivation &amp; Objectives of the solution</td>
<td>Chapter 2</td>
<td>RQ1, RQ2, RQ3</td>
</tr>
<tr>
<td>Design and development</td>
<td>Chapters 3 - 6</td>
<td></td>
</tr>
<tr>
<td>Demonstration</td>
<td>Chapter 7</td>
<td>RQ3</td>
</tr>
<tr>
<td>Evaluation</td>
<td>Chapters 8</td>
<td>RQ1, RQ2, RQ3</td>
</tr>
<tr>
<td>------------</td>
<td>------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Communication</td>
<td>Chapters 1-9</td>
<td></td>
</tr>
</tbody>
</table>
4. Amazon and Google Cloud Services

The biggest and most popular cloud providers are Amazon and Google. The Amazon cloud service is called Amazon Web Service (AWS), while Google cloud can be accessed using the Google Cloud Platform (GCP). This chapter describes the characteristics of these two services, namely it answers the following questions. What is AWS? What is GCP? What are these two infrastructure made of? What has contributed to their widespread adoption across different sectors? Is the structure resilient and secure? These are important questions that must be answered before setting out to use or migrate to AWS or GCP. Each infrastructure is structured differently and contains different components. It is possible to select which component to use and which one not to use, so it is not compulsory to make use of all components of AWS or GCP for your application. In each infrastructure, the name of the components might contain technical abbreviations that are listed in the abbreviation section of this paper. In addition, this paper delves into details to explain the components that are used in the application hosting in Chapter 8.

4.1 Amazon Web Services (AWS)

AWS is a cloud computing platform that is made up of collections of web services provided by Amazon. Amazon Web Services consist of many services that are either available to the public or not (Varia & Mathew, 2014). DeepField reported in 2012 that one-third of all the Internet traffic accesses Amazon’s cloud services. There are many advantages organisations and developers could derive from using AWS. (Lee, 2012)

According to Brantner, Florescu, Graf, Kossmann and Krasker (2008), AWS is flexible, scalable, and elastic, secure and have long time experience of IT infrastructure (see also Varia & Mathew, 2014). Organisations that wish to switch to AWS as its CC provider have the benefits and support for the technology, programming languages, and operating systems they have been using previously. With AWS one would not have to abandon the existing hardware and architecture when migrating to the cloud. In addition, it is also possible to select other tools that fit a client project.

Additionally, AWS cloud architecture provides its customers the ability to scale resources up and down without any hassle. Due to dynamic business environment, not being bordered about resources that your organisation/business would use over the year is a cost efficient and investment opportunity for small companies. That is, AWS provides the ability for reducing/increasing resources when ones business is gaining traffic or traffic is reducing.

To add more, security is an important aspect of CC as it involves trust and confidentiality of customers’ data. AWS had incorporated security into its services according to best security practices. Also, there are extensive documentations that explain how customers could use the security features. The long-term operation of Amazon enables AWS to be built based on experience of more than 15 years and Amazon continues to develop its skills and expertise. (Varia & Mathew, 2014)

The knowledge of the structure, component and usage of AWS will help in deriving best practises and guidelines for answering the research questions.
4.1.1 Structure

AWS structure is divided into four main blocks, namely, AWS Global infrastructure, Foundation services, Application Platform Services, and Management and Administration. Geographical infrastructure (namely the AWS Global infrastructure) is the main building block of AWS. Foundation services are services on which all other services sit. Application Platform services perform specific functions as part of the greater application. Finally, Management and Administration block consists of services that manage, monitor, and control all other AWS services. The Application section contains AWS customer applications. Figure 3 shows the architecture of AWS components. (AWS GI, 2015 and Mathew, 2014) This subchapter explains the functionalities of the services in each level of AWS structure.

The backbone of Amazon Web Services architecture is the **Global Infrastructure**, which has several availability zones that serve as datacenters. There are 9 distinct geographic regions (see Figure 4) that Amazon uses to host AWS across the world to serve many customers. The regions include: US West Oregon, US West California, US East North Virginia, Europe, Asia Pacific Singapore, Asia Pacific Tokyo, Asia Pacific Sydney, AWS GovCloud (AWS GI. AWS Global Infrastructure, 2015). In addition, there are also about 40 edges, which form part of the Amazon content delivery network.

![Figure 3. AWS Architecture (Mathew, 2014)](image-url)
Foundation Services represents the basis on which all other AWS services is founded and it consist of four categories; Compute, Storage, Database and Networking. EC2 means Elastic Compute Cloud, it consist of virtual private servers, which is an example of IaaS. EC2 will be used to host the application framework and more detail will be discussed in the next chapter. Auto Scaling is used to increase/decrease EC2 dynamically, i.e. scalability of applications when the need arises. The storage has many services that add to the strength of Amazon global network. Simple Storage Service (S3) is used to store any data of any format. It can store data such as images, style sheets, and document archives. On the other hand, Elastic Block Storage is used for storing EC2 instances continuously. It basically stores databases and file systems. AWS Storage Gateway connects file servers with S3, and also provides off site storage and backups. The database section of the AWS structure is used for scalable relational database systems such as MS SQL, MySQL, and Oracle. Amazon RDS is a relational database management system that will be used to store data in the application framework. More so, the database in the structure also contains scalable NoSQL databases that stores messages, image metadata and cache database lookups. Finally, the networking part of the above figure contains components that allow AWS to be used in a virtual network, scalable DNS services and so on. (Varia & Mathew, 2014; Mathew 2014)

Application Platform Services (APS) perform specific services that contribute to the general AWS functionalities. They include content delivery network (CDN) that allows distribution of both small and large file sizes (Robinson, 2008). This section also consists of services that could be used to send different notification via HTTP, emails and others. SNS and SQS are used for sending notification and file upload management in the application framework that will be developed. Also, this section is involved with services that are used for automated messaging between computers, services that are serves dynamic search platform, and services that host Hadoop framework used for data analysis are all part of the functionalities of the APS section. Further, Libraries and SDK that are used to interact with AWS are also part of Amazon application platform services (Varia & Mathew, 2014). AWS SDK for PHP will be used to develop the web framework. (Varia & Mathew, 2014; Mathew 2014)

Management and Administration section represents controlling services that include: web interface services for managing AWS, managing access to AWS, and monitoring AWS resources.
The last section of the AWS architecture is represented by **Applications**. This section contains either AWS customer applications or applications written and deployed by oneself on AWS. The customer applications are commercial software that is being charged hourly and could be delivered either as Amazon Machine Image (AMI) or as SaaS (Varia & Mathew, 2014). The cloud providers’ framework applications are developed and hosted on AWS. (Varia & Mathew, 2014; Mathew 2014)

### 4.1.2 Components

The components evolved for the better use of the customer and to provide best functionalities in one place. The paragraphs in this subchapter will explain functionalities and performance of different components of AWS that will be used for management and development of the application framework. Amazon divides the component into seven groups, which include; Compute and Network, Storage and CDN, Database, App Services, Cross-Service, Analytics and Deployment and Management (Varia & Mathew, 2014; Mathew, 2014).

Amazon EC2 fully means Elastic Compute Cloud. It belongs to the Compute and Network group of components. It allows renting of machine by clients to perform services for a specified period. EC2 provides scalable and easier compute capacity for both developers and system administrators (Mathew, 2014). Furthermore, upgrading or downgrading of server instances can be done in less than a minute with Amazon Auto Scaling flexible interface (Mathew, 2014). Amazon EC2 is not free (Brantner et al., 2008). However, you only pay for the resources used. It is worth mentioning that all requests made internally between EC2, S3 and SQS are free of charge. Therefore, according to Brantner et al., (2008) in terms of performance it makes sense to host applications on EC2 if some data will be saved on S3. In addition, applications hosted on EC2 are resilient to failure and are secluded from many situations that cause failure (Mathew, 2014).

As said earlier, EC2 is used for hosting the web framework application. The set up process and activities for hosting and maintaining applications on EC2 instance are explained in Chapter 7 of this thesis.

Amazon S3 means Simple Storage System. According to Baron and Kotecha (2013), S3 provides an easy web interface for storing and retrieving any amount of data, at any time from anywhere on the Internet (Brantner et al., 2008; Varia & Mathew, 2014). However, the size of each object in a container cannot exceed 5TB. All data stored on Amazon S3 will have access to the same scalable, reliable, fast, secure and economical infrastructure, which Amazon uses to power its websites globally (Varia & Mathew, 2014; Mathew, 2014). S3 uses container to store objects that are called buckets. S3 can store unlimited amount of objects. In S3, the objects can be retrieved using a unique identifier called key. It is important to minimize the number of metadata in the bucket in order to achieve fast and scalable object retrieval. It is possible to retrieve object in S3 by using a REST or SOAP protocol (AWS S3, 2015).

In the application I build, I use S3 to store images and style sheets and the REST protocol to retrieve objects in S3. Data consistency in S3 is achieved eventually. That is, S3 is not suitable for storing frequently changed data. In other words, updates that are made to objects in S3 are not visible to all clients at the same time (Brantner et al., 2008). For this reason, I use another component of AWS called RDS to store data.
Amazon Relational Database Service (RDS) is a relational database management system used for storing persistent data in the cloud. It means that it supports “JOIN” statement as opposed to NoSQL database system (Baron and Kotecha, 2013). According to Mathew (2014), RDS is economical and resizable storage system that manages difficult database tasks efficiently. Currently, prominent databases such as MySQL, Oracle, SQL Server, and PostgreSQL are all supported by Amazon RDS. That is, existing codes and tools used with the aforementioned databases can also work with Amazon RDS. In case of failure, RDS ensures safe and quick recovery of data (Robinson, 2008). With RDS, it will be easy to set up, operate and scale the data in the cloud.

In the proposed application, RDS is used to store data such as criteria for accessing cloud migration, list of cloud service providers, and list of all components, and so on. Using RDS, we do not have to worry about patches, and backing up of the data. Unlike EC2, it is not possible to connect via SSH and there is no root access for RDS (Baron & Kotecha, 2013).

