# A Social Network Approach to Provisioning and Management of Cloud Computing Services for Enterprises

Kuada, Eric

*Publication date:*
2014

*Document Version*
Accepted author manuscript, peer reviewed version

Link to publication from Aalborg University

*Citation for published version (APA):*
Kuada, E. (2014). *A Social Network Approach to Provisioning and Management of Cloud Computing Services for Enterprises*. Aalborg University.

# A Social Network Approach to Provisioning and Management of Cloud Computing Services for Enterprises

A Thesis Submitted in Partial Fulfilment of the Requirement for the Degree of Doctor of Philosophy

**Eric Kuada**

April 2014

Center for Communication, Media and Information Technologies (CMI)
Aalborg University Copenhagen, Denmark

AALBORG UNIVERSITY
DENMARK

# Mandatory page in PhD theses

**Thesis Title**: A Social Network Approach to Provisioning and Management of Cloud Computing Services for Enterprises

**PhD Student**: Eric Kuada

**Supervisor**: Henning Olesen (Associate Professor)

**List of published papers:**

Kuada, Eric, and Henning Olesen. 2011. "A Social Network Approach to Provisioning and Management of Cloud Computing Services for Enterprises." In proceedings of CLOUD COMPUTING 2011, The Second International Conference on Cloud Computing, GRIDs, and Virtualization , Rome, Italy, pp. 98-104.

Kuada, Eric, and Henning Olesen. 2012. "Incentive Mechanisms for Opportunistic Cloud Computing Services." In *2012 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, Pittsburg, USA, Oct. 2012, pp. 127-136. IEEE.

Kuada, Eric, Henning Olesen, and Anders Henten. 2012. "Public Policy and Regulatory Implications for the Implementation of Opportunistic Cloud Computing Services for Enterprises." In *9th International Workshop on Security in Information Systems*, pp. 3-13. Wroclaw, Poland.

Kuada, Eric, Kwami Adanu, and Henning Olesen. 2013. "Cloud Computing and Information Technology Resource Cost Management for SMEs." In *Proceedings of IEEE Region 8 Conference EuroCon 2013*, pp. 258-265. University of Zagreb, Croatia: IEEE. http://www.eurocon2013.org/index.html.

Kuada, Eric. 2013. "Trust Management System for Opportunistic Cloud Services." In *2013 IEEE 2nd International Conference on Cloud Networking (CloudNet)*, pp. 33–41. San Francisco, USA: IEEE.

Kuada, Eric. 2014. "Towards Trust Engineering for Opportunistic Cloud Services: A Systematic Review of Trust Engineering in Cloud Computing", submitted to Journal of Grid Computing.

This thesis has been submitted for assessment in partial fulfilment of the PhD degree. The thesis is based on the submitted or published scientific papers, which are listed above. Parts of the papers are used directly or indirectly in the extended summary of the thesis. As part of the assessment, co-author statements have been made available to the assessment committee and are also available at the Faculty. The thesis is not in its present form acceptable for open publication but only in limited and closed circulation as copyright may not be ensured.

## Acknowledgment

If there be any glory in this, it all belongs to the Almighty God (my Father in heaven) who guides my steps through life. He has ordered my steps through this journey of undertaking this PhD study. Many thanks go to my parents and the rest of my Danish family for their support in making my stay in Denmark much more enjoyable. The support of my family in Ghana also deserves to be acknowledged for bearing with my long absence from them, and their continuous prayers for me while I was away.

Thanks to CMI and all its employees for providing a conducive work environment for me to carry out my research work. CMI has also played a major part in reducing my financial challenges in my employment as a lecturer at the university. Agitect ApS has also played a major role in ameliorating my financial burdens during this three year period with a job position as a systems architect for a year; the experience on the projects I worked on at Agitect ApS has also contributed immensely to the quality of my research work.

## Abstract

The motivation for undertaking this study is that there are spare IT resources available at many institutions, enterprises and other organisations while their counterparts at other parts of the world, particularly in the developing world, lack basic IT resources. It has been difficult or even impossible in most cases to make these spare resources available to those who need them most. This should however be less difficult to accomplish with the advent of cloud computing.

Individuals and companies have begun using free cloud services from commercial cloud service providers to meet some of their IT resource needs. Since these free cloud services do not sufficiently meet the IT resource needs of businesses, and commercial cloud service providers have the tendency of changing their policies of providing free resources, there is a need for finding a sustainable way in which enterprises can contribute their spare IT resources to a platform so that other members on the platform can utilize them as and when needed. This leads to the need for my new concept of Opportunistic Cloud Services (OCS) for enterprises. Opportunistic Cloud Services is about enterprises leveraging cloud services to meet their business needs without paying or paying a minimal fee for these services. The OCS network is modelled as a social network of enterprises collaborating strategically in contributing and utilizing cloud services without entering into any business agreements.

The feasibility of OCS has been the focus of this PhD study. I have been interested in finding the sufficient enabling conditions for the implementation and operation of such a platform. It was almost immediately evident from the start that, among other factors, there is a need for a platform that enables the contribution and utilization of such spare IT resources. The other identified factors are that there should be no legal or regulatory prohibitions for enterprises and other organisations to either contribute their resources or utilize the resources that have been donated by others. Thirdly, there need to be economic and other forms of benefits for enterprises in participating on the OCS platform. The objective of the study was therefore to explore these factors with the main goal of laying the foundation for the development of OCS management platforms. This translated into setting up project objectives aimed at exploring to what extent the arguments supporting or weakening these factors could be found.

The first among the main objectives of the project was therefore to develop a conceptual framework and reference architecture for the implementation of OCS networks for enterprises. The second objective is to develop incentives schemes that ensure sustainability of Opportunistic Cloud Services networks. The third among the project objectives is to develop a trust management system that will ensure that resources and services on the OCS platform will be useful and remain reliable for the users. Another objective was to contextualize cloud computing as a business model for developing economies, and highlight the role of OCS in such a model. The six papers that are included in this dissertation have been targeted towards these objectives.

The adopted research methodology for this study is system development. The idea of system development as a research methodology fits into the category of applied science; it belongs to the categories of engineering, developmental, and formulative types of research. Based on this research methodology, the main research methods that have been employed in undertaking this research work include system design, systematic review, mathematical proofs, and simulations.

The targeted project objectives have been mostly achieved. Firstly, detailed reference architecture for OCS platforms has been developed. Secondly, incentive mechanisms have been designed for the OCS platform. Thirdly, a model for the concept of trust for cloud computing has been developed; this model was then applied to the OCS context to develop a trust management system for the OCS platform. A major sub-system of the trust management system is the pseudo SLA system for managing expectation of service users and quality of service guarantees that service providers promise to offer. The above-mentioned results, together with other major findings on cloud APIs and cloud abstraction APIs which should simplify the implementation of the required functionalities of the OCS management platform, demonstrate the technical feasibility of the implementation of the OCS platform.

The included papers, which explored the implications of current public policies and regulations for the implementation of OCS networks, revealed that there are no regulations or policies prohibiting the implementation of opportunistic cloud services. The lack of a single globally accepted data protection law will however pose some challenges for the OCS platform. Furthermore, even though I have successfully designed incentive mechanisms for the OCS platform, I can’t answer fully if there are enough monetary or other forms of benefits for enterprises to join the OCS platform to contribute or utilize resources. Despite these limitations, the presented results demonstrate both the technical feasibility and the existence of enabling conditions for the implementation of opportunistic cloud services for enterprises.

## Resumé

Motivationen for at gennemføre dette projekt er, at mange institutioner, virksomheder og organisationer råder over ubenyttede it-ressourcer, mens der i andre dele af verden, især i udviklingslandene, er mangel på de samme it-ressourcer. Der har indtil nu ikke eksisteret nogen nem måde at gøre disse ubenyttede ressourcer tilgængelige for dem, der kunne have brug for dem, men udviklingen inden for cloud computing åbner nye muligheder for at adressere dette problem.

Personer og virksomheder benytter sig i stigende grad af gratis cloud-tjenester fra kommercielle cloud-udbydere til at opfylde deres behov relateret til it-ressourcer. Men disse gratis cloud-tjenester opfylder ikke i tilstrækkelig grad virksomhedernes behov, og kommercielle cloud-udbydere har en tendens til at ændre deres politik med hensyn til at levere gratis ressourcer. Der er således behov for at udvikle en holdbar måde til overskudsdeling af ressourcerne, så virksomheder kan overføre deres frie it-ressourcer til en platform, hvor andre medlemmer på platformen kan tilgå og forbruge dem, når og hvis det er nødvendigt. Dette leder frem til det nye koncept kaldet Opportunistic Cloud Services (OCS), som er emnet for denne afhandling. Som udgangspunkt er ideen bag OCS, at virksomheder kan gøre brug af cloud-tjenester til at opfylde deres forretningsmæssige behov uden at betale eller mod at betale et minimalt gebyr. Et OCS-netværk kan betragtes som et socialt netværk af virksomheder, der samarbejder strategisk om at bidrage til og udnytte cloud-tjenester, uden at indgå nogen forretningsmæssige aftaler.

Muligheden for at realisere dette OCS-koncept har været hovedemnet for mit phd-studium, herunder at identificere de nødvendige og tilstrækkelige betingelser for implementering og drift af en sådan platform. Det var fra starten åbenbart, at der kræves en dedikeret platform, der kan håndtere tilførsel og forbrug af de frie it-ressourcer. Derudover har undersøgelsen identificeret en række andre faktorer. Der bør ikke være nogen juridiske eller reguleringsmæssige forhindringer for virksomheder og andre organisationer til at bidrage med deres ressourcer eller udnytte ressourcer, der er blevet doneret af andre. Desuden skal der være økonomiske og andre former for fordele, inden virksomhederne vil vælge at deltage på OCS-platformen. Arbejdet har derfor haft til formål at undersøge disse faktorer og lægge fundamentet for udviklingen af OCS management platforms.. I praksis har projektet således undersøgt, i hvilken udstrækning der kan findes argumenter eller modargumenter for disse faktorer.

Det første mål for projektet var at udvikle en begrebsramme og en referencearkitektur for implementering af OCS-netværk for virksomhederne. For det andet har arbejdet handlet om at udvikle incitamentsmekanismer, der kan understøtte og sikre eksistensen af OCS-netværk. En tredje målsætning har været at udvikle et system til trust management, der kan sikre, at ressourcer og tjenester på OCS platformen til enhver tid vil være nyttige og pålidelige for brugerne. Endelig har det været målet at kontekstualisere cloud computing som en forretningsmodel for udviklingsøkonomier og fremhæve betydningen af OCS i sådan en model. De seks videnskabelige artikler, der er inkluderet i denne afhandling, har været rettet mod disse mål.

Den valgte forskningsmetodik for projektet er systemudvikling, som indgår i kategorien anvendt videnskab, og den hører yderligere til kategorierne engineering, udviklings- og formulativ forskning. Baseret på denne metode er der arbejdet med systemdesign, systematisk review, matematiske beviser og simuleringer.

Projektet har i det væsentlige opnået de stillede mål. For det første er der udviklet en detaljeret referencearkitektur for OCS-platforme. For det andet er der designet incitamentsmekanismer for OCS platformen. For det tredje er der udviklet en model for begrebet trust i relation til cloud computing, og denne model er efterfølgende anvendt i en OCS-kontekst til at udvikle et trust management system for OCS-platformen. En væsentlig del af trust management-systemet er et ”pseudo SLA”-system til at håndtere brugernes forventninger og de ”Quality of service”-garantier, som tjenesteudbyderne tilbyder. De ovennævnte resultater, suppleret med andre vigtige resultater for cloud- og ”cloud abstraction” APIs, demonstrerer den tekniske gennemførlighed af OCS-platformen.

Den indeholdte videnskabelige artikel om de reguleringsmæssige aspekter af OCS efterviste, at der ikke er nogen regulativer eller politikker, der forbyder gennemførelsen af opportunistiske cloud-tjenester. Den manglende harmonisering af lovgivningen vedrørende databeskyttelse på globalt plan vil dog medføre nogle udfordringer for OCS- platformen. Selv om mit projekt har resulteret i design af incitamentsmekanismer for OCS-platformen, kan jeg ikke klart dokumentere, om de kommercielle eller øvrige fordele for virksomhederne ved at deltage i OCS-platformen og bidrage eller udnytte ressourcerne, er tilstrækkeligt store. På trods af disse begrænsninger demonstrerer de opnåede resultater dog både den tekniske gennemførlighed af konceptet og tilstedeværelsen af de nødvendige forudsætninger for implementering af opportunistiske cloud-tjenester for virksomhederne.

# Table of Contents

## Table of Figures

# 1 Introduction

The first part of this introduction section provides an overview of the entire dissertation. Next, the motivation for undertaking the study is presented. The main project goals and the expected outcomes are then presented. Based on the above, the research questions and their associated project objectives aimed at providing some answers to them follow respectively. Because it was possible for the study to take different possible paths, it was necessary to delimit the study appropriately to provide the needed focus. The introduction therefore ends with the presentation of the delimitation of the study.

## 1.1 Overview of the Dissertation

The dissertation begins with this introduction, which tries to justify the need for the study by providing the motivation for undertaking it. Next, the state-of-the-art of the technologies and concepts around which the study revolves follows. The state-of-the-art is mainly on cloud computing. It presents the characteristics of cloud computing, cloud service and deployment models and the major technologies facilitating cloud computing. The state-of-the-art also presents the current challenges in cloud computing, and finally ends with an overview of the Opportunistic Cloud Services (OCS) concept by discussing some of the concepts and principles that have inspired it.

The third part of the dissertation is the methodology for carrying out the study. The methodology has three main parts: these are the research approach, the specific methods that have been applied, and the main theories that the study makes use of. The fourth part of the dissertation is the obtained results and the discussions on them. This dissertation is in the form of a collection of published and publishable articles. The results are mainly the collection of papers that have been included. The discussion focuses on the main findings in the included papers and other major findings during the research period that haven't been included in the collection of papers. The section also discusses the implications of these findings to the implementation of the proposed concept of opportunistic cloud services for enterprises. The concluding remarks follow after the discussions, and finally the collection of papers is presented in the Appendices section.

## 1.2 Motivation for the Study

This section gives the motivation for undertaking this study. It argues that while spare Information Technology (IT) resources are available at enterprises and other organisations, Small and Medium

Enterprises (SMEs) and even larger companies, particularly those in the developing world, lack basic IT resources. It has been difficult or even impossible to make these spare IT resources available to those who need them; but with the advent of cloud computing, this should be less difficult to accomplish. Furthermore, individuals and companies are currently making use of free cloud services from major commercial cloud service providers to meet some of their IT resource needs; but what happens when these companies change their policies (as they often do) of providing these free resources? There is therefore a need for finding a sustainable way in which enterprises can contribute their spare IT resources to a platform so that other members on the platform can utilize them as and when needed. They normally will do this with the hope that they will also find services provided by others on the platform useful for their needs. The remainder of this subsection gives some further details on the above-mentioned motivational factors for undertaking this study.

### 1.2.1 Spare IT Resources at Enterprises and other Organisations

The study of De Blanche and Mankefors-Christiernin (2010) was based on a survey on the availability of resources in an ordinary office environment, with the aim of determining if there were truly usable under-utilized networked desktop computers available for non-desktop tasks during the off-hours. They found that in more than 96% of the cases, the computers in their investigation were available for the formation of part-time (night and weekend) computer clusters (De Blanche and Mankefors-Christiernin, 2010). Research indicates that organizations are increasingly discovering that many companies tend to underutilize the substantial investments made in IT resources. One such survey of six corporate data centres found that most of the servers were using just between ten and thirty percent of their processing power, while the average capacity utilization of desktop computers is less than five percent (Marston et al., 2011).

### 1.2.2 SMEs and other Enterprises in the developing world need IT resources

A characteristic that is common to most SMEs is that they are often proprietor managed. This entrepreneurship owner managed style of management has a direct impact on the company's adoption of Information and Communications Technologies (ICT). This is because as the owner is the centre of business, making almost all of the decisions, the owner's attitude towards technology has a direct impact on the company's adoption of ICT. Another important characteristic feature of SMEs is the financing model. This is usually self-financing by the owner or with the support of finances from close family relations. Access to finances by SMEs especially those in sub-Saharan

Africa is extremely limited and more so now that the world is still trying to recover from the financial crises of 2008/2009. As Collier (2009), puts it:

> *"Providing finance for Africa is generally rated as riskier than for other regions. Providing finance for small firms is globally rated as riskier than for large firms. The provision of finance for small African firms unfortunately brings together these two high-risk characteristics. During the recent global economic boom investors chasing returns in the face of high asset prices became willing to accept higher risks. Small African firms were just beginning to benefit from this reassessment when the boom collapsed, inducing a massive reaction in the opposite direction: the appetite for risk evaporated and with it the scope for private finance for Africa's small firms"* (Collier, 2009).

These factors make it a challenging hurdle for SMEs in the developing world to get access to the needed IT resources for their efficient production. According to Oshikoya and Hussain (1998), African countries can draw on the stock of "out-of-trend" computers, which becomes available when companies, organizations and individuals world-wide regularly update their computer sets. As a result, there is a large stock of unused systems, which mostly creates problems of storage and discarding. They suggested that local non-governmental organisations (NGOs) could be encouraged to design and implement schemes of collecting such systems and organizing their distribution at minimal costs (Oshikoya and Hussain, 1998). The scheme might contribute to reducing the cost of importing computers and increasing the numbers of computer users. The processes and cost involved in getting these "out-of-trend" computers to those who may need them most is, however, usually a daunting challenge.

### 1.2.3 Advent of Cloud Computing

The past five years have seen the offering of computing resources as a service rather than as a product. Cloud computing is essentially the packaging of traditional information technology infrastructure and software solutions such as storage, processing capacity, network resources, applications, services, etc., as virtualized resources that are then delivered by a service provider to its customers as an on-demand pay-per-use self-provisioned service through a web portal over a network such as the Internet. There have been major technological advancements as well as social and business demands driving this new trend of computing. The technological factors facilitating cloud computing include the availability and drastic increase in reliable broadband Internet access, advancements in virtualization technologies, and the shift of development of majority of both desktop and enterprise applications as web services and web applications (Kuada and Olesen, 2012).

The adoption of cloud computing is also being driven by the flexibility that it offers enterprises in scaling up their resources when user demands increase and dynamically scaling down when the demands drop. It additionally gives the opportunity for businesses to obviate high upfront cost in acquiring IT infrastructure and services. Furthermore, businesses over the past decade have gotten accustomed to outsourcing part of their IT services to external entities which makes them to be open to the idea of adopting cloud computing, since cloud computing is yet another form of outsourced IT services.

### 1.2.4 The Cellular Network Impact on the Developing World

While the telecommunications industry in the United States, Canada and Europe invested in landlines before moving to cellular networks, the cellular network has effectively leapfrogged the landline in Africa. Mobile phone coverage in Africa has grown at staggering rates over the past decade. Only 10 percent of the African population had mobile phone coverage in 1999, primarily in Northern African countries and South Africa. By 2008, about 60 percent of the population had mobile phone coverage (Aker and Mbiti, 2010).

Existing micro- and macro-level evidence suggests that mobile phones can improve consumer and producer welfare in developing countries (Jensen, 2007), (Aker and Mbiti, 2010), (Aker, 2008). Furthermore, the adoption of mobile phones has also engendered innovative services to meet the needs of the developing world. For example, in March 2007, Kenya's largest mobile network operator, Safaricom, which is part of the Vodafone Group, launched a mobile money payment service (M-PESA), this was an innovative payment service for the unbanked; and within the first month, Safaricom had registered more than 20,000 M-PESA customers, well ahead of their targeted business plan  (Hughes and Lonie, 2007). Six million customers have since registered with the service, which represents nearly half the customer base of Safaricom (Mas and Morawczynski, 2009). This rapid uptake is a clear sign that M-PESA fills a gap in the market. The question then is, can cloud computing help bypass costly IT resource needs for the developing world once again? There are indications that this will most likely be the case.  However, deliberate efforts need to be made in order to achieve this.

### 1.2.5 Free Cloud Services Patronage

Individuals and companies are increasingly using free cloud storage nowadays. Whether it is important documents, music, photos, or other files that need to be shared across multiple devices, using a cloud storage option is often the easiest way to do it. Some of the major free cloud storage services are Google Drive, Dropbox (Dropbox, 2013), Apple iCloud, Microsoft SkyDrive (Microsoft, 2013a), and Box (Box, 2013), whereas all the others are equally patronized by individuals and enterprises, Box is embraced slightly more by enterprise companies compared to individuals who want personal cloud storage options because it comes with a lot of other features and tools for businesses. Some of its features are content management, task management, online workspaces for better collaboration, a built-in editing system, and much more (Moreau, 2013). Other free cloud services being patronized by institutions and enterprises are Google Apps for Education (Google Inc., 2013), and Google App Engine. Most of these free cloud services normally do not sufficiently meet the IT resource needs of businesses because they are normally limited in their features since they were not designed to suit the needs of businesses. Additionally, the options of free cloud services that businesses could choose from are limited so they are not likely to find services that suit their needs. Furthermore, what happens when these big companies change their policies of providing these free resources? There is therefore a need for finding a sustainable way in which enterprises can contribute their spare IT resources to a platform so that other members on the platform can utilize them as and when they need. This leads to the need for my concept of Opportunistic Cloud Services (OCS), which is also expected to provide a larger pool of free cloud services to be chosen from.

## 1.3 Brief Overview of Opportunistic Cloud Services

Opportunistic Cloud Services is a social network approach to the provisioning and management of cloud computing services for enterprises; it deals with the concept of enterprises leveraging cloud services to meet their business needs without having to pay or paying a minimal fee for the services (Kuada and Olesen, 2011). The OCS network is modelled and implemented as a social network of enterprises collaborating strategically for the contribution and usage of cloud services without entering into any business agreements. It is a social network approach because, firstly, unlike commercial cloud service provisioning, no formal agreement exists between members on the OCS network. Secondly, it is about enterprises behaving in a manner similar to the behaviour of individuals in social networks. This is, acting in way of providing services without the sole or direct purpose of financial gains, but rather expecting some other forms of gain in the future.

The OCS platform consists of a set of services each belonging to a category; each service has a non-monetary cost that varies dynamically. The service or resource contributed by a member is of a certain finite capacity and the resources to a particular service may be contributed by multiple members. Members will normally only contribute resources that they have spare capacity of (e.g. processing capacity, storage, application that they have developed internally, etc.). That is, they package their spare IT resources as cloud services and make them available to the OCS platform. Members are free to provide and discontinue one or more services at will at any point in time. They are likewise free to use or discontinue the usage of one or more services at will at any point in time (Kuada and Olesen, 2012).

Figure 1-1 shows a high level overview of the coexistence of commercial cloud services with OCS services. The figure depicts a number of enterprises (A, B, …, Z) that can purchase cloud services from a number of commercial cloud service providers. In this figure, enterprise A has outsourced the resources needed for some of its business processes to Amazon and Microsoft. It is therefore making use of public cloud services. Enterprise A, however, also has its own internal IT resources that serve the rest of its business processes and organizational units. It has spare resources from its internal IT resources, which it has virtualized and made available as RsA to the OCS platform. Enterprise B serves all its organisational units and business processes from its own internal IT resources. Like Enterprise A, enterprise B also has spare IT resources which it has virtualized and made available as RsB to the OCS platform. Enterprise Z serves some of it business processes and organisational units from its own internal IT resources and outsource the rest to cloud vendor N. unlike enterprise A and B, it does not have spare resources that it can contribute to the OCS platform. As members of the OCS platform, all these enterprises can utilize resources on the OCS platform for meeting some of their IT resource needs as and when needed.

## 1.4 Opportunistic Cloud Services vs. Opportunistic Cloud Computing Services

During the nearly three-year period of undertaking this study, there has been a slight change in the tagging of my concept from Opportunistic Cloud Computing Services (OCCS) to Opportunistic Cloud Services (OCS). The removal of "computing" from the tagging is because over the short period of the introduction of the concept of cloud computing, it is becoming a norm to refer to the associated services as "cloud services" instead of "cloud computing services" since it gives the same meaning and probably because it is simpler. In accordance with the rest of the research community and the others in the industry, I found it more appropriate to retag my concept as opportunistic cloud services.

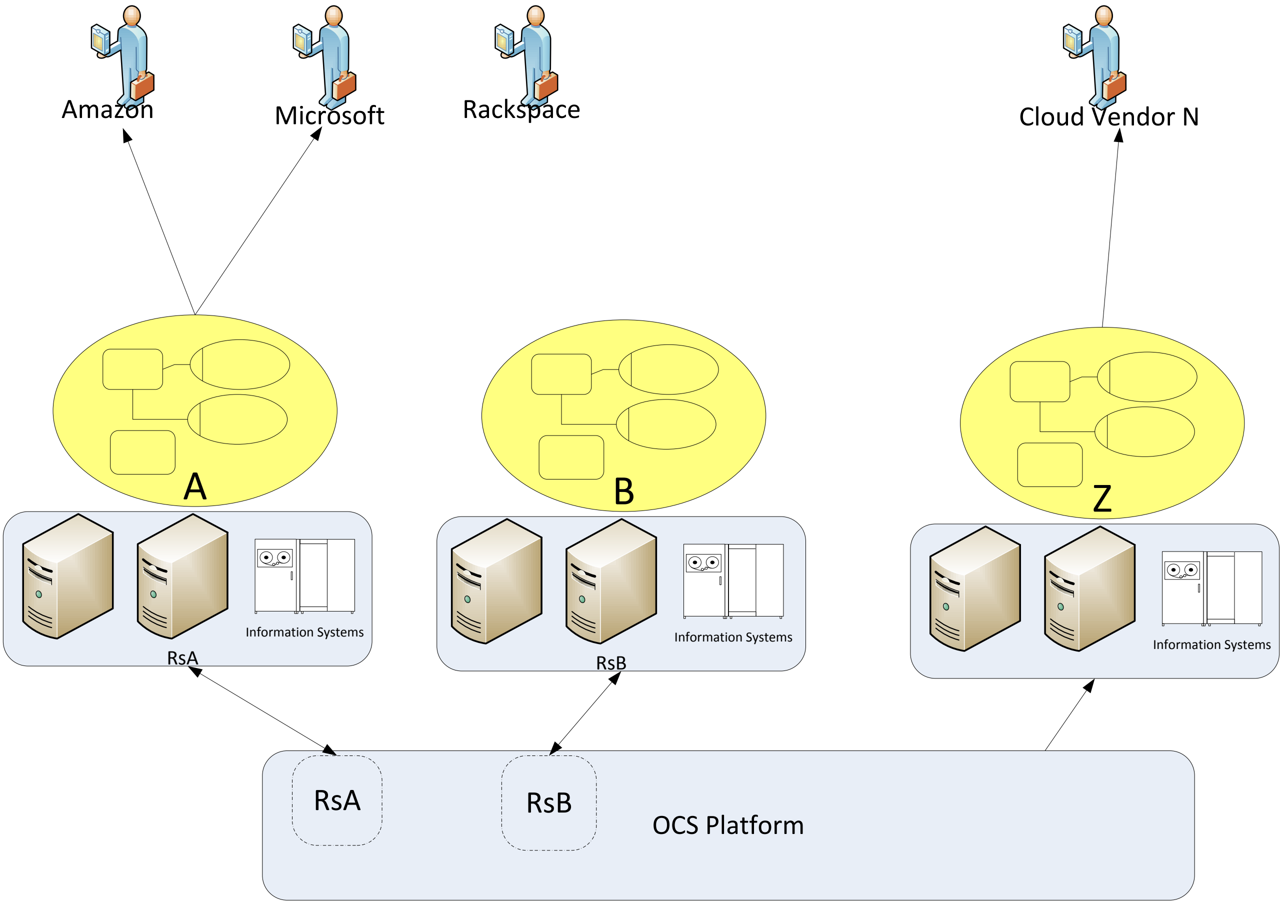


Figure 1-1: High Level overview of coexistence of Commercial Cloud Services and OCS.

## 1.5 Project Goals

The main goal of the project is to lay the foundation for the development of Opportunistic Cloud Services (OCS) management platforms that will engender the implementation of OCS networks. The main expected outcome, the research questions, and the specific project objectives are therefore derived based on this main goal.

## 1.6 Main Expected Outcomes of the Study

This main expected outcome section outlines only the major anticipated results which the study hoped to achieve at the end of the study when the study began. These expected outcomes included:

i. A conceptual framework, and a reference architecture for the implementation of opportunistic cloud services network for enterprises

ii. Incentive schemes that ensure sustainability of OCS platforms

iii. Trust management systems that will ensure that resources and services on OCS platforms will be useful and remain reliable.

## 1.7 Research Questions

The main project goal as mentioned above is the overriding factor that will guide this study. Based on this project goal, the derived research question is

*"What are the sufficient enabling conditions for the provisioning and utilization of opportunistic cloud services by enterprises?"*

From the stated main project objective and its derived research question, there is a need for an OCS platform to provide management of resources and services for such a network. I noticed that work on the design and development of the needed OCS platform alone was going to be very involving so it became necessary to focus the research appropriately. This initial research question therefore became too broad after the delimitation of the research phase which is presented in Section 1.9 below. A much focused research question was therefore derived based on the development of the OCS platform. The new research question which later became the main research question is

*"How can the platform for the opportunistic provisioning and utilization of cloud services by enterprises be designed and developed?"*

Because the delimitation of the research took a rather evolutionary path, this has slightly affected the research design and the associated methods employed in writing the individual included papers in this thesis. Additionally, even though the development of the OCS concept, together with the reference architecture and the design of incentive schemes for it form major parts of this research work, its main contribution has been the design of a trust management system for OCS platforms. This is evident from the contributions of the included papers. Thus, although the individual papers targeting these aspects of the research work form part of what I have termed the core papers, the natural progression of applying the results and lessons learnt from each paper as they are published has led to the main contributions of this research project coming in the later stages of the work. Furthermore, even though the work in the preceding papers can stand alone as major contributions in their own rights, the results from these papers were necessary in providing the details of the OCS platform needed in the work on the trust management for OCS platforms.

Based on the above analysis of the main contributions of this research project, without loss of generality, the main research question could be reformulated as

> *"How can a trust management system for opportunistic cloud services for enterprises be designed and developed?"*

## *1.8* Project's Objectives

The specific project objectives as outlined below are derived based on the main project goal of laying the foundation for the development of OCS management platforms. The objectives are therefore targeted towards aspects of the project that were considered essential in investigating the design and implementation of OCS management platforms. The specific objectives are to:

i. Develop a conceptual framework and reference architecture for the implementation of opportunistic cloud services network for enterprises
ii. Develop incentive schemes that ensure sustainability of the network
iii. Develop a trust management systems that will ensure that resources and services on the OCS platform will be useful and remain reliable
iv. Develop a framework for managing cloud services in the context where enterprises are both services providers and consumers; and subsequently tools that will aid enterprises in adopting, monitoring and managing cloud computing services
v. Evaluate current cloud computing management tools for their suitability for OCS
vi. Contextualize cloud computing as a business model for developing economies, and highlighting the role of opportunistic cloud services in such a model

The first five project objectives are aimed at providing answers to the main research question; and the last project objective stated above is aimed at providing some answers to the initial broader research question. The included papers that form part this thesis are targeted towards addressing the above mentioned project objectives.

## 1.9 Research focus: Delimiting the Research

Based on the expected project outcomes and the project objectives, some of the possible research directions that this study could have taken are: user centred, economic or commercial centred, and systems engineering centred. A systems engineering centred approach will be adopted for this study. Details about the research domains that the systems engineering research approach falls under are presented in the Methodology section in Section 3. A brief overview of what each of the other two possible research directions might have entailed had they been chosen for this study are also presented below.

### 1.9.1 User Centred

A user centred focus on this project will look at the usability of the OCS resources by enterprises in light of their expectations, and how the OCS platform should support them. This will require both quantitative and qualitative evaluation of the likelihood of the adoption of opportunistic cloud services by enterprises and other organisations. This will require conducting surveys and interviews to evaluate the factors that will influence their adoption of opportunistic cloud services. These factors can then be used in the design of the user facing aspects of the OCS platform, which will focus on what processes and tools must be put in place to support the usability of the services on an OCS network.

### 1.9.2 Economic or Commercial Centred

A commercialization centred focus on OCS might explore the economic benefits for enterprises, the fostering of business-to-business co-operation, and the innovation driven business ecosystem that it could bring about. Of particular interest is the type of innovation that the OCS concept could engender when it comes into fruition. This subsection therefore explores this topic.

A general conclusion that was reached at a PhD course on the discussion of inventions and types of innovations is that PhD projects may normally end up only as inventions; even if it manages to get past the invention stage into an innovation, it could mostly lead to an incremental innovation rather than a radical innovation (Gertsen, 2012). Some of the arguments that are made to support this claim are that, the very nature of the academic environment and the structure of PhD study having to build on existing work do not encourage radical innovations. Another generally accepted notion is that most originators of inventions and innovations did not foresee how their inventions will turn out to be in the future. I was therefore curious if my PhD research project could lead to a radical innovation sometime soon or in the near future and have begun exploring this possibility.

#### *1.9.2.1 Types of Innovation*

Innovation is an iterative process initiated by the perception of a new market and/or new service opportunity for a technology based invention which leads to development, production, and marketing activities that are aimed at the commercial success of the invention (Garcia and

Calantone, 2002). Innovations usually involve the commercialization of whatever invention it is; and it usually is the interplay between renewal of product and/or service, market, technology, organisation, business process, with the purpose of increasing the stakeholder values.

One way of categorizing innovation is by the level of impact they have on technology and business models in the value chain. These could be incremental innovations, semi-radical (breakthrough) innovations and radical (transformational) innovations (Gertsen, 2012). Incremental innovations are typically extensions to current product offerings or minor extensions to existing processes (McDermott and O'Connor, 2002). This normally involves minor changes in technology or product improvements that minimally improves the current performance of an existing product (Zhou et al., 2005). Radical innovations are normally novel or state-of-the-art technological advances in a product category that significantly change the consumption pattern of a market. There are basically two types of breakthrough innovations based on the advances in current technology and extent of departure from existing market segments (Zhou et al., 2005). Whereas a breakthrough innovation may either have significant impact on the existing technology or significant impact on the current market segments, radical innovations have both significant impact on the current technologies and market segment.

#### *1.9.2.2 Opportunistic Cloud Services as a Radical Innovation*

This section presents preliminary thoughts on the potential of Opportunistic Cloud Services (OCS) becoming a radical innovation. It begins with a look at if OCS can even be considered as an innovation and if so the type of innovation it is. The section ends with the discussion of its potential of becoming a radical innovation.

The OCS concept is built on cloud computing, which in itself is an incremental innovation. The first element of innovation that OCS brings is the paradigm shift of treating a company as both provider and a consumer of cloud services on the OCS platform. This in turn brings about a related requirement that cloud computing management tools now must be developed to meet this new demand. We can therefore conclude that the implementation of opportunistic cloud services for enterprise as it is today is an incremental innovation. Furthermore, the potential of OCS becoming a totally radical innovation is also very limited because the level of technological change introduced is minimal. Secondly, it is designed to be compatible with future cloud computing technologies and so cannot introduce or support disruptive technologies. OCS, however, has the potential of becoming breakthrough innovation as the concept can provide drastic reduction in cost for IT solutions and provide a platform that promotes the creation of an eco-system of cloud services and the creation of new business models based on it. This could have significant impact on the cloud computing market. Getting to this stage of a market-based breakthrough innovation however, will be a daunting challenge as the OCS platform will have to provide similar levels of security and reliability as services provided by commercial cloud service providers for the reduction in the cost of IT services that it promises to provide to be very attractive to companies.

### 1.9.3 Systems Engineering Centred

In order to delimit the research work and give the study clear focus, a system engineering focus is chosen. This was based mainly on the evaluation of the expected outcomes and the project objectives to identify the key objective around which the others revolve; furthermore the other possible research foci will still need a functional OCS platform in the long run and hence are dependent on the results of the system engineering approach which will lay the foundation for developing such a system. This decision, however, has also partly been influenced by the background and previous experiences of the researcher. Further justifications for taking a system engineering focus and how this has affected the methodology for this research work are discussed later in section 3.2. More details about some of the methodologies and research methods applicable to the systems engineering research direction are presented under the Methodology section in Section 3.

To ensure a holistic approach to this research project, a limited attention has still been given to some elements that fall under the commercial centred focus, by devoting one of the papers to researching on the implication of public policy and regulations on enterprises using resources on an opportunistic cloud services network; and another that examined the cost implications for cloud computing adoption by enterprises. Another reason why some attention has been given to aspects of the study that are not based on the system development approach is the main project objective of laying down the foundation for the development of OCS management platform that will hopefully lead to the implementation of OCS networks.

# 2 State-of-the-art

The larger part of this section deals with the state-of-the-art of selected areas of cloud computing. It also includes the state-of-the-art of the other concepts such as peer-to-peer (P2P) networks and online social networks, on which the OCS concept has been derived. The state-of-the-art on the P2P networks and Grid computing part is focused on incentive schemes and trust management because these topics are central to my work on the design of incentive schemes and development of a trust management system for the OCS concept.

## 2.1 State-of-the-art of Cloud Computing

The following subsections give a presentation of the state-of-the-art in cloud computing. It includes the characteristics of cloud computing, its service models, deployment models, major cloud computing technologies, and finally, the current challenges in cloud computing.

### 2.1.1 Characteristics of Cloud Computing

The roots of cloud computing can be traced by observing the advancement of several technologies. Some of these advancements that have had major impact on the development of cloud computing include advancements in hardware technologies such as hardware virtualization and multi-core chips, Internet technologies such as Web services, service-oriented architecture, and their associated Web 2.0 services. Advancements in other computing technologies that have equally impacted on the development of cloud computing are distributed computing technologies such as clusters, and grid computing; additionally, progress that have been made in systems management concepts and technologies such as autonomic computing, and data centre automation, have also contributed to the evolution of cloud computing. It is the maturity and convergence of these technology fields that significantly advanced and contributed to the advent of cloud computing (Buyya et al., 2010). This evolutionary path to cloud computing has made its definition and properties blurry. There are however, some essential characteristics that cloud computing can be identified with. Some of these main characteristics are on-demand self-service, network access, resource sharing, rapid elasticity, and metered (pay-per-use) service (Mell and Grance, 2011).

Based on the factors stated above, I have defined cloud computing as the repackaging of traditional IT resources such as storage, processing capacity, network services, user applications, and enterprise applications as virtualized resources which are then delivered by a service provider to customers as an on-demand pay-per-use service that can be self-provisioned by the user through a web portal over a network such as the Internet (Kuada et al., 2013).

### 2.1.2 Cloud Service Models

The three components of a regular computing environment, namely the hardware infrastructure, the operating system platform, and end user application software, have respectively translated into Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) delivered in cloud computing (Sultan, 2010), (Mell and Grance, 2011), (Beimborn et al., 2011), (Xu, 2012). However since cloud computing involves the repackaging of traditional IT resources, any IT solution can translate into a service in cloud computing. This leads to the concept of anything as a service (XaaS). Some of these examples are Data as a Service (DaaS), Confidentiality as a Service (CaaS) (Fahl et al., 2012), and Data Integrity as a Service (DIaaS) (Nepal et al., 2011).

### 2.1.3 Cloud Deployment Models

The cloud deployment models are a categorization of cloud computing services based on ownership and scale of the resources. Four main cloud deployment models can be identified. These are private cloud, community cloud, public cloud, and hybrid cloud. An organization can virtualize its data centre resources and deliver it to the users in its business units as cloud services. This kind of deployment is referred to as a private cloud. The creation and operation of a private cloud can normally be done by the organization’s own IT department or outsourced to a commercial cloud service provider. It is also typical for an organization to engage a cloud service provider in the building of the private cloud but then the operation is handled by their own internal IT department.

Public clouds are owned and managed by commercial cloud service providers. They are open to the general public to subscribe to and purchase resources on a pay-per-use basis.

Community clouds are similar to private clouds but whereas a private cloud is owned by a single administrative unit, a community cloud is created by multiple institutions with similar usage and security policies that come together on agreement to build a cloud solution to be used by all the participating parties. For example since universities generally have similar IT resource needs and security policies, the universities in the Copenhagen area may come together to create a community cloud to serve their users.

An organization operates in a hybrid cloud environment if it uses any combination of private, public, and community clouds. An organization can for example build its own private cloud but will also make provision for the usage of public cloud resources for services which its private cloud may not immediately be able to offer. Another reason that an organisation may want to operate a hybrid cloud is that because of security and other forms of risks, the organisation will want to handle sensitive data and business processes in its own private cloud but will at the same time want to process less critical data and business processes in the public cloud domain because it is cheaper.

### 2.1.4 Major Cloud Computing Technologies

The advancement of several technologies, especially in hardware such as hardware virtualization and multi-core chips, together with the maturity of Internet technologies such as Web services, service-oriented architecture, and their associated Web 2.0 technologies have contributed to the advent of cloud computing. It is the convergence of these technologies with other Internet computing concepts and technologies such as distributed computing, and systems management concepts such as autonomic computing, and data centre automation have all culminated into the evolution of cloud computing. Virtualization technologies and web services however stand out as two of these key technologies. This section highlights how these two technologies impact on cloud computing.

#### *2.1.4.1 Virtualization Technologies*

The Cloud model is based on two key characteristics: multi-tenancy, which is the sharing of the same service instance by multiple tenants, and elasticity, which allows tenants to scale the amount of their allocated resources based on current demands. Cloud services are therefore usually backed by large-scale data centres that are composed of thousands of computers. These data centres are design and built to serve many users and host many disparate applications. Hardware virtualization can be considered as a perfect fit to overcome most operational issues of data centre building and maintenance (Buyya et al., 2010).

The concept of virtualizing a computer system's resources such as processors, main memory (Random Access Memory (RAM)), and I/O devices, has been well established for decades. Hardware virtualization allows running multiple operating systems and software stacks on a single physical machine. A software layer termed the virtual machine monitor (VMM) or a hypervisor mediates access to the physical hardware by presenting to each guest operating system a virtual machine (VM); this is a set of virtual platform interfaces that operates as a complete server in its own right. Figure 2-1 illustrates the relationship between the physical hardware, the hypervisor, the virtual machines, the guest operating systems, and the user software that run on it. The advent of several innovative technologies such as multi-core chips, para-virtualization, and hardware-assisted virtualization, together with the live migration of virtual machines, has contributed to an increasing adoption of virtualization on server systems.