4.2 Google Cloud Platform (GCP)

GCP is Google’s infrastructure that has been developed by Google over the past 15 years. It is the same platform that is being used for Google’s websites and apps. It is made up of physical network, thousands of servers and miles of fiber cable (Levy, 2012). It also, contains constantly updated software that is coded and re-coded for maximum efficiency. In 2012, Steven Levy of Wired called GCP as “The Mother of All Clouds”. Web applications, mobile apps and games can leverage virtually all the various parts of GCP (GCP docs, 2015). Google Cloud Platform provides extremely low cost utility computing in terms of computing cost and storage cost. (Xiaojing, 2010)

Google’s network is fast and vast. It is fast partly because it is optimized for speed and partly because it is vast. And the network that utilizes multiple enormous data centers across the globe is possible in part because it is so fast. GCP allows you to quickly develop and deploy mobile apps across multiple devices including iOS and Android. Large size files such as videos and high-resolution images are easily handles in GCP. Apps deployed on Google Cloud Platform scale up easily when required and no initial cost is required to use GCP, as it is pay as you go model. (Krishnan & Gonzales, 2015)

In addition, GCP can handle games apps impressively. All game media and large data files can be maintained through Google’s storage infrastructure. The rapid computation in the gaming environment is available on demand. GCP also provides flexible backends for both iOS and Android games. With GCP, it is possible to synchronize game states across multiple devices and one could distribute and monetize games through numerous Google related services. (Krishnan & Gonzales, 2015)

4.2.1 Structure

As with Amazon Web Service, Google Cloud Platform has 3 distinct regions that serve its global infrastructure and worldwide users. Each region has 3 or 4 zones. According to Figure 5 below, US central region is located in Council Bluffs, in Google’s Iowa data center and has 4 availability zones. The data centers in Western Europe (Belgium) and Eastern Asia (Taiwan) have 3 availability zones each, serving Google Cloud Platform’s European and Asian customers respectively.

Each availability zone runs either Ivy Bridge, Sandy Bridge, or Haswell processors, therefore, all instances in a zone make use of the processor that is supported in that zone.
In addition, each zone has 32-core machine types used for virtualization of hardware that is available for each instance and a local solid-state drive (SSD) for each virtual machine instance. According to the GCP website, it is possible to create many instances across many availability zones in many regions (GCP Regions & Zones, 2015; Krishnan & Gonzales, 2015. p.14). Google Cloud Platform components are structured into 4 groups of components; these include Compute and Hosting, Storage, Big Data and Services (see Figure 6; GCP Structure, 2015).

Figure 5. Google Cloud Platform Regions and Availability Zones (GCP Regions & Zones, 2015)
Figure 6. GCP Structure. (GCP, 2015)

**Compute and Hosting** section of the GCP is made up of both Platform as a service and Infrastructure as a service components. These components are integrated with other components of GCP to deliver robust hosting and computing infrastructure for different web and mobile applications, and games. Additionally, they provide high computing power when your application needs to support large scale computing workload. Compute and Hosting consists of App Engine, Compute Engine and the newly added Container Engine, which at the time of this project, is still in Beta state. (Krishnan & Gonzales, 2015. pp. 25-24). According to Severance (2009) these components are used for building, deploying and maintaining applications.

In the proposed framework application, App Engine is used for hosting. In order to present the application, I use 2 components from the Storage section to store the application’s assets and data. The next subchapter describes each component in detail.

GCP is a major service provider in the **data storage** field backed by its robust group of storage components that are available throughout the globe. As it is with the Compute and Hosting section, most, if not all, the storage components work seamlessly with other components. For example, if the application framework is hosted on App Engine, it will pull assets and data from Cloud Storage and Cloud SQL. The Storage components provide support for storing relational and non-relational data as well as storing files, archives and backups. With Storage components, there is support for MySQL, NoSQL and ACID (Atomicity, Consistency, Isolation and Durability) transactions with automatic replication of assets and data across multiple datacenters. This section also responsible for handling huge amount of data ranging from terabytes to petabytes with minimal latency. (Krishnan & Gonzales, 2015. pp. 25-24; Cohen, Hurley & Newson, 2015)

**Big Data** section contains components that are rapidly evolving. They include BigQuery, Cloud Dataflow and Cloud Pub/Sub. Google Cloud Pub/Sub is used for managing asynchronous publication subscription messaging between applications. GCP BigQuery component is a fast data analyzer that is based on the artificial intelligence of online
analytical processing called OLAP. BigQuery is not a relational database, meaning it does not support INSERT, UPDATE or DELETE and indexes.

In the application framework, none of the components in this section will be used therefore, not many details will be provided. The final section provides Google services such as Prediction API for data analyses and trend forecasting via normal RESTful interface and Translate API for generating multilingual output. (Krishnan & Gonzales, 2015. pp. 25-24)

4.2.2 Components

It is important to understand what each GCP component does and how they work together in order to make the best use of it. GCP integrate full spectrum of products and services that is constantly evolving (Abrams, 2014). Xiaojing (2010) ascertained that Google provides the lowest services in terms of computing and data storage utility due to its architecture and components. This section will describe the functionalities and accessible of these GCP components that are used in developing and hosting the application framework. App Engine, Google Cloud Storage and Cloud SQL will be explained.

As said in the previous subchapter, Google App Engine is a platform as a service (PaaS) cloud service model. It is used for building, deploying and maintaining web applications, mobile applications and games. With App Engine, its users gets complete server side management tool that automatically scales application for an unlimited number of users (Severance, 2009). According to GAE (2015) web page, App Engine provides a secure, robust platform to make applications a reality. In App Engine, automatic scaling is not limited to only needed instances, but also shared databases and expanded bandwidths are scaled as necessary. Getting started with App Engine does not require start-up costs, although, there is a limit to the amount of resources that can be used for free. Each GCP account could create 10 different applications with App Engine and each application can have up to 1GB of storage and 5million page view per month. App Engine offers application developers the ability to control Compute Engine cores and provides a web facing front end for Compute Engine data processing applications. On the other hand, Compute Engine provides direct and full operating system management for App Engine’s virtual machines. Furthermore, App Engine currently supports application development with 4 different runtime environments that includes Python, Java, Go and PHP (GAE Features, 2015).

To add more, App Engine relies on Cloud Storage and Cloud SQL for storing assets and data. It has API for managing, storing, retrieving, displaying and deleting files in Google Cloud Storage. Naturally, App Engine has adequate connectivity to both Cloud SQL and Cloud Datastore through its APIs and many modules. (GAE Features, 2015; Krishnan & Gonzales, 2015. Pg. 83-89)

The application framework proposed in this thesis uses PHP as the development language, and API to access data stored in Cloud SQL.

As the digital storage business is getting bigger and bigger nowadays, Google Cloud Storage is useful for storing digital content, and retrieval from any capable computer or device. Many personal storage companies like Dropbox, ICloud, OneDrive and Google Drive provides limited storage. However, Google Cloud Storage for application storage does not have limit. It uses Google’s infrastructure that is made up of datacenters around the globe as said in the previous section. In other words, it means near infinite storage,
high security, and very fast retrieval due to the fact that Cloud Storage implements Edge Caching that serves content closest to the user for minimal latency. Cloud Storage is a pay as you use service with its 2 different pricing levels, the standard level, which has high data availability, costs 0.026-dollar cents for each gigabyte per month, while the Durable Reduce Availability (DRA) attribute costs 0.02 dollar cents per gigabyte per month. (GCS Pricing, 2015) It is possible to assign different pricing level for different group or bucket. That is, one could use the standard pricing for most assets and DRA for backup data. To elaborate more, Cloud Storage is bucket and object oriented and it is highly structured. Within projects are containers called buckets and each bucket is made up of objects. That is, a bucket is the main storage container in Google Cloud Storage. In GCS, buckets are project-specific and each provides access level control, meaning it is not possible to share objects from one project another and it is not possible to next objects. (Krishnan & Gonzalez Google Cloud SQL Overview, 2015, p. 185; Cohen, Hurley & Newson 2015) Chapter 7 illustrates the use of buckets and objects in Cloud Storage for the application development proposed in the thesis.

Additionally, Google Cloud Storage (GCS) provides secure digital sharing due to its support for Access Control Lists (ACLs) with full OAuth2.0 authorisation so that users can be sure that the data is being shared with intended users. Moreover, GCS can also be used to host static website contents. GCS is accessible through many numbers of ways that includes GCS manager: - a web based interface for managing Google Cloud Storage, standard HTTP protocol access that supports normal file transfer commands like GET, PUT and POST, programmatic transfer through REST interface and finally, through gsutil: - a command line interface written in Python. (Krishnan & Gonzalez, 2015. pp. 185-210; GCS overview, 2015)

GCP offers two separate data-oriented storage services, namely Cloud SQL and Cloud Datastore. Both services belong to Storage section of the GCP structure (see Figure 6 above), however, details about Cloud Datastore will not be done in this thesis because the application framework uses the relational Cloud SQL. Cloud SQL is based on MySQL relational database. Cloud SQL supports all power of MySQL across multiple datacenters with automatic data replication. In addition, it is possible to import and export existing databases through MySQL dump. With Cloud SQL, applications can be hosted in US, Europe or Asia with 100GB of storage and up to 16GB of RAM for each database instance. As it is with other components, Cloud SQL can be accessed through number of ways, these include either via Java or Python, via Java-based command line tool and also through SQL prompt that is available in the Google API console. (Krishnan & Gonzales, 2015. pp. 159-183; Google Cloud SQL Overview, 2015)

However, as at the time of this thesis, Cloud SQL implementation has some restrictions such that it can only be used with MySQL 5.5 or higher and instance size is limited to 500GB for each instance. In addition, some functions like LOAD FILE(), SUPER() and SHA() use for encryption does not work on Cloud SQL. To add more, Cloud SQL integrates easily with other components of Google Cloud Platform. It uses Cloud Storage to handle import and export and it is also accessible from either App Engine or Compute Engine. (Krishnan & Gonzales, 2015. pp. 159-183; Google Cloud SQL Overview, 2015)
4.3 Comparison between AWS & GCP

This section discusses the similarities and differences between AWS and GCP in terms of pricing, performance, structure, interface, and business model. Most of the findings are based on my experience and related research.

4.3.1 Structure and technical capabilities

Regarding the structure, AWS have 9 regions and 40 edge locations, which is different from GCP with 3 regions and 10 availability zones. That is, AWS have wider geographical presence than GCP. This mean AWS is able to serve its customers across the globe faster than GCP.

In addition, an area that has caught attention of many AWS users is the size of persistent disk (PD), which could contribute to choosing GCP over AWS. AWS provide 1TB in its Elastic Block Storage while GCP supports 10TB. Furthermore, customers are able to connect many instances in read-only mode to persistent disk, which allows proper distribution of workforce. However, this opportunity is not available in AWS. Still on performance, AWS provides 6 categories of instances that consist of 29 instance types, while GCP splits instances into 4 categories with 15 types of instances. Finally, at the moment it is possible to run GCP compute engine on Linux and Windows. This means it is possible for 53% of AWS customers to use Google’s Compute Engine. (Wexler, 2014)

4.3.2 Performance

In terms of performance, GCP compute engine is best suited for unexpected rise in traffic because it does not require users to configuring pre-warning before responding to unexpected increase in traffic. That is, Compute Engine load balancers scale immediately to an increase in traffic. Opposing, AWS Elastic Load Balancer (ELB) require setting pre-warning in order to respond to spike in traffic (GCP Compute Engine, 2016). In addition, with AWS customers need to subscribe to AWS support to get notification during traffic increase and because of this, AWS associates AutoScaling to ELB.