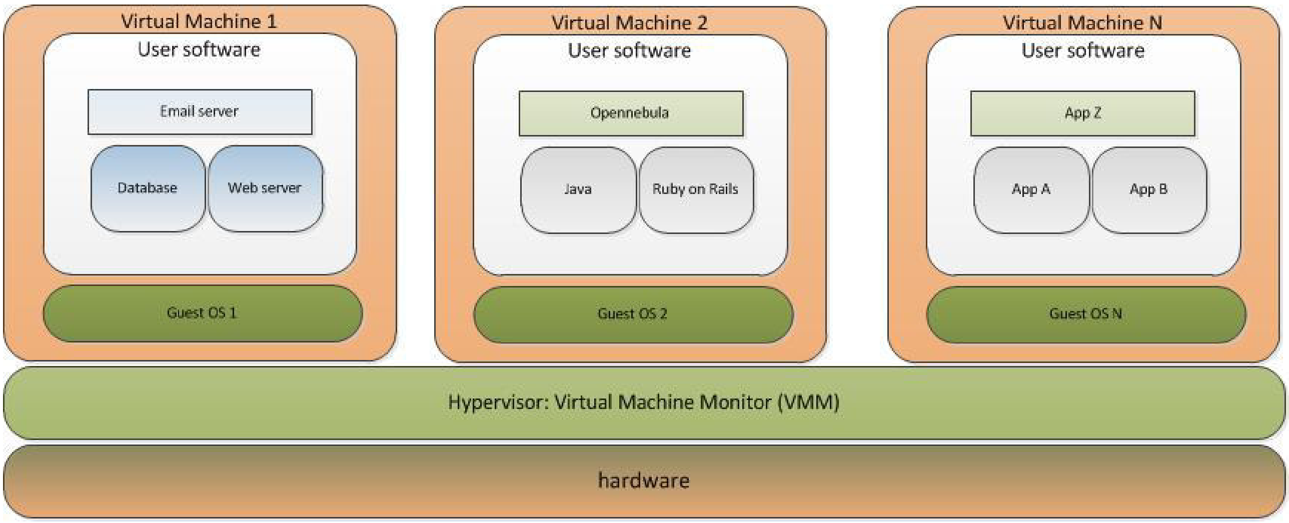


Figure 2-1: A hardware virtualized server showing the hypervisor and a number of virtual machines[1]

The traditional perceived benefits of hardware virtualization were improvements on sharing and utilization, higher reliability, and better manageability. More recently, three basic capabilities regarding management of workload in a virtualized system are being emphasized; these are namely isolation, consolidation, and migration (Uhlig et al., 2005). Workload isolation is achieved since all program instructions are fully confined inside a VM and leads to improvements in security. Furthermore, better reliability is also achieved because software failures inside one VM do not affect others. Better performance control is also attained since execution of one VM should not in theory affect the performance of another VM. The consolidation of several individual and heterogeneous workloads onto a single physical platform leads to better system utilization. Workload migration, also referred to as application mobility, targets at facilitating hardware maintenance, load balancing, and disaster recovery (Buyya et al., 2010). This is done by encapsulating a guest OS state within a VM, and allowing it to be suspended and fully serialized so that it can be migrated to a different platform. The migrated VM can then be resumed immediately or preserved to be restored at a later date. A virtual machine's state includes a full disk or partition image, necessary configuration files, and an image of its RAM. A number of VMM platforms exist; the most notable among them are VMWare (VMware, 2013), Xen (Xen® Hypervisor, 2013), and KVM (KVM, 2013), Citrix XenServer, Microsoft Hyper-V(Microsoft, 2013b), Oracle VM (Oracle VM, 2011).

[1] Image is adapted from (Voorsluys et al., 2011)

#### *2.1.4.2 Web Services*

The anything as a service paradigm of cloud computing has a direct inspiration from the Service Oriented Architecture (SOA) which began over two decades ago. Web services (WS), which is a particular and most prominent implementation of the service-oriented architecture is a major technology on which cloud computing hinges. A rich WS software stack has been specified and standardized over the years. This has resulted in a multitude of technologies to describe services, compose, and orchestrate them, package messages and transport them between services, publish service and discover published ones, represent quality of service parameters, and also ensure security in service access (Voorsluys et al., 2011).

WS standards have been created on top of existing technologies such as HTTP and XML to provide a common mechanism for delivering services; and make them ideal for implementing a service-oriented architecture. The purpose of a SOA is to address requirements of loosely coupled, standards-based distributed computing that are independent of the specific protocol. The software resources in SOA are packaged as "services," which are well-defined, self-contained modules that provide standard business functionality and are independent of the state or context of other services (Erl, 2004), (Buyya et al., 2010), (Jamil, 2009). Services are described in a standard definition language and have a published interface. Many service providers, such as Amazon, Facebook, and Google, make their service APIs publicly accessible using standard protocols such as Simple Object Access Protocol (SOAP) and REST (Buyya et al., 2010). Consequently, one can put an idea of a fully functional Web application into practice just by gluing pieces with few lines of code.

Cloud applications can be built as compositions of other services from the same or different providers to form new SaaS applications. Services such as user authentication, e-mail, payroll services, and calendars are examples of building blocks that can be reused and combined in a business solution. An important aspect of cloud computing is the management of cloud services and the self-provisioning of cloud services from a Web portal. These operational challenges, together with interoperability of cloud services are solved through cloud service providers publishing the APIs to their services as cloud APIs to the general public. An example is Amazon's Amazon Web Services (AWS) published cloud APIs (Amazon Web Services, 2013).

### 2.1.5 The Move toward Cloud APIs

Some of the major findings of this study that I have not yet published include maturity of open source cloud management systems, progress in interoperability and standardization efforts, and emergence of dominant cloud APIs. The factors that are described below have played a significant role in the development of Cloud APIs, which as will be discussed later, forms the bases for Cloud abstraction APIs. The development of Cloud abstraction APIs is a major factor that will facilitate the implementation of OCS management platforms.

#### *2.1.5.1 Maturity of Open Source Cloud Management*

The main cloud infrastructure management products offer similar core features such as support for different cloud deployment models (often referred to as hybrid clouds); support for the on-the-fly creation and provisioning of new objects and the destruction of unnecessary objects like virtual machine instances, storage, and/or application instances. These management tools also provide a suite of reports on status (uptime, response time, quota usage, etc.), and have a dashboard for displaying these information. Some of the major vendors meeting these three criteria by offering approaches in handling provisioning and managing metrics in hybrid environments are RightScale (RightScale, 2013), Kaavo (Kaavo, 2013), etc.

There are also open source cloud management projects providing cloud management tools suitable for enterprise scale deployment. Of particular interest to my project, is the continuous maturity of open source cloud management tools such as ***OpenNebula*** (Opennebula, 2013), ***OpenStack*** (OpenStack, 2013), ***CloudStack*** (CloudStack, 2013), ***Eucalyptus*** (Eucalyptus, 2013), etc. These open source tools support major hypervisors such as VMware, Oracle VM, KVM, XenServer and Xen. They also provide or support one or more Cloud APIs in the industry. This allows for the extension of the functionalities of their provided dashboards with new features, or the development of new tools that interact with cloud deployments using these tools. They also have large community support spearheading their development and are backed by major organisations and cloud vendors. For example ***CloudStack*** and ***OpenStack*** are both projects under the Apache Software Foundation (Apache Software Foundation, 2013), but are backed by different cloud vendors.

#### *2.1.5.2 Interoperability and Standardization*

There is a move towards interoperability and standardization of cloud management technologies. Open standards in the cloud benefit everyone and can dramatically reduce the time it takes to move between public and private clouds or operate a hybrid cloud, and to deploy software in cloud environments. It promotes interoperability, which allows cloud service consumers to easily switch among multiple cloud services providers when necessary. Various standards, together with proprietary and open APIs have been proposed to provide interoperability up and down the three layers (IaaS, PaaS, and SaaS) of the cloud computing stack.

Some of these interoperability efforts are the open cloud computing interface, and the cloud computing interoperability forum. The Open Cloud Computing Interface (OCCI) (Open Cloud Computing Interface, 2011) comprises a set of open community-lead specifications delivered through the Open Grid Forum. The Cloud Computing Interoperability Forum (CCIF) is an open, supposedly vendor neutral community of technology advocates and consumers that are dedicated to driving the rapid adoption of cloud services (Pattenden et al., 2011).

#### *2.1.5.3 Cloud APIs*

Major Application Programming Interfaces (APIs) for cloud computing are beginning to emerge. Most of these APIs are based on open standards; but there are proprietary ones, such as AWS Query API, that have gained industry support. This subsection outlines some of these main Cloud APIs.

***OpenNebula Cloud APIs***

OpenNebula has been designed to be easily adapted to any infrastructure and extended with new components so that it can implement a variety of Cloud architectures and can interface with multiple datacentre services (Opennebula, 2013). OpenNebula implements two different interfaces, namely EC2-Query API which is Amazon's proprietary Cloud API, and the OCCI API which is based on the OCCI standard. This means that any EC2 Query tool can be used to access an OpenNebula Cloud.

***CloudStack Cloud APIs***

CloudStack provides an API that gives programmatic access to all the management features available in the user interface; it enables the creation of command line tools and new user interfaces to suit particular needs (CloudStack, 2013). It also provides an API that is compatible with AWS EC2 and S3 for organizations that wish to deploy hybrid clouds. The API is REST like and returns responses to request in JSON or XML format.

***Amazon Web Services Cloud API***

Amazon Web Services (AWS) provide a web services based Query API for Amazon EC2. This together with their software development kits (SDK) for AWS enable users to access Amazon EC2 from their preferred programming language (Amazon Web Services, 2013). Eucalyptus implement the AWS API on top of Eucalyptus cloud management tool, so cloud management tools in the cloud ecosystem that can communicate with AWS can use the same API with Eucalyptus. On March 22, 2012, AWS and Eucalyptus announced an agreement that enables customers to more efficiently migrate workloads between their existing data centres and AWS while using the same management tools and skills across both environments. A part of this agreement is that, Amazon will support Eucalyptus as they continue to extend compatibility with AWS APIs and customer use cases (Eucalyptus, 2013).

It is evident from the above that Eucalyptus, CloudStack and OpenNebula's implementation of Amazon's proprietary EC2-Query API shows that it has gained wide industry support even though it is a proprietary API.

### 2.1.6 Current Challenges in Cloud Computing

The major challenges in cloud computing can be categorized into trust challenges, security challenges, privacy challenges, and other risk management challenges like weak and ambiguous SLAs and their associated legal issues, vendor lock-in, etc. As vendors start to deploy cloud services, and users upload data in the Cloud to utilize them, new privacy concerns arise because data owners would like to preserve the confidentiality of their data, and even their identities private from the software provider. While cloud service providers pledge to preserve data privacy, the current Software as a Service (SaaS) architecture makes it difficult to provide any assurance that the software in the Cloud will not be able to make copies or redistribute the data it used (Song et al., 2010). Apart from these consumer concerns, cloud architectures also introduce new classes of security risks and attacks over the resources of cloud service providers. These include poisoned virtual machines, attacks against the cloud service provider's management console, attacks based on knowledge of default security settings, abuse of billing systems, and data leakage via uniform resource locators. Cloud service providers do not currently have robust technical solutions that can protect their cloud resources from harmful malware, virus infection, botnets, distributed denial of service attacks, or other types of cyber-attacks. Furthermore, there is no effective mechanism to help cloud users evaluate the security measures of their service providers and ensure the protection of their data while taking into consideration industry standards or personal preferences (Park et al., 2012).

Another important issue in cloud computing is the accountability of the resource usage data. Who should be responsible for performing the measurements required for resource usage data? Is it the provider, the consumer, a trusted third party or some combination of them? Although there are currently no equivalent facilities of consumer-trusted metering as is the case in traditional utility services like electricity and water, provider side accountability is the basis for cloud service providers. Cloud service consumers have no choice but are forced to take whatever usage data made available by the service provider as trustworthy (Mihoob et al., 2010). This situation leads to lack of transparency and hence lack of trust in the cloud services environment.

Some of the other challenges that cloud service consumers are faced with are vendor lock-in and dealing with multiple cloud service providers and their management tools. This is mainly due to cloud service providers' proprietary cloud management tools which are incompatible with each other. Cloud service consumers therefore find it difficult to move their data and business processes from on cloud service provider to another. Furthermore, this leads to cloud service consumers having to master multiple cloud management environments in order to make such a switch or use the services of multiple cloud providers at the same time.

## 2.2 Incentive Schemes in Peer-to-Peer Networks and Grid Computing

Peer-to-peer (P2P) networks are a community of users (nodes/agents) in which the individual user acts as a resource provider in one transaction and in another instance acts as a resource utilizer. That is, they act as both a server and client on the network. The resources offered on P2P networks are usually file sharing services in which users contribute files to the network and are able to download files available on the network that they are interested in. The fundamental premise of P2P networks is that of voluntary resource sharing among peers, but this is usually not the case because of the fact that the individual peers tend to be strategic in their resource sharing; this in turn tends to threaten the sustainability of P2P networks (Feldman and Chuang, 2005). That is, the individual rationality of the peers tends to lead to the free-riding problem in which some peers only use resources without providing any. This usually also result in the tragedy of the commons problem of only a few peers supporting the P2P network (Ma et al., 2006).

It is in the attempt of reducing the above-mentioned problems and also to provide service differentiation on P2P systems that many research works have looked at the design of incentive schemes in P2P networks. For example, a survey by Feldman and Chuang on incentive schemes for P2P systems outlines some of the proposed incentive schemes for P2P systems as inherent generosity, monetary payment schemes (in the form of micropayments), direct reciprocity (tit-for-tat) based schemes, and indirect reciprocity (reputation) based schemes (Feldman and Chuang, 2005).

Mekouar et al. proposed a reputation based contribution management scheme for partially decentralized P2P systems and introduced a concept of contribution behaviour for service differentiation (Mekouar et al., 2006). Zhao et al.'s work on adaptive incentive protocols for P2P networks presented a general analytic framework to analyse and design incentive protocols; and demonstrated the relation between their analytic framework and evolutionary game theory in identifying the robustness of an incentive scheme (Zhao et al., 2009) (Zhao et al., 2012).

The unique characteristics of multimedia streaming systems caused researchers in the field of P2P streaming systems to develop incentives schemes particularly targeting these systems because they argue that those designed for the traditional P2P file sharing services are not suitable for live streaming P2P services. For example, Pianese et al.'s work on P2P live streaming tries to use incentive schemes to promote cooperation among users in supporting the content distribution systems with their resources (Pianese et al., 2007). Silverston et al.'s work proposed incentives for peers not only for providing resources but also for supporting other peers to discover potential resource providers (Silverston et al., 2008).

Another collaborative computing model is Grid computing. The motivation for computational Grids was initially driven by large-scale resource intensive scientific applications that require more resources than a single computer (workstation, supercomputer, or cluster) could provide in a single administrative domain. Grid computing enables the sharing, selection, and aggregation of a wide variety of geographically distributed resources including supercomputers, storage systems, data sources, and specialized devices owned by different organizations for solving large-scale resource intensive problems in science, engineering, and commerce (Buyya et al., 2005). Its origins in the scientific and research community made many assume that resource owners would altruistically make their resources available for others to use; but this was not the case so it became necessary for incentive mechanism to be designed to make Grid environments sustainable.

Although the incentive mechanisms for P2P networks are applicable to Grid environments, the mission critical nature of the services in Grid environments as compared to P2P networks has attracted the proposal and application of economic models in Grid environments. For example Buyya et al.'s, work on "The Grid Economy", identified distributed resource management challenges and requirements of economy-based Grid systems, and discussed various representative economy-based systems for cooperative and competitive trading of resources such as CPU cycles, storage, and network bandwidth (Buyya et al., 2005). Bapna et al.'s work on, market design for Grid computing, developed a market-based resource-allocation model that adds an economic layer to the then current approach of treating resource allocation as primarily a scheduling issue. They designed a value-elicitation and allocation scheme that provide the economic incentives for buyers and sellers of computing resources to exchange assets (Bapna et al., 2008).

## 2.3 Trust Management in Peer-to-Peer Networks and Grid Computing

The open nature of P2P systems and the anonymity that peers usually enjoy in P2P networks make them susceptible to malicious users flooding the network with self-replicating unauthentic and malicious files. This poses trust problems for P2P environments. It is in attempt to solve this problem that many studies have proposed diverse trust management systems for P2P network environments. For example the *EigenTrust* reputation system presented an algorithm to decrease the number of downloads of inauthentic files in a P2P file-sharing network by assigning each peer a unique global trust value based on its history of uploads. These global trust values help users to choose the peers from whom they download, and also allow the network to effectively identify malicious peers and isolate them from the network (Kamvar et al., 2003).

Although reputation based trust management systems such as *EigenTrust* have been proven to be very useful, this deals with only one aspect of the problem of trust because according to Singh and Liu, an important challenge in managing such trust relationships is to design a protocol to secure the placement and access of these trust ratings. They therefore proposed the *TrustMe* protocol as a secure and anonymous underlying protocol for trust management in P2P networks (Singh and Liu, 2003). The analysis of the *TrustMe* protocol by Li and Singhal in their work on trust management in distributed systems claims that although *TrustMe* protects trust-holding-agent (THA) peers' identities, a malicious THA peer can disseminate negative trust information for a peer. They also identified the prevention of malicious peers from becoming THA peers and preventing peers from reporting a wrongful trust value for another peer as some of the problems with the *TrustMe* protocol (Li and Singhal, 2007).

Different studies have applied different mathematical modelling in computing trust values in trust management systems for P2P systems, but central to all these studies is that they normally have a reputation system as the base for these trust management systems. For example, Zhou and Hwang's work on *PowerTrust* justifies the applicability of the power-law distribution to user feedbacks in any dynamically growing P2P systems, and employs it in their *PowerTrust* reputation system for P2P networks (Zhou and Hwang, 2007). Despotovic and Aberer also identified two broad areas of reputation systems for P2P networks. These are social networks that rely on aggregating the entire available feedback in the network in the attempt of achieving as much robustness against possible misbehaviour as possible, and probabilistic models that rely on the well-known probabilistic estimation techniques but use only a limited fraction of the available feedback (Despotovic and Aberer, 2006).

Even though anonymity is not a key characteristic of Grid computing as it is in P2P networks, the collaborative and voluntary nature of Grid computing makes it not immune to malicious and poor resource offerings. There have therefore been several studies on trust management systems for Grid environments. Some of these works include Lin et al.'s work to enhance security in Grid environments by proposing the incorporation of their trust management architecture into security solutions for Grid computing (Lin et al., 2004). Another such work on trust management for Grid computing is Dominigues et al.'s work on sabotage tolerance and trust management with a mechanism for sabotage detection and a protocol for distributed trust management which is targeted at volunteer-based computing commonly used in desktop Grids (Domingues et al., 2007).

## 2.4 Opportunistic Cloud Services Concept Formation

The OCS concept has been inspired by observing the trends of evolution of Internet services, particularly the sharing of resources in P2P networks. It therefore leverages properties of P2P networks such as being built on architecture of participation and in harnessing the power of the crowd. Another Internet phenomenon that relies on this architecture of participation is online social networks (OSN). These two concepts have inspired a new generation of P2P networks - OSN-P2P environments (Pouwelse et al., 2008), (Olteanu and Pierre, 2012), (Fast et al., 2005), (Yang et al., 2004) that try to leverage the social relationships in OSN in providing better services and improved performance in P2P networks. Other studies such as (Schiöberg, 2008), (Buchegger and Datta, 2009), (Buchegger et al., 2009) also try to deal with some of the security and privacy challenges in OSN by taking advantage of the distributed nature of P2P networks. These new generation OSN-P2P networks, like their traditional P2P networks, only provide basic file sharing services to members. The OCS concept goes beyond this simple file sharing services as is presented in the next section below.

## 2.5 Beyond State-of-the-art

The approach of the OCS concept is in two parts. The first is to leverage the advent of cloud computing and the supporting technologies in providing cloud services to members in an environment similar to that of OSN-P2P environments. The cloud services being considered here include the entire cloud services stack. It includes fundamental cloud services such as SaaS, PaaS, IaaS, and any arbitrary cloud service. The second distinguishing feature of the OCS concept compared to OSN-P2P environments is that it focuses on the provisioning and utilization of these cloud services in the context of enterprises instead of individual personal usage which normally requires no or minimal quality of service compared to the enterprise usage scenario.

# 3 Methodology

This methodology section presents the adopted research approach and the justification for it. It also presents the main methods and theories that have been used in undertaking this research work. The section begins with specifying the adopted research approach as system engineering and puts it in the context of the type of research it falls under; this is followed with a justification for adopting this research approach. The major methods that have been employed based on the research approach are system design, systematic review, mathematical proofs, and simulations. The section ends with a brief overview of the major theories that have been made use of in this project; these include game theory and grounded theory.

## 3.1 Systems Engineering as a Research Approach

The adopted research methodology for this study is system development. The idea of system development as a research methodology fits comfortably into the category of applied science. It belongs to the categories of engineering, developmental, and formulative types of research (Nunamaker Jr and Chen, 1990). The main processes involved in system development as a research methodology are requirements analysis and specification, system analysis, system design, system implementation, and the evaluation of the developed system.

### 3.1.1 Applied Research

Basic research involves developing and testing theories and hypotheses in response to the intellectual interests of the researcher rather than for practical reasons. Applied research, on the other hand, is the application of knowledge to solve problems of immediate concern (Bailey, 2008), (Blake, 1978), (Nunamaker Jr and Chen, 1990), (Jain et al., 2010). That is, basic or fundamental research is experimental or theoretical work undertaken primarily to acquire new knowledge of the underlying foundation of phenomena and observable facts without any particular application or use in view; whereas applied research is also original investigation undertaken in order to acquire new knowledge, contrary to basic research, it is directed primarily towards a specific practical aim or objective.

### 3.1.2 Engineering Research

Both engineers and pure scientists employ similar research methods such as applying the systematic approach in their investigations, and are concerned with confirming their theoretical predictions. They however differ in the scale of their experiments and their motives. In comparison with the pure science research approach, the artistry of design and having the edge of developing something functional are essential and therefore emphasised in the engineering research approach (Brooks, 1994), (Nunamaker Jr and Chen, 1990), (Wallace and Blessing, 2007).

### 3.1.3 Development Research

Design and development research (at times simply known as developmental or development research) seeks to create knowledge grounded in data systematically derived from practice. It is defined by Richey and Klein (2007) as "the systematic study of the design, development and evaluation processes with the aim of establishing an empirical basis for the creation of instructional and non-instructional products and tools and new or enhanced models that govern their development" (Richey and Klein, 2007). It is the systematic use of scientific knowledge directed toward the production of useful materials, devices, tools, systems, or methods. This may include the design and development of prototypes and processes (Blake, 1978). It is normally classified as being advanced, exploratory, engineering, and operational (Nunamaker Jr and Chen, 1990).

### 3.1.4 Formulative Research

Another categorization of research approaches is formulative vs. verificational research. The main aim of formulative or exploratory research is to identify problems for more precise investigation, to develop hypotheses, and/or to gain insights and to increase familiarity with the problem area. In contrast, the goal of verification research is to obtain evidence to support or refute formulated hypotheses (Grosof and Sardy, 1985), (Nunamaker Jr and Chen, 1990).

## 3.2 Justification for Adopting System Development Methodology

The reason for adopting system development as the research methodology for this study is that it fits the main project objective of laying the foundation for the development of Opportunistic Cloud Services (OCS) management platforms that will engender the implementation of OCS networks. Consequently, the system development methodology is therefore applicable to the main individual objectives of the study as stated in Section Project's Objectives1.8 above.

## 3.3 Applied Methods

The main research methods that have been employed in writing the individual papers that constitute the main results of this project include system design, systematic review, mathematical proofs and simulations. The following subsections give brief overview of these research methods.

### 3.3.1 System Design

It is important to have a defined design procedure that finds good solutions. Design processes have two main complementary parts - design science and design methodology (Wallace and Blessing, 2007). Design science uses scientific methods to analyse the structures of technical systems and their relationships with the environment; with the aim of deriving rules for the development of these systems from the system elements and their relationships. Design methodology, however, is a concrete course of action for the design of technical systems that derives its knowledge from design science and cognitive psychology, combined with practical experience in different domains. The design methodologies applicable to this project are that of software system design (engineering) methodologies (Fitzgerald, 1996); particularly, the reductionist subdivision of development process which is broken into the broad categories of analysis and specification of requirements, the design of a solution, and the implementation of that solution (Fitzgerald, 1996), (Olerup, 1991).

### 3.3.2 Systematic Literature Review

The systematic literature review methodology will be employed during some of the stages of this project to give adequate insight into specific topics of interest when needed. A systematic literature review is a means of identifying, evaluating, and interpreting available research relevant to a particular research question, topic area, or phenomenon of interest (Kitchenham and Charters, 2007) (Brereton et al., 2007). It is a comprehensive systematic and reproducible method for identifying and evaluating the existing body of completed and recorded work produced by researchers, scholars, and practitioners on a specific topic of interest; and then synthesizing the results to produce new knowledge on the topic (Okoli and Schabram, 2010). The accumulation of evidence through secondary studies can be very valuable in offering new insights or in identifying where an issue might be clarified by additional primary studies. The systematic literature review process consists of three main stages; these are namely: inputs, processing, and outputs (Levy and Ellis, 2006). The eight step guide of (Okoli and Schabram, 2010) to conducting systematic literature review are: purpose of the literature review, protocol and training, searching for the literature, practical screen, quality appraisal, data extraction, synthesis of studies, and finally writing the review report. They recommend all these steps as essential for a review to be scientifically rigorous.

According to Kitchenham and Charters (2007), the stages in a systematic literature review can be summarized into three main phases: planning the review, conducting the review, and then reporting the results of the review. The stages associated with *planning the review* include identifying the need for a review; i.e. there need to be a justification for the review because those who will commission such a review will normally need such a justification in order to invest funds for undertaking the study. The next item in *planning the review* phase is the specification of the research questions that are of interest to be answered during the study; and thirdly, developing a review protocol to be followed in undertaking the review. The first of the main stages associated with *conducting the review* is the identification and selection of relevant existing research which will serve as primary studies. Next, quality assessment must be performed on these studies. The relevant data needed for answering the research questions are then extracted and synthesized to prepare it for the next stage. The stages associated with *reporting the review* are: specifying the dissemination mechanisms, and formatting the main report accordingly to suit the relevant audience.

### 3.3.3 Mathematical Proofs

The systematic nature of research implies the need for research methods or systems that provide a model or structure for logical argument. These structures of making logical arguments are namely proof by demonstration, empiricism, mathematical proof, and hermeneutics (Johnson, 1997).

When I began my study on the design of incentive mechanism for the opportunistic cloud services platform, my first intuitive approach was to find simulators that will demonstrate the existence of a Nash-equilibrium. Due to the limitations of simulators in performing this kind of tasks when the number of agents involved is relatively large, I had to rethink that approach and finally had to employ mathematical proofs. A typical mathematical proof is a sequence of inferential steps between statements so that it can be seen as an argument because each statement except the first one is supported by the reasons given by previous ones (Dufour, 2013), (Harrison, 2008), (Hales, 2008).

Though there are many problems that limit the usefulness of mathematical proofs as a general research approach such as it being prone to mistakes in the argumentation, and the complex nature of the mathematics that are often used makes such mistakes very difficult to detect (Johnson, 1997); these limitations notwithstanding, hermeneutics and proof by demonstration are usually not suitable options in the systems engineering research field. This is because hermeneutics requires that an artefact be observed within its intended environment; and since this artefact in normally non-existent during a system engineering research project, but rather may come as an outcome at the end of the study, hermeneutics is therefore normally not a suitable research method.

Furthermore, until an artefact is developed, the proof by demonstration, besides its other limitations, is neither feasible. This leaves mathematical proofs as the option in the systems development research methodology. Furthermore, in certain specific cases, besides the infeasibility of developing prototypes for demonstration, availability of emulators or even simulators is non-existent. The only option then is the employment of formal methods through mathematical proofs to show the existence of certain desired design properties of the system being developed.

### 3.3.4 Simulation

The unprecedented successes of modern computer simulation approaches and the potential of their enhancement of the utility, accuracy, and reliability of simulation due to advances in data related systems has led to the emergence of a discipline, referred to as Simulation Based Engineering Science (SBES) (Belytschko et al., 2004). Simulation-Based Engineering Science involves the use of computer modelling and simulation to solve mathematical formulations of physical models of engineered and natural systems (Glotzer et al., 2009). The use of computer simulation in engineering systems began many decades ago, but only in the last two decades or so has it become an essential scientific methodology for research and education in nearly all areas of engineering and in many branches of science. There are several factors contributing to this progress. The first is the steady advances in computational science that made it possible to extend the range and depth of applications of simulation as a key methodology. Secondly, in almost all areas of engineering and science, computer simulation has enabled researchers to study and predict the occurrence of physical events from the results of their theoretical investigations. It also provides an alternative to the experimental science when phenomena are not observable or measurements are impractical or too expensive (WTEC, 2007).

Simulations will be employed in this project to study characteristics of the mathematical models developed in studying some of the subsystems in the designed systems.

## 3.4 Applied Theories

This section provides a brief overview of the major theories that have been employed or are of relevance to the central concept in this project. These include game theory and grounded theory.

### 3.4.1 Game Theory

Game theory (Leyton-Brown and Shoham, 2008), (Osborne and Rubinstein, 2009), (Camerer, 2011) is a bag of analytical tools designed to help in the understanding of the phenomena that are observed when decision-makers interact. It has been found to be applicable to diverse categories of problems in different fields over the past several decades. The models of game theory are highly abstract representations of classes of real-life situations. The basic assumptions that underlie the theory are that decision-makers pursue well-defined exogenous objectives (they are *rational*) and take into account their knowledge or expectations of *other* decision-makers' behaviour (they *reason strategically*) (Camerer et al., 2001). The abstract nature of the models allows them to be used to study a wide range of phenomena (Camerer, 2011). For example, the theory of Nash equilibrium has been useful in the study of oligopolistic and political competitions; the theory of mixed strategy equilibrium has been used to explain the distributions of tongue length in bees and tube length in flowers; the theory of repeated games has been used to illuminate social phenomena like threats and promises (Camerer, 2011).

Mechanism design is the sub field of microeconomics and game theory that considers how to implement good system-wide solutions to problems that involve multiple rational agents, each with private information about their preferences (Kuada and Olesen, 2012), (Nisan, 2007). It is best to view the goals of the designed mechanisms in the very abstract terms of social choice. A social choice is an aggregation of the preferences of the different participants towards a single joint decision. Mechanism design attempts to implement desired social choices in a strategic setting. Such strategic design is necessary since usually the preferences of the participants are private (Nisan, 2007).

The concepts of game theory - particularly those of mechanism design - are employed in the design of incentives for encouraging resource contribution and efficient usage of these resources for the design of certain parts of the subsystems that constitute the systems in this project.

### 3.4.2 Grounded Theory

A philosophy that will generally guide this research is grounded theory (Corbin and Strauss, 1990), (Walker and Myrick, 2006). This is a methodology that was first laid out in 1967 by Glaser and Strauss, and since then tends to be a popular form of inquiry in the areas of education and health research (Dawson, 2002). The emphasis in this methodology is on the generation of theory, which is grounded in the data; i.e. theory that has emerged from the analysis of empirical data. With grounded theory, data collection and analysis are interrelated processes; this enables the research process to capture all potentially relevant aspects of a topic as soon as they are discovered. It is a theory of discovery and one that grounds a theory in reality. Concepts are the basic units of analysis; the researcher works with the conceptualizations of data, but not the data itself per se. Analysis makes use of constant comparison. Making comparisons assists the researcher in guarding against bias. He is then challenging concepts with fresh data.

Grounded theory has in recent years been adapted to suit interpretive research in fields of Information Systems evaluation. Jones and Hughes outline some of the research projects that typify the way grounded theory has been used in a variety of Information Systems projects (Jones and Hughes, 2001). Hansen and Kautz (Hansen and Kautz, 2005) also show how grounded theory was used to study whether and how system developers use information systems development methodologies in practice.

Aspects of the processes involved in grounded theory have been used in my paper on the systematic review of trust related studies in cloud computing. This systematic review sought to find state-of-the-art knowledge on trust engineering in cloud computing; this knowledge was then applied in the design of trust management system for opportunistic cloud services.

# 4 Results and Discussions

The processes that are involved in the system development methodology are requirements analysis and specification, system analysis, system design, system development, and the evaluation of the developed system. This dissertation covers the first three processes of the requirements analysis and specification, system analysis, and the system design of the OCS management platform. The magnitude of work required for the analysis and design of the OCS platform, together with limitations on current technologies has made the development of the OCS platform and its evaluation to be outside the scope of the results included in this dissertation. These, however, remain future work to be done and are currently being explored.

The main results of the research work are the six papers that form part of the dissertation. Table 4-1 shows the publication information of each of the included papers and their current status. It also has remarks on the sequence in which they have been published.

The first part of this section gives the relationship between the included papers. This is followed by the summary of the each of the papers. The section ends with how each of these papers contributes to the fulfilment of the project objectives.

**Table 4-1: The six included papers that make part of this dissertation.**

| Paper | Publication Information | Status | Remarks |
|---|---|---|---|
| Paper #1 | E. Kuada and H. Olesen, “A Social Network Approach to Provisioning and Management of Cloud Computing Services for Enterprises,” presented at the CLOUD COMPUTING 2011, The Second International Conference on Cloud Computing, GRIDs, and Virtualization, 2011, pp. 98–104. | Published | This is the first paper that was published |
| Paper #2 | E. Kuada and H. Olesen, “Incentive mechanisms for Opportunistic Cloud Computing Services,” in *2012 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, 2012, pp. 127 –136. | Published | This is the third paper that was published |
| Paper #3 | Kuada, Eric. 2014. “Towards Trust Engineering for Opportunistic Cloud Services: A Systematic Review of Trust Engineering in Cloud Computing.” Submitted to J. of Grid Computing. | Under review | |
| Paper #4 | Kuada, Eric. 2013. “Trust Management System for Opportunistic Cloud Services.” In *2013 IEEE 2nd International Conference on Cloud Networking (CloudNet).* San Francisco, USA: IEEE. | Published | This is the fifth paper to be published |
| Paper #5 | E. Kuada, H. Olesen, and A. Henten, “Public Policy and Regulatory Implications for the Implementation of Opportunistic Cloud Computing Services for Enterprises,” in *Workshop on Security in Information Systems*, Wroclav, 2012. | Published | This is the second paper to be published |
| Paper #6 | Kuada, Eric, Kwami Adanu, and Henning Olesen. 2013. “Cloud Computing and Information Technology Resource Cost Management for SMEs.” In *Proceedings of IEEE Region 8 Conference EuroCon 2013*, 258 – 265. University of Zagreb, Croatia:IEEE. | Published | This is the fourth paper to be published |

Figure 4-1 shows the relationships between the included papers. It tries to give a high level overview of how the objectives and content of the individual papers are related; and the level of impact of a paper on the others.

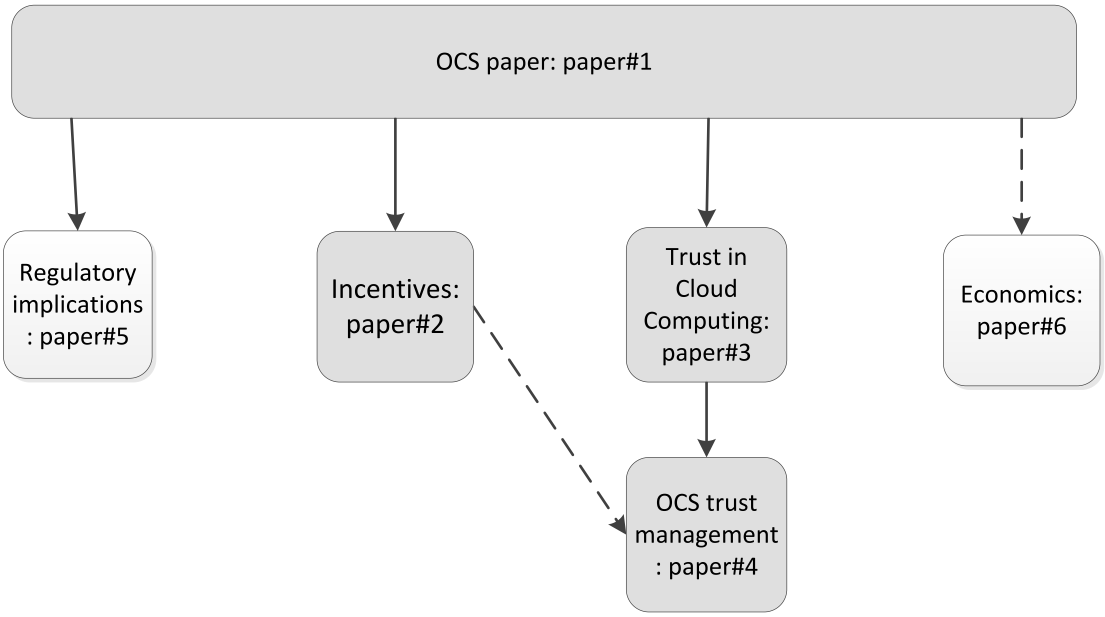

OCS paper: paper#1
Regulatory implications : paper#5
Incentives: paper#2
Trust in Cloud Computing: paper#3
Economics: paper#6
OCS trust management : paper#4

Figure 4-1: Relationship between the included papers.

Paper #1 laid the foundation for the other papers. It forms part of the core papers together with paper #2, paper #3 and paper #4. These are targeted at answering the main research question of

*"How can the platform for the opportunistic provisioning and utilization of cloud services by enterprises be designed and developed?"*

These papers therefore try to address the core objectives of this study. They are shaded grey as shown in Figure 4-1.

Paper #5 and paper #6 are targeted towards the other project objectives of the study. They are aimed at answering the much broader research question of

*"What are the sufficient enabling conditions for the provisioning and utilization of cloud services by enterprises?"*

## 4.1 Summary of the Papers

This subsection provides a summary of the six papers that are included in this dissertation. A summary of each of the six papers is given below. Each summary includes the title, the purpose and the essential points of the paper.

### 4.1.1 Paper on Opportunistic Cloud Services Concept: paper#1

The title of this paper is *"A Social Network Approach to Provisioning and Management of Cloud Computing Services for Enterprises"*. The purpose of this paper was to introduce the OCS concept, and present the reference architecture for the implementation of OCS networks. It also served to present some of the main research areas that needed to be addressed for the successful implementation of OCS networks. The paper proposed a social network approach to the provisioning and management of cloud computing services for enterprises in which members can take advantage of cloud services to meet their business needs without having to pay or paying a minimal fee for these services. An OCS platform is made up of enterprises and institutions collaborating strategically for the provisioning and usage of cloud services without entering into any business agreements.

According to paper #1 (Kuada and Olesen, 2011), some of the major challenges of cloud computing receiving research attention include legal and compliance risk management, migration of legacy enterprise applications into the Cloud, development of new applications to meet the demands of the cloud computing paradigm, meeting of SLA requirements by cloud service providers, management of cloud services, and security concerns. It further mentioned that the introduction of OCS brings new research issues and adds a complexity dimension to some of the existing ones. The paper went further to outline some of these research issues and the intuitive approaches of addressing them, and then stated that these issues have to be researched carefully for the successful implementation of OCS networks. The first among the outlined research issues is the sustainability of OCS networks; this demands the development of appropriate incentive mechanisms to ensure members are motivated to contribute resources to the platform. The second research issue that needs to be addressed is that of security and trust on the OCS platform. Another research issue mentioned by paper #1 is that of service differentiation and service quality management. The fourth issue requiring research as was outlined is the reliability of services under normal operation, and their resilience to recover from faults and malicious attacks. Other outlined research issues include regulatory implications for enterprises joining OCS networks; and finally provisioning and management of services on OCS platforms. The research areas that were identified in paper #1 are tackled in the proceeding papers.

### 4.1.2 Paper on Incentive Mechanisms: paper#2

The title of this paper is *"Incentive Mechanisms for Opportunistic Cloud Computing Services"*. This paper explored the design of incentive schemes that encourage the contribution of resources to the OCS platform as well as the efficient usage of such resources. Game theory was employed to model and design the incentive schemes with two game models presented. The existence of a pure strategy Nash equilibrium for both the cooperative and non-cooperative games was shown. Three base incentive schemes were also presented. These schemes are the Dominant Strategy Scheme (DSS), Equi-Profit Scheme (EPS) and Dominant Equi-Profit Scheme (DEPS). An analytical evaluation of the incentive schemes was performed with mathematical proofs, and it was concluded that the schemes met the desired properties of budget-balance, incentive compatibility, individual rationality, allocative efficiency, robustness, and flexibility to accommodate changing user behaviour on the OCS platform. Though these incentive schemes have been designed for the OCS platform, they can also be applicable to a general incentive and resource allocation problem in cloud computing in which the service contributors will be one or more commercial cloud service providers servicing a collection of clients with their spare capacities on which they put no fixed price tag (Kuada and Olesen, 2012).

### 4.1.3 Paper on Trust Engineering in Cloud Computing: paper#3

The title of this paper is *"Towards Trust Engineering for Opportunistic Cloud Services: A Systematic Review of Trust Engineering in Cloud Computing"* (Kuada, 2014). One of the major challenges that OCS platforms face is data security and trust management issues. In order to design and develop a trust management system for the OCS platform, I needed to look at the current trust engineering issues in cloud computing. I decided to perform a systematic literature review on this topic because since the OCS concept itself is new, any trust design models of its subsystems must be based on and guided by exhaustive knowledge of the state-of-the-art in the field. The rigorous methodological approach offered by systematic literature reviews will ensure that the topic is adequately covered. This paper therefore has reviewed identified primary studies on trust engineering approaches in cloud computing.

The central motivating objective of this work has been to lay the foundation for designing a trust management system for opportunistic cloud services platforms, and provide summary of trust engineering approaches in cloud computing for easy reference by the research community. It has been of specific interest to find the main approaches towards trust engineering in cloud computing; the objectives of the identified primary studies, and the contexts within which these trust management systems were being developed. Another specific objective for conducting this systematic literature review was to find the major trust models and trust management systems for cloud computing. Some of the main findings of the study are that employing trusted computing technologies and reputation based approaches are the two major means towards trust engineering in the cloud computing marketplace. Also, trusted third party approaches play a significant part in enhancing trust between cloud service providers and their consumers. Additionally, the adopted cloud deployment model plays a significant role in improving trust in cloud environments. It was also observed that the concept of trust is used loosely without any formal specification in cloud computing discussions and trust engineering in general. As a first step towards addressing this problem, I have contextualized the formal trust specification in multi-agent environments for cloud computing.

### 4.1.4 Paper on Trust Management Systems for OCS: paper#4

The title of this paper is *"Trust Management System for Opportunistic Cloud Services"*. As was indicated in paper #3, one of the major challenges that OCS platforms face is data security and trust management issues. This paper has therefore looked at the design of a trust management system for OCS platforms. It has applied knowledge and experiences from paper #3 to model trust for the OCS platform, designed a trust management system for OCS platforms, and verified the trust model and the trust management system through the simulation of the computation of the trust values with IaaS, and SaaS usage scenarios. Even though the trust management systems as presented in this paper contain the complete elements, it has focused mainly on the modelling of trust for the OCS platforms and the trust analysis components in the architecture (Kuada, 2013). The paper also addressed the issue of service differentiation and service quality management through the concept of Pseudo Service Level Agreement (Pseudo SLA) and demonstrated its applicability as implemented for OCS networks. The Pseudo SLA approach uses SLA templates that have been prepared for specific cloud services to ensure a uniform view of quality of service parameters by all members on the OCS platform. The Pseudo SLA concept also ensures that the services on the OCS platform adhere to certain minimum quality of service and hence these services will be suitable for business use.

### 4.1.5 Paper on Public Policy and Regulatory Implications for OCS: paper #5

The title of this paper is *"Public Policy and Regulatory Implications for the Implementation of Opportunistic Cloud Computing Services for Enterprises".* This paper discussed the impact that public policy and current regulations, together with corporate strategies towards cloud computing adoption, will have on the implementation of opportunistic cloud services. The paper examined the various strategic policies being adopted by North America, Europe, Asia Pacific and Africa. It was observed from the study that cost reduction and efficiency in the delivery of IT services is a central motivation in all the policies. On the other hand, meeting security and privacy requirements for the protection of both national and citizen data are major issues that the policies try to address. It was also found that the move by companies to the Cloud is a senior management level driven strategic technology shift for organizations as they look to lower costs and evolve their computing models to deliver competitive advantage to their business (Kuada et al., 2012).

The paper concluded that there are regulatory challenges on data protection that raise issues for cloud computing adoption in general; and the lack of a single globally accepted data protection standard poses some challenges for the successful implementation of OCS for companies. However, the direction of current public and corporate policies on cloud computing makes a good case for them to want to try out the usage of opportunistic cloud services.

### 4.1.6 Paper on Economics of Cloud Computing for SMEs: paper#6

The title of this paper is *"Cloud Computing and Information Technology Resource Cost Management for SMEs"*. This paper analysed the decision making problem confronting SMEs considering the adoption of public cloud services as an alternative to in-house computing services provision (Kuada et al., 2013). The economics of choosing between in-house computing and a cloud alternative is analysed by comparing the total economic costs of the two options given that the quality of service is identical across the options. The paper emphasized the importance of accounting for implicit costs in the cost analyses. One of the essential result of this paper is that the choice between in-house and cloud computing is not always clear-cut and cannot be solely based on numerical estimates. Several other factors that may be difficult to monetize may prove to be relatively more important in making such decisions. In most cases, however, this is the exception rather than the norm. Reliable estimates of the costs and benefits of adopting alternative computing service options can and should therefore be produced, and then utilized to guide such decisions.

The initial objective of this paper was to contextualize cloud computing for developing economies and highlight the role of opportunistic cloud services in such a business model. It was, however, necessary to gain a thorough knowledge of the decision making process of the adoption of cloud services by enterprises before the effect of opportunistic cloud services could be factored in. Additionally, the models for the cost of cloud adoption and the cost of in-house service provision that were obtained are applicable not only to enterprises in developing countries.

## 4.2 Fulfilment of Project Objectives

The extent to which the main project objectives, as has been outlined in Section 1.8, have been fulfilled is shown in Table 4-2. Most of the project objectives have been met through the included papers. The remainder of the project objectives that are not completely covered by the included papers are supplemented with discussion on these particular project objectives. It also gives remarks on the level of completion of these objectives.

**Table 4-2: Fulfilment of project objectives by the included papers**

| **Project Objective** | **Paper** | **Remarks** |
|---|---|---|
| Develop a conceptual framework and reference architecture for the implementation of opportunistic cloud computing services network for enterprises | Paper #1: A Social Network Approach to Provisioning and Management of Cloud Computing Services for Enterprises" | Satisfactorily completed |
| Develop incentives schemes that ensure sustainability of the network | Paper #2: "Incentive Mechanisms for Opportunistic Cloud Computing Services" | Satisfactorily completed |
| Develop a trust management systems that will ensure that resources and services on the OCS platform will be useful and remain reliable | Paper #3: "Towards Trust Engineering for Opportunistic Cloud Services: A Systematic Review of Trust Engineering in Cloud Computing" and<br>Paper #4: "Trust Management System for Opportunistic Cloud Services" | Satisfactorily completed |
| Develop a framework for managing cloud services in the context where enterprises are both services providers and consumers; and subsequently tools that will aid enterprises in adopting, monitoring and managing cloud computing services | Included in discussion section of the dissertation, and<br>Paper #4: "Trust Management System for Opportunistic Cloud Services" | Fairly completed |
| Evaluate current cloud computing management tools for their suitability for OCS | Included in discussion section of the dissertation | Satisfactorily completed |
| Contextualize cloud computing as a business model for developing economies, and highlighting the role of opportunistic cloud services in such a model | Paper #6: "Cloud Computing and Information Technology Resource Cost Management for SMEs" | Satisfactorily completed |
| Assess the feasibility of enterprises adopting opportunistic cloud computing services for their businesses | Paper #5: "Public Policy and Regulatory Implications for the Implementation of Opportunistic Cloud Computing Services for Enterprises" | Satisfactorily completed |

Most of the project objectives have been met through the included papers. The remainder of the project objectives that are not completely covered by the included papers are supplemented with discussion on these particular project objectives. It also gives remarks on the level of completion of these objectives.

## 4.3 Analysis of the Implementation of OCS Platform

The systems analyses, mathematical analyses, and qualitative data (in the form of primary studies) analyses that were involved in undertaking this research work are as are presented in the included papers. These analyses have mainly been on the design of incentive mechanisms and trust management system for OCS platforms. Since paper #1, which presented the OCS concept and the reference architecture, was the very first paper to be published over two years ago, there has been considerable insight into how the OCS management platform and associated tools can be developed, which is worth further explanation. This section therefore revisits the OCS reference architecture and how the various subsystems can be implemented.

According to Kuada and Olesen (2011), in light of the fact that the OCS concept is built on the principles of user-generated content, architecture of participation, and openness; a successful implementation of an OCS network will have to in the barest minimum satisfy the requirements outlined in Table 3.

**Table 3: Opportunistic Cloud Services Network Requirements Specification**

| Requirement | Requirement specification |
|---|---|
| R1 | It should provide support for the management of fundamental cloud services (SaaS, PaaS, IaaS) |
| R2 | It should provide support for the management of any arbitrary cloud service – anything as a service (XaaS) |
| R3 | It should provide interoperability with major cloud computing standards |
| R4 | It should provide interoperability with major cloud computing management tools |
| R5 | It should possess the flexibility to support future cloud computing technologies, management tools, cloud standards, etc. |

The above mentioned requirements constitute the requirements analysis and specification phase of the systems development methodology which has been adopted for this research work. The analysis of the requirements R1 and R2 reveals that the OCS platform must be able to support the easy contribution and utilization of the basic cloud services together with any arbitrary cloud service that members will want to contribute. This is because, without this capability, members will find it difficult to contribute resources to the platform, which will undermine the principle of participation on which the OCS concept based; and this will result in lack of participation, and eventually cause the collapse of the OCS platform. Furthermore, failing to provide support for the major cloud standards and major cloud management tools will result in forcing members to use tools that are solely meant for the OCS platform. Failure to provide support for requirements R3 and R4 will undermine the second principal principle (openness) on which the OCS concept is based. This will in turn force members to adopt new management tools before they can participate on the OCS platform. This obviously will hinder the adoption of OCS and the active participation of new members in contributing services or even utilizing services on the platform. Additionally, failure to anticipate that cloud computing technologies, management tools, and standards will be changing with time will be disastrous for the OCS platform because such changes can make the platform outdated and lead to members abandoning the platform. So requirement R5 is imperative for the success of the OCS platform.

To achieve these design objectives, the implementation of the OCS management platform must be based on open source cloud management tools and non-proprietary Cloud APIs. The implementation and development of the OCS platform and Cloud APIs are discussed next.

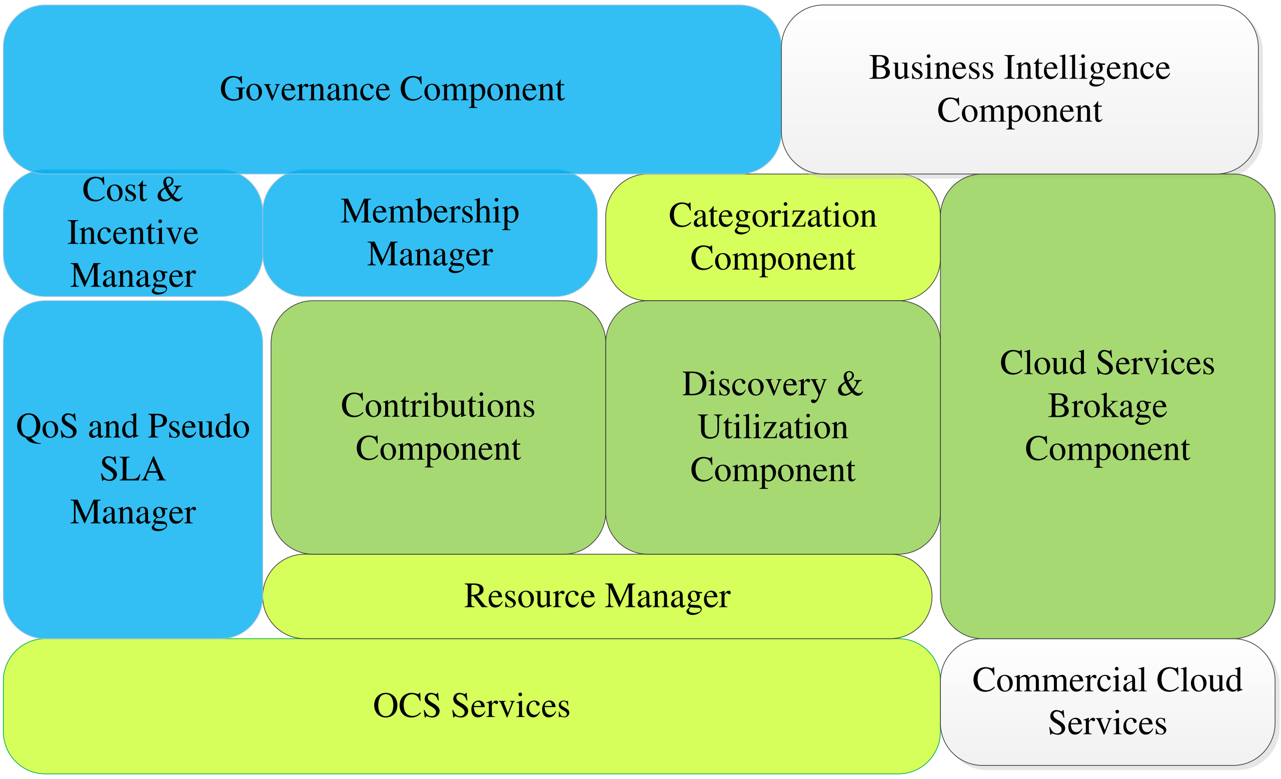
### 4.3.1 Development of the OCS Management Platform

The analysis of the functionalities required for achieving the features of the OCS platform as specified in the requirement specifications phase revealed that three main subsystems are needed. That is, the necessary functionalities of an OCS platform, as is illustrated in Figure 4-2, can be grouped into three main parts:

- The core OCS platform components
- Auxiliary OCS platform components
- Cloud management components

Details about each of these categories are elaborated upon in subsections 4.3.2, 4.3.3, and 4.3.4 respectively. The figure shows the main components that form the OCS reference architecture. The elements in the reference architecture are place based on how they complement each other in accomplishing their roles and which layer that they will logically fit, in the software development architecture, from the perspective of the users and developers of the OCS management platform.

The core OCS platform components are shaded light blue, the cloud management components are shaded light green, and the auxiliary OCS platform components are shaded with light yellow as show in Figure 4-2. These colours, however, do not have any special significance; but are only meant for maintaining visual consistency in all the figures that are concerned with explaining the role of the components in the reference architecture. The same colour scheme is therefore retained in Figure 4-3 and Figure 4-4 in order to maintain visual consistency of the categories in the three figures. It should be noted that the figure (Figure 4-2) has elements that are not shaded with any colour. This is because, from the perspective of a reference architecture, these elements do not form part of any of the three categories mentioned above. However, as will be seen later, from a software development perspective, these components can be placed under the appropriate categories and developed as part of the components in that category.



Governance Component
Business Intelligence Component
Cost & Incentive Manager
Membership Manager
Categorization Component
Cloud Services Brokage Component
QoS and Pseudo SLA Manager
Contributions Component
Discovery & Utilization Component
Resource Manager
OCS Services
Commercial Cloud Services

Figure 4-2: Opportunistic Cloud Services reference architecture[2].

Figure 4-3 shows a simplified depiction of the main parts of an OCS platform. This figure shows only two distinct parts instead of the three. This is because whereas the core OCS platform components (Governance Component, Trust Management Component, Cost & Incentive Manager, QoS and Pseudo SLA Manager, and Membership Manager) are exclusive to the OCS platform; and the cloud management modules (Contributions Component, Discovery & Utilization Component, and Cloud Service Brokerage Component) are also exclusive to the supported cloud management tools; the auxiliary OCS platform components (Categorization Component, Resource Manager,

[2] Based on OCS reference architecture of (Kuada and Olesen, 2011)

OCS Services) are logical functionalities that cut across the core OCS platform component and the cloud management components through the Cloud APIs as is depicted in Figure 4-4.

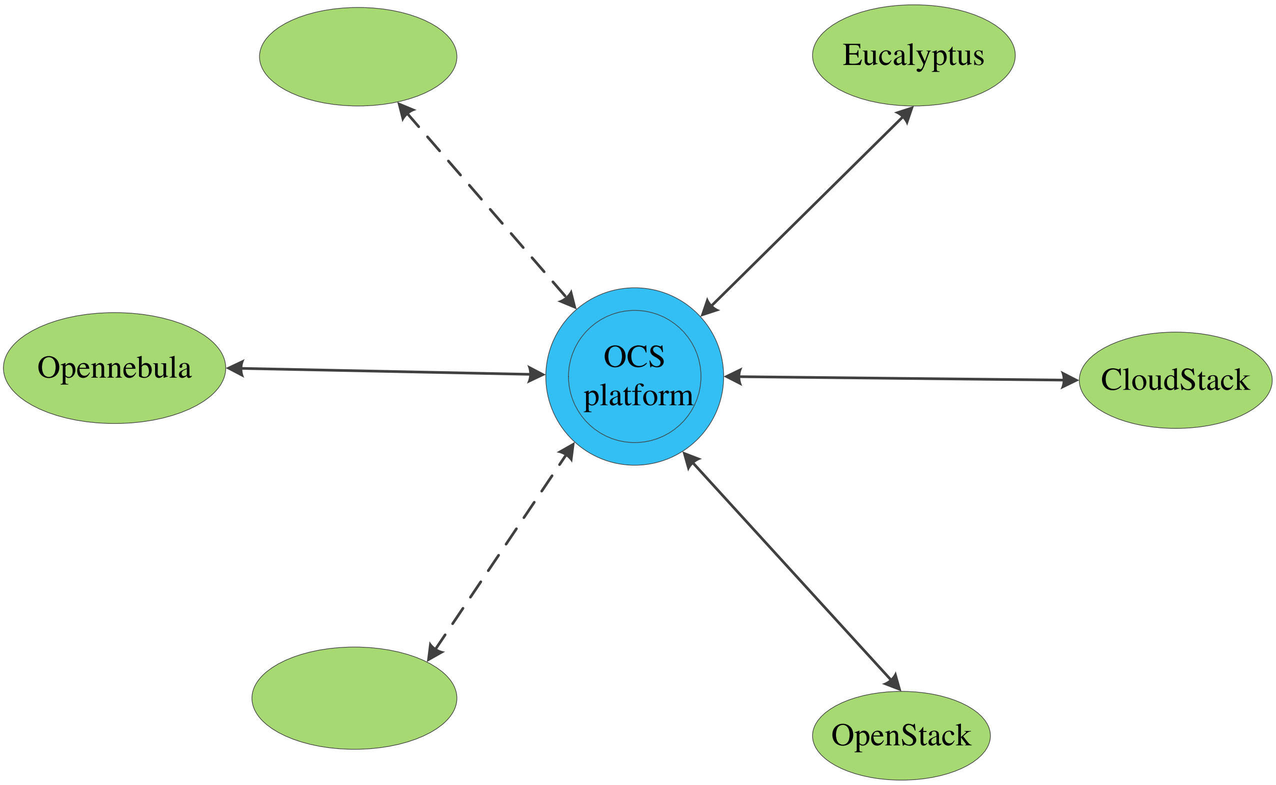


Figure 4-3: Current and future cloud management tools can be plugged onto the OCS platform.

The purpose of Figure 4-3 is to illustrate how current and future cloud management tools can be easily plugged onto the OCS platform.  The OCS components are developed as cloud management software modules, and use is made of Cloud APIs to allow information exchange between the OCS modules and the actual cloud management tools that will be handling the management of the cloud services on an OCS network. This approach allows for openness, supports interoperability with major cloud standards and cloud management tools; and the required flexibility for future cloud technologies is also met.  The support for the management of fundamental cloud services and any arbitrary cloud service is taken care of by the cloud management tools. So the figure in Figure 4-3 shows how current and future cloud management tools that will be developed can be plugged into the OCS cloud management platform. That is, the modules that form the core OCS platform components, together with the auxiliary OCS platform components constitute the main OCS management platform. This then communicates with the cloud management tools through Cloud APIs. This means that the cloud management tools are linked to the OCS management platform through Cloud APIs as is depicted in Figure 4-4.

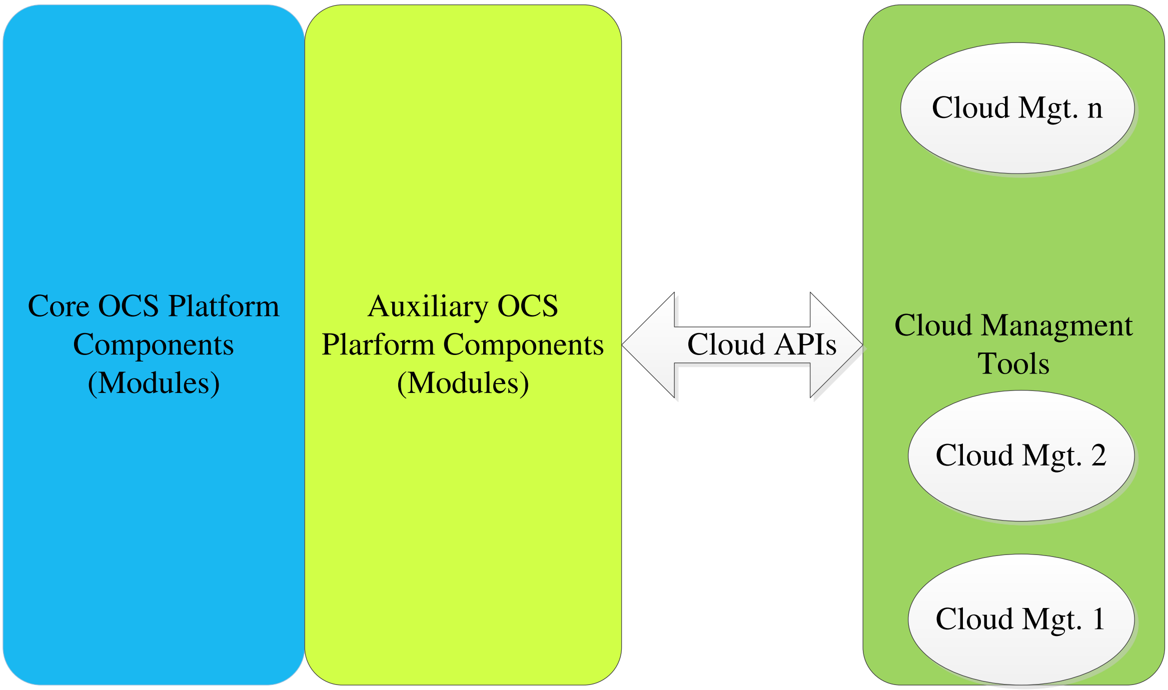

Core OCS Platform Components (Modules)
Auxiliary OCS Plarform Components (Modules)
Cloud APIs
Cloud Mgt. n
Cloud Managment Tools
Cloud Mgt. 2
Cloud Mgt. 1

Figure 4-4: The Cloud management tools are linked to OCS platform through Cloud APIs.

### 4.3.2 Core Components of the OCS Platform

The core components of the OCS management platform are those functionalities that are specific to the OCS management software around which the other modules revolve. These include the trust management component, cost & incentive manager, QoS and pseudo SLA manager, the membership manager, and finally the governance component. Other modules that do not necessarily form part of the OCS platform core elements because they are not needed for its operation can also be identified and implemented as part of the OCS platform when such functionalities are found to be desirable to have on the platform. For example the initial OCS reference architecture of (Kuada and Olesen, 2011) did not include the trust management module but when work on its design was completed, analysis of its functionality shows that it fits with the core OCS platform components.

#### *4.3.2.1 Trust Management Component*

This module makes the analysis of the trust values to be computed from information on the expectation of the potential user of a service, and then computes the necessary trust values based on the available data on the reputation of the service and the service provider. Details of the trust value computation process can be found in paper#4 (Kuada, 2013).

#### 4.3.2.2 Cost & Incentive Manager

This is responsible for the implementations of the incentive schemes (Kuada and Olesen, 2012) on to OCS platform. It is therefore responsible for the dynamic re-computation of cost in real time to be credited to service contributors and debited to resource users. Details about the OCS credit points cost computation and the various types of incentive schemes can be found in paper#2 (Kuada and Olesen, 2012).

#### 4.3.2.3 QoS and Pseudo SLA Manager

It uses information from the Cost Manager and the trust management module to provide service differentiation to members. It is also responsible for the management of the pseudo SLA creation, storage and administration process. Details about the pseudo SLA implementation and management process can be found in paper#4 (Kuada, 2013).

#### 4.3.2.4 Membership Manager

This is the main social network user management module for the OCS platform. It is responsible for managing existing users and the registration of new members (institutions, enterprises, organisations, etc.). It handles membership requests and in cooperation with the Governance Component, performs company profile verification based on data provided by enterprises during registration to make decisions on membership approval or rejection.

#### 4.3.2.5 Governance Component

This is the logical module that provides supervision for all the other components in the OCS management platform. It is implemented as the interfaces through which platform administrators interact with the platform to make governance decision. The creation of pseudo SLA templates and the enforcement of QoS guarantees on the OCS platform through the detection of SLA violation, and malicious behaviours, and then taking appropriate remedial actions are some of the examples of governance actions that can be taken on the OCS platform.

#### 4.3.2.6 Business Intelligence Component

In addition to the components that have been described above, other functionalities or components that do not necessarily form part of the OCS platform, but are targeted towards specific functions that may be desirable can be implemented as part of the core OCS platform components. A typical example of such a component could be the business intelligence component. It should be noted that this component is not shaded in Figure 4-2 as has already been explained above in Section 4.3.1. Some of the features of this component can include data mining based on activities of the OCS platform. For example, services that are being requested for but are not currently provided, or even supported on the OCS platform can serve as a good indication for the platform administrators to guide members to providing such services. This information can also be very useful for making other decisions.

### 4.3.3 Auxiliary OCS Platform Components

The auxiliary OCS platform modules are logical functionalities that cut across the functionalities of the core OCS platform modules and the cloud management modules through the Cloud APIs. This means that the actual functionality resides with the cloud management modules but information will be queried from them to create these auxiliary OCS modules. These include the categorization component, resource manager, and OCS Services.

#### *4.3.3.1 Categorization Component*

This component is needed to ensure that the OCS platform supports arbitrary services while also ensuring easy management of these services. It is responsible for the categories creation process. It handles service category creation requests from members; these requests are then evaluated for approval by the OCS platform administrators. The OCS platform administrators can then create pseudo SLA templates for the service categories. If this categorization component is not implemented but rather members are allowed to simply create new services without it belonging to a service category, the management of the services will be cumbersome and the pseudo SLA implementation can also fail.

#### *4.3.3.2 OCS Services*

These consist of all the services and resources that have been contributed by members to the OCS platform. These could be coming from the contributing member’s data centre, private cloud, etc. Services are mainly fundamental cloud services such as SaaS, PaaS, and IaaS; but can also be any other arbitrary cloud services (XaaS) and resources. This module is placed under the auxiliary OCS modules because the contribution of the resources to the services on the OCS platform will actually be done through the cloud management tools. The information about these services are, however, queried from whatever cloud management tool the service contributor is using and the information is presented in a suitable format to other members on the OCS platform. This presentation of the services in a suitable format will be done by the Resource Manager.

#### *4.3.3.3 Resource Manager*

This module, together with the cloud deployment and management tools found in the *Contributions Component* and the *Discovery & Utilization Component*, abstract the contributed services from the OCS services layer and interface it to the OCS management platform. This is, when resources are contributed to the OCS platform, it is possible that it can be used by other members in the state in which it was provided, otherwise if any further processing needs to be done before it is suitable for use by other members, then the Resource Manage will handle this task. It is also responsible for managing the usage of the services of the OCS platform.

### 4.3.4 Cloud Management Components

These modules are functionalities that are exclusive to the supported cloud management tools. These modules include contributions component, discovery & utilization component, and the cloud services brokerage component. More details about these modules are presented below.

#### *4.3.4.1 Contributions Component*

This component is responsible for handling the resource contribution process. Its main objective is to simplify and make it easy for members to contribute resources to the OCS platform. It performs two sub functions - providing cloud computing management tools and service life cycle management. It consists of cloud computing deployment and management tools for all types of services and resources. The service management involves service creation, service certification and service profiling which includes service review and ranking by users, and service ranking by platform administrators. These features associated with the service life cycle management will however be done in conjunction with the *trust management module*.

#### *4.3.4.2 Discovery & Utilization Component*

This component's role is to simplify the services and resources discovery and utilization process. It performs service recommendation by taking service requirements description by members and matching these with service properties description by contributors. These service recommendation features will be done in conjunction with the QoS and Pseudo SLA management module. It also consists of cloud computing management tools for services and resources provisioning and utilization.

#### *4.3.4.3 Cloud Service Brokerage Component*

This component consists of cloud management tools and processes that interface the OCS network to commercial cloud services and provide cloud brokerage services to members. This component is necessary because the resources on the OCS platform may not always meet all the needs of the members; so they may need to use commercial cloud services. For example, Enterprises A and Z in the co-existence scenario in Figure 1-1 in Section 1.3, make use of commercial public cloud services to serve some of their business processes while still making use of services on the OCS platform to meet some of their other needs. It is the responsibility of this module to ensure that the processes involved in members combining services on the OCS platform with commercial cloud services, or switching between the utilization of a particular service on the OCS platform with a commercial version is handled smoothly.

### 4.3.5 OCS Reference Architectural Elements Usage Scenario

From the co-existence scenario in Section 1.3, before any of the Enterprises A, B, …, and Z, or for that matter, any other entity gets accepted as a member on the OCS platform, it needs to submit information about itself in request to be a member on the OCS platform. It provides information such as the name, address, appropriate registration number, etc., that may be required. This process will be handled by the *Membership Manager*. The approval or rejection of this request is handled by the *Governance Component,* where OCS platform administrators verify the submitted information and make a decision. Once an entity becomes a member, its continuous existence and activities are handled by the *Membership Manager*.

When a member, e.g. Enterprise A, with some spare IT resources decides to contribute it to the OCS platform, this process will be handled by the *Contributions Component*. This resource or service becomes part of the pool of *OCS Services* which other members on the platform could use. The *Resource Manager* will then perform any processing that might be needed before a contributed resource or service becomes usable on the OCS platform. For example, if resources RsA and RsB from the co-existence scenario in Section 1.3 are pure virtual disks. These resources may need to be processed into Storage as a Service before it can be usable on the OCS platform. The *Discovery & Utilization Component* will then handle the process for members finding services on the OCS platform and utilizing such services. The *Resource Manager* will also monitor and manage how resources on the OCS platform are being used.

The *Cost & Incentive Manager* performs computation of the OCS credit points to be awarded to the contributors of a service and debited to users of that service at any point in time based on the demand for that service. For example, when Enterprise Z decides to use resource RsA, the cost involved at the time will be computed by the *Cost & Incentive Manager.* When Enterprise Z decides to switch from using RsA to a commercial cloud service, for example because RsA is no longer available on the OCS platform, or it wants to combine it with some commercial public cloud service to meet a specific business process needs, the *Cloud Service Brokerage Component* handles the aiding of members in such situations.

The *QoS & Pseudo SLA Manager* ensures that the services on the OCS platform meet some minimal QoS for them to be reliable enough for the members on the platform to use. When a member wants to contribute a resource or service that does not fall under any of the fundamental cloud services (IaaS, PaaS, or SaaS) or any existing service category on the OCS platform, it submits a request to create a new service category on the platform. This process is handled by the *Categorization Component*. For example, if a member wants to provide data security services on the OCS platform, it may request a service category creation through the *Categorization Component* for the creation of service category (e.g. Security as a Service). This request will then be appraised and a decision made by the OCS platform administrators through the *Governance Component*.

## 4.4 Discussions

This discussion section covers the discussion of the main findings during the time span of this research project and their implications for the OCS concept. It also covers discussions about the importance of these main findings on the implementation of the OCS management platform. The section also discusses the possible impact of the OCS concept on cloud computing in general. Finally, some limitations with the study are discussed together with some further work to be done.

### 4.4.1 Main Findings and Their Implications for OCS

The main findings are categorised into two groups: those that form part of the published papers that have been included in this dissertation, and those that are major findings during the project but have not yet been published.

#### *4.4.1.1 Main findings in the included papers*

The most essential parts of the results that have already been published in the included papers are as follows. The first one is the detailed reference architecture for the implementation of OCS networks that has been developed. As was presented in paper#1 and its revised version presented in section 4.3 above, this reference architecture forms the basis for the necessary functionalities required for the implementation of OCS networks. This has shown the feasibility of the design of subsystems that will allow for the development of the OCS management platform. With this reference architecture, anyone who wants to develop an OCS management platform will simply have to ensure that the functionalities of the components that constitute the core OCS platform modules are developed. In following with the requirement specifications of the OCS platform, the core OCS modules can be developed to be able to integrate with any open source cloud management tools that will provide the necessary functionalities required by the cloud management modules as have been described in subsection 4.3.4 above. With these two subsystems of the OCS management platform developed, the auxiliary OCS modules can now be implemented by querying the cloud management tools to develop the necessary auxiliary OCS modules as have been presented in subsection 4.3.3.

The creation of this detailed reference architecture is the first step towards answering the main research question of

*"How can the platform for the opportunistic provisioning and utilization of cloud services by enterprises be designed and developed?"*

In line with the research agenda of this study, the second thing was to explore the possibility of the design of incentive schemes for OCS networks. This has been achieved with the design of incentive mechanisms that meet the desired properties of budget-balance, incentive compatibility, ex-post individual rationality, allocative efficiency, robustness, and flexibility to accommodate changing user behaviour on the OCS platform as has been discussed in paper#2 (Kuada and Olesen, 2012). The successful design of these incentive schemes also demonstrates the support for the feasibility of the development of OCS for enterprises. Failure to demonstrate the existence of suitable incentive schemes that meet the desired properties mentioned above would have undermined the feasibility of the implementation of OCS for enterprises because there wouldn't have been any justifiable motivation for enterprises to join the OCS platform, and yet alone contribute services to the platform.

Thirdly, a model of the concept of trust for the OCS platform has been developed. In following with addressing the research agenda specified in the project objectives in Section 1.8, and by extension answering the formulation of the main research question specific to the main contribution of this thesis, there was a need to develop a trust management system for the OCS platform to ensure that the resources/services on the OCS platform are useful and remain reliable. This however became a challenge because it became evident from the systematic literature review that I performed on trust engineering in cloud computing that the concept of trust was used loosely without any formal specification in the discussion of trust engineering in cloud computing or trust engineering in general. To develop effective trust management systems, however, required a formal specification for the concept of trust. So the first thing I did was to adapt the model of trust for multi-agent environments, reformulated it for cloud computing environments and then applied it in the context of the OCS platform. After this was done, it was then easier for me to design a trust management system for OCS platforms as is presented in paper#4 (Kuada, 2013). The trust model and the trust management system have been verified through the simulation of the computation of the trust values with IaaS and SaaS usage examples. The issue of service differentiation and service quality management has been addressed through the incentive mechanisms and the concept of Pseudo Service Level Agreement (Pseudo SLA) and its applicability to OCS networks.

The work on the systematic literature review on trust engineering in cloud computing, and the work on trust management system for opportunistic cloud services sought to answer the research question of *"How can a trust management system for opportunistic cloud services for enterprises be designed and developed?";* which is a reformulation of the main research question of *"How can the platform for the opportunistic provisioning and utilization of cloud services by enterprises be designed and developed?"* in order to highlight the main contribution of this thesis.

The results and discussions above support the case of the feasibility of implementing OCS platform for enterprises. Additionally, these discussions and results also provide answers to the main research question of *"How can the platform for the opportunistic provisioning and utilization of cloud services by enterprises be designed and developed?"* This happens to be the case because it can be observed from the project objectives in Section 1.8 that some of them have specifically been designed to explore answering the main research question while the remaining project objectives are targeted towards the broader research question of *"What are the sufficient enabling conditions for the provisioning and utilization of opportunistic cloud services by enterprises?"*. It should also be noted that both the research questions and the project objectives have been derived from the main project goal of laying the foundation for the development of OCS management platforms to engender the implementation of OCS networks.

The purpose of the work on paper#5 (Kuada et al., 2012) was to explore if there were any existing national or international laws or regulations that will prevent enterprises from joining the OCS platform whether to contribute services or make use of services on the OCS platform. The results from that study were that there are regulatory challenges on data protection that raise issues for cloud computing adoption in general. There were, however, no international laws or regulations that prohibit enterprises or organisations from using free cloud services. The existence of such a law or regulation would have been very detrimental to making a case for the implementation of the OCS concept irrespective of the fact that its technical feasibility has been demonstrated. It was good to find out that such laws or regulations were non-existent; the study, however, found that there is a lack of a single globally accepted data protection standard. This poses some challenges for the very successful implementation of OCS for companies. This is because, since providers of the contributed services and the potential users of their services may be under different jurisdictions, this coupled with the fact that no formal business agreement exist between the service providers and service consumers, getting the service providers to adhere to data protection laws of the consumers of their services will still be a major challenge irrespective of the fact that I have developed a pseudo SLA management process to deal with some of these challenges.

Another result from paper#5 (Kuada et al., 2012) on the study on public policy and regulatory implications for the implementation of  OCS is that the direction of current public and corporate policies on cloud computing make a good case for them to try out opportunistic cloud services. Even though it was evident from this study that various nations were very wary of putting citizen data and other national data in public clouds, they were still pushing for the use of cloud computing by government agencies mainly because of the potential cost savings that cloud computing offers. They were therefore building national private clouds for such purposes. They were also making deliberate efforts of encouraging enterprises to adopt cloud services. Thus, even though services on the OCS platform may not be directly attractive to government agencies' direct usage because it is a much open platform than even public clouds, their attempts at promoting the adoption of cloud services by enterprises and in some cases investing national funds on such endeavours will make the use of services on the OCS platform attractive to them since they will not have to directly invest in such a platform.

Three main factors, among others, can be identified in answering the broader research question of *"What are the sufficient enabling conditions for the provisioning and utilization of opportunistic cloud services by enterprises?"*

The first is the technical feasibility; the second is non-existence of laws and regulation that will prohibit its deployment; and finally, it must make economic sense to the users of the OCS platform. The first and the second conditions have been satisfied as discussed above. The question now is: will it make economic sense to users of the OCS platform? What about the risks that enterprises will face in using the services on the OCS platform? Will the risks outweigh the benefits or vice versa? In paper#2, the design of the incentive schemes for the OCS platform shows all the beautiful results of budget balance, incentive compatibility, individual rationality, allocative efficiency, robustness, and flexibility to accommodate changing user behaviour on the OCS platform. It must be noted, however, that this has been done in the context of OCS credit points and not monetary value. It can be argued that monetary or the economic value have been factored into the OCS credit points because the members will report their anticipated benefits in OCS credit points based on the actual economic benefits they expect to get; but the subjective nature of this and exactly how monetary value translates into OCS credit points for each member on the OCS platform is still not clear. This means that even though the results of the design of the incentive schemes is excellent, it doesn't totally guarantee that it will make full economic sense to the users.

The question of the risks that enterprises face in using services on OCS platform and whether such risks outweigh the benefits or vice versa can be simplified to a comparison of OCS services to that of commercial public clouds. This is because the same risks and benefits exist with using public cloud services. The only difference is that the OCS platform is a more open environment, and even though I have designed a trust management system to make resources and services on the platform reliable, the effectiveness of this still needs to be verified in the real world before it can be said that the reliability of services on the OCS platform is comparable that of commercial public cloud services, and hence they present the same risks to the users.

#### *4.4.1.2 Cloud APIs*

One of the major findings during this project's time span that has not yet been included in any of my papers that form part of this dissertation is the emergence of major Cloud APIs. A Cloud API is a set of application development libraries, popularly known as Applications Programming Interface (API), that are specifically targeted towards cloud management tools. This offers cloud service consumers the opportunity to extend the features of their favourite cloud management tools by querying for more specific information from the cloud service providers' back-end systems. The maturity of open source cloud management tools, and the progress in interoperability and standardization efforts as were presented in Section 2.1.5, has led to the emergence of dominant Cloud APIs. The emergence of these major Cloud APIs is very important for the development of the OCS platform because without this, it will be impractical to develop the OCS platform with the needed interoperable cloud management tools as is required by the requirement specifications for the successful implementation of OCS platforms.

The section takes the discussion further with Cloud Abstraction APIs, which will further simplify the development of the OCS platform. This is because even though it will be possible to develop the OCS platform with the major Cloud APIs, programming against multiple Cloud environments with the need to support multiple Cloud APIs at the same time can be a very challenging endeavour. Cloud abstraction APIs help to simplify developers' need to support multiple cloud APIs.

The application developers' need to support programming against multiple Cloud APIs has brought about projects that aim at solving this problem by providing abstraction to the major cloud APIs. *Deltacloud API*[3] and *jClouds API*[4] are the only two of such projects currently in the public domain.

Deltacloud API is an open source Apache project. It enables management of resources in different Clouds by the use of one of three supported APIs. Its supported APIs are the Deltacloud classic API, the Cloud Infrastructure Management Interface (CIMI) API, and the EC2 API (Deltacloud, 2011). This means that a cloud service user can start an instance on an internal private Cloud, and start another on EC2 or Red Hat Enterprise Virtualization Manager (RHEV-M) with the same code. The Apache Deltacloud is a REST-based cloud abstraction API that enables management of resources in different IaaS clouds by using a single API. This is possible because there are back-end drivers communicating with each supported cloud vendor's native API; and the Deltacloud Core Framework provides the basis for implementing drivers to new IaaS clouds (Deltacloud, 2011). Some of the currently supported drivers are Amazon EC2, Opennebula, OpenStack, Eucalyptus, Rackspace, VMware vSphere, IBM, GoGrid, etc. (Deltacloud, 2011).

The jClouds project is also an open source library that provides support for about thirty cloud services vendors and cloud software stacks including OpenStack, Amazon, GoGrid, Microsoft Azure, Ninefold, and Rackspace. It offers several API abstractions as Java and Clojure libraries (jClouds, 2011).

In comparing these two Cloud abstraction APIs that are currently in the public domain, my impression is that though both are still at their nascent stages, Deltacloud is a more matured project considering the fact that it supports multiple programming languages and development frameworks. It also has the potential becoming a fully matured project because it is an Apache Software Foundation[5] project. Since jClouds is JAVA specific but Deltacloud supports different programing languages such as Ruby, C/C++, etc., it is advisable to use Deltacloud to simplify supporting

[3] (Deltacloud, 2011)

[4] (jClouds, 2011)

[5] (Apache Software Foundation, 2013)

multiple cloud APIs in developing OCS management platforms. It is, however, imperative that the OCS platform development nonetheless supports implementing with Cloud APIs directly so that it will still have the power of access to individual Cloud APIs for functionalities that are not immediately supported by Deltacloud.

### 4.4.2 Possible Impact of OCS on Cloud Computing

One of the factors that contribute to cloud service consumers' distrust of cloud service providers and the services they provide is that even though cloud service providers provide similar services, the service level agreements are very disparate from one service provider to the other. This creates complications for cloud service consumers in understanding what the parameters in an SLA mean. This kind of confusion contributes to the lack of trust cloud service consumers have in cloud service providers and their services. OCS deals with disparities in SLAs of services that offer similar features. The design and implementation processes of a pseudo SLA with its associated SLA templates, as is discussed in paper #4 (Kuada, 2013), can be applicable to commercial cloud services. The use of SLA templates by even commercial cloud service providers is likely going to be the norm in the near future, where cloud services offering similar features will have a standardized SLA that is independent of the cloud service provider. This will therefore reduce the confusion that cloud service consumers are faced with and thereby improve their trust in cloud services and the providers of these services.

Another possible impact of the OCS concept on cloud computing is that the OCS platform can act as a trusted third party for even commercial cloud services providers so that the OCS trust management system could be applied to commercial cloud services. In this way, commercial cloud services could be registered to a platform similar to that of the OCS platform so that the processes involved in the pseudo SLA implementation and management on the OCS platform can be applied to the SLAs between the cloud service providers and their consumers. The other processes involved in the trust management on the OCS platform are also applicable; this is, the processes involved in the expectation management, trust value computation, data monitoring, and data management could all be applied on such a platform. The decision support process of the OCS trust management system can also be applied to such a platform to assist cloud services consumers in making decisions on whether to use a specific service from a particular cloud service provider or in choosing among multiple cloud services or cloud service providers.

The third possible impact that the OCS concept can have on cloud computing is that, the OCS platform can cause a change in the paradigm of how cloud management tools are designed and developed. The way cloud management tools are currently designed and developed is to have distinct roles for the cloud service provider and the cloud service consumer. Even though some cloud management tools do give some level of delegated authority to the users, the role of the provider of the services and that of the users of the services are separate. When the concept of OCS and its principles becomes part of the norms for cloud computing, the cloud management tools developers will begin to design their tools with the view that the same entity can be a consumer of some services and a provider of other services with the same cloud management tool.

The incentive schemes that have been designed for the OCS platform can also be applicable to a general incentive and resource allocation problem in cloud computing, in which, the service contributors will be one or more commercial cloud service providers servicing a collection of clients with their spare capacities on which they put no fixed price tag. That is, because commercial cloud service provides need to have theoretically infinite capacity, they can choose to place some of their current resources that haven't been allocated under their normal price plan on a platform similar to OCS to trade. The incentive schemes that have been developed for the OCS platform can be applied to this environment with the only difference being that resources get traded in normal currency instead of OCS credit points. As is in the case of the OCS platform, the prices of resources will vary dynamically with demand on the platform. The cloud service providers will be at liberty to withdraw these resources and place them back on their normal price plan as and when it suits them best.

### 4.4.3 Limitations of the Study and Further Work

Much attention has not been given to the cloud brokerage component in the OCS reference architecture which is required for ensuring the coexistence of the OCS platform with commercial cloud services. The processes necessary for facilitating OCS members' dynamic switch between services on the OCS platform with substitutable services of commercial cloud service providers are not yet developed. Even those processes required for aiding members to combine resources on the OCS platform with resources of commercial cloud service providers are not yet developed. Developing the details for this component will, however, require some amount of work which could pass for writing a whole new conference or journal article; so this is best left as a future work for now.

The project objective on contextualizing cloud computing as a business model for developing nations in the context of opportunistic cloud services has not been completed. The paper on "Cloud Computing and Information Technology Resource Cost Management for SMEs" (Kuada et al.,

2013) that was targeted towards this objective could not factor in the role of opportunistic cloud services due to practical reasons. This paper ended up with models for cost of cloud adoption and in-house IT services provisioning. These models were, however, not specific to developing countries and could therefore be applicable to SMEs in any country. In any case, this part of the work needed to be done to gain a thorough knowledge on the decision making process of cloud service adoption confronting enterprises before the role of opportunistic cloud services could be fully factored into these models. Therefore, work on actual business models for cloud computing in developing nations and the role opportunistic cloud services will play in such business models is left as a future studies.