As shown in this chapter, AWS provide numerous services, which mean great opportunities for different purposes. From database services for relational, non-relational and NoSQL databases to streaming services and media transcoding services for its customers to convert any media files to playback version that could be used on mobile devices and personal computers. In addition, AWS also provides a managed service called AWS Directory Service that allows users to connect and manage their already existing Microsoft Active Directory on AWS cloud resources. AWS provides a complete and well integrated cloud platform and lets you have high productivity with cloud services from one vendor. On the contrary, GCP have limited services providing mainly IaaS and PaaS services. Although, GCP provides normal services that serves most common needs. From computing services, object storage, relational and non-relational databases to Endpoints that is made up of tools and libraries for generating API from App Engine and DNS services for managing and publishing DNS records. Though, GCP provides excellent massive amount of data analysis with the Big Query component and it is easy to use.

4.3.3 Business models and pricing

In the area of pricing, GCP is preferable in my opinion. Both GCP and AWS provide pay per use services, however AWS charge based on hour for EC2 instance even if one use 5
hours and 5 minutes. It means you will end up paying full hour for using a fraction of an hour. On the other hand, GCP offers real pay per use billing model with the Google Compute Engine. Therefore, you only pay for the minutes used in GCE except the 10 minutes minimum charge (GCP Pricing, 2015 and Krishnan & Gonzales, 2015. p. 11). In addition, GCP offers sustained use discount that basically means customers get automatic savings when an instance or machine is being used for more than 25% in a month. (GCP discount, 2015). Because of this, AWS struck back with price reduction in its Reserved Instance pricing in order to level up with Google’s sustained use model (AWS RI, 2015).

4.3.4 User interface and learning costs

Regarding the user interface, in my opinion, GCP Compute Engine user interface is clearer compared to AWS EC2 interface. Historically, AWS has been the front-runner and GCP started late after Amazon. Therefore, GCP is able to build on AWS lapses so as to gain edge (Krishnan & Gonzales, 2015. p. 11).
5. Cloud Computing Survey

This section contains analysis of the data gathering method used in order to answer the first research question, What are the opinions and attitudes of IT experts with regard to CC, CC adoption, and CC migration? The aim was to establish that the problem that this thesis set out to solve does exist, namely that IT practitioners find it difficult to migrate existing applications to the cloud, and it is not easy to select a suitable CSP. Also, the survey is used to find out the perspective of IT expert about cloud computing paradigm.

5.1 Survey Data

The survey contains twenty-two questions covering cloud migration, cloud service providers, cloud services, and expert’s background information. The survey gathers the experience with and opinions about CC in work-related context. Cloud migration survey questions asked IT experts about their perception about migrating a project to the cloud. Moreover, questions covering CSPs asked about what providers IT experts have worked with. The background information asked about the background and location of the respondents’ industry.

The table 2 below shows summary of the survey questions, percentage of responses and total number of responses for each survey question.

<table>
<thead>
<tr>
<th>Survey Questions</th>
<th>Responses (%)</th>
<th>Number of respondents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Do you have work-related experience with a cloud computing system or application?</td>
<td>Yes: 90, No: 10</td>
<td>10</td>
</tr>
<tr>
<td>2. Which of the following cloud service providers or vendors have you worked with?</td>
<td>Amazon: 30, Google: 50, Others: 20</td>
<td>9</td>
</tr>
<tr>
<td>3. In your opinion, what are the benefits of using or migrating to cloud computing?</td>
<td>Network speed: 80, Price: 20</td>
<td>9</td>
</tr>
<tr>
<td>4. How have you chosen a suitable cloud provider or vendor for your project?</td>
<td>Friends: 40, Trial: 40, Own initiative: 20</td>
<td>10</td>
</tr>
<tr>
<td>5. Are you considering changing your cloud provider(s) / vendor(s)? If No, why?</td>
<td>No: 80, There are many CSP, not easy to select.</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Yes: 10, No answer: 10</td>
<td></td>
</tr>
<tr>
<td>6. How would you rate your knowledge/awareness of the services offered by your current cloud provider(s) and/or vendor?</td>
<td>Knowledgeable: 30, No answer: 10, Not knowledgeable: 60</td>
<td>10</td>
</tr>
<tr>
<td>7. In your opinion, how suitable is cloud computing for your project(s) or</td>
<td>Suitable: 70, Don’t know: 20</td>
<td>8</td>
</tr>
</tbody>
</table>
organisation?

8. How did you find the documentation or guidelines for Amazon Web Services and/or Google Cloud Platform? Documentation at the vendor’s site and available community forum.

Find GCP better: 50,
Find AWS better: 40,
Others: 10

9. For the future, are you considering using or migrating your project(s) to the cloud?

Yes: 90,
No: 10

10. What is your main reason for using or migrating to a cloud computing system?

Network speed: 80,
Price: 20

11. In your opinion, what are the benefits of using or migrating to cloud computing?

Flexibility: 40,
Network speed: 40,
Price: 20

12. In your opinion, what are the disadvantages of using or migrating to cloud computing? Please describe them briefly.

Security: 50,
Privacy: 30,
Price: 10,
Awareness: 10

13. Regarding the migration of an existing application to the cloud, what difficulties and concerns have you experienced in the past and/or do you think you would experience? Please describe them briefly.

Complexity: 30,
Documentation: 40,
Technical know-how: 30

14. How helpful would you consider a system that could suggest the type of cloud provider/vendor that best suits your project(s)?

Helpful: 80,
Not helpful: 10,
No answer: 10

5.2 Survey Administration

The survey was carried out between the end of September and October 2015. The first week of the survey month, I sent the survey link through email to IT experts I have worked and studied with. Not many responses where gotten, then I went to an IT hub in the city of Oulu, Finland and I distributed printed copy of the survey and got their responses within two hours. More than twenty printed copy of the survey was distributed but only six responses were gotten. In total I obtained 10 responses (see Table 2).

5.3 Survey Participants

The ten survey responses capture the view from different experts from three countries, many of the respondents are from Finland, some from Nigeria, and others from England. More than 60 per cent of the respondents work in information technology and communication field and others work in media and entertainment field but have an IT role. It is worth saying that based on the responses, above 50 per cent of the responses are from IT experts who have worked in their respective companies for an average of 3 years. Also, more than 50 per cent of the respondents’ works in companies that have been in existence for 10 years, therefore a higher percentage have a dedicated technical or server administrator.
5.4 Discussion of Survey Findings

The most important findings from the survey is the concern about security (question 12) in the cloud and acknowledgement that there are so many cloud service providers (question 5) that is hard to select a suitable one for a project due to lack of information about cloud vendor’s services. The findings in question 14 of the survey support the need to have the framework and information presented in Chapter 6.

Furthermore, 90 per cent of experts wish to migrate their projects or organisation computing to the cloud in the future. They mentioned benefits such as price, scalability, availability, accessibility, maintainability, network speed, and centralisation. However, some disadvantages and concern were mentioned, like security, which is the main concern, privacy, inadequate knowledge or information, vendor lock in and some percentage of experts complained of the price and customization. Although higher percentage of respondents consider migrating their existing application or project to the cloud, the disadvantages such as documentation, learnability, technical know-how, platform compatibility and server and connection problems make them see the need for an application or system that could suggest suitable vendor for their project. Thus, these survey results confirm the importance of the problem this thesis sets out to solve.

In addition, the survey did not assume that because the respondents were experts they would have adequate knowledge of CC. Cloud is still a new technology paradigm in many organisations. Therefore, I provided background information about CC in the survey. Many of the respondents have used Software as a service, Platform as a service, and Infrastructure as a service from vendors such as Google and Amazon services. Others have used Microsoft, IBM and so on. Also from the survey, it was found that experts chose a cloud provider based on recommendations from colleagues, and others used their own judgement. Further, many difficulties and concerns about their current cloud system were mentioned such as poor documentation, interoperability problem, not a flexible system, poor usability and user experience, security, ability to choose suitable one from many available providers, and so on. This is also reflected in the response about willingness to change cloud service providers because of the stress and contract lock in. To add more, more than 50 per cent of these experts cannot boast of having knowledge of all the services being offered by their cloud vendors and they consider cloud computing as a very suitable tool for their organisation. Many of the respondents find Google cloud services documentation and guidelines easier to understand without prior knowledge compare to Amazon web service documentations.
6. Migration to the Cloud

This chapter provides cloud migration criteria, guidelines, and a framework that summarizes the criteria and guidelines for using AWS and GCP as cloud service provider.

6.1 Cloud Migration Criteria and Guidelines

This section provides criteria for cloud migration by examining previous research, analysis of AWS and GCP services, and the cloud computing survey in Chapter 5. Based on these criteria and guidelines, I deduce the factors that will make up the cloud vendor framework in section 6.2.

To answer the second research question, what criteria are relevant for companies when migrating to a cloud service provider?, and to identify which CSP is suitable for an organisation, one has to consider the organization’s details such as the configuration and nature of the organization or application being migrated to the cloud. Olson (2012) proposed answering a series of questions that help deciding on the suitability of a vendor for own organization. The questions by Olson (2012) reflect possible criteria for vendor selection, which I will further simplify and discuss in Section 6.2 for proposing the selection framework. The criteria derived from the Olson’s questions are:

1. Vendor track record in CC,
2. Availability of trial period of testing,
3. Types of Service Level Agreement and Service Level Performance,
4. Price protection and type of contract provided,
5. Transparency record as a CSP,
6. Sharing of the provided infrastructure with many customers,
7. The existence of a disaster recovery plan,
8. Compliance with the International Standard Organization (ISO) in terms of security,
9. Configuration of migration choice, and
10. Integration of support availability.

According to Chapter 5, criterion 2 is related to the opinions of survey participants for Q5 in the survey regarding the changing of CSP, by having a trial period people can decide easier to change to or migrate to a cloud. Also, criterion 8 is necessary as it shows in the survey Q12 where security is among the reasons many experts will avoid migrating to the cloud. Integration of support availability could mean a flexible cloud services. Flexibility happens to be one main reason for using cloud infrastructure according to the survey.