As was indicated in the Section 1.9 on Research focus: Delimiting the Research, this study could have taken three possible routes: a user-centred focus, an economics centred focus, or a system engineering centred focus. I have justified the reasons for adopting a system engineering focus; but this addresses mainly the technical feasibility aspects of the OCS concept. Much of the work in the included papers has been targeted towards addressing this technical feasibility of the OCS concept. Even though paper#5 and paper#6 have been targeted towards more general objectives of answering the broader research question of what the sufficient enabling conditions for the opportunistic provisioning and utilization of cloud services by enterprises are, they don't tackle what the perspective of the potential users of the OCS platform is. Since the actual success of the OCS platform will not only be based on its technical feasibility, it will be useful to explore some of the activities that were stated in the user-centred focus in subsection 1.9.1. The results from that study should help present a much holistic view of the success outlook of the OCS concept.

The actual research phase of the project is now in its final stages. The project is currently at the implementation of OCS management platform phase. Cloud abstraction APIs that will help to simplify the development are, however, still at the nascent stages and will need to reach a level maturity. Development and maturity of these cloud abstraction APIs will therefore need to be monitored. The project can, however, begin programming directly against cloud APIs for creating a proof of concept of the OCS management platform.

# 5 Conclusion

The feasibility of the implementation of opportunistic cloud services for enterprises has been the subject of this PhD study. I have been interested in finding the sufficient enabling conditions for the contribution and utilization of spare IT resources that have been packaged as cloud services by enterprises. It was almost immediately noticed from the early stages of the study that, among other factors, there is a need for a platform that enables the contribution and utilization of such spare IT resources. The other identified enabling factors are that there should not be legal or regulatory prohibitions for enterprises and other organisations to either contribute their resources or utilize the resources that have been donated by others. Thirdly, there need to be economic and/or other forms of benefits for enterprises in participating in both the contribution of resources and the utilization of these resources. The objective of the study was therefore to explore these factors and with the aim of laying the foundation for the implementation of opportunistic cloud services platforms for enterprises. Since the existence of a platform that will enable such open contribution and usage of cloud services was identified as key among the others, the initial much broader research question of *"What are the sufficient enabling conditions for the provisioning and utilization of opportunistic cloud services by enterprises?",* had to be carefully rethought through in order to delimit the boundaries of the study appropriately. This process led to a sub research question from the one above, and was targeted towards the design and implementation of opportunistic cloud services. This new research question which rather became the main one is *"How can the platform for the opportunistic provisioning and utilization of cloud services by enterprises be designed and developed?"*

A sub research question of: *"How can a trust management system for opportunistic cloud services for enterprises be designed and developed?"* was further derived from the main research question in order to put more focus on the expected main contribution of trust management in opportunistic cloud services and cloud computing in general as central to this research work.

The attempt to answer these questions resulted into setting up project objectives aimed at exploring to what extent the arguments supporting or weakening these factors could be found. This together with the fact that the main goal of the study is to lay the foundation for the development of OCS management platforms that will engender the implementation of OCS networks, led to the creation of the project objectives as stated in Section 1.8. The majority of these project objectives were therefore targeted towards the technical feasibility of the implementation of OCS management platforms. The other project objectives have been targeted towards the much broader research question of finding the sufficient enabling conditions for the implementation of OCS.

The six papers that are included in this dissertation have been targeted towards these objectives. Table 4-2 shows the list of the included papers with their contribution towards fulfilling the stated project objectives. The summary of the main results of the study, as has been presented in the included papers and this dissertation are: First and foremost, detailed reference architecture for OCS platforms has been developed. This reference architecture contains the details of the functionalities that are required in developing a management tool for OCS platforms. Secondly, incentive mechanisms have been designed for the OCS platform. These incentive schemes should ensure that the enterprises and other organisations are encouraged to join the OCS platform both to contribute resources and also utilize resources that have be donated by others. The third major result is that a model for the concept of trust for cloud computing environments has been created; this model has been applied to the OCS context to develop a trust management system for the OCS platform. This trust management system will ensure that the contributed resources on the OCS platform are useful to the members and will remain reliable. A major sub-system of the trust management system is the pseudo SLA system for managing expectation of service users and quality of service guarantees that service providers promise to offer. The above mentioned results, together with the presented discussions on the cloud APIs and the cloud abstraction APIs which should simplify the implementation of the required functionalities of the OCS management platform to communicate with open source cloud management tools, demonstrate the technical feasibility of the implementation of the OCS platform. These therefore provide a satisfactory answer to the main research question of how to design and develop a platform for the opportunistic contribution and utilization of cloud services for enterprises.

The included paper, which explored the implications of current public policies and regulations for the implementation of opportunistic cloud services, revealed that there are no regulations or policies prohibiting the implementation of opportunistic cloud services. The lack of a single globally accepted data protection law will however pose some challenges for the OCS platform since the resource providers, and the potential utilizers of their resources and their clients can potentially all be in different data protection jurisdictions.

As has been discussed in subsection 4.4.1.1, even though I have successfully designed incentive mechanisms for the OCS platform, I can’t answer fully if there are enough monetary or other forms of benefits for enterprises to join the OCS platform to contribute or utilize resources. I must also admit that contrary to the expectation of the project objective aimed at contextualizing cloud computing as a business model for developing economies with the hope of highlighting the role of opportunistic cloud services in such business models, has not fully been completed because the anticipated business models have not been developed yet. Upon thorough reflections, the development of such business models is now considered to be outside the boundaries of this PhD dissertation. These limitations notwithstanding, the presented results demonstrate both the technical feasibility and the existence of enabling conditions for the implementation of opportunistic cloud services for enterprises. I can therefore make a case that, there are good reasons for the implementation of opportunistic cloud services.

## References

Aker, J., 2008. Does Digital Divide or Provide? The Impact of Cell Phones on Grain Markets in Niger (SSRN Scholarly Paper No. ID 1093374). Social Science Research Network, Rochester, NY.

Aker, J., Mbiti, I., 2010. Mobile Phones and Economic Development in Africa (SSRN Scholarly Paper No. ID 1693963). Social Science Research Network, Rochester, NY.

Amazon Web Services, 2013. Amazon Elastic Compute Cloud User Guide: API Version 2013-02-01.

Apache Software Foundation, 2013. Welcome to The Apache Software Foundation! [WWW Document]. The Apache Software Foundation. URL http://www.apache.org/ (accessed 4.24.13).

Bailey, K., 2008. Methods of Social Research, 4th Edition. Simon and Schuster.

Bapna, R., Das, S., Garfinkel, R., Stallaert, J., 2008. A market design for grid computing. INFORMS Journal on Computing 20, 100–111.

Beimborn, D., Miletzki, T., Wenzel, S., 2011. Platform as a Service (PaaS). Business & Information Systems Engineering 3, 381–384.

Belytschko, T., Fish, J., Hughes, T.J.R., Oden, T., 2004. Simulation Based Engineering Science. National Science Foundation.

Blake, S.P., 1978. Managing For Responsive Research And Development. W.H.Freeman & Company Limited.

Box, 2013. Box for Business: What Can Box Do For Your Business? | Box [WWW Document]. Box. URL https://www.box.com/business/ (accessed 4.7.13).

Brereton, P., Kitchenham, B.A., Budgen, D., Turner, M., Khalil, M., 2007. Lessons from applying the systematic literature review process within the software engineering domain. Journal of Systems and Software 80, 571–583. doi:10.1016/j.jss.2006.07.009

Brooks, H., 1994. The relationship between science and technology. Research Policy 23, 477–486. doi:10.1016/0048-7333(94)01001-3

Buchegger, S., Datta, A., 2009. A case for P2P infrastructure for social networks-opportunities & challenges, in: Sixth International Conference on Wireless On-Demand Network Systems and Services, 2009. WONS 2009. pp. 161–168.

Buchegger, S., Schiöberg, D., Vu, L.-H., Datta, A., 2009. PeerSoN: P2P social networking: early experiences and insights, in: Proceedings of the Second ACM EuroSys Workshop on Social Network Systems. pp. 46–52.

Buyya, R., Abramson, D., Venugopal, S., 2005. The Grid Economy. Proceedings of the IEEE 93, 698–714. doi:10.1109/JPROC.2004.842784

Buyya, R., Broberg, J., Goscinski, A.M., 2010. Cloud Computing: Principles and Paradigms. John Wiley & Sons.

Camerer, C.F., 2011. Behavioral Game Theory: Experiments in Strategic Interaction. Princeton University Press.

Camerer, C.F., Ho, T.-H., Chong, J.K., 2001. Behavioral game theory: Thinking, learning and teaching. JOURNAL OF RISK AND UNCERTAINTY 19, 7–42.

CloudStack, 2013. CloudStack Documentation [WWW Document]. CloudStack Open Source Cloud Computing. URL http://cloudstack.apache.org/docs/en-US/Apache_CloudStack/4.1.1/html/Admin_Guide/feature-overview.html (accessed 8.16.13).

Collier, P., 2009. Rethinking Finance for Africa's Small Firms. SME financing in Sub-saharan Africa, Private Sector & Development Issue 1, 3–4.

Corbin, J.M., Strauss, A., 1990. Grounded theory research: Procedures, canons, and evaluative criteria. Qualitative sociology 13, 3–21.

Dawson, C., 2002. Practical research methods: a user-friendly guide to mastering research techniques and projects. How To Books Ltd.

De Blanche, A., Mankefors-Christiernin, S., 2010. Availability of Unused Computational Resources in an Ordinary Office Environment. Journal of Circuits, Systems & Computers 19, 557–572.

Deltacloud, 2011. About Deltacloud [WWW Document]. Deltacloud. URL http://deltacloud.apache.org/rest-api.html; http://deltacloud.apache.org/about.html (accessed 8.16.13).

Despotovic, Z., Aberer, K., 2006. P2P reputation management: Probabilistic estimation vs. social networks. Computer Networks 50, 485–500.

Domingues, P., Sousa, B., Moura Silva, L., 2007. Sabotage-tolerance and trust management in desktop grid computing. Future Generation Computer Systems 23, 904–912. doi:10.1016/j.future.2006.12.001

Dropbox, 2013. Get more space - Dropbox [WWW Document]. Dropbox. URL https://www.dropbox.com/getspace (accessed 4.7.13).

Dufour, M., 2013. Arguing Around Mathematical Proofs, in: Aberdein, A., Dove, I.J. (Eds.), The Argument of Mathematics, Logic, Epistemology, and the Unity of Science. Springer Netherlands, pp. 61–76.

Erl, T., 2004. Service-Oriented Architecture: A Field Guide to Integrating Xml and Web Services. Prentice Hall.

Eucalyptus, 2013. The Eucalyptus Cloud [WWW Document]. EUCALYPTUS. URL http://www.eucalyptus.com/eucalyptus-cloud/iaas (accessed 4.24.13).

Fahl, S., Harbach, M., Muders, T., Smith, M., 2012. Confidentiality as a Service – Usable Security for the Cloud, in: 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). Presented at the 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 153 –162. doi:10.1109/TrustCom.2012.112

Fast, A., Jensen, D., Levine, B.N., 2005. Creating social networks to improve peer-to-peer networking, in: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining. pp. 568–573.

Feldman, M., Chuang, J., 2005. Overcoming Free-riding Behavior in Peer-to-peer Systems. SIGecom Exch. 5, 41–50. doi:10.1145/1120717.1120723

Fitzgerald, B., 1996. Formalized systems development methodologies: a critical perspective. Information Systems Journal 6, 3–23. doi:10.1046/j.1365-2575.1996.00100.x

Garcia, R., Calantone, R., 2002. A critical look at technological innovation typology and innovativeness terminology: a literature review. Journal of Product Innovation Management 19, 110–132. doi:10.1111/1540-5885.1920110

Gertsen, F., 2012. Management of Research and Development.

Glotzer, S.C., Kim, S., Cummings, P.T., Deshmukh, A., Head-Gordon, M., Karniadakis, G., Petzold, L., Sagui, C., Shinozuka, M., 2009. Interantional Assessment of Research and Development in Simulation-Based Engineering and Science, WTEC Panel Report. World Technology Evaluation Center, Inc., Baltimore, Maryland.

Google Inc., 2013. Google Apps for Education Guide to Going Google [WWW Document]. URL http://eduguide.googleapps.com/ (accessed 4.9.13).

Grosof, M.S., Sardy, H., 1985. A research primer for the social and behavioral sciences. Academic Press.

Hales, T.C., 2008. Formal proof. Notices of the AMS 55, 1370–1380.

Hansen, B.H., Kautz, K., 2005. Grounded theory applied-studying information systems development methodologies in practice, in: System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference On. p. 264b–264b.

Harrison, J., 2008. Formal proof—theory and practice. Notices of the AMS 55, 1395–1406.

Hughes, N., Lonie, S., 2007. M-PESA: Mobile Money for the "Unbanked" Turning Cellphones into 24-Hour Tellers in Kenya. Innovations: Technology, Governance, Globalization 2, 63–81. doi:10.1162/itgg.2007.2.1-2.63

Jain, R., Triandis, H.C., Weick, C.W., 2010. Managing Research, Development and Innovation: Managing the Unmanageable. John Wiley & Sons.

Jamil, E., 2009. What really is SOA. A comparison with Cloud Computing, Web 2.0, SaaS, WOA, Web Services, PaaS and others. White Paper, Soalib Incorporated.

jClouds, 2011. WHAT IS JCLOUDS? [WWW Document]. jClouds. URL http://www.jclouds.org/documentation/gettingstarted/what-is-jclouds/ (accessed 4.27.13).

Jensen, R., 2007. The Digital Provide: Information (Technology), Market Performance, and Welfare in the South Indian Fisheries Sector. The Quarterly Journal of Economics 122, 879–924. doi:10.1162/qjec.122.3.879

Johnson, C., 1997. What is Research in Computing Science?

Jones, S., Hughes, J., 2001. An exploration of the use of grounded theory as a research approach in the field of IS Evaluation, in: Proceedings of the 8th European Conference on Information Technology Evaluation-2001. p. 49.

Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H., 2003. The eigentrust algorithm for reputation management in p2p networks, in: Proceedings of the 12th International Conference on World Wide Web. pp. 640–651.

Kitchenham, B., Charters, S., 2007. Guidelines for performing Systematic Literature Reviews in Software Engineering.

Kuada, E., 2013. Trust Management System for Opportunistic Cloud Services, in: 2013 IEEE 2nd International Conference on Cloud Networking (CloudNet). Presented at the 2013 IEEE 2nd International Conference on Cloud Networking (CloudNet), IEEE, San Francisco, USA, pp. 33 – 41.

Kuada, E., 2014. Towards Trust Engineering for Opportunistic Cloud Services: A Systematic Review of Trust Engineering in Cloud Computing.

Kuada, E., Adanu, K., Olesen, H., 2013. Cloud Computing and Information Technology Resource Cost Management for SMEs, in: Proceedings of IEEE Region 8 Conference EuroCon 2013. Presented at the EUROCON 2013 International conference on computer as a tool, IEEE, University of Zagreb, Croatia, pp. 258 – 265.

Kuada, E., Olesen, H., 2011. A Social Network Approach to Provisioning and Management of Cloud Computing Services for Enterprises, in: Proceedings of The Second International Conference on Cloud Computing, GRIDs, and Virtualization. Presented at the CLOUD COMPUTING 2011, The Second International Conference on Cloud Computing, GRIDs, and Virtualization, pp. 98–104.

Kuada, E., Olesen, H., 2012. Incentive mechanisms for Opportunistic Cloud Computing Services, in: 2012 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom). Presented at the 2012 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), IEEE, Pittsburgh, PA, USA, pp. 127 –136.

Kuada, E., Olesen, H., Henten, A., 2012. Public Policy and Regulatory Implications for the Implementation of Opportunistic Cloud Computing Services for Enterprises, in: 9th International Workshop on Security in Information Systems. Presented at the ICEIS 2012, Wroclaw, Poland, pp. 3 – 13.

KVM, 2013. Kernel Based Virtual Machine [WWW Document]. KVM. URL http://www.linux-kvm.org/page/Main_Page (accessed 10.12.13).

Kaavo, 2013. Cloud Computing Made Easy [WWW Document]. Offerings. URL http://www.kaavo.com/products-and-services (accessed 4.24.13).

Levy, Y., Ellis, T.J., 2006. A Systems Approach to Conduct an Effective Literature Review in Support of. INFORMATION SYSTEMS RESEARCH. INFORMING SCIENCE JOURNAL 9, 181212.

Leyton-Brown, K., Shoham, Y., 2008. Essentials of Game Theory: A Concise Multidisciplinary Introduction. Morgan & Claypool Publishers.

Li, H., Singhal, M., 2007. Trust management in distributed systems. IEEE Computer 40, 45–53.

Lin, C., Varadharajan, V., Wang, Y., Pruthi, V., 2004. Enhancing grid security with trust management, in: 2004 IEEE International Conference on Services Computing, 2004. (SCC 2004). Proceedings. Presented at the 2004 IEEE International Conference on Services Computing, 2004. (SCC 2004). Proceedings, pp. 303–310. doi:10.1109/SCC.2004.1358019

Ma, R.T., Lee, S., Lui, J., Yau, D.K., 2006. Incentive and service differentiation in P2P networks: a game theoretic approach. IEEE/ACM Transactions on Networking (TON) 14, 978–991.

Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J., Ghalsasi, A., 2011. Cloud computing—The business perspective. Decision Support Systems 51, 176–189.

Mas, I., Morawczynski, O., 2009. Designing mobile money services lessons from M-PESA. Innovations: Technology, Governance, Globalization 4, 77–91.

McDermott, C.M., O’Connor, G.C., 2002. Managing radical innovation: an overview of emergent strategy issues. Journal of Product Innovation Management 19, 424–438. doi:10.1111/1540-5885.1960424

Mekouar, L., Iraqi, Y., Boutaba, R., 2006. Handling free riders in peer-to-peer systems, in: Agents and Peer-to-Peer Computing. Springer, pp. 58–69.

Mell, P., Grance, T., 2011. The NIST definition of cloud computing (draft). NIST special publication 800, 145.

Microsoft, 2013a. Microsoft SkyDrive - Microsoft Windows [WWW Document]. windows.microsoft.com. URL http://windows.microsoft.com/en-us/skydrive/download (accessed 4.7.13).

Microsoft, 2013b. Why Hyper-V? Competitive Advantages of Microsoft Hyper-V Server 2012 over the VMware vSphere Hypervisor.

Mihoob, A., Molina-Jimenez, C., Shrivastava, S., 2010. A Case for Consumer-centric Resource Accounting Models, in: 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD). Presented at the 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD), pp. 506 –512. doi:10.1109/CLOUD.2010.44

Moreau, E., 2013. 5 of the Best Free Cloud Storage Providers and Their Features [WWW Document]. About.com Web Trends. URL http://webtrends.about.com/od/office20/tp/Free-Cloud-Storage-Providers-Services.htm (accessed 4.7.13).

Nepal, S., Chen, S., Yao, J., Thilakanathan, D., 2011. DIaaS: Data Integrity as a Service in the Cloud, in: 2011 IEEE International Conference on Cloud Computing (CLOUD). Presented at the 2011 IEEE International Conference on Cloud Computing (CLOUD), pp. 308 –315. doi:10.1109/CLOUD.2011.35

Nisan, N., 2007. Introduction to mechanism design (for computer scientists) : Algorithmic Game Theory, Algorithmic Game Theory. Cambridge University Press, New York, USA.

Nunamaker Jr, J.F., Chen, M., 1990. Systems development in information systems research, in: System Sciences, 1990., Proceedings of the Twenty-Third Annual Hawaii International Conference On. pp. 631–640.

Okoli, C., Schabram, K., 2010. A Guide to Conducting a Systematic Literature Review of Information Systems Research, in: Working Papers on Information Systems.

Olerup, A., 1991. Design approaches: a comparative study of information system design and architectural design. The Computer Journal 34, 215–224.

Olteanu, A., Pierre, G., 2012. Towards robust and scalable peer-to-peer social networks, in: Proceedings of the Fifth Workshop on Social Network Systems. p. 10.

Open Cloud Computing Interface, 2011. Open Cloud Computing Interface [WWW Document]. URL http://occi-wg.org/ (accessed 4.24.13).

Opennebula, 2013. OpenNebula Key Features [WWW Document]. OpenNebula.org Open Source Data Center Virtualization. URL http://opennebula.org/about:keyfeatures (accessed 4.24.13).

OpenStack, 2013. Open source software for building private and public clouds. [WWW Document]. OpenStack Cloud Software. URL http://www.openstack.org/ (accessed 4.24.13).

Oracle VM, 2011. Oracle VM 3: Application-Driven Virtualization.

Osborne, M.J., Rubinstein, A., 2009. A Course in Game Theory (Levine's Bibliography No. 814577000000000225). UCLA Department of Economics.

Oshikoya, T.W., Hussain, M.N., 1998. Information Technology and the Challenge of Economic Development in Africa. African Development Review 10, 100–133. doi:10.1111/j.1467-8268.1998.tb00099.x

Park, J.S., Spetka, E., Rasheed, H., Ratazzi, P., Han, K.J., 2012. Near-Real-Time Cloud Auditing for Rapid Response, in: 2012 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA). Presented at the 2012 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp. 1252 –1257. doi:10.1109/WAINA.2012.78

Pattenden, S., Young, R., Zelm, M., 2011. FInES Standardization Task Force Report.

Pianese, F., Perino, D., Keller, J., Biersack, E.W., 2007. PULSE: an adaptive, incentive-based, unstructured P2P live streaming system. Multimedia, IEEE Transactions on 9, 1645–1660.

Pouwelse, J.A., Garbacki, P., Wang, J., Bakker, A., Yang, J., Iosup, A., Epema, D.H., Reinders, M., Van Steen, M.R., Sips, H.J., 2008. TRIBLER: a social-based peer-to-peer system. Concurrency and Computation: Practice and Experience 20, 127–138.

Richey, R.C., Klein, J.D., 2007. Design and Development Research: Methods, Strategies, and Issues. Routledge.

RightScale, 2013. Why RightScale [WWW Document]. RightScale Cloud Management. URL http://www.rightscale.com/products/why-rightscale.php (accessed 4.24.13).

Schiöberg, D., 2008. A peer-to-peer infrastructure for social networks. Diplom thesis, TU Berlin, Berlin, Germany.

Silverston, T., Fourmaux, O., Crowcroft, J., 2008. Towards an Incentive Mechanism for Peer-to-Peer Multimedia Live Streaming Systems, in: Eighth International Conference on Peer-to-Peer Computing , 2008. P2P  '08. Presented at the Eighth International Conference on Peer-to-Peer Computing , 2008. P2P  '08, pp. 125–128. doi:10.1109/P2P.2008.25

Singh, A., Liu, L., 2003. TrustMe: anonymous management of trust relationships in decentralized P2P systems, in: Third International Conference on Peer-to-Peer Computing, 2003. (P2P 2003). Proceedings. Presented at the Third International Conference on Peer-to-Peer Computing, 2003. (P2P 2003). Proceedings, pp. 142–149. doi:10.1109/PTP.2003.1231514

Song, Z., Molina, J., Strong, C., 2010. Trusted Anonymous Execution: A Model to Raise Trust in Cloud, in: 2010 9th International Conference on Grid and Cooperative Computing (GCC). Presented at the 2010 9th International Conference on Grid and Cooperative Computing (GCC), pp. 133 –138. doi:10.1109/GCC.2010.37

Sultan, N., 2010. Cloud computing for education: A new dawn? International Journal of Information Management 30, 109–116. doi:10.1016/j.ijinfomgt.2009.09.004

Uhlig, R., Neiger, G., Rodgers, D., Santoni, A.L., Martins, F.C., Anderson, A.V., Bennett, S.M., Kagi, A., Leung, F.H., Smith, L., 2005. Intel virtualization technology. Computer 38, 48–56.

VMware, 2013. VMware vSphere Hypervisor$^{TM}$: Virtualization made free and easy [WWW Document]. VMware. URL http://www.vmware.com/products/vsphere-hypervisor/overview.html (accessed 5.5.13).

Voorsluys, W., Broberg, J., Buyya, R., 2011. Introduction to Cloud Computing, in: Cloud Computing Principles and Paradigms. John Wiley & Sons, Inc., Hoboken, New Jersey, pp. 3–41.

Walker, D., Myrick, F., 2006. Grounded theory: An exploration of process and procedure. Qualitative Health Research 16, 547–559.

Wallace, K., Blessing, L., 2007. Engineering design: a systematic approach. Springer.

WTEC, 2007. International Assessment of Research and Development in Simulation-based Engineering and Science [WWW Document]. URL http://www.wtec.org/sbes/ (accessed 10.3.13).

Xen® Hypervisor, 2013. The open source standard for hardware virtualization: What is the Xen® Hypervisor? [WWW Document]. The Linux Foundation Collaborative Projects. URL http://www.xen.org/products/xenhyp.html (accessed 5.5.13).

Xu, X., 2012. From cloud computing to cloud manufacturing. Robot. Comput.-Integr. Manuf. 28, 75–86. doi:10.1016/j.rcim.2011.07.002

Yang, M., Chen, H., Zhao, B.Y., Dai, Y., Zhang, Z., 2004. Deployment of a large-scale peer-to-peer social network, in: Proc. of WORLDS.

Zhao, B.Q., Lui, J.C., Chiu, D.-M., 2009. Analysis of adaptive incentive protocols for P2P networks, in: INFOCOM 2009, IEEE. IEEE, pp. 325–333.

Zhao, B.Q., Lui, J.C.-S., Chiu, D.-M., 2012. A Mathematical Framework for Analyzing Adaptive Incentive Protocols in P2P Networks. IEEE/ACM Transactions on Networking 20, 367–380. doi:10.1109/TNET.2011.2161770

Zhou, K.Z., Yim, C.K. (Bennett), Tse, D.K., 2005. The Effects of Strategic Orientations on Technology- and Market-Based Breakthrough Innovations. Journal of Marketing 69, 42–60.

Zhou, R., Hwang, K., 2007. PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing. IEEE Transactions on Parallel and Distributed Systems 18, 460–473. doi:10.1109/TPDS.2007.1021

## Glossary of Abbreviations

| | |
|---|---|
| Amazon Web Services (AWS) | 28, 30 |
| anything as a service (XaaS) | 25, 53 |
| Application Programmers Interfaces (APIs) | 30 |
| Cloud Computing Interoperability Forum (CCIF) | 30 |
| Cloud Infrastructure Management Interface (CIMI) | 67 |
| Confidentiality as a Service (CaaS) | 25 |
| Data as a Service (DaaS) | 25 |
| Data Integrity as a Service (DIaaS) | 25 |
| Distributed Management Task Force (DMTF) | 67 |
| Information and Communications Technologies (ICT | 13 |
| Information Technology (IT) | 12 |
| Infrastructure as a Service (IaaS) | 25 |
| online social networks (OSN) | 35 |
| Open Cloud Computing Interface (OCCI) | 30 |
| Opportunistic Cloud Services (OCS) | iv, 16, 18, 38, 64 |
| Peer-to-peer (P2P) | 32 |
| Platform as a Service (PaaS) | 25 |
| Pseudo Service Level Agreement (Pseudo SLA) | 49, 64 |
| quality of service (QoS) | 28 |
| Random Access Memory (RAM) | 26 |
| Red Hat Enterprise Virtualization Manager (RHEV-M) | 67 |
| Simple Object Access Protocol (SOAP) | 28 |
| Simple Storage Service (S3) | 31 |
| Small and Medium Enterprises (SMEs) | 13 |
| Software as a Service (SaaS) | 25, 31 |
| software development kits (SDK) | 30 |
| virtual machine (VM) | 26 |
| virtual machine monitor (VMM) | 26 |

## Appendices

This appendices section contains a copy of each of the included papers that form part of this dissertation. Below is a list of the details of the included papers. This is followed with a copy of each of the papers in their original published form.

I. Kuada, Eric, and Henning Olesen. 2011. “A Social Network Approach to Provisioning and Management of Cloud Computing Services for Enterprises.” In proceedings of CLOUD COMPUTING 2011, The Second International Conference on Cloud Computing, GRIDs, and Virtualization , Rome, Italy, pp. 98–104.

II. Kuada, Eric, and Henning Olesen. 2012. “Incentive Mechanisms for Opportunistic Cloud Computing Services.” In *2012 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, Pittsburg, USA, pp. 127 – 136. IEEE.

III. Kuada, Eric. 2014. “Towards Trust Engineering for Opportunistic Cloud Services: A Systematic Review of Trust Engineering in Cloud Computing”, submitted to Journal of Grid Computing.

IV. Kuada, Eric. 2013. “Trust Management System for Opportunistic Cloud Services.” In *2013 IEEE 2nd International Conference on Cloud Networking (CloudNet).* San Francisco, USA: IEEE.

V. Kuada, Eric, Henning Olesen, and Anders Henten. 2012. “Public Policy and Regulatory Implications for the Implementation of Opportunistic Cloud Computing Services for Enterprises.” In *9th International Workshop on Security in Information Systems*, pp. 3 – 13. Wroclaw, Poland.

VI. Kuada, Eric, Kwami Adanu, and Henning Olesen. 2013. “Cloud Computing and Information Technology Resource Cost Management for SMEs.” In *Proceedings of IEEE Region 8 Conference EuroCon 2013*, pp. 258 – 265. University of Zagreb, Croatia: IEEE. http://www.eurocon2013.org/index.html.

## Appendix 1: Paper #1

Kuada, Eric, and Henning Olesen. 2011. “A Social Network Approach to Provisioning and Management of Cloud Computing Services for Enterprises.” In proceedings of CLOUD COMPUTING 2011, The Second International Conference on Cloud Computing, GRIDs, and Virtualization , Rome, Italy,  pp. 98–104.

# A Social Network Approach to Provisioning and Management of Cloud Computing Services for Enterprises

Eric Kuada, Henning Olesen
Center for Communication, Media and Information Technologies
Aalborg University
Copenhagen, Denmark
kuada@cmi.aau.dk, olesen@cmi.aau.dk

*Abstract* - **This paper proposes a social network approach to the provisioning and management of cloud computing services termed Opportunistic Cloud Computing Services (OCCS), for enterprises; and presents the research issues that need to be addressed for its implementation. We hypothesise that OCCS will facilitate the adoption process of cloud computing services by enterprises. OCCS deals with the concept of enterprises taking advantage of cloud computing services to meet their business needs without having to pay or paying a minimal fee for the services. The OCCS network will be modelled and implemented as a social network of enterprises collaborating strategically for the provisioning and consumption of cloud computing services without entering into any business agreements. We conclude that it is possible to configure current cloud service technologies and management tools for OCCS but there is a need for new approaches that view enterprises as both service providers and consumers to facilitate the easy implementation of OCCS networks**.

***Keywords-cloud service brokerage; social networking; and opportunistic cloud computing services.***

## I. INTRODUCTION

Though faced with several challenges which are mostly security and risk management related, cloud computing adoption is gaining grounds with enterprises [1] because of the flexibility, scalability, elasticity, and potential cost savings that it offers to businesses [2]. With the support of industry analysts (e.g., Gartner, PricewaterhouseCoopers) and companies such as Amazon, Google, IBM, VMware, Microsoft, Sun, Dell, etc., this trend is not expected to change. Additionally, Vinod, et al. [3][4] suggest that instead of perceiving cloud computing simply as a way to make internal Information Technology services cheaper and efficient, businesses could take advantage of cloud computing to drive business growth by developing a new business model which is termed as the extensible enterprise.

The benefits of cloud computing has caught the attention of all stakeholders in research efforts to address its challenges to pave the way for an accelerated adoption of cloud computing services. There are therefore currently numerous research efforts by Information Technology industry giants, academic institutions, governments and union of countries (e.g., European Union) to promote the adoption of cloud computing services [5][6][7]. These efforts are resulting in diverse cloud computing service offerings from cloud service providers which have left enterprise consumers trying to make sense of the offerings of service providers. This situation is increasingly necessitating the services of a special group of cloud service providers that offer brokerage services for enterprise consumers on the more fundamental services such as Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) provided by cloud service providers.

This research work proposes a social network approach to the provisioning and management of cloud computing services termed Opportunistic Cloud Computing Service (OCCS) that has some resemblance to Cloud Service Brokerage (CSB). OCCS deals with the concept of enterprises taking advantage of cloud computing services to meet their business needs without having to pay or paying a minimal fee for the services.

This innovative approach of OCCS can facilitate the adoption process since enterprises will require no financial commitments to begin using cloud computing services, and discovery of services on an OCCS network will be easier in light of how information spreads on social networks. Commercial cloud service providers can benefit tremendously in the long run by introducing some of their services onto such a network; especially new services can be introduced onto the OCCS network for a period of time to gain popularity before being withdrawn later. Additionally OCCS can promote SaaS collaboration, scalability for resource aggregation for particular services when needed, fostering of business collaboration and further reduction of cost in Information Technology services. Since the idea of OCCS will be to provide a governance platform and its associated cloud management tools with which interested enterprises will provision SaaS, PaaS, IaaS and other resources that would be used by other interested enterprises, but not necessarily create new technologies, the platform is compatible with future cloud computing technologies and solutions.

The remainder of the paper is organised as followers: Section II explains the OCCS concept and outlines some of the background ideas and concepts that have inspired it. Section II also presents cloud service brokerage and outlines the similarities of OCCS in functionality with CSB. We present the research challenges that must be addressed for the implementation of OCCS in Section III. Section IV discusses some unintended advantages that could be leveraged from OCCS implementation and Section V concludes the paper.

## II. OPPORTUNISTIC CLOUD COMPUTING SERVICES

This section begins with an overview of the opportunistic cloud computing services concept, an outline of some background developments inspiring it, then a discussion of its Cloud Services Brokerage features and then presents detailed reference architecture for its implementation.

### A. *Overview*

Opportunistic Cloud Computing Service (OCCS) is a social network approach to the provisioning and management of cloud computing services for enterprises. Previous works that link cloud computing with social networks such as [8], looked at leveraging the pre-established trust formed through friend relationships within social networking sites to enable friends to share resources; and most other examples use Cloud platforms to host social networks or create applications within the social network. There is however no literature on a social network infrastructure for enterprises currently; and this is where OCCS comes in. OCCS deals with the concept of enterprises taking advantage of cloud computing services to meet their business needs without having to pay or paying a minimal fee for the services. The OCCS network will form a social network of enterprises collaborating strategically (possibly selfishly or even maliciously) for the provisioning and consumption of cloud computing services without entering into any business agreements. Unlike social networking sites for individual use where users creates their own network of friends, in an OCCS network, members do not explicitly create ties with other members but these ties comes indirectly through the services and resource contribution and consumption mechanism.

This concept is derived from the combination of the concepts of peer-to-peer network services and the utility model of cloud computing. As in peer-to-peer networks where users are both resource providers and consumers, the idea will be to provide a governing platform that serves as the social networking platform for the enterprises and also consisting of interoperable Cloud management tools with which interested enterprises will provision SaaS, PaaS, IaaS and other resources that would be used by other enterprises interested in these services. A major challenge besides risk management and security issues that such a network will face is how to develop incentive schemes that ensure sustainability of the network.

It is anticipated that such a network will not always provide all the cloud service needs of an enterprise; hence OCCS will also seek to explore the utility model of cloud computing for enterprises to consume services provided by commercial cloud computing service providers at specific times, geographic locations, and Service Level Agreement (SLA) requirements for which a utility function defined by the enterprise is minimized. Here again the framework will try to employ open source brokerage tools instead of employing the services of a commercial Cloud Service Broker (CSB) for arbitrating between the cloud service providers and the enterprises.

Furthermore, preliminary investigations indicate that the OCCS network will not be most ideal for large corporation and financial institutions but will be well suited for small and medium sized enterprises. There have however been indications of larger corporations joining an OCCS network mainly as services and resource contributors in promoting their businesses.

Figure 1 shows an overview of the major parts in an OCCS network. It consists of two layers – the service layer and the management layer. The service layer consists of all the services contributed by members. These will normally be fundamental cloud services such as SaaS, PaaS, and IaaS; but, it can also include value added services normally provided by cloud service brokers. The management layer consists of two main components – the governance component that manages the services from members and CSB component that serves as an interface between the OCCS network and commercial cloud services providers and cloud service brokers.

OCCS is derived from two main concepts: peer-to-peer network services and the utility model of cloud computing. It however has also been inspired by equally important phenomenon such as social network theory, social networking, Web2.0, and the open source movement.

Social network theory has been used to examine how companies interact with each other, characterizing the many informal connections that link executives together, as well as associations and connections between individual employees at different companies. These networks provide ways for companies to gather information, deter competition, and even collude in setting prices or policies. It forms the basis of the OCCS feature of having no formal business agreements between the participating member enterprises. The other characteristics of OCCS stem from concepts and ideas such as user-generated content, harnessing the power of the crowd, architecture of participation, data on a epic scale, and openness [9] that characterises Web 2.0, social networking and the open source movement. OCCS however focus on corporate organisations instead of individual users and deals with replacing simple data and files as resources with cloud computing services that would normally have been provided by commercial cloud service providers.

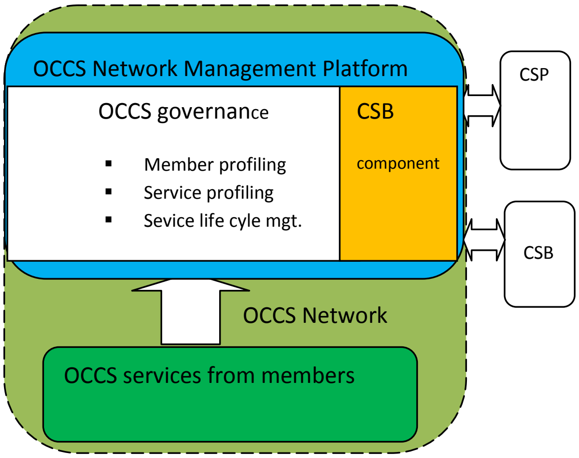


Figure 1. Major components of an OCCS network

### B. Cloud Service Brokerage Functionalities

Cloud services brokerage is a business model where a company or other entity adds value to one or more (generally public or hybrid, but possibly private) cloud services on behalf of one or more consumers of those services [10]. The major functionalities that CSB provide to enterprises include service aggregation, customization, governance, new applications, services billing and arbitration, security, and insurance services. The services of CSB are becoming increasingly necessary to both enterprises and cloud service providers because of their different perspectives, objectives, and expectations from the cloud computing industry, coupled with the challenges enterprises have to deal with in selecting from cloud service providers and using the diverse cloud computing services.

An OCCS network consists of two main components - a platform for managing the services provisioned by members and a brokerage component for interfacing with commercial cloud service providers. The OCCS concept thus inherently provides new applications, service aggregation to multiple consumers, governance, and service arbitration and billing.

### C. OCCS Architecture

In light with the principles on which the OCCS concept is built – namely: user-generated content, architecture of participation and openness; a successful implementation of an OCCS network will have to in the barest minimum provide the following features

- Support for the management of fundamental cloud computing services (SaaS, PaaS, IaaS)
- Support for the management of any arbitrary cloud computing service – anything as a service (XaaS)
- Interoperability with major cloud computing standards
- Interoperability with major cloud computing management tools
- Support for future cloud computing technologies

These factors have been considered in the design of the OCCS network reference architecture shown in Figure 2.

***OCCS Services***: these consist of all the services and resources that have been contributed to the network by members. These could be coming from contributing member’s data center, private cloud, etc. Services are mainly fundamental cloud computing services such as SaaS, PaaS, IaaS; and other cloud computing services (XaaS) and resources.

***Resource Manager***: this together with the cloud computing deployment and management tools found in the Contributions Component and the Discovery & Utilization Component abstract the contributed services from the services layer and interface it to the OCCS management platform.

***Contributions Component:*** it is responsible for handling the resource contribution process. Its main objective is to simplify and make it easy for members to contribute resources to the network. It performs two sub functions – providing cloud computing management tools and service life cycle management. It thus consists of cloud computing deployment and management tools for all types of services and resources. The service management involves service creation, service certification and service profiling which includes service review and ranking by users and service ranking by platform administrators.

***Discovery & Utilization Component:*** its role is to simplify services and resources discovery and utilization process. It performs service recommendation by taking service requirements description by members and matching these with service properties description by contributors together with the profile rank of services. It also consists of cloud computing management tools for services and resources provisioning and utilization.

***Categorization Component:*** this component is needed to ensure OCCS network supports arbitrary services while also ensuring easy management of these services. It is responsible for the categories creation process. It handles service category creation requests from members which is evaluated for approval by the platform administrators; and also delegates privileges of categories creation given to some level of membership.

***Membership Manager:*** this is the main social network user management module for the OCCS network management platform. It is responsible for managing existing users and the registration of new members (enterprises, companies, institutions, etc.). It handles membership requests and in cooperation with the Governance Component performs company profile verification based on data provided by enterprises during registration to make decisions on membership approval or rejection.

***Incentives Manager:*** Dynamic re-computation of cost in real time to be credited to service contributors and debited to resource users. Cost of service or resource utilization is dependent on demand.

***QoS & Pseudo SLA Manager:*** it uses information from the Incentives Manager to provide service differentiation and pseudo SLA management to members.

***Governance Component:*** it is the logical module that provides supervision for all the other components in the OCCS network management platform. It is implemented as the interfaces through which platform administrators interact with the platform to make governance decision.

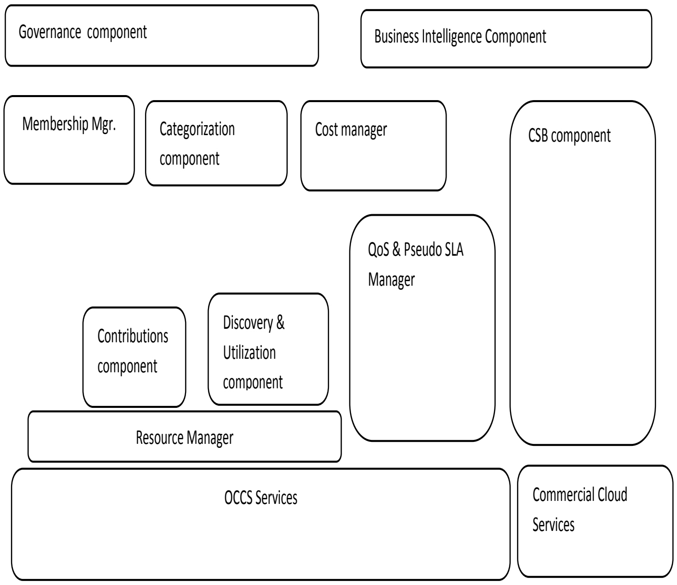


Figure 2. OCCS network Reference Architecture

***Business Intelligence Component:*** this module is not essentially required for the operation of the OCCS network but provides means for the gathering of business intelligence from the platform and may include for example:

- Analysis of services contributed, their categories, utilization and their profile performance
- Analysis of member profiles with their contributed services and the enterprises that are utilizing these services
- Analysis of the services and resources requests that are not currently being provided by the platform

***CSB Component:*** it consists of cloud computing management tools and processes that interface the OCCS network to commercial cloud computing services and provide cloud brokerage services to members.

### *D. Implementation Strategy for OCCS*

To ensure that the barest minimum features required for a successful implementation of a OCCS network is met, a typical OCCS network implementation will use the feature requirements of support for the management of fundamental cloud computing services, support for the management of any arbitrary cloud computing service, interoperability with major cloud computing standards and cloud computing management tools, and support for future cloud management technologies, in selecting a suitable cloud management tool (likely a non proprietary cloud management tool) which will form the base on which other functionalities can be added. The various components outlined in the OCCS reference architecture in Section III *C* above can then be developed on this base cloud management tool.

## III. RESEARCH ISSUES WITH OCCS

Some of the major challenges of cloud computing receiving research attention currently include legal and compliance risk management, migration of applications, meeting SLA requirements, managing cloud services, and security concerns. The introduction of OCCS brings new research issues and adds a complexity dimension to some of the existing ones. This section outlines some of these research issues and the intuitive approaches of addressing them, which will have to be researched carefully for the successful implementation of OCCS networks.

### *A. Sustainability and Pseudo SLA*

The sustainability of an OCCS network revolves around the concepts of architecture of participation and harnessing the power of the crowd. A potential problem that such a network will face is that of free-riding where member enterprises will want to only use services on the network without contributing [10]. The challenge here will be to develop appropriate incentive mechanisms for the sustainable operation of the network.

Another challenge is that of service differentiation and service quality management. Unlike conventional cloud computing service offerings by commercial service providers, no SLA exist between the participating members in an OCCS network, hence such service quality differentiation must be handled through the incentive mechanisms that will be designed so that when limited resources are being contended for by multiple candidates those that have supported the system more can be given some form of preference. Additionally, there will be the need for transparency in dynamic demands and cost of service utilisation. Several research efforts have applied game theoretic approach to the modelling of incentives in peer-to-peer networks to solve the free-riding problem in peer-to-peer networks. [12] presents a resource allocation mechanism based on a distributed algorithm to enable service differentiation in peer-to-peer networks that also increases the aggregate utility in the whole network. Work on incentives for sharing in peer-to-peer networks by [13] analyzes several different payment mechanisms designed to encourage file sharing in peer-to-peer systems. The game theoretic approach can be explored in the design of incentive mechanisms for OCCS networks and the concept of pseudo SLA introduced for service differentiation and service quality management.

### *B. Reliability and Fault resilience*

An OCCS network will need to provide a certain level of reliability to its members under normal operations and must be resilient enough to recover from faults. The reliability and resilience is however threatened by poor quality of services provisioned by members, failure and withdrawal of services from members, and the introduction of malicious services. Dynamic algorithms are required for detection, notification and responding to faults and poor quality services. Of particular importance is how to respond to faults in the network. A simple approach will be to notify service consumers of problematic events for them to take their own decisions; it may however be necessary to develop mechanisms that reassign alternative services to consumers based on certain usage policies and preferences indicated by the service consumers. The challenge here is the precise capturing of the properties of services in service descriptors and effectively matching these to the usage policies and SLA requirements of potential service consumers so that the entire process is transparent to them and their customers; and more so this transparency in fault handling must be achieved in the context of the fact that no SLA exists between the contributors of the services and the consumers of these services.

### *C. Network Governance*

The purpose of the OCCS network governance will be to promote the overall quality of the system. Of particular research interest is the development of community management enabling technologies for profiling, service life cycle management and transparency in the pseudo SLA management. Both network members (enterprises) and the services they provision will have to be profiled to maintain trust in the individual services, member enterprises and the entire system platform. For example service provisioning will have to be in phases such as testing, and various levels of certification through continual ranking of services. Both central ranking by the platform administrators and peer review ranking by the members may have to be adopted. The service ranking and certification will need to promote new services from good profiled enterprises while quickly identifying malicious and poor quality services and revoking their certification.

### *D. Security*

Security is the ability to protect information and information systems from unauthorized access, use, disclosure, disruption, modification or destruction and to respond and recover in case of a fault or incident. The implementation of OCCS will not bring any new technical demands on security in terms of confidentiality and data integrity apart from what is already necessary in ordinary cloud computing implementations. An area of research interest however is how to harness the available resources on the platform and the collaboration of members in combating security threats. If we consider the introduction of malicious services onto the OCCS platform, the OCCS network governance which includes member profiling, service profiling and life cycle management should prevent such occurrences. In the event of such an occurrence however, the system has to respond and recover quickly. It is therefore useful to research into mechanisms for harnessing the available resources on the platform and the collaborative efforts of members in dealing with such a threat.

### *E. Other Research Issues*

Some other issues that are of importance and worth looking at are regulations and service provisioning. Current cloud computing vendor technologies and management tools assume distinct roles for the service providers and service consumers. But with some cloud management tools offering features such as delegated control and autonomous virtual enterprises [14]; and support for the technologies of most of the major cloud solution providers [15], it will be possible to configure these cloud management tools for OCCS. There may however be a need for new approaches for cloud management that view enterprises as both resource providers and consumers to facilitate the easy implementation of an OCCS network.

An issue with regulatory authorities for enterprises joining the OCCS platform could be that of tax evasion implications. This is because enterprises will be offering and using services, which are not being paid for and hence may not be subject to taxes depending on the country in which they are. Also most enterprises have internal policies that need adherence, and there may be industry specific laws and regulations that they need to comply with. Furthermore, different countries have their own laws concerning user data handling. Storing data in the Cloud therefore presents enterprises and service providers with several risk management challenges. These challenges are further compounded by the concept of OCCS and hence can hamper its successful implementation.

## IV. POSSIBLE FUTURE BENEFITS

This section gives brief discussions on some of the unintended benefits that can be leveraged from the implementation of an OCCS network. Some of benefits as discussed below include platform for new business models, promotion of SaaS collaborations, and promotion of cloud computing standardization.

### *A. Platform for new Business Models*

OCCS can serve as a platform for enterprises to adopt new business models such as the extensible-enterprise model (deep B2B integration and highly modular web services). The adoption of cloud computing by any two companies in general reduces the complexities in business-to-business (B2B) integration. Companies can therefore leverage cloud computing by exposing their business processes to potentially large ecosystems of partners who often find ways of joining and integrating their business processes in the value chain. It is envisaged that OCCS will promote the adoption of cloud computing by enterprises and hence indirectly promoting such new business models. Secondly, enterprises on an OCCS network would already have been using similar services with similar cloud management tools; this should facilitate the integration of their business processes.

Additionally, the platform can foster the creation of new business that will provide commercial cloud brokerage services to members on the OCCS network.

### *B. Promotion of SaaS collaborations*

The implementation of an OCCS network can promote SaaS collaborations. Enterprises on an OCCS network are very likely to participate in collaboration efforts in the development of software solutions that they deem useful to their own business. As an example, a construction company in need of a specialized software for design simulation that is currently not being provided by any member on an OCCS platform can initiate a SaaS project to involve other interested members in the development of the software which can then be contributed to the platform upon completion. Such SaaS collaborations could also come about by a member enterprise identifying an application of interest and providing the development platform with specific tools and providing it as a PaaS on the OCCS network; this could spark interest in the development of such an application by other members and can eventually lead to collaboration by interested members in its development.

### *C. Promotion of Cloud Computing Standardization*

As already indicated in Section II C and Section II D, a successful implementation of an OCCS network must provide support for the management of fundamental cloud computing services, support for the management of any arbitrary cloud computing service, interoperability with major cloud computing standards and cloud computing management tools, and support for future cloud management technologies. Thus to start with, the OCCS concept must carefully follow cloud computing standards; the situation is however reversed if OCCS network implementations become successful. Thus those standards that are dominant on the OCCS platform will then be followed closely by cloud management tool developers and cloud service providers. This will further promote the success of the OCCS platform; and hence the promotion of cloud computing standardization and promotion of the OCCS implementations will be in a virtuous cycle.

## V. CONCLUSION

Support for major hypervisors and role-based delegated control make it possible to configure current cloud computing technologies and management tools for OCCS even though they assume distinct roles for the service providers and service consumers. There is however a need for new approaches to cloud management that view enterprises as both resource providers and consumers which when complemented with standards for interoperability will facilitate the easy implementation of an OCCS network.

Successful implementation of OCCS networks can result is some unintended benefits such as serving as a platform for new business models, promotion of SaaS collaborations, and promotion of cloud computing standardization. These benefits together with providing a platform for enterprises to start using cloud computing services without any initial financial commitment will however be possible only if the research challenges identified in Section III (namely, developing appropriate incentive mechanisms and the associated quality of service differentiation, security, reliability and fault resilience, network governance and regulatory issues) are carefully dealt with.

## REFERENCES

[1] Justin Pirie, "Setting the Standards," *European Communications*, pp. 30-31, Autumn 2010.

[2] Vinod Baya and Randy Myers, "How CFOs should audit the cloud balance sheet," *PricewaterhouseCoopers Technology Forecast*, no. 4, pp. 44-53, 2010.

[3] Vinod Baya and Galen Gruman, "Making the Extensible Enterprise a reality," *PricewaterhouseCoopers Technology Forecast*, no. 4, pp. 26-35, 2010.

ISBN: 978-1-61208-153-3

[4] Vinod Baya, Bud Mathaisel, and Bo Parker, "The cloud you don’t know: An engine for new business growth," *PricewaterhouseCoopers Technology Forecast*, no. 4, pp. 4-13, 2010.

[5] OPTIMIS: Optimized Infrastructure Services. (2011, March) OPTIMIS. [Online]. http://www.optimis-project.eu/project 11-07-2011

[6] Contrail consortium 2010. (2011, March) Contrail. [Online]. http://http://contrail-project.eu/objectives;jsessionid=E1961F3D98F3B602D34CFC9445D63DC7 11-07-2011

[7] RESERVOIR. (2011, March) RESERVOIR. [Online]. http://www.reservoir-fp7.eu/ 11-07-2011

[8] Kyle Chard, Simon Caton, Omer Rana, Kris Bubendorfer, "Social Cloud: Cloud Computing in Social Networks," in *2010 IEEE 3rd International Conference on Cloud Computing*, Miami, Florida, 2010, pp. 99 - 106.

[9] Paul Andeson, "What is Web 2.0? Ideas, technologies and implications for education," JISC , JISC Technology and Standards Watch Feb. 2007.

[10] Benoit Lheureux. (2010, December) Gartner Inc. [Online]. http://www.gartner.com/it/content/1461800/1461813/december_2_cloud_services_brokerage_blheureux.pdf 12-07-2011

[11] Markus Hofmann and Leland R. Beaumont, *Content Networking: Architecture, Protocols, and Practice*, Rick Adams and Karyn Johnson, Eds. San Francisco, USA: Elsevier, 2005.

[12] Richard T. B. Ma, et al, "A Game Theoretic Approach to Provide Incentive and Service Differenciation in Peer-to-Peer Networks," in *SIGMETRICS/Performance’04*, New York, NY, USA., June 12–16, 2004.

[13] Philippe Golle, Kevin Leyton-Brown, Ilya Mironov, and Mark Lillibridge, "Incentives for Sharing in Peer-to-Peer Networks," in *Springer WELCOM 2001, LNCS 2232*, Verlag Berlin Heidelberg, 2001, pp. 75-87.

[14] Abiquo, Inc. (2011, March) abiquo. [Online]. http://www.abiquo.com/products/features-and-benefits.php?lang=en 11-07-2011

[15] enStratus Networks LLC. (2011, April) enStratus. [Online]. http://www.enstratus.com/page/1/cloud-providers.jsp 11-07-2011

## Appendix 2: Paper #2

Kuada, Eric, and Henning Olesen. 2012. “Incentive Mechanisms for Opportunistic Cloud Computing Services.” In *2012 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, Pittsburg, USA, pp. 127 – 136. IEEE.

# Incentive Mechanisms for Opportunistic Cloud Computing Services

Eric Kuada, Henning Olesen
Center for Communication, Media and Information Technologies
Aalborg University
Copenhagen, Denmark
{kuada, olesen}@cmi.aau.dk

***Abstract*— Opportunistic Cloud Computing Service (OCCS) is a social network approach to the provisioning and management of cloud computing services for enterprises. The OCCS network may suffer from the free riding problem where members are selfish and will only want to use services on the platform without ever contributing resources. It may also suffer from resource wastage from members or external entities trying to attack the system so that genuine users are deprived of valuable resources. The purpose of this paper is to design incentive schemes that will encourage the contribution of resources to the OCCS platform as well as the efficient usage of these resources. We employ game theory and mechanism design to model and design the incentive schemes. We present two game models and show the existence of a pure strategy Nash equilibrium for both the cooperative and non-cooperative games. Three base incentive schemes are presented and two advanced schemes one based on discount factor and the other a stochastic scheme are also presented. We perform analytical evaluation of our incentive schemes and conclude that the schemes meet the desired properties of budget-balance, ex-post individual rationality, incentive compatibility, allocative efficiency, robustness, and flexible to accommodate changing user behavior on the platform.**

***Keywords- game theory; mechanism design; opportunistic cloud computing services***

## I. INTRODUCTION

A lot of Small and Medium size Enterprises (SMEs) and even relatively big companies lack the adequate Information Technology (IT) resources necessary for their business processes. Spare IT resources such as computing power (CPU and RAM) and storage capacity are however available at some large corporations and even relatively smaller companies. It should be useful to have a way by which companies and other organizations can make use of these spare resources to get their required IT resources necessary for their business processes. The advent of cloud computing within the past few years technically supports this concept. Cloud computing is mainly the packaging of traditional information technology infrastructure and software solutions such as storage, CPU, network, applications , services, etc. as virtualized resources and delivered by a service provider to its customers as an on-demand pay-per-use self-provisioned service through a web portal over a network such as the Internet. There have been major technological advancements as well as social and business demands driving this new trend of computing. The technological factors facilitating cloud computing include the availability and drastic increase in reliable broadband Internet access, advancements in virtualization technologies and the shift of development of majority of both desktop and enterprise applications as web services and applications software.

Our hypothesis is that a social network approach can be used for companies with underutilized spare resources to provide IT resources to SMEs and other organizations that may need them by leveraging the cloud computing paradigm and supporting technologies. Though the concept of taking advantage of spare computing resources to meet certain demands – generally referred to as volunteer computing (e.g. Folding@home, SETI@home, and MilkyWay@home) has existed for a some years now, it focuses on gathering large numbers of donated computing cycles to form a large-scale virtual supercomputer; and only has support for embarrassingly parallel applications and does not offer enough varied and enticing alternative use-cases [1].

The goal therefore is to identify the sufficient enabling conditions for the successful implementation of this social network approach to IT resource sharing. Reference [2] introduced the Opportunistic Cloud Computing Services (OCCS) concept, presented the research issues that are needed to be addressed for its successful implementation and presented the detailed reference architect for it. Because OCCS promises an accelerated adoption of cloud computing services and further reduction in IT cost for small companies, we have been working on the feasibility of its successful implementation in terms of the technical feasibility, impact of public policy and regulations on its implementation, and its acceptance and support by all stakeholders. Additionally the data center and Cloud management competences that companies develop over time through managing their own resources can be of value to others lacking such competences (e.g. SMEs needs, especially in the developing world). OCCS also has the potential of fostering business collaboration, offering further reduction of cost in IT services and by design is compatible with future cloud computing technologies and solutions.

The OCCS network like peer-to-peer networks may suffer from the free riding problem where members are selfish and will only want to use services on the platform without ever contributing resources to it. It may also suffer from resource wastage from members on the platform or external entities trying to attack the system so that genuine

978-1-936968-36-7 © 2012 ICST

DOI 10.4108/icst.collaboratecom.2012.250403

users are deprived of valuable resources. The purpose of this paper is therefore to design incentive schemes that will encourage the contribution of resources to the OCCS platform as well as the efficient usage of these resources.

Additionally there should be enough motivation for managing the OCCS governance platform. Furthermore because enterprises will be making use of these resources on the platform to offer services to their customers, they may require some level of quality of service (e.g. availability). However, unlike as is the case with services provided by commercial cloud service providers, no SLA exists among the members on the OCCS network. It is therefore necessary to devise a way for offering services differentiation so that when scarce resources are being contended for by multiple members those that have contributed more resources to the OCCS network can be given some level of preferential treatment in offering these scarce resources to them.

The approach to addressing the challenges that are of interest in this work is to employ game theory [3] [4] to model and design incentive mechanisms that will encourage the contribution of resources to the OCCS platform as well as their efficient usage. The contribution of this work is the design of suitable incentive schemes for resource sharing of generic cloud computing services, and also providing the foundation for the implementation of Pseudo Service Level Agreement (Pseudo SLA) on the OCCS platform which is discussed later in the sequel to this paper on "Trust engineering and Pseudo SLA in Opportunistic Cloud Computing Services". It should be evident from the title of the sequel to this paper that we consider data privacy, security and trust as major issues that OCCS and cloud computing in general has to address. We are therefore currently working on trust engineering in cloud computing and how to adapt it for the OCCS environment.

The remainder of the paper is organized as follows: Section II gives the background for this work, the first part of which is an overview on opportunistic cloud computing services and the second part discusses related works. Section III deals with the incentives modeling process and the results obtained. We present two game models and show the existence of a pure strategy Nash equilibrium for both the cooperative and non-cooperative games. Section III also presents a base incentive scheme with its three variations and two advanced schemes one based on discount factor and the other a stochastic scheme. Section IV presents the resource allocation process and the evaluation of the presented incentive mechanisms. Section V concludes the paper and also touches on future work. We also include an appendix on the proof of the existence of a pure strategy Nash equilibrium.

## II. Background

The first part of this section gives an overview on cloud computing mainly on the types of services and the deployment models. The second part gives an overview of the OCCS concept regarding its main components. Its details and reference architecture can be found in [2]. The possible impact that public policy and data privacy regulations may have on its implementation can also be found in [5]. In the final part of this section we consider some related works that has a bearing on a at least two of the topical areas of cloud computing, resource allocation, game theory, and incentive mechanism design.

### *A. Cloud Computing Overview*

Cloud computing is essentially the packaging of traditional information technology infrastructure and software solutions such as storage, CPU, network, applications, services, etc. as virtualized resources and delivered by a service provider to its customers as an on-demand pay-per-use self-provisioned service through a web portal over a network such as the Internet. There have been major technological advancements as well as social and business demands driving this new trend of computing. The technological factors facilitating cloud computing include the availability and drastic increase in reliable broadband Internet access, advancements in virtualization technologies and the shift of development of majority of both desktop and enterprise applications as web services and applications.

The three main components of a regular computing environment, namely the hardware infrastructure, the operating system platform and user application software, have respectively translated into Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) delivered in cloud computing. Additionally, there is an inexhaustible list of other cloud computing services due to the concept of "Anything as a Service" (XaaS) being the main driving idea of cloud computing. Thus, virtually all IT products and solutions are potential cloud computing services. These services are normally deployed in four main cloud deployment models namely public, private, community, and hybrid cloud computing deployment models [6].

A public cloud is one in which the infrastructure and other computational resources that it comprises are made available to the general public over the Internet. It is owned by a cloud service provider selling cloud services and by definition, is external to an organization. Private clouds are at the other end of the spectrum; a private cloud is one in which the computing environment is operated exclusively for an organization. It may be managed either by the organization itself or a third party such as a commercial cloud services provider, and may be hosted within the organization's data center or outside of it. The community clouds and hybrid clouds fall between public and private cloud deployment models. A community cloud is similar to a private cloud, but the infrastructure and computational resources are shared by several organizations that have common privacy, security, and regulatory considerations, rather than for the exclusive use of a single organization. A hybrid cloud deployment model is a combination of two or more of the other cloud deployment models (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technologies that enable interoperability.

### B. *Opportunistic Cloud Computing Services*

Opportunistic cloud computing service is a social network approach to the provisioning and management of cloud computing services for enterprises. OCCS deals with the concept of enterprises taking advantage of cloud computing services to meet their business needs without having to pay or paying a minimal fee for these services [2]. The OCCS network is a social network of enterprises collaborating strategically for the contribution and usage of cloud computing services without entering into any business agreements. Unlike social networking services provide by social networking sites for individual use where users create their own network of friends, in an OCCS network, members do not explicitly create ties with other members but these ties come indirectly through the resource contribution and consumption process.

The OCCS network platform is a governing platform that serves as the social networking platform for enterprises and also includes interoperable Cloud management tools with which member enterprises can provision resources that will be used by other enterprises interested in these services. The OCCS platform consists of two main layers – the service layer and the management layer. The service layer consists of all the services contributed by members. These will normally be fundamental cloud computing services such as SaaS, PaaS, and IaaS; but it can also include value added services normally provided by cloud service brokers. The management layer consists of two main components – the governance component that manages the services from members and cloud services brokerage component that serves as an interface between the OCCS network and commercial cloud services providers and cloud service brokers.

### C. *Related Work*

It this section we consider related works that has a bearing on at least two of the topical areas of cloud computing, resource allocation, game theory, and incentive mechanism design. In [7] they looked at a single commercial IaaS provider service provisioning problem for multiple SaaS providers who in turn are trying to maximize their profit on services provided to their end-user customers subject to SLA constraints. SaaS providers want to maximize their revenues from SLAs, while minimizing the cost of use of resources supplied by the IaaS provider. They modeled the service provisioning problem as a Generalized Nash game, and proposed an algorithm for the run time management and allocation of IaaS resources to competing SaaS providers.

Reference [8] proposed the incorporation of mechanism design to enforce and achieve efficient resource utilization among selfish VMs in non-cooperative cloud systems. They looked at the problem that on cloud platforms, computing resources are allocated on-demand dynamically and the application hosted on a virtual machine (VM) usually has the illusion of complete control of resources. Thus, a selfish VM may strategically compete for resource with other VMs to maximize its own benefit while at the cost of overall system performance; this problem poses new challenges to cloud providers, who must thwart non-cooperative behavior as well as allocating resource among selfish VMs efficiently. They proposed to use mechanism design to allocate resource among selfish VMs in a non-cooperative cloud environment.

Reference [9] considered applying game theory to the resource allocation problem where service demanders intend to solve parallel computing problem by requesting the usage of resources across a cloud-based network while service providers schedule and allocate resources to the individual subtasks. Other related works such as [10] have used game theory in the modeling of incentives in P2P systems.

The related works mentioned above and other previous works on resource allocation and mechanism design on cloud services all look at the resource allocation problem of already existing resources of commercial cloud services. We are in addition to this, looking at incentive mechanism design for the schemes to also promote the contribution of resources to the platform together with the efficient usage of these contributed resources.

## III. Incentive Mechanisms for OCCS

Mechanism design is the sub field of microeconomics and game theory that considers how to implement good system-wide solutions to problems that involve multiple self-interested agents, each with private information about their preferences. It is best to view the goals of the designed mechanisms in the very abstract terms of social choice. A social choice is an aggregation of the preferences of the different participants towards a single joint decision. Mechanism design attempts to implement desired social choices in a strategic setting. Such strategic design is necessary since usually the preferences of the participants are private [11]. We employ the concepts of game theory – particularly those of mechanism design in the design of incentives for encouraging resource contribution and efficient usage of these resources for OCCS platforms in the remainder of the sections of this paper.

### A. *The Modeling Process Methodology*

The methodology for the modeling process is as follows: We begin the modeling process by defining key properties of the real world system that is to be modeled. Next we outline the desired properties that the incentive schemes should possess and hence the modeling process should achieve. We then create a model of the real world system and design the incentives schemes. We also review related work involving incentive design for cloud computing resource allocation and game theory and refine the model with some inspiration from the reviewed related work. We perform analysis to compare the results with the desired properties of the incentive mechanisms. Finally, we iteratively refine the model and the incentive mechanisms until the result approaches the desired properties.

The key properties of the OCCS platform to be modeled include the nature of the members and services, the credit points and benefits, knowledge of information on the platform, and the resource allocation.

#### 1) *Nature of Members and Services*

The OCCS network consists of a set of strategic members contributing and utilizing cloud computing services. The platform consists of a set of services each belonging to a category; each service has a non-monetary cost that varies dynamically. The service or resource contributed by a member is of a certain finite capacity and the resources to a particular service may be contributed by multiple members. It should be noted that though there may be members that are supposedly altruistically offering services without directly benefiting from using services provided by others, they may still be doing it strategically since they may be doing it for example for advertisement purposes.

Members will normally only contribute resources that they have spare capacity of (e.g. CPU, storage, application that they have developed internally, etc.). We term these resources contributable resources. It should be obvious that a member's IT resources that it has need of using for its own production will constitute its non-contributable resources. Contributable resources incur a base personal (non OCCS platform) maintenance cost whether or not they are contributed; and also incur additional utilization cost when they have been contributed and are being utilized by other members. Members are free to provide and discontinue one or more services at will at any point in time. They are likewise free to use or discontinue the usage of one or more services at will at any point in time.

#### 2) *OCCS Platform credit Points and Benefits*

Members are credited with OCCS platform points for contributing resources towards a service and debited OCCS platform points for resource usage. The only way to obtain credits on the platform is by accumulating the credit points through contributing resources to the platform. Members are interested in maximizing their OCCS platform credit points obtained for resource contribution and minimizing OCCS platform cost incurred in resource usage. Members receive a certain personal (non OCCS platform) benefit from utilizing a service; e.g. the benefit they obtain in using the service for providing services to their customers or in some cases simply as a form of advertisement.

#### 3) *Knowledge of Information on Platform*

Members are aware of the activities of other players, and actions are sequential. Since the history about a service is known only to a certain extent they may or may not have perfect information about a particular service they might be interested in.

#### 4) *Resource Allocation*

When the demand on a service is higher than its resources can support, interested users of the service with currently higher accumulated OCCS credit points are giving higher priority to use such resources. They may however choose not to use the service now and rather use it when the demand and hence the cost go down. However since their business processes may depend on using the service now in order to provide a certain level of quality of service (e.g. availability) to their customers, they may be constrained to use the service at peak demand. Members can be allocated recourses even when their OCCS credit is zero or negative.

### B. *Desired Features of the Incentive Schemes*

The main features of the incentive schemes are as follows: It needs to be incentive compatible. That is it induces cooperation among an otherwise rational or selfish players. Secondly, it does not only encourage contribution of resources but also the efficient usage of these resources. Furthermore it must ensure the existence of sustainable equilibrium under normal operation and also ensure that the platform quickly arrives at a stable equilibrium when genuine disturbances take place (e.g. discontinued services or increase in demand for a particular service). Additionally, it must be robust enough to prevent or foil the effect of malicious attacks on the incentive scheme either by members on the OCCS platform or external entities. The collection of incentive scheme must also be flexible enough to adapt to changing user behavior on the platform. Ideally the individual schemes must be modular so that a combination of them can be applied simultaneously to achieve specific effects depending on the observed characteristics of their effects on the platform. This is important so that it will not be necessary to continually call for the redesign of the incentive scheme.

### C. *Model of Incentive Schemes for OCCS*

This section begins with a detail description of the system model. It is followed by the adopted notations together with the utility payoff matrix that will be used throughout the rest of the paper. Two models of the type of games namely non cooperative and cooperatives games are presented and the existence of pure strategy Nash-Equilibrium shown for them.

#### 1) *System Model*

Each service or group of substitutable services on the platform constitutes a separate game. Multiple games are being played simultaneously. A member participates in a particular game by either contributing resources to the service, using resources of the service, or express interest in the service by subscribing to it. A user may play in multiple games at the same time. We however assume players resource contribution or usage on one service is not dependent on that of other services and hence the games are independent. Their strategies in a particular game are therefore independent of those in others being played concurrently.

Service contribution and utilization are continuous in time. We however make analysis on an arbitrary time slot (e.g. 1hr, 1day etc. depending on the nature of the service and the corresponding billing model). The games are thus inherently repeated games. The time horizon consist of the active slot period corresponding to the time slot for which analysis and payoffs are going to be made, and the preparation period. All time slots preceding an active slot and has not already expired are potentially part of the preparation period for that active slot. Resources contributed towards a service during a particular slot are immediately available in the next time slot unless otherwise specified by the contributor when the resources will be available, in which case the OCCS platform does not consider it a resource until the specified time. Potential users of a service explicitly specify when they want to use the service and hence the active slot(s) they are interested in.

### 2) *Notation*

We have a nonempty, finite set I of $n \in \mathbb{N} \equiv \{1, 2, 3, \ldots\}$ players in a game. Contributable resources incur a base maintenance cost of $m_i$ whether or not they are contributed and also incur additional utilization cost $u_i$ when they have been contributed and are being utilized by other players. Players receive a certain benefit $b_i$ from utilizing a service e.g. the benefit they obtain in using the service for providing services to their customers. To clarify the various costs involved in the discussion of the notations above we use a simplified scenario where player **A** has contributable resources to a service for which player **B** has need of and hence is a potential user of this resource. If player **A** chooses to contribute the resource, then resource $R_a$ is available to the OCCS platform which if player **B** decides to use will pay a cost $C$ to the platform of which $\alpha C$ is credited to player **A**. In this case player **A** also incurs a utilization cost $u_a$ in addition to the maintenance cost $m_a$ while player **B** derives a benefit $b_b$ from the usage of this resource.

Table 1 show the payoffs $\mu(s)$ on the OCCS platform where the row player is assumed to have a contributable resource while column player has need of such a resource. This scenario is also depicted in figure 1. The payoffs in the table are for per unit capacity contributed or used. Thus the actual credit points $\varphi_i$ obtained or debit incurred on a particular service (game) during the active slot under consideration is the product of per unit payoff and the quantity of resource contributed toward or utilized on that service. $\varphi_i(s) = q_i \; \mu_i(s)$ where $q_i$ , is the quantity contributed or requested for use by player $i$ and $\mu_i(s)$ is it's per unit payoff. The **Off** state in Table 1 for the row player indicates that though it has contributable resources, it chooses not to offer them to the OCCS platform for the active time slot under consideration. Similarly the **Off** state for the column player indicates that though it has need of that particular resource it chooses not to use this service in this active time slot under consideration. It can be seen from table 1 that resource contributors are only awarded OCCS credit points when their resources have been utilized by others. This will ensure that resource contributors are encouraged to contribute only resources that other users on the platform will find useful to use thereby encouraging cooperation between the contributors of resources and the corresponding potential users of these resources.

The role of parameter α is to determine the importance (1-α) attached to the administration of the OCCS governance platform itself. The resources required for the management of the OCCS platform can be viewed in one of two ways. One view will be to treat it as any other service on the platform in which case α can be set to 1; on the other hand it could be argued that since the whole concept collapses without the OCCS platform then it is a critical service which needs special attention.

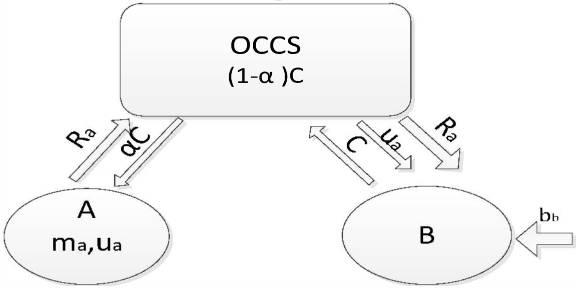


Figure 1: Scenario of member A having a contributable resource to a service of which player B have need of and is therefore a potential user of this resource.

TABLE 1: UTILITY PAYOFFS ON OCCS PLATFORM

| Strategies | Use | Off |
|---|---|---|
| **Supply** | $\alpha C$ - $m$ - $u$, $b$ - $C$ | - $m$,0 |
| **Off** | - $m$,0 | - $m$, 0 |

Row player is assumed to have a contributable resource while column player has need of the service. $0 < \alpha \leq 1$

The strategy $s_i$ of player $i$ is a non-negative real valued function of the quantity $q_i$ that the player chooses to supply or request to use and the reported type $\hat{\theta}_i$ of its true type $\theta_i \in \Theta_i \rightarrow \mathbb{R}_+$ .The set of types $\theta$ of all players is given by $\theta = (\theta_1, \theta_2, \ldots \theta_I)$ where

$$\theta_i = \begin{cases} m_i + u_i \;, \; i \in supply \\ \\ b_i \;, \; i \in use \end{cases}$$

$$s_i = (q_i, \hat{\theta}_i) \rightarrow \mathbb{R}_+^2 \qquad (1)$$

### 3) *Game Models*

Two types of game models are considered; the non-cooperative game and the cooperative game models.

The goal of a player in the non-cooperative game model is to maximize only its own profit.

$$max.q_i\mu_i(s) \quad s.t.\ q_i \leq K_i,\ K_i \text{ is the capacity of } i. \qquad (2)$$

In the cooperative game, the goal of the player is to maximize its own profit together with that of its collaborating partners $P(i)$.

$$max.\ q_i\mu_i(s) + \sum_{j\in P(i)} q_j\mu_j(s) \quad s.t.\ \ q_i \leq K_i$$

$$and \forall j, q_j \leq K_j \qquad (3)$$

Though the OCCS platform does not require members to explicitly establish ties with other members, over time such ties may evolve due to the incentive compatibleness of the incentive schemes as is discussed later, whereby player $i$ identifies key members on the platform whose strategies has significant effect on its own payoffs and therefore will want to guard their interest. The relationship in this case in not necessarily mutual, that is for two players $i$ and $j$

$$j\in P(i) \nRightarrow i\in P(j)$$

A second possibility on the other hand is the scenario where two or more members decide to collaborate for their collective interest; these could be a group of contributors or resource users of a service or a mixture of both. In this case

$$j\in P(i) \Rightarrow i\in P(j)$$

### 4) *Existence of Equilibrium*

The strategy $s_i$ of player $i$ is a non-negative real valued function of the quantity $q_i$ that the player chooses to supply or request to use and the reported type $\hat{\theta}_i$ of its true type $\theta_i \in \Theta_i \to \mathbb{R}_+$

$$s_i = \left( q_i, \theta_i \right) \to \mathbb{R}^2_+$$

This lies within the closed ball $\mathcal{B}_i \subset \mathbb{R}^2_+$ which is defined by $q_i \leq K_i$ , $\theta_i \in \Theta_i \leq \Theta_{ki}$

We now define $s_{-i}$ as the strategies of all other players other than *i*. This is a vector of the individual strategies $s_{i'}$ $i' \neq i$ . We now formulate *theorem1* which shows the existence of a Nash-equilibrium.

***Theorem1***: for both the non-cooperative and the cooperative game models there exists at least one pure strategy Nash equilibrium, i.e. a set of strategies $\bar{s}_i$ such that

$$\varphi_i(\bar{s}_i, \bar{s}_{-i}) \geq \varphi_i(s_i, \bar{s}_{-i}) \quad for\ i \in I,\ s_i \in \mathcal{B}_i$$

*Proof*: The proof of the theorem is based on Kakutani's fixed point theorem [12] which has been used by a number of authors in the proof of the existence of Nash-equilibrium [13], [14]. The proof is shown in the Appendix1.

## D. *Incentive Schemes*

From the perspective of the OCCS platform to ensure the sustainability of the network through encouraging contribution of resources and the efficient utilization of these resources, the objective is to maximize the total utility of every service on the platform subject to the constraints that for each service the utilized resources are less or equal to the contributed resources. The social choice function $f : (S_1,\Theta_1)x(S_2,\Theta_2)\ldots x(S_I,\Theta_I) \to O$ choses an outcome $f(s,\theta) \in O$ given the types $\theta = (\theta_1, \theta_2, \ldots \theta_I)$

$$f(s,\theta) = max.\sum_{i=1}^{n}\varphi_i = max. \sum_{i=1,\ i\in supply}^{n} q_i\mu_i(s,\theta) + \sum_{i=1,\ i\in use}^{n} q_i\mu_i(s,\theta)$$

$$s.t.\, q_i \leq K_i \ \forall i$$

$$and\ L = \sum\nolimits_{i=1,\ i\in use}^{n} q_i \leq \sum\nolimits_{i=1,\ i\in supply}^{n} q_i = G \qquad (4)$$

Mechanism design is the sub field of microeconomics and game theory that considers how to implement good system-wide solutions to problems that involve multiple self-interested agents each with private information about their preferences. A social choice is an aggregation of the preferences of the different participants towards a single joint decision which is literally the common good of the participants as a whole in contrast to their individual interests. So in our case on the OCCS platform, the social choice is to have resources offered by members to the platform and have these resources used efficiently (not abused by other members). The social choice function $f(s,\theta)$ is therefore to maximize the total utility payoff of all services on the OCCS platform which will ensure that the desired social choice is achieved.

For notational simplicity we drop $\theta$ as $s_i = (q_i, \hat{\theta}_i)$ and the reported type $\theta_i$ is a function of agent $i$'s actual type $\theta_i$ . Equation (4) is rewritten to be

$$f(s) = max.\sum\nolimits_{i=1,\ i\in supply}^{n} q_i\mu_i(s) + \sum\nolimits_{i=1,\ i\in use}^{n} q_i\mu_i(s)$$

$$s.t.\, q_i \leq K_i \ \forall i \qquad (4b)$$

This reduces to an optimization problem in which the cost *C* that maximizes the total utility payoff of all the players is computed and then resources allocated. Equation (4) is a combination of linear functions and any suitable optimization algorithm such as Linear Programming can be applied in solving it.

### 1) *Variations of the Base Scheme*

The scheme in equation (4) is in our context the simplest form of incentive scheme which is based on the utility payoffs in table 1 that ensures that service resource contributors only get credit points when their resources have been used by interested members. This will form the base scheme. This section will present variations of this base scheme that add new constraints to it that seek to achieve specific features of the incentive mechanism. These new schemes are the Dominant Strategy Scheme (DSS), Equi-Profit Scheme (EPS) and Dominant Equi-Profit Scheme (DEPS).

*a) Dominant Strategy Scheme*

This scheme introduces new constraints on the base scheme in (4) so that the set of contributors collectively and the set of utilizers of the resources collectively have dominant strategies. In game theoretic terms a player is said to have a dominant strategy $s_i' \in S_i$ if $s_i'$ gives player $i$ a higher expected utility payoff than does every other $s_i \in S_i$ for every possible deleted pure strategy profile $\boldsymbol{s}_{-i} \in \boldsymbol{S}_{-i}$ which the opponents could play. That is if player $i$ has contributable resources it will always choose to contribute if $\alpha C - u_i \geq 0$; and will always use resources if $b_i - C \geq 0$ when it has need of such a service. The dominant strategy scheme's constraint therefore is that

$$\frac{\sum_{i=1,i\in supply}^{n} q_i\left(\alpha C - u_i\right)}{G} \geq 0 \text{ and } \frac{\sum_{i=1,i\in use}^{n} q_i(b_i - C)}{L} \geq 0$$

(5)

It must be noted that equation (5) does not attempt to achieve a dominant strategy equilibrium in the classical game theoretic sense as all agents likely will not have a dominant strategy but rather it is the set of contributors collectively and the set of utilizers collectively that must have a dominant strategy.

*b) Equi-Profit Scheme*

The equal profit scheme also introduces a new constraint on the base scheme in (4). This constraint is that the average per unit utility payoff for contributors of resources is equal to that of the utilizers of these resources. Let

$$\varphi_u = \sum_{i=1,i\in use}^{n} q_i \mu_i(s) = \sum_{i=1,i\in use}^{n} q_i(b_i - C) \text{ and}$$

$$\varphi_s = \sum_{i=1,i\in supply}^{n} q_i \mu_i(s) = \sum_{i=1,i\in supply}^{n} q_i(\alpha C - m_i - u_i)$$

$$\text{Then } \frac{\varphi_s}{G} = \frac{\varphi_u}{L} \tag{6}$$

The scheme seeks to ensure fairness towards both contributors of resources and their utilizers thereof, and thereby guarding against any single player or collaborating players of a particular category teaming up against the other category. Hence this should force collaboration between suppliers of the resources and the users of these resources.

*c) Dominant Equi-Profit Scheme*

This scheme combines both of the constraints of dominant strategy and the equi-profit schemes to the base scheme in (4) such that the set of contributors of resources and the set of utilizers of these resources have dominant strategy; and additionally the average per unit utility payoff is the same for both contributors and utilizers of the service.

*2) Advanced Schemes*

In addition to the variations in the base scheme, we have two schemes also targeted at achieving specific features in the desired properties of the incentive mechanism. However unlike the variations of the base scheme, they do not add constraints but apply game theory concepts in achieving this.

*a) Discount Factor Scheme*

This scheme introduces the consideration of a discount factor $(0 < \delta < 1)$ element to the variations of the base schemes. In game theory the discount factor denotes how much a future payoff is valued at the current period. On the OCCS platform it is also a measure of a player's perception of the continuity of both the platform and the services that it is interested in. For a contributor of resources to a particular service, $\delta_i$ is an indication of the value placed on OCCS credits points obtained now as compared with when it could make use of these credits points later on the platform when it is interested in using a service. On the other hand for a player currently using a service, it potentially may never need to provide services to compensate the platform for the debits incurred now. If it even does, it is similar to obtaining the service on high purchase.

The discount factor can be applied to any of the base schemes discussed above. $\varphi_u$ and $\varphi_s$ in (6) become

$$\varphi_u = \sum_{i=1,i\in use}^{n} q_i(b_i - \frac{C}{\delta_i}),$$

$$\varphi_s = \sum_{i=1,i\in supply}^{n} q_i(\alpha\delta_i C - m_i - u_i)$$

*b) Stochastic Scheme*

The stochastic scheme allows for flexibility in the specification of the maintenance cost $m_i$ and utilization cost $u_i$ for contributors of resources and the benefits $b_i$ for the utilizers of resources. This scheme is useful in specifying these parameters as random variables for two reasons. The first is that utilizers of resources on the platform may be uncertain about the specific value they derive from the resource usage and contributors may also be uncertain about their maintenance and utilization costs. Secondly it may be needful for the OCCS platform to automatically predict these parameters based on the history of members on the platform and their resource contribution and utilization patterns. Thus in addition to offering flexibility in specifying these parameters it can also guard against members specifying arbitrary values for these parameters.

The stochastic scheme together with the discount factor scheme discussed above lays the foundation of our future work on performing repeated game analysis and dynamic mechanisms design for incentive schemes on the OCCS platform.

## IV. Resource Allocation

The outcome $o$ of a game is the resource allocation function $g(s)$ and the transfer function $t(s)$. The resource allocation function $g(s)$ allocates resources to resource demanders and assigns these allocations to contributions. The transfer function $t(s)$ credits OCCS points to resource contributors and debits utilizers of these resources according to the allocations made.

### A. The Resource Allocation Process

The resource allocation process has an inbuilt robustness scheme that guard against the exploitation of the resource allocation process by utilizers of resources with transient membership – that is member with the intension of being on the platform for only a few active time slots. It must be noted that the application of game theory in modeling the incentive mechanisms would have naturally thwarted this had it not been that one of the desired features of the schemes as stated in Section III A.4 is that members can be allocated recourses even when their OCCS credit is zero or negative.

The resource allocation process redefines the set of resource demanders (***use)*** to be only genuine utilizers of resources. It applies various criteria in determining the genuine utilizers. For example a user could potentially exploit the resource allocation process by specifying arbitrarily high $b_i$ thereby always getting resources allocated to it for which it may never have to later compensate the platform for. This is guarded against for example by requiring that reported $b_i$ by a resource demander in a particular active time slot, be less than twice the average benefits $\bar{b}$ indicated by all utilizers of resources for this particular service during this active time slot, in order for it to be among the set of genuine users.