To answer the third research question, what are the guidelines to consider when configuring application on AWS & GCP?, it is important to discuss general things that could be achieved in both AWS and GCP but in different ways. The following subsections and paragraphs discuss some of the best practises for configuring organisation’s infrastructure in AWS and GCP cloud services. First, I provide general guidelines such as thinking about failure, that is; putting failure at the back of the mind as you operate in the cloud, elasticity, loose coupling, etc. Then, security guidelines are
provided. These form the guidelines to configure all applications migrating to AWS & GCP.

6.1.1 General Guidelines for Migrating to AWS and GCP

The guidelines are based on Varia (2014), and the structure of AWS and GCP explained in Chapter 4.

Think about failure. Werner Vogels as cited in Varia (2011) claims “everything fails all the time”. In my opinion, he sounds pessimistic. However, from people that have much experience in technology, it is obvious that this is true. His statement is to get cloud users in the right mindset to be thinking about failure and design accordingly for it. The idea here is that while things will always fail at the individual component level, if you architect the overall system which often compose of multiple and often redundant components along with implementing good practises to minimize dependency between these components, it is possible to create a very stable and fairly fail-proof system. So the idea is, if you build the system expecting part of it to fail and mitigate that risk to failure in your architecture, your overall system will not fail.

Implementing Elasticity. Another general guideline for migrating to cloud refers to elasticity. Elasticity can be defined as the ability to scale cloud resources be it up or down (Varia, 2014). This scaling can be achieved in a few ways such as 1) schedule-based proactive scaling; 2) event-based proactive scaling, and 3) automatic scaling (Varia, 2014). Schedule-based proactive scaling resizes the resources in a regular fixed basis, for instance, daily, weekly, or quarterly. This could work in the case of predictable traffic patterns and known demand for hubs and flows. Event-based proactive scaling scales the system components just when some events are expected to happen. Typically, this will be scaling out or up due to some forthcoming marketing campaign or events that predict sharp increase in demand. Automatic scaling is a method highly recommended to set up monitoring of certain metrics, for example, central processing utilization, or network Input/Output; when the metrics reach certain thresholds, the components scale automatically either out or in based on whether the metric rose above or fell below some predefined thresholds.

Loose Coupling. In software architecture, loose coupling means a design principle concerned with minimizing dependencies between components. Also refers to loosely coupled components in order to improve the scalability of the system. The idea is to strive for a system where if one of the components in the architecture would have failed for some reason, the other component can keep matching happily on as if no failure occurred at all (Varia, 2014). A common example of this can be seen in a typical web application architecture where there are several web servers that connect to several application servers, which in turn connect to a database. The application servers are typically not aware of the web servers and similarly the web server do not necessarily needs to know much about the application servers. Especially where they are connected via a load balancer or some other components that facilitates distribution. Achieving loose coupling allows the system to easily add more components or remove them to match demand without impacting the other components that are happily working away.

Keeping Things Secure. The idea of hosting and storing data in the cloud is still relatively new. With everything new, especially when it comes to business and organisation data, security is of utmost importance. There are all sorts of rumours and
tales of cloud security challenges that have spread, which have caused some to be reluctant to warm up to cloud computing. Many perceptions about security of the cloud stem from a lack of understanding that the customer is still responsible for a lot of the security implementation (Bamiah, Brohi, Chuprat, & Brohi 2012; Hamouda, 2012). In general, the physical security of the building, the infrastructure, the equipment, and keeping customers separate from each other are the responsibility of providers; for example, Amazon is responsible in terms of AWS users, and Google is responsible with respect to GCP users. However, on the other hand, the users are responsible for the security implementation in the network and application level.

**Cloud infrastructure.** According to Tran et al. (2011), after considering and solving programming language supports, data storage, and differences in software environment and performance qualities, the migration can be started. In my opinion, there is no need to consider the development state of the cloud. The state of cloud computing in 2006 was different from its stage at the moment, and cloud computing is the future of computing (Linthicum, 2015) so it is better to migrate so as to have infrastructure efficiency and better utilization, and operation cost reduction (Keung and Kwok, 2012). Undoubtedly, security and privacy are the greatest challenge according to research by Bamiah et al. (2012) and security and privacy is not limited to cloud infrastructure but also traditional IT system. (Bamiah, Brohi, Chuprat, & Brohi, 2012; & Hamouda, 2012; Tran et al., 2011)

6.1.2 Security Guidelines for Migrating to AWS and GCP

Varia (2014) recommends the following things to keep in mind in order to establish a secure cloud based applications:

**Protecting data in motion by using SSL:** manage SSL certificates either directly on an EC2 instance or the load balancer level, which is recommended if you intend to use multiple EC2 instances to serve similar Hypertext Transfer Protocol, secure (HTTPS) request.

**Protect data at rest:** encrypt data before storing them on any storage devices; you also encrypt the entire file systems. Both AWS and GCP protect data at rest. However, AWS file system storage comes in two general varieties; Elastic Block Storage which persists beyond the lifetime of the underlying instance, and Local Storage which will not survive after termination of the storage on which it resides. Both types of storage can be encrypted.

**Protect access credentials:** As demonstrated in Section 7.3, AWS applies two types of security credentials; AWS Access Keys and X509 certificates. The AWS access key has two parts to it (1) A public access key ID and (2) A secret access key. When using API, one has to use the secret access keys in the request for authentication. As such all API requests sends from the public Internet should be sent over HTTPS. If an application uses other AWS services which require secret access key, rather than having the secret access key to be part of your application you should build your application in a manner such that this key is passed as an argument during launch of the application and encrypted before sending or you can launch your instance in an IAM role, as such the instance will have access to the credentials associated with that role. If your secret access key becomes compromised, you should obtain a new one by rotating to a new access key ID. It is strongly recommended that you rotate your keys often to make sure compromised key will not live forever. You also want to use IAM to manage access control. IAM is AWS service that allows you to create users and manage their permission by assigning them
role and placing them into group, this has been demonstrated in the chapter that demonstrated how to host applications on AWS. Rather than handing out the root account information for everyone that needs access, it is strongly recommended that one create separate user for people needing access and then granting them access only to the services they need access to. You also need to secure your own application. AWS provides security groups, which act as firewall to the associated instances. Always make sure you lock this down and restrict access to the instances. You also want to make sure you update all software packages and apply all security patches (Varia, 2014). GCP too uses end-to-end encryption, which involve the use of private and public key encryption that allow only intended user to have access to the data.

6.2 Cloud Provider Selection Model (CPSM) Framework

Cloud Provider Selection Model is a framework derived from the survey in Chapter 5, the criteria and guidelines in Section 6.1, my experience as a software developer, structure of AWS & GCP and previous research to help with selecting CSP.

The challenges involved in choosing between AWS and GCP, as a public cloud provider to suit a project is the main focus of this paper. CPSM is deduced based on the criteria in the above Section 6.1, to help prospective cloud users select between these major vendors in order to have good application performance and efficiency. CPSM is made up of six main characteristics of AWS and GCP services that must be considered before deciding to pick either AWS or GCP. They are:

1. Years of operation and experience of the CSP.
2. Application support.
3. Amount of available services.
5. Pricing model.

Each of these main characteristics is made up of different factors that must be taken into account while determining vendor suitability. The weight of each factor is calculated from the position of the main characteristic. For example, geographical presence is the fourth characteristics with regions and availability zones as factors. Many availability zones can exist in a region, which means no availability zone (AZ) without region. That is why AZ has weight of 1.5 and region has 2.5 making a total of 4.

6.2.1 Years of Operation and Experience

In my own opinion, number of times a cloud vendor had experienced downtime and recover from the disaster is a factor that will contribute the quality of services and supports provided by the vendor. Equally, number of years of services signifies more cloud infrastructure experience. Olson (2012) also stressed this in subchapter 6.1. AWS started with cloud infrastructure in 2002 while GCP started in 2006 (Arif, 2016). Of these years, AWS have experienced more outages than GCP according to downdetector.com. Thus, two sub-factors are identified:

1. Disaster recovery
2. Number of years of public cloud offering
6.2.2 Application support

This is more important for projects or organisations that do not have in-house IT department. Thus, having a good documentation and customer support to solve any issue in regards to the cloud vendor are important. Programming languages support is also important in order to extend or customize already existing services by the provider. Based on my experience with both service providers, GCP has supportive community and online forum coupled with fast customer service response. Also, according to the survey response many experts finds GCP (fifty per cent) better than AWS (forty per cent) in terms of documentation and supports. The guideline 10 in proposed by Olson (2012) in subchapter 6.1 also matches these criteria. Two sub-factors are identified:

1. Documentation and customer support
2. Programming language support

6.2.3 Amount of available services

The number of services and interoperability of cloud services is another aspect of cloud vendor suitability that is vital in my own opinion based on the survey responses. This is important because if one could get all required services in a vendor, there is no need to have cross vendor services, which may cause additional overhead. As at 2016, AWS have more than 60 services with each service able to interact with one another without using any third-party application (AWS Dashboard Home, 2016). On the other hand, GCP have around 35 services though some are still in beta stage. (GCP Products, 2016). the two sub-factors identified are:

1. Number of services
2. Interoperability with other services or application

6.2.4 Geographical presence

According to Varia (2014) and GCP Regions & Zones (2015), choosing regions and availability zones closer to the cloud users will guarantee faster speed, high availability and performance. This is why considering the regions and availability zones closer to ones business is important for choosing suitable cloud vendor. Two sub-factors are in this category.

1. Regions
2. Availability zones

6.2.5 Pricing model

The two kinds of pricing models that are being used by cloud vendors (Gorelik, 2013) should be considered because this may reduce the cost of cloud adoption. GCP offers both pay per use and pay per hour pricing model while AWS only have pay per use model. (GCP Pricing, 2015 & AWS Pricing, 2016)

1. Pay per use
2. Pay per hour
6.2.6 Security

According to Tianfield (2012), security is an integral part and the main reason why there is slow cloud adoption (Tianfield, 2012). Also, Olson (2012) recommends making sure a cloud vendor satisfy security audit before hoping for the vendor. Based on my experience, it is also important to find out the last time the security was updated and if an updated security certification is available. The two factors to consider under security are;

1. Security certifications
2. Last updated

The table 3 below gives summary of factors and the maximum weight for each factor.