Resources are allocated to maximize the total allocated payoffs. First the amount of allocatable resources is computed as the minimum of the total supplied resources and the total demanded resources by those interested in using the service. The set of ***use*** is then sorted in descending OCCS accrued points and then by profitability $(b_i - C)$. Resources are then allocated while the amount of allocated resources is less than the allocatable limit. Next the set of contributors (***supply)*** is arranged in descending profitability $(\alpha C - m_i - u_i)$ and demands from resource utilizers assigned to them while the allocated supply is less than allocatable.

### B. Evaluation of the Incentive Mechanisms

#### 1) Budget Balance

We show weak budget balance of the mechanisms by showing that the net transfer to the mechanisms is non-negative at equilibrium $\sum_i t_i(s^*) \geq 0$

$$\sum_i t_i(s^*) = \sum_{i=1, i \in use}^{n} t_i(s^*) + \sum_{i=1, i \in supply}^{n} t_i(s^*)$$

$$\sum_i t_i(s^*) = \sum_{i=1, i \in use}^{n} q_i^* C + \sum_{i=1, i \in supply}^{n} (-\alpha C q_i^*)$$ where $q_i^*$ is the quantity of resource allocated or assigned to a player $i$ at equilibrium.

$$\sum_i t_i(s^*) = C\left(\sum_{i=1, i \in use}^{n} q_i^* + \sum_{i=1, i \in supply}^{n} (-\alpha q_i^*)\right)$$

But $\sum_{i=1, i \in use}^{n} q_i^* = \sum_{i=1, i \in supply}^{n} q_i^* = Q^*$ hence

$$\sum_i t_i(s^*) = CQ^*(1-\alpha) \geq 0 \text{ , } since\, C \geq 0, Q^* \geq 0, 0 < \alpha \leq 1$$

#### 2) Individual-Rationality

To show individual rationality [11] [15] [16] (actually ex-post individual rationality), we show that for all agents $i \in I$ the total utility $\varphi_i^*$ of agent $i$ in the equilibrium outcome of the mechanism is always greater or equal to the agent's utility $\hat{\varphi}_i$ for not participating in the mechanism. We first show for $i \in use \subset I$ then after we show for $i \in supply \subset I$.

For $i \in use, \hat{\varphi}_i = 0, \varphi_i^* = q_i^*(b_i - C^*)$

The allocation function $g(s)$ maximizes $\sum_{i=1}^{n} \varphi_i$ and decides on $q_i^*$ based on $(b_i - C^*)$. $g(s)$ sets

$$q_i^* = \begin{cases} q_i^* \geq 0, if\,(b_i - C^*) \geq 0 \\ q_i^* = 0, if\,(b_i - C^*) < 0 \end{cases} \quad hence\, \varphi_i^* \geq 0$$

Similarly,

for $i \in supply, \hat{\varphi}_i = -m_i, \varphi_i^* = q_i^*(\alpha C^* - m_i - u_i)$,

$g(s)$ sets

$$q_i^* = \begin{cases} q_i^* \geq 0, if\,(\alpha C^* - m_i - u_i) \geq 0 \\ q_i^* = 0, if\,(\alpha C^* - m_i - u_i) < 0 \end{cases} \quad hence\, \varphi_i^* \geq 0 > \hat{\varphi}_i$$

#### 3) Incentive Compatibility and Allocative Efficiency

In an incentive compatible mechanism the equilibrium strategy profile $s^* = (s_1^*, s_2^*, \ldots s_I^*)$ has every agent reporting its true preference to the mechanism at equilibrium. We prove incentive compatibility [17] of our mechanisms by showing that truth revelation is equilibrium (a pure strategy Nash equilibrium) of the games induced by the mechanisms; and that the outcome rule $g(s)$ implements the social choice function $f(s)$.

*Proof:* We proved in section III.C.4 as is shown in appendix1 the existence of a pure strategy Nash equilibrium for both the non-cooperative and cooperative games induced in the mechanisms. In the mechanism implementations the allocation function $g(s)$ maximizes $\sum_{i=1}^{n}\varphi_i$ (and is therefore Allocative efficient) and decides on the allocated resource $q_i'$ for resource demanders based on $(b_i - C)$ and the assigned resources to contributors $q_i'$ based on $(\alpha C - m_i - u_t)$. Thus $g(s) = f(s)$; the outcome rule is precisely the social choice function.

## V. Conclusion and Future Work

This paper has looked at the design of incentive schemes that encourage the contribution of resources to the OCCS platform as well as the efficient usage of such resources. Game theory has been employed to model and design the incentive schemes with two game models presented. The existence of a pure strategy Nash equilibrium for both the cooperative and non-cooperative games has been shown. Three base incentive schemes have also been presented. These schemes are the Dominant Strategy Scheme (DSS), Equi-Profit Scheme (EPS) and Dominant Equi-Profit Scheme (DEPS). We performed analytical evaluation of our incentive schemes and conclude that the schemes meet the desired properties of budget-balance, ex-post individual rationality, incentive compatibility, allocative efficiency, robustness, and flexibility to accommodate changing user behavior on the platform.

Though these incentive schemes have been designed for the OCCS platform, they can also be applicable to a general incentive and resource allocation problem in which the service contributors will be one or more commercial cloud service providers servicing a collection of clients with their spare capacities on which they put no fixed price tag.

As was stated in the systems model in Section III.C.1, the service contribution and utilization are continuous in time. We however made analysis on a single time slot for which payoffs are to be made. Since the real world system induces inherently repeated games, the stochastic scheme and the discount factor scheme discussed in Section III.D.2 lay the foundation of our future work on performing repeated game analysis and dynamic mechanisms design for incentive schemes on the OCCS platform. We will also consider these further works in the context of a decentralized OCCS platform.

Finally, this work has provided the foundation for the implementation of Pseudo Service Level Agreement (Pseudo SLA) on the OCCS platform which is discussed later in the sequel to this paper on “Trust engineering and Pseudo SLA in Opportunistic Cloud Computing Services”. It is evident from the title of the sequel to this paper that we consider data privacy, security and trust as major issues that OCCS and cloud computing in general has to address. We are therefore currently working on trust engineering in cloud computing and how to adapt it for the OCCS environment.

## References

[1] Fernando Costa, Luis Silva, and Michael Dahlin, "Volunteer Cloud Computing: MapReduce over the Internet," in *2011 IEEE International Parallel & Distributed Processing Symposium*, 2011, pp. 1855-1862.

[2] Eric Kuada and Henning Olesen, "A social network approach to provisioning and management of cloud computing services for enterprises," in *CLOUD COMPUTING 2011 : The Second International Conference on Cloud Computing, GRIDs, and Virtualization*, Rome, Italy, Sep.2011, 2011, pp. 98 - 104.

[3] Martin J Osborne and Ariel Rubinstein, *A course in game theory*, 1st ed. Cambridge, Massachusetts, London, England: Massachusetts Institute of Technology Press, 1994.

[4] Kevin Leyton-Brown and Yoav Shoham, *Essentials of game theory: A concise, multidisciplinary introduction*, Ronald J Brachman and Tom Dietterich, Eds.: Morgan & Claypool Publishers, 2008. [Online]. http://www.morganclaypool.com/doi/pdf/10.2200/S00108ED1V01Y200802AIM003

[5] Eric Kuada, Henning Olesen, and Anders Henten, "Public policy and regulatory implications for the implementation of opportunistic cloud computing services for enterprises," in *9th International Workshop on Security in Information Systems*, Wroclaw, 2012, pp. 3-13, http://www.iceis.org.

[6] Wayne Jansen and Timothy Grance, "Guidelines on security and privacy in public cloud computing," Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, Draft Special Publication 800-144 800-144, 2011.

[7] Danilo Ardagna, Barbara Panicucci, and Mauro Passacantando, "A game theoretic formulation of the service provisioning problem in cloud systems," in *WWW 2011 – Session: Monetization II*, Hyderabad, India, 2011. [Online]. http://www.www2011india.com/proceeding/proceedings/p177.pdf

[8] Zhen Kong, Cheng-Zhong Xu, and Minyi Guo, "Mechanism design for stochastic virtual resource allocation in non-cooperative cloud systems," in *2011 IEEE 4th International Conference on Cloud Computing*, 2011, pp. 614 - 621. [Online]. http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=06008762

[9] Guiyi Wei, Athanasios V Vasilakos, Yao Zheng, and Naixue Xiong, "A game-theoretic method of fair resource allocation for cloud computing services," *Springer*, vol. J Supercomput, no. 54, pp. 252 - 269, 2010.

[10] Chiranjeeb Buragohain, Divyakant Agrawal, and Subhash Suri, "A Game Theoretic Framework for Incentives in P2P Systems," in *Third International Conference on Peer-to-Peer Computing (P2P 2003)*, 2003, pp. 48-56.

[11] Noam Nisan, "Introduction to mechanism design (for computer scientists)," in *Algorithmic Game Theory*. New York, USA: Cambridge University Press, 2007, ch. 9, pp. 209-241, 978-0-521-87282-9.

[12] Shizuo Kakutani, "A generalization of Brouwer's fixed point theorem," *Duke Mathematical Journal*, vol. 8, no. 3, pp. 457-459, 1941.

[13] John F Nash, "Equilibrium points in n-person games," in *Proceedings of the National Academy of Science(NAS)*, vol. 36, 1950, pp. 48-49.

[14] Juan Pablo Torres-Martinez, "Fixed points as Nash equilibria," *Fixed Point Theory and Applications*, vol. 2006, pp. 1-4, October 2006, Article ID 36135.

[15] David C Parkes, "Online mechanisms," in *Algorithmic Game Theory*. New York, USA: Cambridge University Press, 2007, ch. 16, pp. 411-439.

[16] Danish Garg, Y Narahari, and Sujit Gujar, "Foundations of mechanism design: A tutorial Part 2 – Advanced concepts and results," *Sadhana, Indian Academy Proceedings in Engineering Sciences*, vol. 33, no. 2, pp. 131-174, April 2008.

[17] Matthew O Jackson, "Mechanism theory," in *Optimization and Operations Research*, Ulrich Derigs, Ed. Oxford, UK: Encyclopedia of Life Support Systems (EOLSS), EOLSS Publishers, 2003, ch. 4, Developed under the Auspices of the UNESCO, http://www.eolss.net.

[18] Gyorgy Dan, "Cache-to-Cache: could ISPs cooperate to decrease peer-to-peer content distribution costs?," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 9, pp. 1469-1482, September 2011, http://doi.ieeecomputersociety.org/10.1109/TPDS.2011.28.

## APPENDICES

### *A. Appendix1*

Lemma 2 (Kakutani): Let $\mathcal{B} \subseteq \mathbb{R}^{|\mathcal{H}|}$, $\mathcal{B}$ compact, convex and non-empty. Let $\mathcal{K}$: $\mathcal{B} \rightrightarrows \mathcal{B}$ be a correspondence (non-empty valued). Such that $\mathcal{K}$(b) is convex ∀ b∈$\mathcal{B}$. assume moreover that $\mathcal{K}$ has closed reduced graph. Then there is a fixed point for $\mathcal{K}$, i.e. $\exists b \in \mathcal{B}$ s.t. $b \in \mathcal{K}\left(b\right)$ [12]. The following proof of Theorem 1 consists of showing that the conditions of Lemma 2 are satisfied. The proof is based on [18] which is also based on [13].

*Proof:* (**Theorem 1**) $\mathcal{B}_i$ is non-empty because for $\mathcal{K}_i > 0$ there is at least one feasible strategy. $\mathcal{B}_i$ is closed and bounded, hence it is compact. Furthermore, $\mathcal{B}_i$ is convex due to the capacity constraint being linear (2). The payoff function that player $i$ tries to maximize is continuous in $q_i$ and $\mu_i\left(s\right)$ in both the non-cooperative game (2) and the cooperative game (3). And it is quasi-concave as it is linear. We define the set valued best response function of player $i$.

$$\mathcal{K}_i(s_{-i}) = \left\{s_i \in \mathcal{B}_i | \varphi_i(s_i, s_{(-i)}) \geq \varphi_i(s_i', s_{(-i)}) \forall s_i \in \mathcal{B}_i\right\}$$

The set $s_i = \mathcal{K}(s_i)$ is non-empty because $\varphi_i$ is continuous and $\mathcal{B}_i$ is compact. It is convex due to the quasi-concavity of the payoff function $\varphi_i$ . The graph of $\mathcal{K}_i$ is closed because of the payoff functions are concave.

Let us define $\mathcal{B}$ = x$_{i\in I}$ $\mathcal{B}_i$ and the correspondence $\mathcal{K}$: $\mathcal{B} \rightrightarrows \mathcal{B}$ as $\mathcal{K}$ = x$_{i\in I}$ $\mathcal{K}_i$. $\mathcal{B}$ is hence compact, convex and non-empty, and $\mathcal{K}$ is convex, non-empty valued and has closed reduced graph. Hence, due to Kakutani's theorem $\mathcal{K}$ has a fixed point such that $s_i = \mathcal{K}(s_i)$ , which proves the existence of a Nash-equilibrium both for the non-cooperative and the cooperative strategies.

## Appendix 3: Paper #3

Kuada, Eric. 2014. “Towards Trust Engineering for Opportunistic Cloud Services: A Systematic Review of Trust Engineering in Cloud Computing”, Journal of Grid Computing.

# Towards Trust Engineering for Opportunistic Cloud Services:
## A Systematic Review of Trust Engineering in Cloud Computing

Eric Kuada
Department of Electronic Systems
Aalborg University, Copenhagen, Denmark
kuada@cmi.aau.dk

***Abstract*** -The systematic review methodology has been employed to review trust related studies in cloud computing. It was observed that trusted computing technologies and reputation based approaches are the main approaches to trust engineering in cloud computing. Also, trusted third party approaches and the deployment model play a significant role in enhancing trust between service providers and consumers. It was observed that the concept of trust is used loosely without any formal specification in cloud computing discussions and trust engineering in general. As a first step towards addressing this problem, we have contextualized the formal trust specification in multi-agent environments for cloud computing.

***Keywords***- security engineering; trust engineering; trust in cloud computing; trust modeling.

## 1 INTRODUCTION

This introductory section begins with the motivation for undertaking this study. Next, a background to the need for trust engineering in cloud computing is provided. An overview to opportunistic cloud services (which is the foundation for the motivation of this study) is also given. Since the methodological approach to this study is systematic literature review, the section ends with a brief introduction to systematic literature reviews and the processes involved in conducting such a review.

### 1.1 Motivation

We have over the past three years been working on the feasibility of Opportunistic Cloud Services (OCS) for enterprises[1] [2]. One of the major challenges that such a platform faces is data security and trust management issues. In order to design and develop a trust management system for OCS platforms, we needed to review the current trust engineering issues in cloud computing. It was decided that we needed to perform a systematic literature review on this topic because since the OCS concept is itself new, any trust design models of its subsystems must be guided by exhaustive knowledge of the state-of-the-art in the field. The rigorous methodological approach offered by systematic literature reviews will ensure that the topic is adequately covered. The objective of this paper is therefore to provide state-of-the-art knowledge on trust engineering concepts and models in cloud computing.

### 1.2 Background to Trust in Cloud Computing

Cloud computing is essentially the packaging of traditional Information Technology infrastructure and software solutions such as storage, CPU, network, applications, services, etc., as virtualized resources and delivered by a service provider to its customers as an on-demand pay-per-use self-provisioned service, which is normally offered through a web portal over a network such as the Internet[3] [4] [5]. While cloud service providers pledge to preserve data privacy, the current Software as a Service (SaaS) architecture makes it difficult to provide any assurance that the software in the Cloud will not be able to make copies or redistribute the data it used[6] . Secondly, the Cloud model is based on two key characteristics: multi-tenancy, where multiple tenants share the same service instance, and elasticity, where tenants can scale the amount of their allocated resources based on current demands. Although both characteristics target improving resource utilization, cost reduction, and service availability, these gains are threatened by multi-tenancy security implications. The sharing of applications that process critical information without sufficient proven security isolation, security SLAs or tenant control, results in "loss-of-control" and "lack-of-trust" problems[7].

Apart from these consumer concerns, cloud architectures also introduce new classes of security risks and attacks over the resources of cloud service providers. These include poisoned virtual machines, attacks against the cloud service provider's management console, attacks based on knowledge of default security settings, abuse of billing systems, and data leakage via uniform resource locators. Cloud service providers still do not currently have sufficiently robust technical solutions that can protect their cloud resources from harmful malware, virus infection, botnets, distributed denial of service attacks, or other types of cyber-attacks. Furthermore, there is no effective mechanism to help cloud users evaluate the security measures of their service providers and ensure the protection of their data while taking into consideration industry standards or personal preferences [8].

### 1.3 Opportunistic Cloud Services

Opportunistic Cloud Services (OCS) is a social network approach to the provisioning and management of cloud computing services for enterprises. OCS is about enterprises leveraging free cloud services to meet their business needs without having to pay or paying a minimal fee for these services [9][10]. An OCS network is a social network of enterprises collaborating strategically for the contribution and usage of cloud services without entering into any business agreements[1]. Members normally will package only their spare IT resources and make them available as Cloud services on the OCS platform so that others interested can utilize them. Since no business agreement and hence no Service Level Agreement (SLA) exist between the service providers and the potential users of their services, service consumers do not enjoy the level of support (in terms of quality of service, reliability, availability, security, billing transparency, etc.) that commercial cloud service providers offer to their clients. Considering the fact that commercial cloud service providers are finding it extremely challenging to provide such a support, coupled with having to provide adequate transparency in their management processes, the OCS platform more so needs a well-crafted and soundly engineered trust management system in order to make resources on the platform suitable for business use.

### 1.4 Systematic Literature Reviews

A systematic literature review is a means of identifying, evaluating and interpreting all available research - that are known to the researcher - and relevant to a particular research question, topic area, or phenomenon of interest [11][12]. It is a systematic, explicit, comprehensive, and reproducible method for identifying, evaluating, and synthesizing the existing body of completed and recorded work produced by researchers, scholars, and practitioners on a specific topic of interest [13]. The accumulation of evidence through secondary studies can be very valuable in offering new insights or in identifying where an issue might be clarified by additional primary studies. The systematic literature review process consists of three main stages - namely inputs, processing, and outputs [14]. The eight step guide of [13] to conducting systematic literature review are: purpose of the literature review, protocol and training, searching for the literature, practical screen, quality appraisal, data extraction, synthesis of studies, and finally writing the review. They recommend all these steps are essential for a review to be scientifically rigorous. According to[11] the stages in a systematic literature review can be summarized into three main phases: planning the review, conducting the review, and reporting the review. The stages associated with *planning the review* are: identification of the need for a review, specifying the research questions, and developing a review protocol; the stages associated with *conducting the review* are: identification of relevant existing research, selection of primary studies, study quality assessment, data extraction and monitoring, and data synthesis; and finally, the stages associated with *reporting the review* are: specifying dissemination mechanisms, and formatting the main report.

## 2 METHODOLOGY

We adopt a blend of the guidelines of [11] and [13] because after a careful analysis, we consider both guides to be suitable for our purpose; and it was evident that their main individual stages are in agreement and refer to the same concepts with slightly different tagging.

### 2.1 Planning the Review

The main activities involved in planning the review are specifying the objectives of the study, specifying the research questions, developing and evaluating the review protocol, and justifying the need for the study.

#### 2.1.1 Need for the Study

There have been efforts on surveys on security issues in cloud computing [15] [16] but not on trust engineering. Also, even though security is a key element of trust, these studies are not systematic reviews and those that attempt a systematic review such as [17] focus only on security; so to the best of our knowledge, this study is the first attempt of summarizing the body of knowledge on trust engineering in cloud computing environments.

#### 2.1.2 Objectives and Research Questions

The first of the main objectives of this study is to provide state-of-the-art knowledge on trust engineering concepts and models in cloud computing. The second objective is to provide a firm grounding for engineering a trust model and trust management system for opportunistic cloud computing services. Based on these objectives, the research questions that are of interest to this study are:

a. What are the main approaches towards trust engineering in cloud computing?
b. What are the major trust models and trust management systems for cloud computing?
c. What are the objectives of the identified primary studies and in what contexts are these trust management systems being developed?

#### 2.1.3 Review Protocol

A review protocol specifies the methods that will be used to undertake a specific systematic review. The components of a protocol include all the elements of the review plus some additional planning information such as the project timeline [11]. The entire methodology section in this paper gives a summary of review protocol that has been applied in undertaking this study. The review protocol has been under constant re-evaluation to ensure that the applied search strings are derived from the research questions; the extracted data properly address the research questions; and the data analysis procedure is appropriate to answer the research questions.

### 2.2 Conducting the Review

The stages associated with conducting the review are identification and selection of relevant existing primary studies, study quality assessment, data extraction and monitoring, and data synthesis**.**

#### 2.2.1 Search Strategy

The adopted search strategy is to search for keywords in standard metadata (i.e. title, abstract, and author keywords). The keywords derived from our topic "Trust Engineering in Cloud Computing" are trust, engineering, and cloud computing. However, because privacy and security are two major elements in trust in cloud computing, we expanded our search keywords to include them. Additionally, to ensure the search strings are derived from our research questions, we further expand the keywords to include model. We then use a combination of two or more of the resulting keywords as search strings in searching for the primary resources for this study. The resulting search strings are: trust engineering, trust cloud computing, trust model, security in cloud computing, privacy in cloud computing, security engineering, and privacy engineering.

#### 2.2.2 Sources

Considering the topic of this study, the major sources that the search strategy was applied in are IEEE Xplore Digital Library, ACM Digital Library, Google scholar, and Journals of Elsevier and Springer Link. These sources were supplemented with the general Internet and the Aalborg University digital library portal, Primo, which is a portal into well-known research databases.

#### 2.2.3 Practical Screen and Quality Appraisal

Refworks [18] was used as the bibliographic management tool for managing the large number of over five hundred studies resulting from the search process. These were taken through practical screening by reading through their abstracts and those that didn't have relation to our research topic were excluded; leaving about 320 primary studies as our base resources. A second and a third round of reviews were performed to select those that had direct bearing on the research questions. This process yielded about 140 articles that have been included in this study. All these articles were then retrieved and processed for the data extraction phase. This process spanned a period of four months, from September to December 2012. Regular update to the list of articles was made during data extraction and synthesis of studies phases in the subsequent months of undertaking this study.

#### 2.2.4 Data Extraction

NVivo10[19] was the choice of tool for the data extraction phase. Even though we are well aware of other qualitative data analysis tools such as Atlas.ti [20], we did not consider them since the university had license to only NVivo. All these relevant primary studies were manually read. The basic methodological steps of constant comparison for coding in grounded theory [21] [22]were applied in the data extraction process with help of Nvivo in the coding of the data as we read through each article. Furthermore the grounded theory methodology allowed extraction of relevant information (e.g. the major challenges of trust engineering in cloud computing) from the primary studies, even though these were not initially part of the focus of our study and hence did not reflect in our research questions.

#### 2.2.5 Synthesis of Studies

During the synthesis stage, major trends that had developed during the coding process were further investigated by searching for new articles on these specific topics in order to shed more light on them. We followed an iterative process of categorization and reorganization of the extracted data, supplemented with finding new articles that support or weaken the trends being observed. Those categories lacking adequate support and could also not fit naturally under other categories did not merit further analysis and were dropped in our discussions as is presented below.

## 3 ANALYSIS & DISCUSSION

We have followed an iterative process of categorization and reorganization of the extracted data, supplemented with finding new articles that support or weaken the trends being observed, in order to obtain the final headings that are discussed in this section. The main areas covered in our analyses and discussions are:

- Trust, security and privacy challenges in cloud computing
- Focus on trust engineering in cloud computing
- Modeling trust: this deals with the modeling of the concept of trust.
- Trust engineering approaches
- Trust management systems

## 3.1 Trust, Security and Privacy Challenges in Cloud Computing

Though the identification of challenges in cloud computing was not part of the research objectives or questions that were spelt out during the “Planning the Review” stage, it was evident during the data extraction process that it is a paramount issue that needed some attention. The major challenges in cloud computing as were reported by the reviewed papers can be categorized into trust challenges, security challenges, and privacy challenges. This categorization however does not mean the categories are mutually exclusive, as it will be seen later that for example, security and privacy issues impact upon the perceived trust of various entities in a cloud computing marketplace.

### 3.1.1 Trust Challenges

An important issue in cloud computing is the accountability of the resource usage data: who performs the measurement to collect resource usage data – is it the provider, the consumer, a trusted third party or some combination of them? Currently, provider side accountability is the basis for cloud service providers, although, as yet there are no equivalent facilities of consumer-trusted metering as is the case in traditional utility services; rather, consumers have no choice but to take whatever usage data made available by the provider as trustworthy [23].

Another issue concerning trust in cloud computing is that, potential customers of cloud services often feel that they lose the control over their data, and they are not sure whether they can trust the cloud service providers. A survey conducted in 2011 among more than three thousand cloud consumers from six countries, shows that 84 percent of the consumers are concerned about their data storage location and 88 percent of the consumers worry about who has access to their data. Though consumer concerns can be mitigated by using preventive measures for privacy (e.g., demonstrating compliance standards) and security (e.g., secure hypervisors, TPM based servers), at present, cloud providers demonstrate their preventive measures by including related descriptions in the SLAs; assurances and compensations for SLA violations are however not convincing enough for the consumers. Especially, SLAs with vague clauses and unclear technical specifications lead the consumers into a decision dilemma when considering them as the only bases to identify trustworthy providers [24].

A third issue concerning trust in cloud computing is that, the SaaS model gives software providers an unprecedented access to data uploaded by users. At execution time the control of the data is handed over from the user (data owner) to the software provider. Furthermore, the results generated from the software execution, in theory, are under the control of the software provider. This raises a new concern about trust on software providers [6]. Data must be decrypted into memory when performing the computation, even though they can be encrypted during storage and transmission. In this case, the privileged administrators of SaaS providers are able to inspect or modify users’ data and computations. As a result, the users are hesitant to trust the SaaS providers [25].

### 3.1.2 Security Challenges

Possible misuse of customers’ data by cloud service providers is a major challenge in cloud computing. The privileged administrators of cloud service providers are able to inspect, modify, or misapply users’ data and computations. Some of the security challenges facing cloud computing are multi-tenancy security implications, security isolation, cloud service providers’ and customers’ need of modeling and enforcing different security requirements (especially at runtime because security requirements may change over time as new risks emerge), and integrating with different security services. After analyzing the cloud computing model security problem, and information security management systems (ISMS) process, [26] has identified the following key problems:

- ♦ Each stakeholder has their own security management process (SMP) that they want to maintain or extend to the cloud hosted assets.
- ♦ No stakeholder can individually maintain the whole security process of the cloud services because none of them has the full information required to manage security and each one has a different perspective.
- ♦ Multi-tenancy requires maintaining different security profiles for each tenant on the same service instance.
- ♦ No Security SLA is available that can be used to maintain agreements related to cloud assets security.
- ♦ The existing standards such as ISO27000 and FISMA do not map well to the cloud model because these standards consider the SMP from the perspective of the platform/asset owner, not from a service provider perspective.

While there might be a multitude of operating systems (OSs) deployed in a single cloud, the majority of such OSs have not been designed for the Cloud. In particular, traditional logging is process and/or event-based (for a particular user or node). In the Cloud, however, there are no clear user or node barriers; instead, logging should be done with respect to the key assets, i.e., data and information. In terms of OSs, this means data-centric logging. Besides provenance, other key concerns mandating data- centric logging include the need for support of consistency assurance, rollback, recovery, replay, backup, and restoring of data. Such functionality is usually enabled by using operational and/or transactional logs. Such logs have also been proven useful for monitoring of operational anomalies. While these concepts are well established in the database domain, cloud computing's characteristics such as eventual consistency, 'unlimited' scale, and multi-tenancy pose new challenges. In addition, secure and privacy-aware mechanisms must be devised not only for consistency logs but also for their backups, which are commonly used for media/node recovery [27].

Data processing clouds, including Hadoop[28], execute untrusted, user-submitted code on trusted cloud nodes during job processing, and must therefore remain vigilant against malicious mobile code attacks. Virtualization technologies, including trusted hardware, hypervisors, secure operating systems, and trusted VMs are the typical means by which such mobile code is secured. However, a variety of studies have shown that clouds introduce significant new security challenges that make mobile code security a non-trivial, ongoing battle. For example, the Cloud Security Alliance has identified insecure cloud APIs, malicious insiders, shared technology issues, service hijacking, and unknown risk profiles all as top security threats to cloud services [29].

Adopting multi-tenancy with SaaS results in a set of requirements that must be addressed by the SaaS application. Two key requirements in the area of SaaS applications' security engineering have been identified by [7]. The first one is the security isolation among tenants' assets at rest (storage), during processing (in memory), and during transient (among application components or between the application and the tenant site). Secondly, it is required to support enforcement of different security requirements on the same service instance at runtime. Application customization approaches do not fit well with runtime and multi-tenant specification and security enforcement because these security requirements may change over time as new risks emerge.

Data integrity is another major security challenge for cloud computing. It is most often assumed that the underlying storage arrays (similar technologies of which are being employed by cloud service providers), receive, store and retrieve data flawlessly. This assumption is however proven to be false in the past, as evident from the CERN report[30] and other studies[31]. Therefore, prompt detection of integrity violations is vital for the reliability and safety of the stored data in the Cloud [32].

#### 3.1.3 Privacy Challenges

In cloud computing, entities may have multiple accounts associated with a single or multiple service providers (SPs). Sharing sensitive identity information (i.e. Personally Identifiable Information (PII)) along with associated attributes of the same entity across services can lead to mapping of the identities to the entity; and this is leads to privacy loss. The major problems regarding privacy in the Cloud include how to secure PII from being used by unauthorized users; how to prevent attacks against privacy (such as identity theft) even when a cloud SP cannot be trusted; and how to maintain control over the disclosure of private information [33].

As has been indicated by [34], there are situations where cloud service providers themselves invade the privacy of their users, so a cloud service provider is generally not the entity to fully rely on in order to protect one's privacy. Consequently, there is a need for additional external measures to protect a user's privacy. This need has been recognized in several previous approaches for protecting data in the Cloud [35] [36]. However, these approaches suffer from bad usability and require too much effort from the users, as shown for example by Whitten and Tygar [37] and subsequent user tests. There are theoretically many cryptographic mechanisms that would perfectly suit the privacy needs of today's Internet users, but their use is avoided due to a lack of good usability and high effort required. For example, Public Key Infrastructures (PKIs) burden the users with handling cryptographic artifacts. Although there are many efforts to simplify the usage of a PKI ,e. g. [38] [39], the majority of users still shy away from the extra work [34].

### 3.2 Cloud Computing Trust Engineering Focus

We now analyze the main objectives of researchers on trust engineering in cloud computing to determine what trust engineering research has focused on within the past few years. We extract the objectives of selected works of which the objectives had been clearly stated (normally stated in the abstract or in the introductory sections), or can be easily inferred from these sections. We have identified five main research focuses on trust engineering in cloud computing. They are performance and Quality of Service (QoS), security related, access and Identity management, user and provider support on trust management, and billing and accountability. We end the section with some concluding remarks on some of the salient points of these research areas together with the context within which these studies had been carried out.

#### 3.2.1 Performance and QoS

The objective of the trust evaluation model of [40] is to configure the complex set of services dynamically in a cloud environment according to the predictive performance in terms of stability and availability of all services that are to be provided; this is with the aim of allowing a system to configure services dynamically and distribute tasks efficiently in such a way that minimizes task failure and task migration rate.

Success of cloud computing requires that both customers and providers can be confident that signed SLAs are supporting their respective business activities to their best extent. The SLAs currently being used fail in providing such confidence, especially when providers outsource resources to other providers. These resource providers typically support very simple metrics like availability, or metrics that hinder an efficient exploitation of their resources. A resource-level metric for specifying fine-grain guarantees on CPU performance has been proposed by [41].

Due to the dynamic nature of cloud computing, how to achieve satisfactory QoS in cloud workflow systems becomes a challenge. Meanwhile, since QoS requirements have many dimensions, a unified system design for different QoS management components is required to reduce the system complexity and software development cost;[42] has therefore proposed a generic QoS framework for cloud workflow systems. The framework covers the major stages of a workflow lifecycle. It consists of QoS requirement specification, QoS-aware service selection, QoS consistency monitoring and QoS violation handling.

### 3.2.2 Security

The aim of [43] is to provide a system that makes it possible to detect that at least the configuration of the cloud infrastructure -as provided in the form of a hypervisor and administrative domain software- has not been changed without the customer's consent. They present a system that enables periodical and necessity-driven integrity measurements and remote attestations of vital parts of cloud computing infrastructures. The objective of [43] is to tackle the problem of protecting entities using the Cloud from malicious or negligent entities providing the cloud infrastructure. They present the *BonaFides* system for remote attestations of security-relevant parts of the cloud infrastructure, which guarantees to service providers at runtime the detection of unintended or malicious modifications of cloud infrastructure configurations. Their approach does not prevent the cloud infrastructure provider from altering crucial components and subsequently stealing data, but these activities will at least be detected by the cloud consumers.

The objectives of [25] is to provide a trusted SaaS platform (TSP) which will guarantee data security during storage and transmission, and also enforce a trusted execution environment (TEE) that guarantees the confidentiality and integrity of the users' data and computations. The objective of [7] is to provide a security management architecture- Tenant Oriented SaaS Security Management Architecture (TOSSMA) - that allows service providers to enable their tenants in defining, customizing and enforcing their security requirements without having to go back to application developers for maintenance or security. The objective of [32] is to offer a secure cloud storage service architecture with the focus on Data Integrity as a Service (DIaaS) based on the principles of Service-Oriented Architecture and Web services. The approach releases the burdens of data integrity management from a storage service by handling it through an independent third party data Integrity Management Service (IMS); it also reduces the security risk of the data stored in the storage services by checking the data integrity with the help of IMS.

In order to address privacy and security issues, and to incorporate security and trust functionalities that complies with EU and government privacy laws, [44] has presented the Cloud Data Security (CloudDataSec) project that aims to design cloud services adhering to government privacy laws. In particular, they introduced a six-layer security model for cloud computing and three level of security assurance for SMEs to take advantage of. Finally, they proposed Security Management as a Service (SMaaS) modules to enable users to apply necessary security and privacy operations, based on the sensitivity of their data.

The objective of [26] is to introduce a cloud security management framework based on aligning the FISMA standard[45][46] to fit with the cloud computing model; this is with the aim of enabling cloud providers and consumers to be security certified through improving collaboration between cloud infrastructure providers, cloud service providers and service consumers in managing the security of the cloud platform and the hosted services.

### 3.2.3 Access and Identity Management

Because available solutions to identity management in cloud computing use trusted third party (TTP) in identifying entities to service providers, and these solution providers do not recommend the usage of their solutions on untrusted hosts, the objective of [33] is to develop a framework for identity management which is independent of TTP and has the ability to use identity data on untrusted hosts. The objective of [47] is to provide a mechanism (Trust Ticket) of ensuring trust and security in Software as a Service (SaaS). Their Trust Ticket, together with the supporting protocols, is a mechanism that helps a data owner in establishing a link between a cloud service provider and a registered user. In this mechanism, a user first gets registered with a data owner before receiving a Trust Ticket and a secret key from that data owner. Each Trust Ticket is unique and encrypted. On completing the registration of each user, the data owner apprises the cloud service provider of the Trust Ticket.

### 3.2.4 User and Provider Trust Management Support

Due to the vast diversity in the available cloud services, from the customers' point of view, it has become difficult to decide whose services they should use and what the basis for their selection is. Currently, there is no framework that can allow customers to evaluate Cloud offerings and rank them based on their ability to meet the user's QoS requirements. Reference [48] has proposed a framework and a mechanism that measures the quality and prioritizes cloud services. The objective of [24] is to support the customers in reliably identifying trustworthy cloud providers. The objective of [49] is to provide personalized trust management in which the user may play any of the three roles of consumer, broker, or provider. The objective of [50] is to provide decision making guidance to service providers to initialize collaborations by selecting trustworthy partners within the context of a cloud marketplace.

The objective of [51] is to provide a framework that enable trust-based cloud customer and cloud service provider interactions within the context of hybrid cloud computing environments. The objective of [27] is to employ a data-centric, detective approach to provide a framework (TrustCloud) to increase trust, security of data, and accountability in the Cloud at all levels of granularity. The aim of [34] is to provide usable confidentiality and integrity, through their Confidentiality as a Service (CaaS) paradigm for the majority of users for whom the current security mechanisms are too complex or require too much effort.

#### 3.2.5 Billing and Accountability

The objective of [23] is to provide openness and transparency . They propose the notion of consumer–centric resource accounting model such that consumers can programmatically compute their consumption charges of a remotely used service. In particular, the notion of strongly consumer–centric accounting model is proposed that requires that all the data needed for calculating billing charges can be collected independently by the consumer (or a trusted third party, TTP).

According to [8], one of the major security obstacles to widespread adoption of cloud computing is the lack of near-real-time auditability. In particular, near-real-time cloud auditing, which provides timely evaluation results and rapid response, is the key to assuring the Cloud. Their objective is therefore to present strategies for reliable cloud auditing.

#### 3.2.6 Concluding Remarks and Contexts of Studies

Usually, cloud providers provide assurances by specifying technical and functional descriptions in SLAs for the services they offer. The descriptions in SLAs are not consistent among the cloud providers even though they offer services with similar functionality. Customers are not sure whether they can identify a trustworthy cloud provider only based on its SLA. To support the customers in reliably identifying trustworthy cloud providers, [24] has proposed a multi-faceted trust management system architecture for a cloud computing marketplace. The context of [50] is the provision of guidance in the selection of trustworthy partners within a cloud computing marketplace. The context of [51] is to provide a framework that enable trust-based cloud customer and cloud service provider interactions within the context of hybrid cloud computing environments, where resource sharing between multiple Clouds to meet cloud user requirements are enabled by peering arrangements established between the participating Clouds. The context of [40] is the scheduling of resources of services in cloud computing environments by adopting a trust model based on Probabilistic Latent Semantic Analysis (pLSA) which analyzes the history information of each node and allocates reliable resources according to user requests.

Based on the findings from the above, the main arrears of trust engineering research focus has been on quality of service, security, access and identity management, user support on trust management, and accountability in in the context of a cloud computing marketplace . A major observation that I made from the reviewed studies is that the concept of trust is treated loosely without any formal specification or definition in the discussion of trust in cloud computing and trust engineering in general. Formal trust modeling and definitions are however very necessary in ensuring a unified view of the concept of trust in the design and engineering of trust management systems for cloud computing; this therefore deserves more attention from the cloud computing research community.

### 3.3 Modeling Trust

Reference [52] has carried out a survey on the trust management systems implemented on distributed systems with a special emphasis on cloud computing. They reported on several trust models such as CuboidTrust [53] , EigenTrust [54] , Bayesian Network based Trust Management (BNBTM) [55], GroupRep [56] , AntRep[57] , Global Trust[58] [59] , Peer Trust [60], and Trust Ant Colony System (TACS)[61]. These models were mainly proposed for systems like clusters, grids and wireless sensor networks, and have not been used or tested in cloud computing environments. Secondly, these models do not model the concept of trust but rather model practical trust management systems for distributed systems and their algorithms for acquiring and computing trust values.

This section is about the actual modeling of the concept of trust with a special focus on trust in cloud computing. We begin with looking at some definitions of trust and move on to obtaining a formalized model of the definition of the concept of trust in the context of cloud computing environments.

#### 3.3.1 Definitions of Trust

Though there has been some work on trust modeling and trust management systems, and even in the new domain of trust management systems for cloud computing environments [62] [51] [63], the subjective nature of trust has made a solid definition elusive. Researchers have most often used the term loosely in their works; more specifically, a rigorous formal definition has not been applied in most cases. A few of the attempts at the definition of trust in the domain of trust engineering for cloud computing that was found during this study corroborates this observation. Salah and Eltoweissy [49] defined trust as the belief or disbelief of a party that another party, for a said subject of trust, in a given context, has the intent, integrity, results and capability to exhibit a set of acceptable actions in the future, for the welfare of the trusting party. Viriyasitavat and Martin [64] has developed trust definition in the application domain of service workflows. They defined trust as "Trust is a subjective mutual measurable between interacting entities willing to act dependably, securely, and reliably, in a given situation within specific context of a given time". Their definition is an adaptation of that of Olmedilla, et al [65] which states that "Trust of a party A to a party B for a service X is the measurable belief of A in that B behaves dependably for a specified period within a specified context (in relation to service X)". It should be noted that whilst in the domain of service workflows, being able to establish trust in both directions is crucial, as one service may need to verify trustworthiness of a subsequent service before passing information, and the subsequent service perhaps requires trust that an outcome must be originated from a trusted source, the definition of Viriyasitavat & Martin contradicts the generally accepted asymmetry property of trust relationships.

Dellarocas' definition of trust [66] is adopted in this work. Its salient points are summarized below and explained in the context of cloud computing.

The level of trust $T_c^s\left(t_i\right)$ of a service consumer $c$ for a service provider $s$ in the context of a transaction $t_i \in T$ is the a priori probability that the utility of $c$ will meet or exceed its minimum threshold of satisfaction $u_0$ at the end of transaction $t_i$, given $c's$ perceived trustworthiness of service provider $s$. Simply stated, trust is the level of confidence of $c$ that the outcome of a transaction with another agent $s$ will be satisfactory for it. More formally: $T_c^s\left(t_i\right) = \int\limits_{U_c(R)\geq u_0} \tau_c^s\left(R,t_i\right).dR$, where $U_c(R)$ is the utility function of service consumer $c$; and $\tau_c^s\left(R,t_i\right)$ - the trustworthiness of service provider $s$ as perceived by consumer $c$ in the context of a transaction $t_i \in T$ is the a priori subjective joint probability distribution function of the critical rating vector $R_c^s\left(t_i\right)$ from the perspective of $c$.

It is not only cloud service consumers that need the consideration of trust in their transactions with the cloud service providers. Most often than not, cloud services providers also need to be wary of the activities of cloud service consumers. Thus, trust modeling is useful in the analysis of the genuine and potentially malicious service consumers. Therefore a trust model is needful for the perceived trustworthiness of service consumers by the providers of the services. So similarly, the level of trust $T_s^c\left(t_i\right)$ of a service provider $s$ for a service consumer $c$ in the context of a transaction $t_i \in T$ is the a priori probability that the utility of $s$ will meet or exceed its minimum threshold of satisfaction $u_0$ at the end of transaction $t_i$, given service provider $s$ perceived trustworthiness of service consumer $c$. Again, more formally: $T_s^c\left(t_i\right) = \int\limits_{U_s(R)\geq u_0} \tau_s^c\left(R,t_i\right).dR$, where $U_s(R)$ is the utility function of service provider $s$; and $\tau_s^c\left(R,t_i\right)$ - the trustworthiness of service consumer $c$ as perceived by service provider $s$ in the context of a transaction $t_i \in T$ is the a priori subjective joint probability distribution function of the critical rating vector $R_s^c\left(t_i\right)$ from the perspective of $s$. Please note that it is for notational simplicity that the critical rating vectors $R_c^s\left(t_i\right)$ and $R_s^c\left(t_i\right)$ are denoted by $R$ (without the full complement of the subscripts) in the denotation of the trustworthiness.