**Table 3. Cloud Provider Selection Framework**

<table>
<thead>
<tr>
<th>Main characteristics</th>
<th>Factors</th>
<th>Weight</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Years of operation</td>
<td>- Disaster recovery</td>
<td>- 0.3</td>
<td>Own experience and Olson (2012)</td>
</tr>
<tr>
<td></td>
<td>- Years of cloud offering</td>
<td>- 0.7</td>
<td></td>
</tr>
<tr>
<td>2. Application support</td>
<td>- Documentation and customer support</td>
<td>- 1.0</td>
<td>Survey response and Olson (2012)</td>
</tr>
<tr>
<td></td>
<td>- Programming language support</td>
<td>- 1.0</td>
<td></td>
</tr>
<tr>
<td>3. Amount of available services</td>
<td>- Number of services</td>
<td>- 2.0</td>
<td>Survey response</td>
</tr>
<tr>
<td></td>
<td>- Interoperability with other services or application</td>
<td>- 1.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Availability zone</td>
<td>- 1.5</td>
<td></td>
</tr>
<tr>
<td>5. Pricing model</td>
<td>- Pay per use</td>
<td>- 2.0</td>
<td>Gorelik (2013) and Olson (2012)</td>
</tr>
<tr>
<td></td>
<td>- Pay per hour</td>
<td>- 3.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Last updated</td>
<td>- 3.5</td>
<td></td>
</tr>
</tbody>
</table>

The findings in chapter 5 contributed to the metric number given to the factors in the Cloud Provider Selection Model (CPSM) table above, that is, security is given higher value in the CPSM framework because it is the main challenge mentioned by experts preventing them from not wanting to use the cloud, documentation and programming language support were mentioned the same number of times by different respondents as being among the difficulties for cloud vendor selection, and so on.
CPSM framework is different from previous research frameworks because it is not based on performance and economics of cloud computing adoption. Rather, the criteria are practical, and measurable. Its analysis takes into account the architectural and components of AWS and GCP. One does not need to be an IT expert to use CPSM framework for selecting cloud vendor, which is not possible with research by Keung and Kwok (2012) and Misra & Mondal (2010) (See Appendices 10 and 11 for overviews of Keung & Kwok model and Misra & Mondal models, respectively).
7. Application Development and Testing

In this chapter, I describe the application developed for selecting suitable CSP, access either to migrate or not, and to demonstrate hosting application on AWS and GCP. The application name is KlaudCelet, which signifies “Cloud Selection”. The chapter details the languages and tools used in developing the application.

KlaudCelet is a web application that suggests suitable cloud service providers for an organisation based on the CPSM factors in Chapter 6 of this paper, which solves one of the purpose of this research (RQ2) as stressed in Chapter 2. The current beta version reported in this thesis suggests cloud providers based on pricing model and industry type. In addition, KlaudCelet suggests cloud deployment models based on the framework deduced by Keung and Kwok (2012), which is the first step that must be fulfilled before migrating to the cloud. KlaudCelet additionally provides cloud migration factor to consider based on the research by Misra and Mondal (2010), which are another criteria for migration to the cloud. (See appendix 11)

KlaudCelet is made with AngularJS, a JavaScript frontend framework; PHP, a server-side scripting language; and MySQL, the server-side database management system that is used mainly on the web. The current beta version of KlaudCelet has not incorporated the proposed CPSM framework, however for future work the final version will contain the CPSM implementation.

7.1 KlaudCelet Interface and Functionalities

Figure 7 below shows the first page that will appear when you visit the application URL after hosting it on GCP, AWS or locally. The application URL could be http://localhost if it is hosted locally or appname.appspot.com if it is hosted on GCP or ec2-203-0-113-25.compute-1.amazonaws.com if hosted on AWS. The current version incorporates four pages: Home, Buy/Rent, Migrate, and About that can be accessed from the top-right menu (see Figure 7).

![KlaudCelet Front Page](image)

**Figure 7:** KlaudCelet front page
The front page (that can be also accessed by selecting Home page) shows the list of all cloud service providers with their images as the provider name, description of their services and the deployment model offered by each provider. To filter the list, typing the industry, provider name and cost (either pay per hour or pay per use) will reduce the displayed providers. In addition, clicking on a provider name image will display more information about the provider’s services (see Figure 8).

![Google Cloud Platform](image1)

**Figure 8**: Cloud service provider’s details page

The Buy/Rent page describes criteria for deciding on renting or buying a cloud service. In this page, the entered figures are used to determine if a company will benefit more in renting a public cloud or buying a ready-made private cloud provider based on, the calculation proposed by Keung and Kwok (2012) (see Chapter 2). Keung and Kwok used the Cloud Deployment Selection Model (CDSM) that has been empirically tested with two real business cases. In the model, the authors used different recommendation factors and analysis by experts and previous researches to come up with this model. Based on the eight factors used in the model (see Appendix 10) if the result is less than or equal to 180, it is advised to rent a public cloud while if the result is less than or equal to 370, buying a private cloud is advised.

![Buy/Rent page](image2)

**Figure 9**: Buy/Rent page
The Migrate page (as in Figure 10) suggests different factors that a company or project needs to consider before migrating to the cloud. It also suggests the suitable of a company for adopting cloud computing. It is based on the research by Misra & Mondal (2010). The paper takes many organisations’ sensitive factors such as financial, data sensitivity, security and so on into account. As said in their paper, the calculation is subjective and many things need to be considered, but the result will give good criteria and characteristics of what to look out for when considering the cloud (See Appendix 11). From the page, a result less than 3760 suggests a company is unsuitable for cloud adoption and greater than 3760 and less than 4600 means more things need to be considered such as cloud provider selection. More so, if the result is greater than 4600 this signifies that the company is suitable for migrating to the cloud. Figure 10 shows the migration page.

Figure 10: Migration page

The page About shows a description of the application and information about this thesis.

7.2 Application development

The application uses an MVC architecture that comprises of PHP, MySQL and AngularJS framework. I have arranged the project files of MVC framework into 6 different folders for better organisation. The following are the folder structure that will appear after cloning the project from the github application repository. (Application repository, 2016)

- js/ – JavaScript library files. e.g. angular.js
- images/ – Custom JavaScript controller files for our project
- partials/ – Small pagelets that are reusable
- classes/ – The PHP files that communicate to server (Connect, Create, Read, Update, Delete)
- css/ – Stylesheet files
• services/ - PHP files that created the API

PHP is used to get the data through the MySQL database server to generate a JSON array, and then AngularJS gets the data and displays it in the browser. Example JSON response:

```json
"Cloud Providers": {
  "title": "Cloud Provider",
  "name": "Amazon Web Services",
  "image": "aws.png",
  "description": 500
}
```

Below I describe the programming languages used for application and their role in my thesis project. The application workflow diagram in Figure 11 shows the data in the cloud is being requested by using PHP & MySQL. PHP then passes the data to AngularJS, the frontend framework. The sample data used are saved in the cloud; PHP and MySQL retrieve data from AWS and GCP service (cloud). Afterwards, PHP converts the data to JSON format and passes it to the AngularJS page, which outputs the data in the browser. The reverse is the when saving data to the cloud. The following tools were used for completing the application: Git, PHP, JSON, MySQL, AngularJS.

![Figure 11: Application Workflow](image)

**Git** is a distributed revision control and source code management system. It controls changes made to project files so that it can be reverted back to its previous versions if there is a problem with the changed version. Git is commonly used for source code management and can be used on any type of computer. It can also be used both online through github.com and offline by installing it on the computer. Everyone that is given access to the repository can clone/download it to their local computer, which makes it possible to work offline even if the central server is unavailable. It provides fast support for non-linear development, distributed and ability to handle large projects. (Chacon, 2013). These features of Git make it suitable for sharing this project source code for testing.

**JSON** (JavaScript Object Notation) is the lightweight open standard data interchange format. It is a text format that is not dependent on any language. JSON syntax is human readable, which makes it an ideal format for data interchange language. It is preferable to
XML in modern browsers as the browser parses it faster and it is supported in GCP and AWS services. There are two fundamental structures of JSON data:

- As a collection of name/value pairs
- As a list of ordered values

In this project, the data are in JSON format and output them in the client side. That is, the AngularJS framework will display the JSON data as HTML. (JSON, 2016)

**PHP** is a server-side scripting language designed specifically for the Web to retrieve data from the server. It can be used to generate different data formats such as XML, JSON, and CSV etc. PHP applications runs on the server and can run on all operating systems and it is supported in GCP and AWS services. PHP is a scalable, fast, open source software and have good support. (PHP & JSON, 2014)

**MySQL** is a relational database management system that is efficient and durable. It is used to store, search, sort, and retrieve data. MySQL server controls access to data to ensure that multiple users can work with it concurrently, to provide fast access to it, and to ensure that only authorized users can obtain access. It runs on the server and it is suitable for small and large applications. MySQL makes use of Structured Query Language (SQL), which is the standard database query language. Both PHP and MySQL work with many major operating systems. To run PHP and MySQL application, it is necessary to have a web and database server. Apache server is the most popular open source web server that is being used to develop PHP and MySQL application. (Welling & Thomson, 2009) As stated earlier, my thesis application outputted JSON data converted by the PHP script. The data are manipulated through the MySQL database management system. AWS and GCP services provide supports for MySQL.

**AngularJS** is one of the best frontend MVC framework. Google is maintaining AngularJS. It can be used to develop different frontend application. AngularJS is extensible and supports many libraries. In my opinion, features such as Two-way data binding, Modules, Controllers, Template, Services, Routing and so on, contributed to the wider adoption and its suitability for many frontend projects. (AngularJS, 2016)

### 7.3 Application Hosting

This chapter provides technical details on how to host applications on AWS & GCP. This is needed in order to demonstrate the importance and how to follow the guidelines stated in Chapter 6.

#### 7.3.1 Hosting on GCP

This section explains the workflow for getting the application online on Google Cloud Platform. As pointed out in Chapter 3, App Engine, Cloud SQL, and Cloud Storage are the main services required for migrating an application to the Google cloud. Before getting started setting up the project environment, one has to take some initial steps such as: 1) sign up for a Google account; 2) sign up for a Google Cloud Platform account; and 3) enable billing. Then one needs to select the programming language for the application; in my thesis project I choose PHP. Currently, other languages supported by App Engine are Java, Python, and Go (see Chapter 3). The next step is to download the SDK available on the download page of Google App Engine for PHP (Google App Engine for PHP, 2016). For setting up the coding environment, it is possible to use any code editors or IDEs. However, there are many IDEs with enhance connectivity to GAE such as; Google
plugin for Eclipse IDE suitable for Java, PHP and Python. Also, PhpStorm and Wind IDE can be used for PHP and Python with good connection to GAE. (Cohen, Hurley & Newson 2015)

GAE is a paid service and after the testing one may want to take advantage of its services, therefore, it is important to understand the available Quotas. There are two Quotas in GAE, Safety Quota and Billable Resource Quota. Safety Quota is used for setting that no one application will use so many resources that other apps are affected. Safety Quotas could be daily or per-minute. Billable resource Quotas unlike Safety Quotas are set by the application administrators with the intention to help their app stays within their own established budget. (Google App Engine Quota, 2016)

---

**Figure 12:** Google Cloud Platform hamburger menu after sign in (GCP, 2015)

After signing up and signing in to Google Cloud Platform (e.g., by using a Google account), App Engine is selected from the Services page (see Figure 8). In the GCP projects console page, the lists of existing own projects are shown and new projects can be created (Figure 13). The free trial account has a limit of ten projects per account.