The above definitions have a number of interesting properties which correspond with the intuitive properties of trust in our everyday life such as trustworthiness is subjective, and it is defined relative to a particular set of critical attributes; trustworthiness is defined at a given point in time, and it is defined as a probability distribution. Some other important intuitive attributes of trust are that trust has duality - it is subjective and objective; that is, some of the critical attributes are subjectively measureable and others are objectively measureable; trust is not always symmetrical; and trust is dynamic, that is, trust is related to environment (context) and temporal factors [67].

### 3.3.2 Cloud Computing Parameters of Trust

When selecting a cloud service provider, multiple important parameters that are of relevance to the cloud service consumer need to be identified properly. Also, there is need for mechanisms to measure those parameters and aggregate these measurements based on the customers' preference regarding the importance of the parameters[68]. Ref. [69] and [68] have identified several of these parameters which have been categorized into quality of service related, security and privacy related, risk management related, and reputation related attributes. These parameters (attributes) are termed critical attributes; more formerly, a *critical attribute* of a service provider $s$, from the perspective of a service consumer $c$, in the context of a transaction $t_i \in T$ is an attribute whose value affects the utility of $c$ and is contingent upon the behavior of $s$ in the course of transaction $t_i$ [66]. A non-exhaustive list of selected set of the potential critical attributes in cloud services are briefly outlined below under each of these categories.

#### 3.3.2.1 Quality of Service Related Attributes

International Telecommunication Union has defined a methodology for capturing the quality requirements of a user of communication services which uses seven general criteria [70]. This view is modified in [71] by adding capability, usability, and fidelity - as a supplement to accuracy. Each of these generic aspects can be applied at different stages of the SLA lifecycle, and are applicable to cloud services. They therefore remain useful dimensions along which to classify cloud services [72]. The QoS related elements are performance metrics such as latency, availability, accuracy, reliability, and capability [72]. These metrics have also been emphasized by [48] and also asserted by [73] to be part of their ten common denominators that must be considered to make cloud storage valuable.

#### 3.3.2.2 Security and Privacy Related Attributes

Some of the security and privacy related parameters that are pertinent to cloud consumers and cloud service providers are data confidentiality and integrity, federated identity management solutions, secure authentication and session management, and secure cryptographic mechanisms. Other prevalent vulnerabilities in state-of-the-art cloud computing offerings that cloud consumers are wary of include SQL injection, command injection and cross-site scripting. Some of the security parameters that are more pertinent to cloud services providers are key management, physical security support, network security support, unauthorized access to management interface, and internet protocol vulnerabilities.

#### 3.3.2.3 Risk Management Related Attributes

Some of the risk management related factors that are of importance to cloud consumers are standardized SLA with unambiguous guarantees, near-real time auditing services [8] and visibility into the security controls and processes employed by the service provider as well as their performance over time that offer transparency, compliance (accreditation or certification), security measures, interoperability, customer support facilities, geographical location of cloud storage (data protection laws and other jurisdictional implication of where data is stored), and cloud service deployment models.

#### 3.3.2.4 Reputation Related Attributes

Reputation related parameters form some of the potential critical attributes that users consider in selecting cloud services. Some of these parameters are recommendation from existing users of the service, feedback and publicly available reviews of the specific cloud services, category of the service and reputation of the cloud service provider.

#### 3.3.2.5 General Cloud Metrics of Trust

In addition to the cloud specific attributes, some general attributes that are dependent on the activities of an entity to be trusted are of relevance for our discussion. The four main attributes of this category are intent, integrity, capability and results. Intent constitutes information about declared agendas (what parties promise to provide through their services), integrity constitutes information about honesty (if parties deliver what they promised), capability constitutes information about owned or outsourced resources, and results constitute information about products they are specialized in [49].

## 3.4 Trust Engineering Approaches

The various major approaches towards trust engineering in cloud computing is presented in this section. It should be evident to readers that any research work that targets one or more of the trust attributes (or other related trust attributes) discussed in Section 3.3.2 above contributes to trust engineering in cloud computing. We identify two broad categories based on whether it is targeted towards benefiting cloud service consumers or the cloud service providers. The identified major approaches to trust engineering in cloud computing are cloud audit based, reputation based, trusted third party based, trusted computing technology based, and cloud services deployment based approaches.

### 3.4.1 End User Support Oriented Trust Engineering

This is about mechanisms that facilitate building cloud consumers' trust in choosing and managing cloud service usage.

#### 3.4.1.1 Cloud Audit Approaches

Reference [23] has proposed the notion of a Consumer–centric Resource Accounting Model for a cloud resource. An accounting model is weakly consumer-centric if all the data that the model requires for calculating billing charges can be queried programmatically from the provider. Further, an accounting model is strongly consumer–centric if all the data that the model requires for calculating billing charges can be collected independently by the consumer (or a TTP); in effect, this means that a consumer (or a TTP) should be in a position to run their own measurement service. They contend that it is in the interest of the providers to make their accounting models at least weakly consumer-centric. Strongly consumer–centric models should prove even more attractive to consumers as they enable consumers to incorporate independent consistency or reasonable checks as well as raise alarms when apparent discrepancies are suspected in consumption figures. Strongly consumer-centric accounting models have the desirable property of openness and transparency, since service users are in a position to verify the charges billed to them.

One of the most common groupings or layers in cloud computing is the view of IaaS, PaaS and SaaS. These abstractions layers are mainly system-centric. In contrast, the *TrustCloud* framework takes a different perspective, i.e., an architectural, data-centric view. Because of the scale of cloud computing, the types of data-centric logs range from system-level file-centric logs to workflow-level audit trail logs. The *TrustCloud* framework attempts to describe the layers of cloud accountability. The five abstraction layers of the types of logs needed for an accountable cloud are system layer – addresses tracking of files across the Cloud, data layer – addresses tracking of change of data and information across the Cloud, workflow layer – addresses data and information flow in the Cloud, law and regulations layer – addresses data-centric logging requirements mandated by external laws and regulations, and finally, policies layer – addresses data-centric audit requirements mandated by internal governance and audit requirements [27].

#### 3.4.1.2 Reputation Based Approaches

Reference [51] presents a fully distributed framework that enable trust-based cloud customer and cloud service provider interactions. The framework aids a service consumer in assigning an appropriate weight to the feedback of different raters regarding a prospective service provider. They developed a mechanism based on their framework for controlling falsified feedback ratings from iteratively exerting trust level contamination due to falsified feedback ratings.

Secure integrity attestation of computation results is the focus of [29]. Whereas AdapTest [74] and RunTest [75] implement cloud service integrity attestation for the IBM ***System S*** stream processing system [76] using attestation graphs in which always-agreeing nodes form a clique in the graph, facilitating detection of malicious collectives; in contrast, the work of [29] considers a reputation-based trust management approach to integrity violation detection in Hadoop clouds. Trust management systems probabilistically anticipate future misbehavior of untrusted agents based on their histories of past behavior.

#### 3.4.1.3 Trusted Third Party Based Approaches

The goal of [43] is the remote assessment of the cloud infrastructure's integrity by a cloud certifier. They hence need to detect all changes in the remote system that can possibly compromise security. All changes in the hardware or software should be reported to the cloud certifier, even if the infrastructure provider has super-user access to the machine. Their *BonaFides* system monitors the infrastructure provider's physical hosts by observing file modifications on a low level and persistently stores the history of these integrity measurements and file changes. Files are measured at regular intervals and whenever changes in the files are detected. *BonaFides* measures the hypervisor, kernel, kernel modules, disk and network utilities, and system configuration files in the Dom0 (the administrative domain of the Xen hypervisor that manages access to the physical host's resources).

#### 3.4.1.4 Trusted Computing Technology Base Approaches

Ref. [77] has presented a multi-tenancy trusted computing environment model (MTCEM) to support the security duty separation between Cloud Service Provider (CSP) and customers. MTCEM is designed for IaaS service delivery model, and it intends to separate the security responsibility of the CSP and their customers on cloud infrastructures. In MTCEM model, CSP is responsible to assure a trusted host and Virtual Machine Monitor (VMM) environment, and customers are responsible for the assurance of trusted virtual instances they rent from CSP. MTCEM uses the two main mechanisms of transitive trust and platform attestation of the trusted computing technology. It uses transitive trust mechanism to build a trusted computing platform and attestation mechanism to improve the customers' confidence on CSP. Ref. [25] shows how to design the Trusted SaaS Platform (TSP) by taking advantage of trusted computing technologies. Conventional trusted computing platforms like Terra [78] are able to prevent the owner of a physical machine from inspecting or interfering with a computation running in a virtual machine (VM) that is hosted in the physical machine, and thus can effectively secure the computation running in the VM. However, these platforms cannot address security and trust issues in SaaS environments due to the following two reasons. First, they do not specify who will launch the VM that is responsible for performing the computation. The approach presented in Towards Trusted Cloud Computing [79] on Trusted Cloud Computing Platform (TCCP) can only be used for IaaS and not suitable for SaaS environments. In TCCP, the protocols are mainly utilized for node registration and securing VM launch and migration. However, in SaaS system, the users' main purpose is guaranteeing that the SaaS providers process their data and respond with the result without inspection or modification, rather than guaranteeing the security of their VMs. To address this problem, [25] proposed a trusted SaaS platform that enables a trusted third party to launch a VM as a trusted execution environment(TEE) on the computation server. Thus though the privileged administrators of SaaS providers can access the physical host of TEE, they cannot access the TEE because the TEE is not launched by them. The TSP leverages the trusted virtual machine monitor (TVMM) [78] so privileged administrators cannot tamper with the TEE. The TEE is also where all of the decryption, computation and encryption take place, so it can ensure the confidentiality and integrity of users' data and computations outsourced to SaaS services.

#### 3.4.1.5 Cloud Service Deployment Approaches

Reference [69] has devised five reference deployment models for cloud computing that progressively address user security concerns and increase users' trust in cloud computing. These are the separation model, availability model, migration model, tunnel model, and encryption model. The Separation Model is the base model for all the other four models. It separates data storage from data processing, requiring at least two independent cloud service providers to process data and to store data, respectively. This can help ease users' concerns on having a single provider in complete control over the data and the services they use. The Availability Model introduces redundancy into the Separation Model, in both the data processing and the data storage. With the redundancy in the Availability Model, failures of one data processing service and one data storage service can be tolerated. The Tunnel Model further enhances the Separation Model by using a Tunnel Service to impose isolation between the Data Processing Service and the Cloud Storage Service. The Tunnel Service prevents collusion by cutting the direct communications between the Data Processing Service and the Cloud Storage Service, assuming that it is very unlikely for two isolated providers to collude. The Cryptography Model augments the Tunnel Model with cryptography support, such as data encryption, decryption, and digital signing.

Even though there are approaches to provide confidentiality for the users' data in the Cloud, these are not widely adopted due to both awareness and usability issues. Therefore, [34] proposed the Confidentiality as a Service (CaaS) paradigm to provide usable confidentiality and integrity for the bulk of users for whom the current security mechanisms are too complex or require too much effort. The CaaS paradigm combines data security with usability by design and integrates effortlessly into available cloud service applications and workflows. They leverage the splitting of trust between the cloud service provider and one or more CaaS providers to improve usability. CaaS focuses on unobtrusive confidentiality by hiding all cryptographic artifacts from the prevalently non-technical users [34].

#### 3.4.2 Service Provider Oriented Trust Engineering

This facilitates building trust between the cloud service providers and their customers in ensuring that their resources and administrative platforms will not be abused by the consumers.

##### 3.4.2.1 Reputation Based Approaches

Reference [50] considers the scenario where a service provider, termed the Master Service Provider (MSP), identifies a great business opportunity or other scenarios which need collaboration with other service providers, termed Guest Service Providers (GSP), to offer a set of new services to the customers. Their approach is to derive trustworthiness of guest service provider *i* (GSP*i*) according to its past behavior.

##### 3.4.2.2 Identity and Access Management

Identity management is one of the core components in cloud privacy and security and can help alleviate some of the user trust issues associated with cloud computing. Available solutions use trusted third party in identifying entities to service providers. The solution providers do not recommend the usage of their solutions on untrusted hosts. Ref. [33] has proposed an approach for identity management that is independent of trusted third parties and has the ability to use identity data on untrusted hosts. The approach is based on the use of predicates over encrypted data and multi-party computing for negotiating a use of a cloud service. It uses active bundle - which is a middleware agent that includes PII, privacy policies, a virtual machine that enforces the policies, and has a set of protection mechanisms to protect it. An active bundle interacts on behalf of a user to authenticate to cloud services using user's privacy policies.

#### 3.4.3 Final Remarks

Ref. [80] has argued that cryptography alone can't enforce the privacy demanded by common cloud computing services, even with such powerful tools as fully homomorphic encryption (FHE). They formally define a hierarchy of natural classes of private cloud applications, and show that no cryptographic protocol can implement those classes where data is shared among clients.

Employing trusted computing technologies and reputation based approaches are two key approaches to trust engineering in cloud computing marketplace. Also the adopted cloud deployment model plays a significant role in improving trust in cloud environments.

### 3.5 Trust Management Systems

Trust management is the activity of creating systems and methods that allow relying parties to make assessments and decisions regarding the dependability of potential transactions involving risk, and that also allow players and system owners to increase and correctly represent the reliability of themselves and their systems [81]. There is a need for methodologies that enable relying parties to determine the trustworthiness of remote parties through computer mediated communication and collaboration. At the same time, trustworthy entities need methodologies that enable them to be recognized as such; developing and applying these methodologies can be called trust management.

A survey on the trust management systems implemented on distributed systems with emphasis on cloud computing has been carried out by [52] . They reported on several trust models such as *CuboidTrust* [53] , *EigenTrust*[54] , *Bayesian Network based Trust Management (BNBTM)* [55], *GroupRep*[56] , *AntRep*[57] , *Global Trust*[58] [59] , *Peer Trust* [60], *and Trust Ant Colony System (TACS)*[61]. These models were mainly proposed for systems like clusters, grids and wireless sensor networks, and have not been used or tested in cloud computing environments; hence their suitability for use in cloud computing cannot be recommended without an extensive evaluation. Though a few work on trust targeting cloud computing environments were considered in [52], it was found that none of the proposed systems was based on solid theoretical foundation and also does not take any quality of service attribute into account for forming the trust scores. This observation may be due in part to the fact that, although the considered studies dealt with elements of trust in cloud computing and hence will pass for approaches to trust engineering in cloud computing, these were not really trust management systems since they do not possess elements for the generic operations of trust management systems which include expectation, data monitoring, data management, analysis, and decision making. Secondly, this observation is also partly due to the fact that the concept of trust itself is still not well understood by the research community due to its loose usage without formal specification. Hence a solid formulation of the concept of trust is essential for the research community, and more especially in the context of cloud computing in order to lay solid theoretical foundation for building trust management systems for cloud computing.

Some of the trust related works in cloud computing that have provided some generic methodologies in developing trust management systems for cloud computing environments are [49] and [24]. The generic operations of trust management include expectation, data monitoring, data management, analysis, and decision making. Separation of these operations supports data privacy, confidentially and integrity, where data can be kept at their sources and accessed only on a need to know basis[49]. The model builds trust using the four parameters: intent, integrity, capability and results. Intent constitutes information about declared agendas about what entities promise to provide through their services. Integrity constitutes information about honesty which is a measure of, to what extent entities deliver on what they promised. Capability constitutes information about owned or outsourced resources; and finally, results constitute information about products and services that entities specialized in through consistently delivering these products and services satisfactorily to their clients.

#### 3.5.1 Final Remarks on Trust Management Systems

The current state-of-the-art in trust management systems are that, they are mainly for peer-to-peer systems. Secondly, current trust systems provide no separation of concern among different trust management operations. Also most current trust management systems provide limited or no customization according to trusting entities' requirements. The focus is skewed towards service providers being evaluated by service consumers for their trustworthiness, but not vice versa[49]. In addition to designing trust management systems that factor in the above mentioned points, the solid formulation of the concept of trust is essential for the research community, and more especially in the context of cloud computing in order to lay solid theoretical foundation for building trust systems for cloud computing environments.

## 4 CONCLUSION & FUTURE WORK

This work has reviewed identified primary studies on trust engineering approaches in cloud computing. The central motivating objective of this work has been to lay the foundation for designing a trust management system for OCS platforms, and provide summary of trust engineering approaches in cloud computing for easy reference by the research community. The study has been specifically interested in finding the main approaches towards trust engineering in cloud computing, the objectives of the identified primary studies and in what contexts these trust management systems are being developed; and finally, the major trust models and trust management systems for cloud computing.

It was observed that trusted computing technologies and reputation based approaches are the main approaches to trust engineering in cloud computing. Also trusted third party approaches and the deployment model play a significant role in enhancing trust between service providers and consumers.

Based on the findings during the study, the main arrears of trust engineering research focus has been on quality of service, security, access and identity management, user support on trust management, and accountability in in the context of a cloud computing marketplace .

We observed that the concept of trust is used loosely without any formal specification in cloud computing discussions and trust engineering in general. As a first step towards addressing this problem, we have contextualized the formal trust specification in multi-agent environments for cloud computing. This should prove very useful for other researchers interested in trust related research in a cloud computing marketplace.

The findings in this paper have been applied in the design of a trust management system for opportunistic cloud services [82]. We will as part of our future work, expand on the concept of composite (group) trust, and provide suitable formal specification and definition for it.

## 5 LIMITATIONS OF STUDY

There could be a possible bias of the authors during the practical screening process towards selecting relevant primary studies based on personal interest in studies that are based on concepts similar to that of opportunistic cloud services. This is because since the central motivating objective of this work is to lay the foundation for designing a trust management system for opportunistic cloud services platforms, studies that have elements of concepts similar to that this are of interest to the authors. With this concern in mind from the beginning of this work, deliberate steps were however taken to ensure that this inherent bias does not affect the selection of the included primary studies.

## REFERENCES

[1] E. Kuada and H. Olesen, “A Social Network Approach to Provisioning and Management of Cloud Computing Services for Enterprises,” in *Proceedings of The Second International Conference on Cloud Computing, GRIDs, and Virtualization*, 2011, pp. 98–104.

[2] E. Kuada, H. Olesen, and A. Henten, “Public Policy and Regulatory Implications for the Implementation of Opportunistic Cloud Computing Services for Enterprises,” in *9th International Workshop on Security in Information Systems*, Wroclaw, Poland, 2012, pp. 3 – 13.

[3] W. Jansen and T. Grance, “Guidelines on security and privacy in public cloud computing,” *NIST Special Publication*, pp. 800–144, 2011.

[4] L. Badger, T. Grance, R. Patt-Corner, and J. Voas, “Draft cloud computing synopsis and recommendations,” *NIST Special Publication*, vol. 800, p. 146, 2011.

[5] P. Mell and T. Grance, “The NIST definition of cloud computing (draft),” *NIST special publication*, vol. 800, p. 145, 2011.

[6] Z. Song, J. Molina, and C. Strong, “Trusted Anonymous Execution: A Model to Raise Trust in Cloud,” in *2010 9th International Conference on Grid and Cooperative Computing (GCC)*, 2010, pp. 133 – 138.

[7] M. Almorsy, J. Grundy, and A. S. Ibrahim, “TOSSMA: A Tenant-Oriented SaaS Security Management Architecture,” in *2012 IEEE 5th International Conference on Cloud Computing (CLOUD)*, 2012, pp. 981 –988.

[8] J. S. Park, E. Spetka, H. Rasheed, P. Ratazzi, and K. J. Han, “Near-Real-Time Cloud Auditing for Rapid Response,” in *2012 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, 2012, pp. 1252 –1257.

[9] E. Kuada and H. Olesen, “Incentive mechanisms for Opportunistic Cloud Computing Services,” in *2012 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, Pittsburgh, PA, USA, 2012, pp. 127 –136.

[10] E. Kuada, K. Adanu, and H. Olesen, “Cloud Computing and Information Technology Resource Cost Management for SMEs,” in *Proceedings of IEEE Region 8 Conference EuroCon 2013*, University of Zagreb, Croatia, 2013, pp. 258 – 265.

[11] B. Kitchenham and S. Charters, “Guidelines for performing Systematic Literature Reviews in Software Engineering,” 2007.

[12] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, “Lessons from applying the systematic literature review process within the software engineering domain,” *Journal of Systems and Software*, vol. 80, no. 4, pp. 571–583, Apr. 2007.

[13] C. Okoli and K. Schabram, “A Guide to Conducting a Systematic Literature Review of Information Systems Research,” in *Working Papers on Information Systems*, 2010.

[14] Y. Levy and T. J. Ellis, “A Systems Approach to Conduct an Effective Literature Review in Support of,” *INFORMATION SYSTEMS RESEARCH. INFORMING SCIENCE JOURNAL*, vol. 9, p. 181212, 2006.

[15] L. M. Vaquero, L. Rodero-Merino, and D. Morán, “Locking the sky: a survey on IaaS cloud security,” *Computing*, vol. 91, no. 1, pp. 93–118, Jan. 2011.

[16] C. Modi, D. Patel, B. Borisaniya, A. Patel, and M. Rajarajan, “A survey on security issues and solutions at different layers of Cloud computing,” *J Supercomput*, pp. 1–32, Oct. 2012.

[17] K. Hashizume, D. G. Rosado, E. Fernández-Medina, and E. B. Fernandez, “An analysis of security issues for cloud computing,” *Journal of Internet Services and Applications*, vol. 4, no. 1, p. 5, Feb. 2013.

[18] RefWorks, “RefWorks,” 2013. [Online]. Available: http://www.refworks.com/.

[19] nVivo10, “Nvivo,” nVivo10. [Online]. Available: http://www.qsrinternational.com/products_nvivo.aspx.

[20] S. Friese, “ATLAS.ti 7 User Guide and Reference: ATLAS.ti 7 USER MANUAL.” ATLAS.ti Scientific Software Development GmbH, Berlin, 28-Jan-2013.

[21] K. Charmaz, “Grounded theory,” *Strategies of qualitative inquiry*, vol. 2, p. 249, 2003.

[22] A. Strauss and J. Corbin, “Grounded theory methodology,” *Handbook of qualitative research*, pp. 273–285, 1994.

[23] A. Mihoob, C. Molina-Jimenez, and S. Shrivastava, “A Case for Consumer-centric Resource Accounting Models,” in *2010 IEEE 3rd International Conference on Cloud Computing (CLOUD)*, 2010, pp. 506 –512.

[24] S. M. Habib, S. Ries, and M. Muhlhauser, “Towards a Trust Management System for Cloud Computing,” in *2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2011, pp. 933 –939.

[25] C. Zhong, J. Zhang, Y. Xia, and H. Yu, “Construction of a Trusted SaaS Platform,” in *2010 Fifth IEEE International Symposium on Service Oriented System Engineering (SOSE)*, 2010, pp. 244 –251.

[26] M. Almorsy, J. Grundy, and A. S. Ibrahim, “Collaboration-Based Cloud Computing Security Management Framework,” in *2011 IEEE International Conference on Cloud Computing (CLOUD)*, 2011, pp. 364 –371.

[27] R. K. L. Ko, M. Kirchberg, and B. S. Lee, “From system-centric to data-centric logging - Accountability, trust amp; security in cloud computing,” in *Defense Science Research Conference and Expo (DSR), 2011*, 2011, pp. 1 –4.

[28] A. Hadoop, *Apache Hadoop*. 2013.

[29] S. M. Khan and K. W. Hamlen, “Hatman: Intra-cloud Trust Management for Hadoop,” in *2012 IEEE 5th International Conference on Cloud Computing (CLOUD)*, 2012, pp. 494 –501.

[30] B. Panzer-Steindel, “Data integrity,” CERN/IT, Draft Draft 1.3, Apr. 2007.

[31] S. Narayan, J. A. Chandy, S. Lang, P. Carns, and R. Ross, “Uncovering errors: the cost of detecting silent

data corruption," in *Proceedings of the 4th Annual Workshop on Petascale Data Storage*, New York, NY, USA, 2009, pp. 37–41.

[32] S. Nepal, S. Chen, J. Yao, and D. Thilakanathan, "DIaaS: Data Integrity as a Service in the Cloud," in *2011 IEEE International Conference on Cloud Computing (CLOUD)*, 2011, pp. 308 –315.

[33] R. Ranchal, B. Bhargava, L. B. Othmane, L. Lilien, A. Kim, M. Kang, and M. Linderman, "Protection of Identity Information in Cloud Computing without Trusted Third Party," in *2010 29th IEEE Symposium on Reliable Distributed Systems*, 2010, pp. 368 –372.

[34] S. Fahl, M. Harbach, T. Muders, and M. Smith, "Confidentiality as a Service – Usable Security for the Cloud," in *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2012, pp. 153 –162.

[35] M. M. Lucas and N. Borisov, "FlyByNight: mitigating the privacy risks of social networking," in *Proceedings of the 7th ACM workshop on Privacy in the electronic society*, New York, NY, USA, 2008, pp. 1–8.

[36] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin, "Persona: an online social network with user-defined privacy," in *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, New York, NY, USA, 2009, pp. 135–146.

[37] A. Whitten and J. D. Tygar, "Why Johnny can't encrypt: a usability evaluation of PGP 5.0," in *Proceedings of the 8th conference on USENIX Security Symposium - Volume 8*, Berkeley, CA, USA, 1999, pp. 14–14.

[38] P. Gutmann, "Plug-and-play PKI: a PKI your mother can use," in *Proceedings of the 12th conference on USENIX Security Symposium - Volume 12*, Berkeley, CA, USA, 2003, pp. 4–4.

[39] D. Balfanz, G. Durfee, D. K. Smetters, and R. E. Grinter, "In search of usable security: five lessons from the field," *IEEE Security Privacy*, vol. 2, no. 5, pp. 19 – 24, Oct. 2004.

[40] H. Kim, H. Lee, W. Kim, and Y. Kim, "A Trust Evaluation Model for QoS Guarantee in Cloud Systems," *International Journal of Grid and Distributed Computing*, vol. 3, no. 1, pp. 1 – 10, Mar. 2010.

[41] íñIgo Goiri, F. Julí́, J. O. Fitó, M. MacíAs, and J. Guitart, "Supporting CPU-based guarantees in cloud SLAs via resource-level QoS metrics," *Future Gener. Comput. Syst.*, vol. 28, no. 8, pp. 1295–1302, Oct. 2012.

[42] X. Liu, Y. Yang, D. Yuan, G. Zhang, W. Li, and D. Cao, "A Generic QoS Framework for Cloud Workflow Systems," in *Proceedings of the 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing*, Washington, DC, USA, 2011, pp. 713–720.

[43] R. Neisse, D. Holling, and A. Pretschner, "Implementing Trust in Cloud Infrastructures," in *2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, 2011, pp. 524 – 533.

[44] F. Doelitzscher, C. Reich, and A. Sulistio, "Designing Cloud Services Adhering to Government Privacy Laws," in *2010 IEEE 10th International Conference on Computer and Information Technology (CIT)*, 2010, pp. 930 –935.

[45] "Federal Information Security Management Act (FISMA)," 2002. [Online]. Available: http://www.dhs.gov/federal-information-security-management-act-fisma. [Accessed: 27-Feb-2013].

[46] G. Stoneburner, A. Y. Goguen, and A. Feringa, "SP 800-30. Risk Management Guide for Information Technology Systems," National Institute of Standards & Technology, Gaithersburg, MD, United States, 2002.

[47] M. Ahmed and Y. Xiang, "Trust Ticket Deployment: A Notion of a Data Owner's Trust in Cloud Computing," in *2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2011, pp. 111 –117.

[48] S. Garg, S. Versteeg, and R. Buyya, "A framework for ranking of cloud computing services," *Future Generation Computer Systems*, Jun. 2012.

[49] H. Salah and M. Eltoweissy, "Towards a personalized trust management system," in *2012 International Conference on Innovations in Information Technology (IIT)*, 2012, pp. 373 –378.

[50] L. Xin and A. Datta, "On trust guided collaboration among cloud service providers," in *2010 6th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, 2010, pp. 1 –8.

[51] J. Abawajy, "Establishing Trust in Hybrid Cloud Computing Environments," in *2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2011, pp. 118 –125.

[52] M. Firdhous, O. Ghazali, and S. Hassan, "Trust Management in Cloud Computing: A Critical Review," *International Journal on Advances in ICT for Emerging Regions (ICTer)*, vol. 4, no. 2, pp. 24–36, 2012.

[53] R. Chen, X. Zhao, L. Tang, J. Hu, and Z. Chen, "CuboidTrust: a global reputation-based trust model in peer-to-peer networks," *Autonomic and Trusted Computing*, pp. 203–215, 2007.

[54] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *Proceedings of the 12th international conference on World Wide Web*, 2003, pp. 640–651.

[55] Y. Wang, V. Cahill, E. Gray, C. Harris, and L. Liao, "Bayesian network based trust management," *Autonomic and Trusted Computing*, pp. 246–257, 2006.

[56] H. Tian, S. Zou, W. Wang, and S. Cheng, "A group based reputation system for P2P networks," *Autonomic and trusted computing*, pp. 342–351, 2006.

[57] W. Wang, G. Zeng, and L. Yuan, "Ant-based reputation evidence distribution in P2P networks," in *Grid and Cooperative Computing, 2006. GCC 2006. Fifth International Conference*, 2006, pp. 129–132.

[58] F. Yu, H. Zhang, F. Yan, and S. Gao, "An improved global trust value computing method in P2P system," *Autonomic and trusted computing*, pp. 258–267, 2006.

[59] W. Wang, X. Wang, S. Pan, and P. Liang, "A new global trust model based on recommendation for peer-to-peer network," in *New Trends in Information and*

*Service Science, 2009. NISS'09. International Conference on*, 2009, pp. 325–328.

[60] L. Xiong and L. Liu, "PeerTrust: supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843 – 857, Jul. 2004.

[61] F. Gómez Mármol, G. Martínez Pérez, and A. F. Gómez Skarmeta, "TACS, a trust model for P2P networks," *Wireless personal communications*, vol. 51, no. 1, pp. 153–164, 2009.

[62] X. Zhang, H. Liu, B. Li, X. Wang, H. Chen, and S. Wu, "Application-Oriented Remote Verification Trust Model in Cloud Computing," in *2010 IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom)*, 2010, pp. 405 –408.

[63] M. Kuehnhausen, V. S. Frost, and G. J. Minden, "Framework for assessing the trustworthiness of cloud resources," in *2012 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, 2012, pp. 142 –145.

[64] W. Viriyasitavat and A. Martin, "Formal Trust Specification in Service Workflows," in *2010 IEEE/IFIP 8th International Conference on Embedded and Ubiquitous Computing (EUC)*, 2010, pp. 703 –710.

[65] D. Olmedilla, O. F. Rana, B. Matthews, and W. Nejdl, "Security and trust issues in semantic grids," *In Proceedings of the Dagstuhl Seminar, Semantic Grid: the convengence of technologies, Volume 05271. 2005. [PD05] [PPI04] Panteli*, pp. 191–200, 2005.

[66] C. Dellarocas, "The Design of Reliable Trust Management Systems for Electronic Trading Communities," in *SLOAN SCHOOL OF MANAGEMENT, MIT, 2000*, 2001.

[67] C. Shen, H. Zhang, H. Wang, J. Wang, B. Zhao, F. Yan, F. Yu, L. Zhang, and M. Xu, "Research on trusted computing and its development," *Sci. China Inf. Sci.*, vol. 53, no. 3, pp. 405–433, Mar. 2010.

[68] S. M. Habib, S. Ries, and M. Muhlhauser, "Cloud Computing Landscape and Research Challenges Regarding Trust and Reputation," in *2010 7th International Conference on Ubiquitous Intelligence Computing and 7th International Conference on Autonomic Trusted Computing (UIC/ATC)*, 2010, pp. 410 –415.

[69] G. Zhao, C. Rong, M. G. Jaatun, and F. E. Sandnes, "Reference deployment models for eliminating user concerns on cloud security," *J Supercomput*, vol. 61, no. 2, pp. 337–352, Aug. 2012.

[70] "Communications quality of service: A framework and definitions," International Telecommunication Union, Recommendation ITU-T Recommendation G.1000, Nov. 2001.

[71] ETSI, "User Group; Quality of telecom services; Part 1: Methodology for identification of parameters relevant to the Users," European Telecommunications Standards Institute, Sophia Antipolis Cedex - FRANCE, Guide ETSI EG 202 009-1 V1.2.1 (2006-11), Nov. 2006.

[72] "CLOUD; SLAs for Cloud services," European Telecommunications Standards Institute, 650 Route des Lucioles F-06921 Sophia Antipolis Cedex - FRANCE, Technical Report ETSI TR 103 125 V1.1.1 (2012-11), Nov. 2012.

[73] PROMISE Technology Inc., "Cloud Computing andTrusted Storage." PROMISE Technology Inc., Q1-2010.

[74] J. Du, N. Shah, and X. Gu, "Adaptive data-driven service integrity attestation for multi-tenant cloud systems," in *2011 IEEE 19th International Workshop on Quality of Service (IWQoS)*, 2011, pp. 1 –9.

[75] J. Du, W. Wei, X. Gu, and T. Yu, "RunTest: assuring integrity of dataflow processing in cloud computing infrastructures," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, New York, NY, USA, 2010, pp. 293–304.

[76] B. Gedik, H. Andrade, K.-L. Wu, P. S. Yu, and M. Doo, "SPADE: the system s declarative stream processing engine," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, New York, NY, USA, 2008, pp. 1123–1134.

[77] X.-Y. Li, L.-T. Zhou, Y. Shi, and Y. Guo, "A trusted computing environment model in cloud architecture," in *2010 International Conference on Machine Learning and Cybernetics (ICMLC)*, 2010, vol. 6, pp. 2843 – 2848.

[78] T. Garfinkel, B. Pfaff, J. Chow, M. Rosenblum, and D. Boneh, "Terra: a virtual machine-based platform for trusted computing," 2003, pp. 193–206.

[79] N. Santos, K. P. Gummadi, and R. Rodrigues, "Towards Trusted Cloud Computing," in *HOTCLOUD*, 2009.

[80] M. Van Dijk and A. Juels, "On the impossibility of cryptography alone for privacy-preserving cloud computing," in *Proceedings of the 5th USENIX conference on Hot topics in security*, Berkeley, CA, USA, 2010, pp. 1–8.

[81] A. Jøsang, C. Keser, and T. Dimitrakos, "Can we manage trust?," in *Proceedings of the Third international conference on Trust Management*, Berlin, Heidelberg, 2005, pp. 93–107.

[82] E. Kuada, "Trust Management System for Opportunistic Cloud Services," in *2013 IEEE 2nd International Conference on Cloud Networking (CloudNet)*, San Francisco, USA, 2013, pp. 33 – 41.

## Appendix 4: Paper #4

Kuada, Eric. 2013. “Trust Management System for Opportunistic Cloud Services.” In *2013 IEEE 2nd International Conference on Cloud Networking (CloudNet)*. San Francisco, USA: IEEE.

# Trust Management System for Opportunistic Cloud Services

Eric Kuada
Department of Electronic Systems
Aalborg University, Copenhagen, Denmark
kuada@cmi.aau.dk

*Abstract* - We have over the past three years been working on the feasibility of Opportunistic Cloud Services (OCS) for enterprises. OCS is about enterprises strategically contributing and utilizing spare IT resources as cloud services. One of the major challenges that such a platform faces is data security and trust management issues. This paper presents a trust management system for OCS platforms. It models the concept of trust and applies it to OCS platforms. The trust model and the trust management system are verified through the simulation of the computation of the trust values with Infrastructure as a Service, and Software as a Service, usage scenarios.

*Keywords*: Opportunistic Cloud Services; Trust Management System; Trust Engineering; Pseudo Service Level Agreement

## I. INTRODUCTION

This introductory section begins with the motivation for this study, and then is followed by the background for trust engineering efforts in cloud computing.

### *A. Motivation for the Study*

We have over the past three years been working on the feasibility of Opportunistic Cloud Services (OCS) for enterprises [1] [2] [3]. We have been working on the feasibility of its successful implementation in terms of the technical feasibility, impact of public policy and regulations on its implementation [3] , and the development of suitable incentive mechanisms for OCS networks [2]. We have also been working on the role of cloud computing for Information Technology (IT) resource cost management for SMEs [4], and how to leverage opportunistic cloud services to lighten the IT resource needs of SMEs, particularly in developing economies.

One of the major challenges that such a platform faces is data security and trust management issues. The OCS network platform is a governing platform that serves as the social networking platform for enterprises and also includes interoperable cloud management tools with which member enterprises can provide resources that will be used by other enterprises interested in these services. Members normally will package only their spare IT resources and make them available as Cloud services on the OCS platform so that others interested can utilize them. For example a member institution that has virtualized its data center into a private cloud can plan its resources such that it can make some of these resources available to the OCS platform.

Since no business agreement and hence no Service Level Agreement (SLA) exist between the service providers and the potential users of their services, service consumers do not enjoy the level of support (in terms of quality of service, reliability, availability, security, billing transparency, etc.) that commercial cloud service providers offer to their clients. Considering the fact that commercial cloud service providers are finding it extremely challenging to provide such a support, together with providing adequate transparency in their services to their clients with the hope of establishing the necessary trust, the OCS platform more so needs a well-crafted and soundly engineered trust management system in order to make resources on the platform suitable for business use.

### *B. Trust Engineering in Cloud Computing*

Cloud computing is essentially the packaging of traditional information technology infrastructure and software solutions such as storage, CPU, network, applications, services, etc. as virtualized resources and delivered by a service provider to its customers as an on-demand pay-per-use self-provisioned service; normally offered through a web portal over a network such as the Internet [5] [6] [7]. As vendors start to deploy Cloud services, and users upload data in the Cloud to utilize them, a new privacy concern arises, because data owners would like to preserve the confidentiality of their data, and under some circumstance even their identities private from the software provider. While cloud service providers pledge to preserve data privacy, the current Software as a Service (SaaS) architecture makes it difficult to provide any assurance that the software in the Cloud will not be able to make copies or redistribute the data it used [8] . The Cloud model is based on two key characteristics: multi-tenancy, where multiple tenants share the same service instance, and elasticity, where tenants can scale the amount of their allocated resources based on current demands. Although both characteristics target improving resource utilization, cost and service availability, these gains are threatened by multi-tenancy security implications. The sharing of applications that process critical information without sufficient proven security isolation, security SLAs or tenant control, results in “loss-of-control” and “lack-of-trust” problems [9].

Apart from these consumer concerns, cloud architectures also introduce new classes of security risks and attacks over the resources of cloud service providers. These include poisoned virtual machines, attacks against the Cloud Service Provider (CSP) management console, attacks based on knowledge of default security settings, abuse of billing systems, attacks that abuse the trust associated with the CSP’s namespace, and data leakage via uniform resource locators. Currently, CSPs do not have robust technical solutions that can protect their cloud resources from harmful malware, virus infection, botnets, distributed denial of service attacks, or other types of cyber-attacks. Furthermore, there is no effective mechanism to help cloud users evaluate the security measures of their service providers and ensure the protection of their data while taking into consideration industry standards or personal preferences [9].

In order to design and develop a trust management system for the OCS platform, we needed to look at the current trust engineering issues in cloud computing. It was decided that we needed to perform a systematic literature review on this topic because since the OCS concept itself is new, any trust design models of its subsystems must be based on and guided by exhaustive knowledge of the state-of-the-art in the field. Some of the main findings of the study are that, employing trusted computing technologies and reputation based approaches are two key approaches to trust engineering in the cloud computing marketplace. Also, trusted third party approaches and the deployment model play a significant part in enhancing trust between service providers and their consumers. Based on the findings during the study, the main areas of trust engineering research focus has been on quality of service, security, access and identity management, user support on trust management,

978-1-4799-0568-3/13/$31.00 ©2013 IEEE

and accountability in in the context of a cloud computing marketplace . Though the objective of [10] is a formal trust specification which covers a wide range of intuitive trust characteristics such as trust transitivity and mutual relationship, much work has not been in this area of formal trust modeling. According to[11], discussions about cloud computing security often fail to distinguish general issues from cloud-specific issues. A similar trend is seen in the discussion of trust in cloud computing and trust engineering in general where the concept of trust in treated loosely without any formal specification or definition in its treatment. Formal trust modeling and definitions are however very necessary in ensuring a unified view of the concept of trust in the design and engineering of trust management systems for cloud computing. As a first step towards addressing this problem, we contextualized the formal trust specification of multi-agent environments for cloud computing environments, and provided a formal definition of the concept of trust as is applicable to the cloud computing marketplace.

The rest of the paper is organized as follows: Section II models the concept of trust for OCS platforms. Section III presents the design of a trust management system for OCS platforms. This is followed with the OCS trust management architecture in Section IV. Section V presents work on the verification of the trust model and the designed trust management system. Section VI concludes the paper and also touches on some insights on future work.

## II. TRUST MODEL FOR OCS PLATFORMS

Opportunistic Cloud Services (OCS) is a social network approach to the provisioning and management of cloud computing services for enterprises; it deals with the concept of enterprises taking advantage of cloud computing services to meet their business needs without having to pay or paying a minimal fee for the services. The OCS network is modelled and implemented as a social network of enterprises collaborating strategically for the contribution and usage of cloud computing services without entering into any business agreements.

### *1) Nature of Members and Services*

An OCS network consists of a set of strategic members contributing and utilizing cloud computing services. The platform consists of a set of services each belonging to a category; each service has a non-monetary cost that varies dynamically. The service or resource contributed by a member is of a certain finite capacity and the resources to a particular service may be contributed by multiple members. Members will normally only contribute resources that they have spare capacity of (e.g. CPU, storage, application that they have developed internally, etc.). That is, they package their spare IT resources as cloud services and make them available to the OCS platform. Members are free to provide and discontinue one or more services at will at any point in time. They are likewise free to use or discontinue the usage of one or more services at will at any point in time [2].

### *2) Trust Model in the context of OCS*

Though there has been some work on trust modeling and trust management systems, and even in the new domain of trust management systems for cloud computing environments [12] [13] [14], the subjective nature of trust has made a solid definition elusive. Researchers have most often used the term loosely in their work; more specifically, a rigorous formal definition has not been applied in most cases.

The adopted definition and model of the concept of trust in this work is an adaptation of [15]: The level of trust, $T_c^p\left(t_i\right)$ of a service consumer $c$ for a service provider $p$ in the context of a transaction $t_i \in T$ is the a priori probability that the utility of $c$ will meet or exceed its minimum threshold of satisfaction $u_0$ at the end of transaction $t_i$, given $c$'s perceived trustworthiness of service provider $p$. Simply stated, trust is the level of confidence of $c$ that the outcome of a transaction with another agent $p$ will be satisfactory for it. More formally:

$T_c^p\left(t_i\right) = \int\limits_{U_c(R)\geq u_0} \tau_c^p\left(R,t_i\right).dR$ , where $U_c(R)$ is the utility function of service consumer $c$; and $\tau_c^p\left(R,t_i\right)$ - the trustworthiness of service provider $p$ as perceived by consumer $c$ in the context of a transaction $t_i \in T$ is the a priori subjective joint probability distribution function of the critical rating vector $R_c^p\left(t_i\right)$ from the perspective of $c$. This definition and model of trust has been adopted by this paper because it provides this formal specification which is necessary in ensuring a unified view of the concept of trust in the design and engineering of trust management systems for cloud computing.