---

**Figure 13.** GCP projects console page (GCP, 2015)

For existing projects, click on the project name and the project page will display and guide you through what to do in order to deploy your application or you could use a sample project provided by Google. In the GCP projects console page, you can manage
different projects and their metadata. Once you have an application in the console, clicking on the project name will open the project page that contain an overview or summary of the application usages and requests in a chart form and you can select different criteria depending on what you want to see. (Figure 14)

Figure 14. App Engine project page (GAE, 2015)

It is important to say that you will need to enable the APIs that your application will need. Clicking APIs on the Google Cloud Platform hamburger menu will show list of all available Google APIs and you can change the status to on/off to enable or disable the APIs your application will be using. After creating a project and exploring the project pages, the next thing is to deploy an existing application. However, before migrating the project, it is required to download and install the App Engine SDK. From developers.google.com/appengine/docs/php page, install the SDK by clicking on the link associated with the operating system and follow the instructions below to install the SDK.

To install the SDK on Windows:

1. Double-click the SDK file you downloaded from the Google App Engine for PHP link provided in the reference section of this paper (GAE, 2015) and follow the prompts to install the SDK.

2. You will need Python 2.7 to use the App Engine PHP SDK, because the Development Server is a Python application. Download Python 2.7.X (do not use a higher version) from official Python web site. (Google App Engine for PHP, 2016).

The SDK includes a web server and software to upload the app. If your application requires database connectivity, you will need to set up MySQL locally. One option is to use MAMP, XAMPP, WAMP or LAMP.

Download the file and open the downloaded .msi file and move it to another location e.g. on the desktop, else it will not work properly. Then double click the icon and start the installation process. Click open on the dialog box. You will be asked if the application can create symbolic links for the App Engine to work properly. The App Engine will tell you the symbolic links that were created and the App Engine launcher should be ready and installation completed.
In the app launcher (Figure 15), go to file and click ”Add existing application” and navigate to the folder containing your application and then select one to upload. Select a runtime environment and click upload. After uploading the project to the launcher, you will see the applications you have uploaded with the admin port and port numbers. The port number will be automatically incremented by the launcher. Once there is a project in the launcher, the menu will be enabled and you can select the project by clicking run to start running the application. A green sign should show next to the app on the left. You can test the app is running by going to the browser and go to the URL, localhost: + the port number. For example, if the port number is 9080 the URL is localhost:9080.

Next, you will need to specify the files to load and their path for the application. This needs to be specified in the app.yaml file. Each project hosted on App Engine must have the yaml file. The yaml file uses pattern matches regular expression that contains all path to hold static files used in the application. It contains path for files such as images, stylesheets etc. The first handler in the file specifies the main opening page using a wildcard. It also contains version changes to the application. Also, the yaml file must be tabbed correctly. (Google App Engine configuration, 2016). Example content of the yaml file:

```yaml
application: klausdelet
version: 1
runtime: php55
api_version: 1
threadsafe: true

handlers:
- url: / static_files: index.html
  upload: (.*\.html)
- url: /css
  static_dir: css
- url: /js
```
After you have gotten the app working locally as you want, you then deploy it on GCP. Before deploying your application online, you will need to register the app first and make sure all statements are correct. Registering the application requires login to Google cloud, entering the application information on App Engine page, which shows the list of all current projects. Click the create application button and enter the details. The application identifier or name must be unique and you can check availability of the name in the GCP console. See Appendix 1 for step by step information on uploading existing application to App Engine. After following the step in Appendix 1 to create the application on App Engine, the next thing is to set up the Google Cloud Storage to save the database file and other assets such as images and style sheets. The main step involves uploading the files to buckets and linking to the file from the application. Appendix 2 contains details steps on how to creating bucket and store files in GCP cloud storage. Having the database file and assets file in the cloud opens to opportunity for experiencing cloud benefits.

After uploading the database file (.sql) to the new Google cloud storage bucket by following the steps in Appendix 2, the application needs to connect to the database. This project will be using PHP PDO to connect to the database with Cloud SQL. In Cloud SQL, create an instance that will connect to the bucket. The connection details below will establish connection to the data in the bucket via the Google Cloud SQL. This will be specified in the application;

**Connecting using the instance URL**

```php
$db = new pdo('mysql:unix_socket=/cloudsql/<your-project-id>:<your-instance-name>;dbname=<database-name>',
'root', // username
'', // password
);
```

**Connecting using the instance IP address**

```php
$db = new pdo('mysql:host=127.0.0.1:3306;dbname=<database-name>',
'<username>',
'<password>'
);
```

It is important to give permission to the application to use the file. In cloud SQL, you will see all the instances that you have created and you can create a new one. As said in Chapter 3, Cloud SQL is a relational database management system. After storing the SQL file in cloud storage, the file must be dump in cloud SQL to create the database with tables and needed privileges. Appendix 3 gives the details on to connect the database file saved in cloud storage with cloud SQL.

In conclusion, having power to host your database driven application in the cloud really opens the door to working with more benefits offered by cloud computing paradigm.
Whether you are working with cloud datastore running non-relational data or relational data on cloud storage. The discussion section will explain things to do in order to have scalable, elastic, secure, eliminate redundancy and a system designed to reduce failure.

7.3.2 Hosting on AWS

To host application on AWS, it is necessary to sign up first. Signing up for AWS is very easy, you create a single account and you have access to the entire landscape of AWS suites of tools and services. All that is needed is an Internet connection, email address, and a valid credit card. Though it does not cost anything to create an account, most of AWS services have cost associated with them so a valid credit card needs to be associated with the account. Since signing up involves all sorts of personal information, I will not work through details. Follow the steps in Appendix 4 to create an account on amazon web services. The first page that appears is the AWS dashboard (Figure 16), which contains list of AWS services. When hosting application in Amazon cloud, the very first thing to do is to create an IAM user account so that you do not use the master account to login. The IAM user will have all permission and privileges to all the services the master account have access to but will not have access to billing information so as to not have access to the credit card and personal information. Appendix 5 explains how to set up an IAM user account on AWS. The process involves mainly setting an account and specifying some privileges for the account. The next paragraph of this section will continue with another steps needed for launching an AWS instance.

Amazon EC2 uses public key cryptography to encrypt and decrypt log information. Public key cryptography uses a public key to encrypt a piece of data such as password. Then the recipient uses the private key to decrypt the data. The private key is known as a key pair. To connect to EC2 instance, you must create a key pair. You specify a name for the key pair when you launch an instance and provide the private key when you want to connect to an instance. Linux and UNIX instances have no password, rather you use this key pair to login using SSH. With Windows instances, you use the key pair to obtain the administrative password and then login using RDP. A key pair is assign to an instance when you spin one up and it controls any SSH access to instances. The bottom-line is that you need to create a key pair when you create an instance. To create a key pair, it is associated with EC2 as stated earlier. Therefore, create a key pair by following the guideline on the AWS services dashboard. It is recommended to have a suitable name.
convention that works for you and your organisation. It is also recommended to rotate this so often as part of a good security practices. It is good to have different key pairs for your users. For instance, application environment might use separate key pair like one for development and one for production. After entering a name and clicking create button, a (.pem) file should be downloaded automatically to your local computer. The .pem file is the private key of the public-private key discussed. It will control access to any server launched with this key pair. It is also important to set proper permission for this file in order to login via SSH. Otherwise, if it is too open i.e. if the permission is not strong enough it will be rejected by the SSH client. To ensure proper permission is set, open Terminal and change the permission on the file by issuing the following command

Chmod 0400 ~/Downloads/filename.pem

The above command will make the file readable to only the owner and can now be used for login. Windows users don’t have to worry about the file permission. After creating an IAM account and a key pair, it is now time to launch the instance. The discussion part of this paper will elaborate more on the importance of the IAM account. Follow the steps in Appendix 6 to spin up a new AWS EC2 instance.

![Figure 17: AWS creating security group page (AWS instance, 2016)](image1)

![Figure 18: EC2 instance page (AWS instance Home, 2016)](image2)

After launching an EC2 instance, it is important to connect to it in order to upload the application to the cloud on AWS. There are two ways to connect to the EC2 instance, via a standalone SSH client and this is the recommended way to connect. In Mac and Linux, an SSH client should already been installed and available from the Terminal. For Windows machine, a free PuTTy application could be used. I will be displaying how to connect via PuTTy on Windows in this project. For Mac and Linux machine, select the running instance and click connect at from the top menu, a guideline for connecting to the running instance should be displayed (see Figure 21). Follow the guideline to establish
connection. The second way is to use the browser Java SSH client. However, this is not recommended due to compatibility issue.

Figure 19: Guideline for connecting to an instance on Mac and Linux (AWS Dashboard Home, 2016)

To connect on a windows machine, the .pem file that was downloaded when creating the key pair needs to be changed to .ppk file. Using a third party application called PuTTY can do this. It is an easy step to complete by following Appendix 7: converting .pem to .ppk on a windows machine and connecting to EC2 instance. After successfully connected remotely to the instance on a Windows machine by following Appendix 7 steps, it signifies that everything has been configured correctly, a page similar to Figure 21 should be seen and one can start moving the project to the cloud on AWS. In addition, Opening the public DNS URL should show Apache default start up page.

Figure 20: PuTTY security popup
Now that the web server is set up and connection has been established, it is now time to set up the database. As said earlier, the project will be doing this with the AWS RDS. It is part of the free tier bundle. The first step is to create an RDS instance, which can be done via the console. After successfully creating the RDS instance with the steps in Appendix 8, the status should show running before connecting to it. When it finished launching, one can see the endpoint and other monitoring activities of the database instance as seen in Figure 22 below.

![RDS instance page](AWS_RDS_instance_2016)

**Figure 22:** RDS instance page (AWS RDS instance, 2016)

After creating the RDS instance, it is possible to connect to the RDS instance via SSH, issue the following the commands to connect and start creating databases, tables and so on. After connecting to the EC2 instance according to the guideline followed previously to see Figure 23, issue the following:

```
Mysql -h endpoint -u masterusername -p
Enter password
```

Now it is possible to query instance and issue any MySQL commands such as: `show databases; create databases;` and so on. In the next paragraph, creating an AWS S3 buckets for storing the images and asset files.

Amazon S3 stores files in buckets and it is worth mentioning that this is part of the free tier bundle available for one year. For this thesis, I have uploaded the images; SQL and
asset files to the bucket by following Appendix 9 steps and this will serve the images, style sheets and JavaScript static files used in the project. The application needs to connect to the files in the bucket in order to use them. Therefore the files in the bucket need to be made available to the public.

Figure 23: Modifying an S3 bucket files (AWS S3, 2015)

In the next chapter, I discuss all the findings from the previous chapters in relation to the research questions and the survey.
8. Discussion

This chapter elaborates more on already discussed issues and draw out some points from previous chapters. These are used to answer the research questions. It is evident from analysis in Chapter 5 and different experts’ papers that there are many cloud providers at the moment and it is hard knowing the best one for a project or organisation. In addition, it is not easy to migrate existing applications due to insufficient guideline and information about existing vendors. Previous research and this thesis showed that security, failure, privacy and centralisations are big concerns for organisations that plan to move to the emerging cloud market. Answers to the research questions are derived from what has been carried out in this paper.

RQ1: What are the opinions and attitudes of IT experts with regard to CC, CC adoption, and CC migration?

According to the findings exposed in Chapter 5, IT experts have optimistic attitude towards cloud computing paradigm. They acknowledged the benefits of CC such as the ability to spend less on IT infrastructure, increase in application speed, and flexibility of cloud services tools. Also, they acknowledged the challenges being faced by cloud computing like security, privacy, and learning curve.

Despite the challenges of CC, according to the survey, more than half of the respondents find AWS & GCP documentation not helpful for knowing the suitability of the CSP services for their project.

As at the time of this paper, there has not been much research on how to select a cloud provider based on CSP services and benefits for a company’s project(s) or organisation. Also, according to the survey responses in Chapter 5, IT experts would appreciate a way to know which of the providers to choose because there are now many public cloud services provider as stated by Gorelik (2013). The CPSM in Section 6.2 addresses the question of what criteria are needed when selecting between AWS and GCP. This model combines the findings from the survey, findings from previous researches and my experience of using the cloud. In addition, future work will address the implementation and automation of the CPSM framework in the next version of KlaudCelet (i.e., alpha version), so that minimal effort will be needed in finding a suitable cloud provider.

RQ2: What criteria are relevant for companies when migrating to a cloud service provider?

It is evident that almost all cloud vendors provide vendor lock-in (Olson, 2012). However, one should be able to answer the questions by Olson (2012) in subchapter 6.1 in order to have much efficiency and satisfaction from changing to the cloud infrastructure. Vendors track record, transparency, trial period, type of service agreement, ease of integration and configuration, security must be provided by the cloud service provider. In addition, it is also very important to consider your organisation’s sensitivity, availability of resources, and risk of failure and organisation nature before moving your infrastructure to the public cloud. (Keung & Kwok, 2012)
Misra and Mondal (2010) analysed different parts of an organisation and deduced a framework to examine whether to join the cloud or use in-house infrastructure. It provides different threshold to determine whether to migrate or not. As pointed by Olson (2012) everything that can be done when hosting ones data center can now be done in the cloud. AWS and GCP provide all kinds of public cloud services with AWS having numerous services than GCP. It is possible to migrate some parts of an application to the cloud and run some other parts in ones data center. (Baron, & Kotecha, 2013; Barr, 2010) Although, there has been some arguments about cloud latency which is not a focus of this paper and this could prevent some organisations from moving all their application’s components to the cloud. In Section 6.1, I showed that all parts of the application could be moved to the cloud. From the databases and images to the assets files. Nowadays, many applications are using content delivery networks (CDN) to host their asset files.

RQ3: What are the guidelines to consider when configuring application on AWS & GCP?

According to Varia (2011), Varia & Mathew (2014) and Abrams (2014) elasticity, data protection and motion and at rest, designing with failure mindset are guidelines that must be considered when using AWS and GCP. Designing for failure, this has been explained briefly in Section 6.1. Thinking about failure upfront causes you to think about recovery strategies at design time, which will lead itself to a better, more stable end product. One major rule of thumb when architecting for failure is to avoid a single point of failure. An example of this having two web servers are connecting to a single database server, the database server is our single point of failure, when the database goes down, so is the entire system. However, with the database configured in previous chapter, where a stand by instance is set up as a secondary database server, I have designed the application with failure upfront. With RDS, doing this is easy; RDS takes care of all the synchronization details. If the primary database fails, the secondary database is enabled because it has been standing by for such an event. The application stays up and customers remain happy. Avoiding single point of failure of course is not just relevant for database; it needs to be considered for every component in the system architecture. Even when single point of failure is avoided, one has to consider how failover happens in the event of a component failure. The failover process itself may entail other hardware, software or network resources and needs to be given attention during the system design. The AWS and GCP provide services to easily eliminate single point of failure and failover.

As explained in Section 6.1 another guideline for AWS and GCP cloud migration is to implement elasticity. Regardless of the methods used to achieve elasticity, the deployment process has to be automated to ensure that initial system configuration allow the application to be streamlined and accommodate scaling. Most cloud services have an API available to make automation possible at every step and much of the needed automation is also available in the console user interface where no scripting is required. The provided tools make the automation of the deployment process straightforward and doing so helps to reduce errors and ensure efficient scaling. One of the keys to automating the deployment process is bootstrapping the instances. Bootstrapping means to create a self-sustaining start up process that can run on its own and in AWS; it basically means the process needed to get the application up and running on an EC2 instance As stated in Section 4.3.2, scalability in GCP happens automatically without configuration.

Finally, as a cloud architect or user, it is important to get familiar with cloud general best practices as well as fully aware of exactly how security responsibilities are shared between the cloud providers and the cloud users. (Dunn, 2010 & Minoli, 2010)
8.1 Limitations

The sample data used in the application are not real; this is a huge challenge for this project. Real data are not available because the cloud is a relatively new technology and it is hard to get a complete list of cloud providers and their services. In addition, due to the lack of data, the application could not be used to make a complete recommender system to predict or suggest a particular cloud vendor for application or organisation. It would be very useful if there were API that could return all the cloud service providers. This is one area that can be improved in the future. It would be nice to use CPSM in KlaudCelet and test it empirically with a real company in order to validate the recommender system developed.

Furthermore, this research could be improved by setting up the application to use custom domain rather than using the Google’s appspot subdomain and Amazon’s public DNS URL. This would require buying a domain name and could not be included in this project because money is not available to buy a domain name. A custom domain name would enhance the application for proper indexing and business functioning.

Another limitation encountered during this research was lack of IT expert participation for the survey. This would have made it possible to use a quantitative data analysis to analyse the data and would add more evidence of the challenges and usefulness of this project.
9. Conclusion

This thesis shows the importance, guidelines for migration, and the usefulness of cloud to the computing discipline, as this is the goal of this thesis. This paper provided answers to the research questions, What are the opinions and attitudes of IT experts with regard to CC, CC adoption, and CC migration? What criteria are relevant for companies when migrating to a cloud service provider? What are the guidelines to consider when configuring application on AWS & GCP? The attitude of IT experts is optimistic even though there are many challenges to cloud adoption. Criteria such as transparency, year of service, geographical location, organisation culture, ease of configuration and so on, must be looked for when planning to migrate to the cloud. Also, failure, security, loose coupling, scalability are configuration criteria that are needed when setting up application on AWS and GCP.

This thesis accompanied the task of creating a recommender framework. The recommender framework was developed using design science approach. The need for solving this problem was shown by conducting a review of previous research and a survey among IT experts. The decision to migrate to the cloud must take into consideration the organisation's state and culture, cloud deployment model, and choosing a suitable cloud service provider. In this thesis, I build an application named KlaudCelet to demonstrate the usefulness of a framework that recommends, based on established criteria, the cloud migration decision. KlaudCelet is a recommender system that is developed based on previous research on cloud deployment frameworks and proposed model (CPSM; Cloud Provider Selection Model). The recommendation by the system is reliable because it is based on research results. KlaudCelet recommends a deployment model, a cloud service provider, and a decision either to migrate or not. CPSM was not used in KlaudCelet but the model by Keung & Kwok and Misra & Mondal were used.

At the moment, the cloud is still young and it is not perfect. If security is a very crucial part of an application or the application is a security dependent, it is better to go for a private cloud. However, with the fast development and improvement in the area of cloud computing, joining or migrating to the cloud is a very important aspect of nowadays IT infrastructure. Either an organisation choose to go with AWS or GCP, it is still running a cloud application.

In conclusion, undoubtedly cloud computing is the future of computing with great benefits and certain challenges such as security, privacy, learning curves and so on, which can be overcome. Therefore, it is a good idea to be optimistic and migrate to the cloud in order to harness its benefits.
References


Alseadoon, I. M., The Impacts of users’ characteristics on their ability to detect phishing emails. Queensland University of Technology Brisbane, Australia 2014.


PUTTY home page URL, Retrieved January 04, 2016, from


Appendices

Appendix 1: App Engine set up
- Fill out other details and hit create.
- The application name must be the same with the name used for the application in the app launcher.
- Go to the app launcher and select the application you wish to deploy.
- Click deploy button on the top right.
- You will be required to sign in to confirm your identity.
- The log console in the app launcher will automatically open, showing the progress of the deployment.
- If there is no error, the app is live now and can be viewed from anywhere around the world.
- Go to the browser and enter the registered app name for example, appname.appspot.com.

Appendix 2: Creating and storing files in cloud storage
- Click cloud storage from the list of services on the left in the cloud console page, login is required (console.cloud.google.com).
- Click new bucket to add new one and give it a unique name.
- You can upload files or folders to the bucket after creating it. In this project, all the images used in the application are stored in a bucket.
- The SQL file that contains data used in the project are stored in a bucket and link to from the cloud SQL service. See the project file on github (Application repository, 2016)
- Select the link that is beside the file and check it if you want others or other application to view the file/folder once they have the link.
- This could be done on an individual file or on the folder containing the file as a whole. It is also possible to edit the file metadata by clicking on the leftmost link to the file.

Appendix 3: Connecting cloud storage with cloud SQL
- Create an instance by clicking create instance button
- Choose import to import the SQL file saved in the cloud storage.
- Give a name and Specify the path to the file and click deploy or okay.
- A status of the import will show.
- You can go to the operations tab to see if the upload was successful or not.

Appendix 4: Creating an AWS account
- Go to the aws.amazon.com website and click sign up. Then enter email and click the new user button.
- Fill the form to complete the process. You will also need a valid phone number in order to finish the sign up process and amazon will call and ask you to type the 4 digits number or pin.
- The automated call will tell you if the account has been verified successfully.
- When you finish creating an account, go to console.aws.com or from aws.amazon.com and click sign in to the console, enter email address and password used for sign up and select returning user and click sign in.

Appendix 5: Creating IAM user on AWS

- To create an IAM account user, in the dashboard services under security and identity, click Identity and Access Management.
- Click on users on the left menu and click the big ‘Create New Users’ button and a dialog or popup will appear and you can enter multiple user names at once.
- You might need to check the checkbox to generate an access key for the user.
- Access keys are needed to make secure REST or query protocol requests to AWS service APIs.
- You can leave this step since this project is not making any REST request with the AWS application.
- After creating the user, you will need to give the user some privileges.
- Click on the user and the permission tab. Then click attach user policy and use the selected policy template and give administrative access to the user.
- Clicking on the user in the IAM page, you will see the URL, when the user was created, and whether the user needs a password for accessing AWS services or not.
- Click on the security credential tab to set a password for the user. Now, the new user can login with the name and password by going to the IAM user sign-in URL. It is found in IAM dashboard.

Appendix 6: Launching an EC2 instance

- Navigate to the AWS console or services.
- Click on EC2 from the list of services.
- Click on launch instance to launch a virtual instance, it should open a window with 4 options.
- Select the Quick start wizard and select the first option on the right. This will give finite option on how the instance should be configured.
- Click next and it opens the instance type page with free tier preselected.
- Click next at the bottom to configure instance details. Because this is a free account, I will go with the default options and click ‘Add storage’ button
- In the storage page, go with the default storage volume to avoid charges, then click ‘Next’ button
- Choose a tag instance or leave it as it is. The tag instance consist of a name-value pair, it is used for tagging EC2 instances.
- Next configure security group (Fig. 17). This is just like a name for a set of firewall rules. Create a security group based on the role of the server i.e., a webservice. In this project no SSL certificate is used, therefore I won’t create a rule for HTTPS. Give a name, and description, SSH rule has been added by default. Change type to HTTP, protocol and port will be shown as TCP and 80 as default. The CIDR could be any IP address, or other security group. In this
Finish the launching, it is important to select a key pair for connecting to the instance and check the acknowledgement checkbox. Finally, click ‘Launch instance’ and the dashboard should display the status of the instance.

- To see the instance just launched, go back to AWS console dashboard and clicking EC2 instance menu will show the newly created instance with the status of running. (Fig. 18)

Appendix 7: Converting .pem file to .ppk and Connecting to EC2 instance

- Go to the PuTTy home page URL available in the reference section and select the putty installer to download PuTTy and its keygen.
- Start the installed PuTTy keygen on your local computer.
- Click load and change the filename dropdown to show all files. Then select the .pem file from its location and click open.
- A notice indicating success should be shown, click OK. This will show the PuTTy keygen application.
- Click save private key and click YES to the passphrase popup question.
- Give a filename to the private key and click save. This will create a .ppk file that will be used for connecting to the EC2 instance.
- To connect to the instance, open PuTTy, enter the hostname, the hostname is the public DNS URL that is found when you click on the running instance in the EC2 instance page.
- Make sure it is port 22.
- Click on Data on the left section of the PuTTy application and give the login username, this is usually ‘ec2-user’ for all EC2 instances.
- Then on the left section, Click SSH and select Auth for authentication.
- A page where you can browse to load the .ppk private key file should appear and finally, click open. This will open up a security alert popup (see Fig. 20) that ask if the hostname should be added to known hostname.
- Click yes. You should see a page similar to the Figure 21.

Appendix 8: Launching an RDS instance

- Click the RDS from the console under the database section.
- Click “Launch DB instance” Then choose a database engine.
- At the moment, RDS supports MySQL, POSTgreSQL, ORACLE, SQL SERVER and MySQL/AURORA launched recently. This project will be using MySQL in this project.
- The next step is to choose either you want to use it in production or development environment.
- We choose development for this project.
- Next step is the database detail, choose GPL license and go with the defaults because this is a free tier account with minimum of 5GB storage.
- For more information about each selection, hovering over the name will show a tooltip with more description.
- Click Launch DB instance and an indication that it is launching should appear.
Appendix 9: Creating and S3 bucket and storing assets file in it

- Click S3 from the storage section of the AWS dashboard and click ‘Create Bucket’ button.
- A dialog box appears and enter a name for the bucket and click create.
- Note that the name should be unique else it will show an error.
- After creating a bucket, click on the bucket name to see its contents.
- Click on ‘upload’ to upload the files or folders to the bucket.
- After uploading, select the file or folder by clicking on the box on the left side of the file/folder.
- Click on ‘Actions’ tab at the top and a page similar to Figure 23 below will appear.
- Click on ‘Make Public’ from the dropdown menu and the file is available to the world.
- Also, clicking on ‘Properties’ from the dropdown menu will show a tab page that displays the permission, details, and metadata for the selected file.
- From the details, you can see the URL to use in order to connect to the file from the application.

Appendix 10: Cloud deployment selection model (CDSM) – Keung and Kwok (2012)

<table>
<thead>
<tr>
<th>Major Aspect</th>
<th>Factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>General aspects for cloud adoption</td>
<td>- Data traffic</td>
</tr>
<tr>
<td></td>
<td>- Data traffic influence to business</td>
</tr>
<tr>
<td></td>
<td>- Current IT capacity</td>
</tr>
<tr>
<td></td>
<td>- Predictability of IT demand</td>
</tr>
<tr>
<td>AS-IS IT condition</td>
<td>- IT maturity</td>
</tr>
<tr>
<td></td>
<td>- IT complexity</td>
</tr>
<tr>
<td></td>
<td>- IT compatibility</td>
</tr>
<tr>
<td></td>
<td>- Dependency on legacy system</td>
</tr>
<tr>
<td>TO-BE IT condition</td>
<td>- Expected level of IT changes</td>
</tr>
<tr>
<td></td>
<td>- Expected workaround assigned to in-house IT</td>
</tr>
<tr>
<td></td>
<td>- Desired IT control</td>
</tr>
<tr>
<td></td>
<td>- Intended coverage of Cloud</td>
</tr>
<tr>
<td>Sensitivity of data</td>
<td>- Data significance</td>
</tr>
<tr>
<td></td>
<td>- Data confidentiality</td>
</tr>
<tr>
<td>Availability of resources</td>
<td>- In-house IT support</td>
</tr>
<tr>
<td></td>
<td>- Manpower</td>
</tr>
<tr>
<td></td>
<td>- IT skills</td>
</tr>
<tr>
<td></td>
<td>- Time</td>
</tr>
<tr>
<td></td>
<td>- Space</td>
</tr>
<tr>
<td>Demand of IT resources</td>
<td>- Customisation of IT solutions</td>
</tr>
<tr>
<td></td>
<td>- Users’ mobility</td>
</tr>
<tr>
<td></td>
<td>- Future IT usage expansion</td>
</tr>
<tr>
<td></td>
<td>- Strategic alignment</td>
</tr>
<tr>
<td>Nature of organisation</td>
<td>- Scale business</td>
</tr>
<tr>
<td></td>
<td>- Development stage</td>
</tr>
<tr>
<td></td>
<td>- IT conservativeness</td>
</tr>
<tr>
<td></td>
<td>- IT decisiveness</td>
</tr>
<tr>
<td>Risk Possibility</td>
<td>- Risk of natural disaster</td>
</tr>
<tr>
<td></td>
<td>- IT failure rate</td>
</tr>
</tbody>
</table>
## Appendix 11: Cloud deployment selection model (CDSM) - Misra & Mondal (2010)

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of IT resources</td>
<td>- Less than 100 servers</td>
</tr>
<tr>
<td>Number of servers</td>
<td>- From 101 to 2000 servers</td>
</tr>
<tr>
<td></td>
<td>- From 1001 to 10,000 servers</td>
</tr>
<tr>
<td></td>
<td>- From 10,001 to 50,000 servers</td>
</tr>
<tr>
<td>Annual revenue from IT offerings</td>
<td>- One country</td>
</tr>
<tr>
<td></td>
<td>- More than one but less than four countries</td>
</tr>
<tr>
<td></td>
<td>- More than equal to four but less than equal to six countries</td>
</tr>
<tr>
<td></td>
<td>- More than six but less than ten countries</td>
</tr>
<tr>
<td></td>
<td>- Above 10 countries</td>
</tr>
<tr>
<td>Number of countries it is spread across</td>
<td>- Less than $100 million</td>
</tr>
<tr>
<td></td>
<td>- Between $(100–500)$ million</td>
</tr>
<tr>
<td></td>
<td>- Above $500 million</td>
</tr>
<tr>
<td>Workload variability</td>
<td>- Peak usage</td>
</tr>
<tr>
<td></td>
<td>- Duration of peak usage/year</td>
</tr>
<tr>
<td></td>
<td>- Peak by average</td>
</tr>
<tr>
<td>Average usage</td>
<td>- Type of services</td>
</tr>
<tr>
<td></td>
<td>- Type of projects undertaken</td>
</tr>
<tr>
<td></td>
<td>- Size of user/customer base</td>
</tr>
<tr>
<td>Amount of data handling</td>
<td>- Above 100 terabytes/month</td>
</tr>
<tr>
<td></td>
<td>- 1–100 terabytes/month</td>
</tr>
<tr>
<td></td>
<td>- 500 gigabytes–1 terabytes/month</td>
</tr>
<tr>
<td></td>
<td>- 100–500 gigabytes/month</td>
</tr>
<tr>
<td></td>
<td>- Below 100 gigabytes</td>
</tr>
<tr>
<td>Sensitivity of data</td>
<td>- Extremely sensitive</td>
</tr>
<tr>
<td></td>
<td>- Very sensitive</td>
</tr>
<tr>
<td></td>
<td>- Sensitive</td>
</tr>
<tr>
<td></td>
<td>- Less sensitive</td>
</tr>
<tr>
<td></td>
<td>- Not sensitive</td>
</tr>
<tr>
<td>Criticality of work done</td>
<td>- Highly critical</td>
</tr>
<tr>
<td></td>
<td>- Critical</td>
</tr>
<tr>
<td></td>
<td>- Less critical</td>
</tr>
<tr>
<td></td>
<td>- Standard</td>
</tr>
</tbody>
</table>