It is not only cloud service consumers that need the consideration of trust in their transactions with the cloud service providers. Most often than not, cloud services providers also need to be wary of the activities of cloud service consumers. Thus, trust modeling is useful in the analysis of the genuine and potentially malicious service consumers. Therefore a trust model is needful for the perceived trustworthiness of service consumers by the providers of the services. So similarly, the level of trust $T_p^c\left(t_i\right)$ of a service provider $p$ for a service consumer $c$ in the context of a transaction $t_i \in T$ is the a priori probability that the utility of $p$ will meet or exceed its minimum threshold of satisfaction $u_0$ at the end of transaction $t_i$, given service provider $p$'s perceived trustworthiness of service consumer $c$. Again, more formally: $T_p^c\left(t_i\right) = \int\limits_{U_p(R)\geq u_0} \tau_p^c\left(R,t_i\right).dR$ , where $U_p(R)$ is the utility function of service provider $p$; and $\tau_p^c\left(R,t_i\right)$ - the trustworthiness of service consumer $c$ as perceived by service provider $p$ in the context of a transaction $t_i \in T$ is the a priori subjective joint probability distribution function of the critical rating vector $R_p^c\left(t_i\right)$ from the perspective of $p$. Please note that it is for notational simplicity that the critical rating vectors $R_c^p\left(t_i\right)$ and $R_p^c\left(t_i\right)$ are denoted by $R$ (without the full complement of the subscripts) in the denotation of the trustworthiness.

The above definitions have a number of interesting properties which correspond with the intuitive properties of trust in our everyday life such as trustworthiness is subjective, and it is defined relative to a particular set of critical attributes; trustworthiness is defined at a given point in time, and it is defined as a probability distribution. Some other important intuitive attributes of trust are that trust has duality - it is subjective and objective; that is some of the critical attributes are subjectively measureable and others are objectively measureable; trust is not always symmetrical, that is, A trusts

B, does not always mean B trusts A; and trust is dynamic, that is, trust is related to context and temporal factors [16].

We now consider the level of trust in a service - this time not the service providers but rather the services themselves. Since a particular service may come into fruition as a combination of resources and services from multiple providers, each service's trust level must be assessed as an autonomous entity even though this trust level is a function of the composite trust level of the providers and the base services from which it has been derived. So we define the trust level of a service $s$ as:

The level of trust $T_c^s\left(t_i\right)$ of a service consumer $c$ in a service $s$ in the context of a transaction $t_i \in T$ is the a priori probability that the utility of $c$ will meet or exceed its minimum threshold of satisfaction $u_0$ at the end of transaction $t_i$, given $c's$ perceived trustworthiness of service $s$.

$T_c^s\left(t_i\right) = \int\limits_{U_c(R)\geq u_0} \tau_c^s\left(R,t_i\right).dR$, where $U_c(R)$ is the utility function of service consumer $c$; and $\tau_c^s\left(R,t_i\right)$ - the trustworthiness of service $s$ as perceived by consumer $c$ in the context of a transaction $t_i \in T$ is the a priori subjective joint probability distribution function of the critical rating vector $R_c^s\left(t_i\right)$ from the perspective of $c$.

The application of this trust modeling for the OCS platform and its usage will be demonstrated in Sections IV and V which respectively discuss the components of the OCS trust management architecture, and the OCS trust model verification by demonstrating how it is applied to an IaaS and SaaS usage scenarios.

### C. Trust Production Approaches

There are generally three basic ways by which communities of interacting entities go about the issue of trust generation. These are: norms backed up by institutional guarantees, indirect cues, and reputational information [15]. *Norms and institutional guarantees* attempt to reduce the uncertainty on the behavior of other agents by prescribing specific allowed behavioral ranges (which usually correspond to satisfactory outcomes for the majority of transaction types and society members) and by providing institutions, which prevent deviations or make such deviations highly unlikely because of quick detection and effective sanctions [17]. *Indirect cues* are attributes of an agent, which we have associated with certain likely behaviors based on our experience, intuition and training. *Reputational information* is information about, or observations of an agent's past behavior on similar situations that is aggregated and distributed by means of word-of-mouth or through trusted third parties, such as credit rating agencies, consumer reports, etc. Reputational information can help agents construct estimates on another agent's trustworthiness under the assumption that agents have an underlying distribution of behavior, which is relatively stable over time[15].

Trust production through indirect cues is however inapplicable in online communities because this is mostly obtained by interacting agents through observing the body language and appearance of the other party they are transacting with. The application of institutional guarantees is also normally infeasible in cyberspace because of the anonymity enjoyed by the interacting entities and the lack of appropriate institutions or their inability to enforce adequate guarantees. However, findings from our previous work has shown that the major approaches to trust engineering in cloud computing are that, employing trusted computing technologies and reputation based approaches are two key approaches to trust engineering in the cloud computing marketplace. Also trusted third party approaches and the deployment model play a significant part in enhancing trust between service providers and their consumers. In the context of trust engineering for OCS environments, two of these come handy. These are the trusted third party based and the reputation based approaches.

#### 1) Trusted Third Party Based

The OCS platform by its very nature can serve as a trusted third party for enhancing trust between service contributors and the utilizers of these services by providing and enforcing OCS institutional guarantees for the allowed normal behavior on the platform. This could be achieved through for example standardized Pseudo SLA templates (see **Section IV** for the discussion on the implementation and management of the Pseudo SLA system and its associated templates), and the detection of problematic services, service providers, and service consumers. Appropriate actions can then be taken to subsequently penalize the offending entities.

#### 2) Reputation Based

The platform also lends itself to the gathering, management, and analysis of reputation information about services, service providers and service consumers. This information can then be used in supporting entities on the platform in making decisions about the trustworthiness of other entities they are interested in transacting with.

#### 3) Trust Values

The platform will provide the following trust values to its members in facilitating their decision making process. The trust values of interest are: trust value of a service or a resource, trust value of the service category to which the service/resource belongs, trust value of service/resource providers, composite trust value of the group to which service /resource provider belongs, and trust value of resource consumers. This list is however extensible as new needs concerning trust values arise.

## III. OCS TRUST MANAGEMENT SYSTEM

Some of the desired properties of the OCS trust management system are reliability, robustness, scalability, and usability. The verification of the system in this work will however focus on its applicability and usability. Demonstration of the other desired properties will be part of our future work when the responsible components are fully implemented.

### A. Main Elements of trust management systems

The generic operations of trust management include expectation, data monitoring, data management, analysis, and decision making.

#### 1) Expectation:

This is a function of the service user's requirement specification, and service provider's declared intensions of supported quality of service. This is normally expressed in SLAs that the cloud service provider and their customers enter into. There is a need for expectation management from both ends to ensure smooth arbitration in case of any breaches in the agreement. Since no formal agreement exists between the service providers and the users of these services, this expectation management is expressed on the OCS platform as Pseudo SLA.

We adapt some of the stages of the work of [18] on the usage of public SLA templates in cloud markets to develop the creation and usage of pseudo SLA templates for the OCS platform. The implementation and management of the pseudo SLA concept on the OCS platform is in three main parts: the creation of pseudo SLA templates for categories of services by the OCS platform, the pseudo SLA specification by service providers, and the quality requirement specification by potential service users. These main parts are respectively outlined below.

*a) Existing OCS platform pseudo SLA templates*

i. Platform administrators create pseudo SLA (pSLA) templates for service categories and their sub-categories. This normally will be based on the cloud service type such as IaaS, PaaS, SaaS, DaaS, etc.
ii. Request SLAs (rSLAs) may be promoted to OCS pSLAs by the OCS platform administrators
iii. Each pSLA is stored in the SLA repository.

*b) Service providers pseudo SLA specification*

The service providers' declared intentions are specified in service SLAs.

i. The service provider specifies the category of service (and may also specify some critical attribute values), and it is then presented with a list of existing pSLA templates that the OCS platform considers as suitable templates.
ii. The service provider assigns the service to a pSLA template.
iii. The service provider creates the service SLA (sSLA) for the service by accepting a presented pSLA template or modifying it to create the sSLA to meet the specification of this service.
iv. The new service SLA (sSLA) is stored in the service SLA repository.

*c) Service users' quality requirements specification*

i. A potential service user specifies the service category together with the values of some critical attributes of the potential service to use.
ii. The user is presented with possible matching sSLAs from the service SLA repository.
iii. The user may accept one of these sSLAs and proceed to use the service or create a request SLA (rSLA) based on one of the sSLAs to suit its usage requirements.
iv. The new rSLA is stored in the request SLA repository which can be presented as an OCS pSLA template to service providers during the service SLA creation stage of the service creating process.

Fig.1 shows the processes involved in the pseudo SLA implementation and management.

### 2) Data Monitoring and Management

The data monitoring and management is concerned with what kinds of data should be monitored and the associated cost of monitoring. It is therefore responsible for defining new trust data that need to be monitored on the OCS platform. The kind of data that should be monitored should facilitate the gathering of information for computing the necessary trust values. These are the trust value of services and the trust value of the service category to which they belong; trust value of service providers, trust value of service consumers, and the composite trust value of the group to which service providers belong. Examples of some of the data that needs to be monitored in the case of computing the trust values of services are: how long the service has been in operation, percentage service uptime, probability distribution of service failures, and user ratings of the service. Some of the important parameters that need monitoring in the case of service providers are: how long the provider has been providing services, the services that the provider has been providing, trust values of all the services the provider has been providing, and the probability distribution of the failure of its services. Some of the parameters of monitoring interest in computing trust value of service consumers are service usage patterns, and reports of malicious behavior.

The data management also deals with defining data storage policies such as for example local storage of trust matrix by members, storage of member interactions by the OCS platform, the type of communication and exchange of information, and what type of data are to be exchanged.

### 3) Data Analysis

The data analysis is concerned with the computation of the requisite trust values based on the information from the expectation of consumers and providers of services.

*a) Cloud Computing Parameters of Trust*

When selecting a cloud service provider, multiple important parameters that are of relevance to the cloud service consumer need to be identified properly. Also, there is need for mechanisms to measure those parameters and aggregate these measurements based on the customers' preference regarding the importance of the parameters[19]. References [20] and [19] have identified several of these parameters which have been categorized into quality of service related, security and privacy related, risk management related, and reputation related attributes. These parameters (attributes) are termed critical attributes; more formerly, a *critical attribute* of a service provider $s$ , from the perspective of a service consumer $c$ , in the context of a transaction $t_i \in T$. is an attribute whose value affects the utility of $c$ and is contingent upon the behavior of $s$ in the course of transaction $t_i$ [15].

*b) Reputation System (computation of reputation)*

The trustworthiness, $\tau_c^p\left(R,t_i\right)$ (reputation, if trustworthiness is from only reputational information) of service provider $p$ as perceived by consumer $c$ in the context of a transaction $t_i \in T$ is the a priori subjective joint probability distribution function of the critical rating vector $R_c^p\left(t_i\right)$ from the perspective of $c$ . Instead of a single value rating, we have rating of metrics of intent, integrity, capability and results of the critical attributes of the entity to be trusted, from which our model computes the trustworthiness; trust values are then computed from this. That is, for each critical attribute (e.g. accuracy, reliability, etc.) in the critical rating vector $R_c^p\left(t_i\right)$ that has been identified from the rSLA of the service consumer, the reputational ratings are based on the intent, integrity, capability and results; where intent constitutes information about declared agendas about what entities promise to provide through their services. Integrity constitutes information about honesty; this is a measure of, to what extent entities deliver on what they promised. Capability constitutes information about owned or outsourced resources (what assets parties have); and finally, results constitute information about products and services that entities specialized in through consistently delivering these products and services satisfactorily to their clients [21].

### 4) *Decision Support*

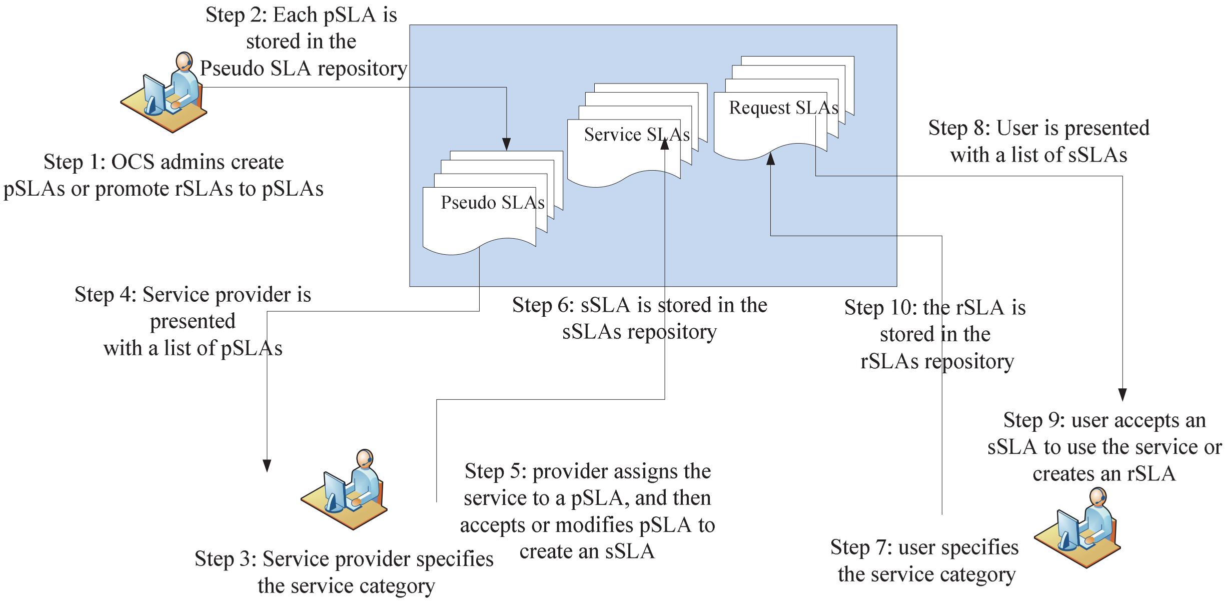
The purpose of the users of a trust management system is to make decisions concerning engaging in particular transaction at a point in time. It is imperative to provide an intuitive representation of trust values for all the types of users of the OCS platform. This will be handled by the Decision Support Manager as is discussed later in the next section.

## IV. OCS TRUST MANAGEMENT ARCHITECTURE

The major components of the OCS trust management system are Expectation Manager (EM), Platform Guarantees Enforcement Manager (PGEM), Data Monitoring Manager (DMoM), Data Management Manager (DMaM), Trust Analysis Manage (TAM), and Decision Support Manager (DSM). Fig. 2 shows the relationship between these components.

### A. *Expectation Manager*

The expectation manager is responsible for handling the creation and maintenance of the OCS Platform pseudo SLA (pSLA) templates, the service provider assignment of services to a particular pSLA template, and the creation of service SLA (sSLA) to meet the specification of each service. It is also responsible for presenting potential service users with possible matching sSLAs from the service SLA repository which they may accept and proceed to use the service or create a request SLA (rSLA) based on one of the sSLAs to suit their specific usage requirements.



Step 2: Each pSLA is stored in the Pseudo SLA repository
Step 1: OCS admins create pSLAs or promote rSLAs to pSLAs
Pseudo SLAs
Service SLAs
Request SLAs
Step 8: User is presented with a list of sSLAs
Step 4: Service provider is presented with a list of pSLAs
Step 6: sSLA is stored in the sSLAs repository
Step 10: the rSLA is stored in the rSLAs repository
Step 9: user accepts an sSLA to use the service or creates an rSLA
Step 5: provider assigns the service to a pSLA, and then accepts or modifies pSLA to create an sSLA
Step 3: Service provider specifies the service category
Step 7: user specifies the service category

Fig. 1: Processes involved in the pseudo SLA implementation and management

### B. *Platform Guarantees Enforcement Manager*

This module is responsible for ensuring good and acceptable behavior on the platform. It applies appropriate sanctions to undesirable behaviors on the platform. It is therefore responsible for malicious conditions detection and the detection of SLA violation, and then taking appropriate remedial actions such as removing offending services from the platform and banning offending users.

### C. *Data Monitoring Manager*

The DMoM is responsible for defining new trust data that needs to be monitored on the OCS platform in order to accommodate for adapting the platform to future needs such as when new service categories and categories of trust values are needed to be computed.

### D. *Data Management Manager*

The data management manager is responsible with defining data storage policies such as for example local storage of trust matrix by members, storage of member interactions by the OCS platform, the type of communication and exchange of information, and what type of data are to be exchanged. It also deals with data reliability, security, recovery in case of problems and maintaining consistency in situations of discrepancies in data from multiple sources.

### E. *Trust Analysis Manager*

This module makes the analyses of the trust values to be computed from information from the expectation manager and the available data from the data management module. It then computes the necessary trust values. It is evident from our model of the definition of trust on the OCS platform that we are interested in personalized trust values for the entities on the OCS platform. For example a potential service consumer will be interested in computing its perceived trust value for a service and its perceived trust value of the provider(s) of that service. This is achieved with equations (1) and (2), and the trust values computation algorithm as described below.

The level of trust $T_c^p(t_i)$ of a service consumer $c$ for a service provider $p$ in the context of a transaction $t_i \in T$ is the a priori probability that the utility of $c$ will meet or exceed its minimum threshold of satisfaction $u_0$ at the end of transaction $t_i$, given $c's$ perceived trustworthiness of service provider $p$.

$T_c^p(t_i) = \int\limits_{U_c(R)\ge u_0} \tau_c^p(R,t_i).dR$ , ------- (1) where $U_c(R)$ is the utility function of service consumer $c$ ; and $\tau_c^p(R,t_i)$ - the trustworthiness of service provider $p$ as perceived by consumer $c$ in the context of a transaction $t_i \in T$ is the a priori subjective joint probability distribution function of the critical rating vector $R_c^p(t_i)$ from the perspective of $c$ .

The level of trust $T_c^s(t_i)$ of a service consumer $c$ for a service $s$ in the context of a transaction $t_i \in T$ is the a priori probability that the utility of $c$ will meet or exceed its minimum threshold of satisfaction $u_0$ at the end of transaction $t_i$ , given $c's$ perceived trustworthiness of service $s$ .

$T_c^s(t_i) = \int\limits_{U_c(R)\ge u_0} \tau_c^s(R,t_i).dR$ , ------- (2) where $U_c(R)$ is the utility function of service consumer $c$ ; and $\tau_c^s(R,t_i)$ - the trustworthiness of service $s$ as perceived by consumer $c$ in the context of a transaction $t_i \in T$ is the a priori subjective joint probability distribution function of the critical rating vector $R_c^s(t_i)$ from the perspective of $c$ .

### *1) Trust Values Computation Algorithm*

1. Identify service dependencies from rSLA
2. Compute trust value of the service based on eq. (2)
3. Compute trust value (based on eq. (1)) of each of the service providers contributing to this service
4. Compute composite trust value, $T_c^{comp}(t_i)$ based on

$$T_c^{comp}(t_i) = \omega_s T_c^s(t_i) + \omega_{p1} T_c^{p1}(t_i) + \omega_{p2} T_c^{p2}(t_i) + \\ \ldots + \omega_{pn} T_c^{pn}(t_i)$$
$$where, \omega_s + \omega_{p1} + \omega_{p2} \ldots + \omega_{pn} = 1$$

5. For each service dependencies in step 1, repeat steps 2, 3 and 4
6. Compute the overall composite trust value by applying the appropriate dependency level weight( $\omega_l$ ) based on the level of the dependency in the dependency chain for all composite trust values as in step 4

$$T_c^{total}(t_i) = \omega_1 T_1^{comp}(t_i) + \omega_2 T_2^{comp}(t_i) + \ldots + \omega_l T_l^{comp}(t_i) ,$$
$$\text{where } \omega_1 + \omega_2 \ldots + \omega_l = 1$$

To compute $T_c^{comp}(t_i)$ from equations (1) and (2), we need the utility function $U_c(R)$ , the critical rating vectors $R_c^p(t_i)$ and $R_c^s(t_i)$ , the weights( $\omega_s$ , $\omega_p$ and $\omega_l$ ) , and the trustworthiness (reputation) distribution functions $\tau_c^p(R,t_i)$ and $\tau_c^s(R,t_i)$ . The critical rating vectors are internal properties of $c$ which is specified in or can be derived from its rSLA; and the utility function is also an internal property of $c$ which it can provide for the algorithm to compute the trust values. All the weights, $\omega_s$ , $\omega_p$ and $\omega_l$ , are parameters that are determined by the OCS platform. The remaining required parameters are the reputation distribution functions; and these are also available from the data monitoring and data management modules on the OCS platform as reputation data. The only issue left is the need to convert the reputation data that have been collected into standardized probability distribution functions to simplify the trust value computations and provided tractable solutions.

### *2) Approximation of reputation (trustworthiness)*

The approximation of the reputation data to standard known probability distribution functions (e.g. Uniform distribution, normal distribution, etc.), can be achieved using appropriate curve fitting algorithms. The reputation data gathered by the DMoM and DMaM modules can be approximated to standard probability distributions function using curve fitting algorithms such as polynomial (linear, quadratic, 3rd order, etc.), and logarithmic algorithms.

## *F. Decision Support Manager*

The decision support manager is responsible for taking results from the trust value computations of the analysis manager and presenting it in a format that simplify visualization for the users. The user-friendly trust value representation together with making recommendations on decisions to be taken by users should facility their decision making process.

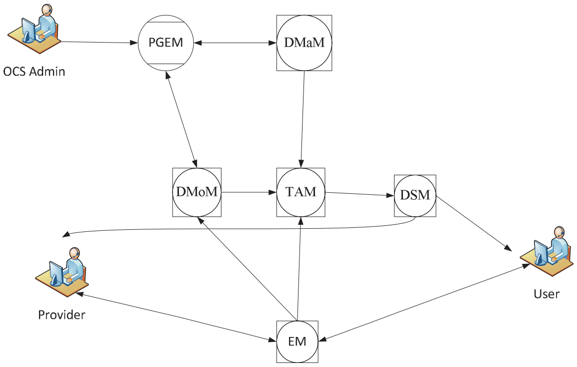


Fig. 2: OCS trust management systems architectural components

## V. OCS TRUST MODEL VERIFICATION

Two usage scenarios are employed for demonstrating the applicability of the OCS trust model and the OCS trust management system. One of the usage scenarios will be on an IaaS and the other on SaaS. The verification process seeks to demonstrate the applicability of the trust model and trust management system to standard cloud computing services.

## *A. IaaS usage scenario*

We look at the case where an OCS member has spare storage space at its data center and has virtualized this storage space to provide a storage service on the OCS platform. The standardized pSLA that may apply to such a service is show in table 1 below.

Table 1: pSLA template for IaaS

| Attribute | value types |
|---|---|
| Service Identification | Service ID & category ID |
| Service Type / category | IaaS & category ID |
| Availability | 50 % uptime |
| Service support | No |
| Service support type | N/A |
| Maintenance notification | Yes |
| SLA dependencies | {} |
| Service location | {} |
| Security | None |
| Data encryption | None |
| Privacy | None |
| Certification | {} |

*1) sSLA created from a pSLA*

In this case, the service provider may for example, change only the availability level to 95%, and provide security support of data backup and recovery as shown in table 2.

Table 2: sSLA created from a pSLA

| Attributes | value types |
|---|---|
| Service Identification | Service ID & category ID |
| Service Type / category | IaaS & category ID |
| Availability | 95 % uptime |
| Service support | No |
| Service support type | N/A |
| Maintenance notification | Yes |
| SLA dependencies | {} |
| Service location | {} |
| Security | Data backup & recovery |
| Data encryption | None |
| Privacy | None |
| Certification | {} |

*2) rSLA created from sSLA or pSLA*

A service user may accept this sSLA in order to use the service or request for additional requirements in its rSLA.

*3) Computation of trust values*

i. Extraction of critical rating vectors from the rSLA If the user in interested in service availability, security and maintenance notification as its criteria for using then service, then the critical rating vector is given by

$$R_c^s(t_i) = \{availability, \mathrm{sec}urity, maintenance\ notification\}$$

ii. Extraction of users' utility function

Assuming the user's utility function increases monotonically with availability above 90% given that security and maintenance notification are provided, then the utility function is given by

$$U_c(R) \geq 90\%$$

iii. Approximation of reputational information into standardize probability distribution functions

We look at two cases, one in which the trustworthiness $\tau_c^s(R,t_i)$ approximates a uniform distribution function, and the second approximates a normal distribution. The empirically collected data is approximated to standard distribution functions using appropriate curve fitting algorithms as mentioned in Section IV.E.2 above.

We compute the trust level of the service when $\tau_c^s(R,t_i)$ is a uniform distribution with parameters $U(a,b)$

$$T_c^s(t_i) = \int_{0.9}^{1} \tau_c^s(R,t_i).dR = 1 - \int_{-\infty}^{0.9} \tau_c^s(R,t_i).dR = 1 - \Phi(\frac{0.9-a}{b-a})$$

a. We compute the trust level of the service when $\tau_c^s(R,t_i)$ is a normal distribution with parameters

$$N(\mu,\sigma) = \frac{1}{\sqrt{2*pi*\sigma^2}} * \exp(\frac{-(x-\mu)^2}{2*\sigma^2})$$

$$T_c^s(t_i) = \int_{0.9}^{1} \tau_c^s(R,t_i).dR = 1 - \int_{-\infty}^{0.9} \tau_c^s(R,t_i).dR = 1 - \Phi(\frac{0.9-u}{\sigma})$$

Fig. 3 shows the characteristic curves of the trust level against the lower limit in the above scenario when the upper limit of the availability is 100%. It shows for the uniform distribution and a normal distribution curve with standard deviations equal to that of the uniform distribution.

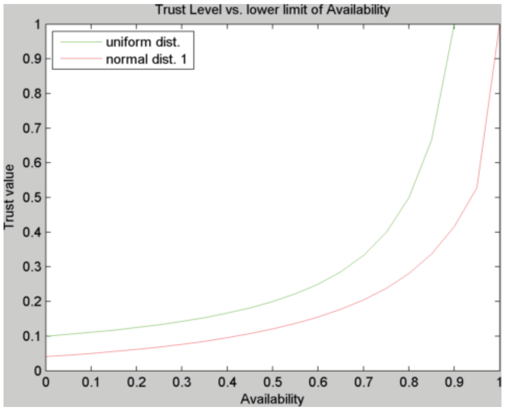


Fig. 3: Trust Level against varying Lower limit of availability

Similarly the trust level for the provider of the services is computed in the same way us above. Fig. 4 shows the composite trust level in our scenario with varying service weight, where the trust level of the service is from the uniform distribution as above and the trust level of the provider is the normal distribution with the same standard deviation as that of the uniform distribution.

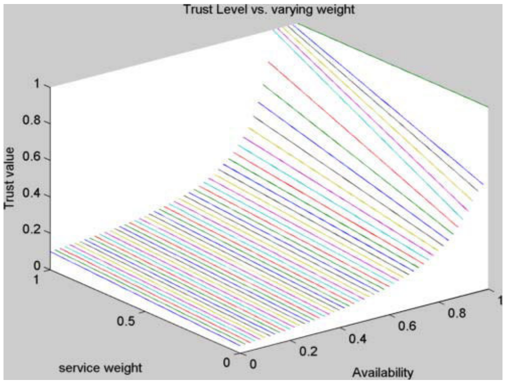


Fig. 4: Trust Level against varying service weight

Fig. 5 show the composite trust level in our scenario with varying user utility, where the trust level of the service is from the uniform distribution as above and the trust level of the provider is the normal distribution with the same standard deviation as that of the uniform distribution. Fig.5a shows when the service and the service provider have equal weight of 0.5 in the composite trust value, Fig.5b shows when the service has a weight of 1(provider has a weight of 0) in the composite trust value, and Fig.5c shows when the service has a weight of 0 (provider has weight of 1) in the composite trust value.

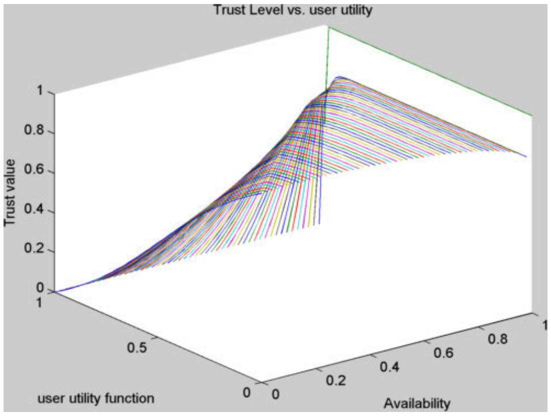


Fig. 5a: Trust level against varying user utility when the service and the service provider have equal weight of 0.5 in the composite trust value

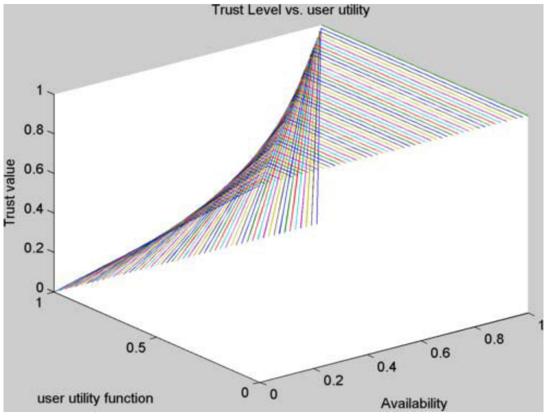


Fig. 5b: Trust level against varying user utility when the service has a weight of 1

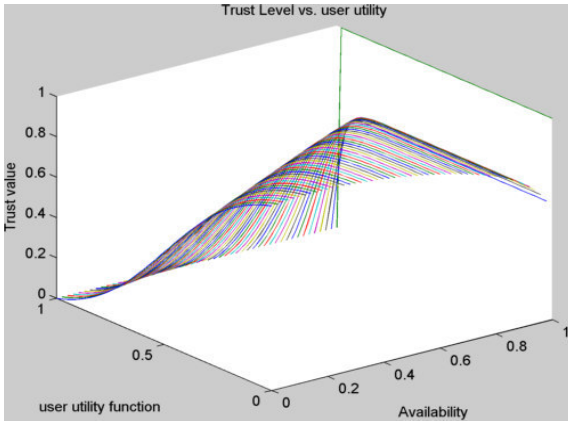


Fig. 5c: Trust level against varying user utility when the provider has a weight of 1

### B. *SaaS usage scenario*

We now show the applicability of the trust management system in the case of a SaaS usage scenario. The only difference of the two processes is only in the pseudo SLA templates. Table 4 shows a pseudo SLA template for SaaS.

Table 3: pSLA for SaaS

| Attribute | value types |
|---|---|
| Service Identification | Service ID &category ID |
| Service Type / category | SaaS & category ID |
| Availability | 50 % uptime |
| Service support | No |
| Service support type | N/A |
| Maintenance notification | Yes |
| SLA dependencies | {} |
| Service location | {} |
| Security | None |
| Data encryption | None |
| Privacy | None |
| Certification | {} |
| Performance (Throughput) | 1Kbps |
| Performance(Response time) | 5sec |

The user's utility increases monotonically with availability above 85% and response time below 2sec $R_c^s\left(t_i\right) = \{availability, security, response\_time\}$

#### 1) *sSLA created from pSLA*

A service provider may accept this pSLA to create the sSLA for the service.

#### 2) *rSLA created from sSLA or pSLA*

A service user may accept this sSLA in order to use the service or request for additional requirements in its rSLA.

The computation of the trust values follow the same process as is in the case of the IaaS above.

## VI. CONCLUSION AND FUTURE WORK

This paper has looked at the design of a trust management system for OCS platforms. It has modeled trust for the OCS platform, designed a trust management system for OCS platforms, and verified the trust model and the trust management system through the simulation of the computation of the trust values with IaaS, and SaaS examples.

Even though our trust management systems contain the complete elements, we have focused mainly of the modeling of trust for the OCS platforms and the trust analysis components in our architecture. The other aspects require further work in terms of the implementation of the data monitoring and data management components. Secondly the decision support system and usability of the pseudo SLA templates in the system needs some further work for their verification. These further works will also require verifying the robustness and scalability of the trust management system.

## REFERENCE

[1] E. Kuada and H. Olesen, “A Social Network Approach to Provisioning and Management of Cloud Computing Services for Enterprises,” presented at the CLOUD COMPUTING 2011, The Second International Conference on Cloud Computing, GRIDs, and Virtualization, 2011, pp. 98–104.

[2] E. Kuada and H. Olesen, “Incentive mechanisms for Opportunistic Cloud Computing Services,” in *2012 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, 2012, pp. 127 –136.

[3] E. Kuada, H. Olesen, and A. Henten, “Public Policy and Regulatory Implications for the Implementation of Opportunistic Cloud Computing Services for Enterprises,” in *Workshop on Security in Information Systems*, Wroclav, 2012.

[4] E. Kuada, K. Adanu, and H. Olesen, “Cloud Computing and Information Technology Resource Cost Management for SMEs,” in *Proceedings of IEEE Region 8 Conference EuroCon 2013*, University of Zagreb, Croatia, 2013, pp. 258 – 265.

[5] W. Jansen and T. Grance, “Guidelines on security and privacy in public cloud computing,” *NIST Special Publication*, pp. 800–144, 2011.

[6] L. Badger, T. Grance, R. Patt-Corner, and J. Voas, “Draft cloud computing synopsis and recommendations,” *NIST Special Publication*, vol. 800, p. 146, 2011.

[7] P. Mell and T. Grance, “The NIST definition of cloud computing (draft),” *NIST special publication*, vol. 800, p. 145, 2011.

[8] Z. Song, J. Molina, and C. Strong, “Trusted Anonymous Execution: A Model to Raise Trust in Cloud,” in *2010 9th International Conference on Grid and Cooperative Computing (GCC)*, 2010, pp. 133 –138.

[9] M. Almorsy, J. Grundy, and A. S. Ibrahim, “TOSSMA: A Tenant-Oriented SaaS Security Management Architecture,” in *2012 IEEE 5th International Conference on Cloud Computing (CLOUD)*, 2012, pp. 981 –988.

[10] W. Viriyasitavat and A. Martin, “Formal Trust Specification in Service Workflows,” in *2010 IEEE/IFIP 8th International Conference on Embedded and Ubiquitous Computing (EUC)*, 2010, pp. 703 –710.

[11] B. Grobauer, T. Walloschek, and E. Stocker, “Understanding Cloud Computing Vulnerabilities,” *IEEE Security Privacy*, vol. 9, no. 2, pp. 50 –57, Apr. 2011.

[12] X. Zhang, H. Liu, B. Li, X. Wang, H. Chen, and S. Wu, “Application-Oriented Remote Verification Trust Model in Cloud Computing,” in *2010 IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom)*, 2010, pp. 405 –408.

[13] J. Abawajy, “Establishing Trust in Hybrid Cloud Computing Environments,” in *2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2011, pp. 118 –125.

[14] M. Kuehnhausen, V. S. Frost, and G. J. Minden, “Framework for assessing the trustworthiness of cloud resources,” in *2012 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, 2012, pp. 142 –145.

[15] C. Dellarocas, “The Design of Reliable Trust Management Systems for Electronic Trading Communities,” in *SLOAN SCHOOL OF MANAGEMENT, MIT, 2000*, 2001.

[16] C. Shen, H. Zhang, H. Wang, J. Wang, B. Zhao, F. Yan, F. Yu, L. Zhang, and M. Xu, “Research on trusted computing and its development,” *Sci. China Inf. Sci.*, vol. 53, no. 3, pp. 405–433, Mar. 2010.

[17] T. Parsons, *The Social System: (New York): The Free Press of Glencoe*. Collier-Macmillan, 1964.

[18] I. Breskovic, M. Maurer, V. C. Emeakaroha, I. Brandic, and S. Dustdar, “Cost-Efficient Utilization of Public SLA Templates in Autonomic Cloud Markets,” in *Proceedings of the 2011 Fourth IEEE International Conference on Utility and Cloud Computing*, Washington, DC, USA, 2011, pp. 229–236.

[19] S. M. Habib, S. Ries, and M. Muhlhauser, “Cloud Computing Landscape and Research Challenges Regarding Trust and Reputation,” in *2010 7th International Conference on Ubiquitous Intelligence Computing and 7th International Conference on Autonomic Trusted Computing (UIC/ATC)*, 2010, pp. 410 –415.

[20] G. Zhao, C. Rong, M. G. Jaatun, and F. E. Sandnes, “Reference deployment models for eliminating user concerns on cloud security,” *J Supercomput*, vol. 61, no. 2, pp. 337–352, Aug. 2012.

[21] H. Salah and M. Eltoweissy, “Towards a personalized trust management system,” in *2012 International Conference on Innovations in Information Technology (IIT)*, 2012, pp. 373 –378.

## Appendix 5: Paper #5

Kuada, Eric, Henning Olesen, and Anders Henten. 2012. “Public Policy and Regulatory Implications for the Implementation of Opportunistic Cloud Computing Services for Enterprises.” In *9th International Workshop on Security in Information Systems*, pp. 3–13. Wroclaw, Poland.

# Public Policy and Regulatory Implications for the Implementation of Opportunistic Cloud Computing Services for Enterprises

Eric Kuada, Henning Olesen and Anders Henten
Center for Communication, Media and Information Technologies, Aalborg University,
Sydhavnsgade 17, Copenhagen, Denmark
{kuada, olesen, henten}@cmi.aau.dk

**Abstract.** Opportunistic Cloud Computing Services (OCCS) is a social network approach to the provisioning and management of cloud computing services for enterprises. This paper discusses how public policy and regulations will impact on OCCS implementation. We rely on documented publicly available government and corporate policies on the adoption of cloud computing services and deduce the impact of these policies on their adoption of opportunistic cloud computing services. We conclude that there are regulatory challenges on data protection that raises issues for cloud computing adoption in general; and the lack of a single globally accepted data protection standard poses some challenges for very successful implementation of OCCS for companies. However, the direction of current public and corporate policies on cloud computing make a good case for them to try out opportunistic cloud computing services.

## 1 Introduction

Chief Information Officers (CIOs) and Information Technology (IT) managers have over the past two decades successfully handled overseeing the convergence of voice and data networks to support the business process of their organisations. IT departments currently face the daunting task of ensuring compliance, data security and cutting down on operational cost amidst tight spending budgets. There is also a growing expectation from Chief Executive Officers (CEOs) and boards of directors for information technology's mission to quickly expand from cost cutting to revenue generation. Enabling revenue growth involves aligning IT strategy and capabilities with the overall business objectives mainly through knowledge management and supporting collaboration with partners.

As CIOs wake up to their new corporate mandate, they are seeking to tap organisational and technical solutions to improve IT's fit with business objectives which is also likely to involve the painful process of the decentralisation of IT departments authority and functions to other business units [15].

But for compliance and data security challenges which are generating serious policy and regulatory challenges for cloud computing adoption, it seems an excellent solution to the challenges that IT departments currently face. Cloud Computing [12],[11] 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 and released with minimal management effort or service provider interaction.

Perhaps the most paramount factor for the adoption of cloud computing by any organisation is that of cost reduction in the purchase and operation of IT solutions. It obviates the need for huge initial capital expenditure involved in the acquisition of IT infrastructure and solutions which are normally underutilised or may later not be suitable for their intended purpose; and allow for the payment for resources as per their actual usage. Cloud computing also offers the agility, scalability and elasticity that IT departments require in coping with changing business needs of the organisations that they serve. These factors together with the transformation of the roles of IT departments and their staff that cloud computing bring to organisations result in efficient IT service delivery.

Enabling revenue growth comes from the transformation of IT departments' roles, promoting business-to-business (B2B) integration with partners and creation of new business models. With the adoption of cloud computing services the role of IT departments must evolve from that of a service and support provider to that of certification and management of cloud services. As a result the roles of IT staff members also shift from task-oriented administrators of infrastructure and application services to that of strategic planners, project managers or business analysts who understands the company's business processes and end-user needs to support them better. The adoption of cloud computing by any two companies in general reduces the complexities involved in B2B integration. Companies can therefore leverage cloud computing by exposing their business processes to potentially large ecosystems of partners who often find ways of joining and integrating their business processes in the value chain.

Compliance risk management, security and privacy issues are however generating serious policy and regulatory challenges for cloud computing adoption. The introduction of the concept of Opportunistic Cloud Computing Services (OCCS) for enterprises will certainly not make these challenges any lighter. Because OCCS promises an accelerated adoption and further reduction in IT cost for small companies, we have been working on the feasibility of its successful implementation in terms of the technical feasibility, developing suitable incentive mechanisms to promote contribution of services to the platform, and its acceptance and support by all stakeholders. This paper discusses how public policy and regulations will impact on OCCS implementation. The rest of the paper is organised as follows: Section 2 gives an overview of cloud computing and the concept of opportunistic cloud computing services. Section 3 presents governments strategic policies towards cloud computing adoptions and the guiding regulations. Based on the results from Section 3, Section 4 discusses how these public and corporate strategies will impact on the implementation of OCCS. Section 5 concludes the paper and Section 6 touches on our future work.

## 2 Background

To put the subsequent sections in context, the first part of this section gives a brief overview of cloud computing in general and the second part briefly presents the opportunistic cloud computing services concept and its motivation. Details about it and its reference architecture can be found in [9].

## 2.1 Overview of Cloud Computing

Cloud computing is essentially the packaging of traditional information technology infrastructure and software solutions such as storage, CPU, network, applications, services, etc. as virtualized resources and delivered by a service provider to its customers as an on-demand pay-per-use self-provisioned service through a web portal over a network such as the Internet. There have been major technological advancements as well as social and business demands driving this new trend of computing. The technological factors facilitating cloud computing include the availability and drastic increase in reliable broadband Internet access, advancements in virtualization technologies and the shift of development of majority of both desktop and enterprise applications as web services and applications.

The three main components of a regular computing environment, namely the hardware infrastructure, the operating system platform and user application software, have respectively translated into Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) delivered in cloud computing. Additionally, there is an inexhaustible list of other cloud computing services due to the concept of "Anything as a Service" (XaaS) being the main driving idea of cloud computing. Thus, virtually all IT products and solutions are potential cloud computing services. These services are normally deployed in four main cloud deployment models namely public, private, community, and hybrid cloud computing deployment models [7].

A public cloud is one in which the infrastructure and other computational resources that it comprises are made available to the general public over the Internet. It is owned by a cloud service provider selling cloud services and by definition, is external to an organization. At the other end of the spectrum are private clouds. A private cloud is one in which the computing environment is operated exclusively for an organization; a private cloud may be managed either by the organization itself or a third party such as a commercial cloud services provider, and may be hosted within the organization's data centre or outside of it. The community clouds and hybrid clouds fall between public and private cloud deployment models. A community cloud is somewhat similar to a private cloud, but the infrastructure and computational resources are shared by several organizations that have common privacy, security, and regulatory considerations, rather than for the exclusive use of a single organization. A hybrid cloud deployment model is a combination of two or more of the other cloud models (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technologies that enable interoperability.

## 2.2 Overview of Opportunistic Cloud Computing Services

Opportunistic cloud computing service is a social network approach to the provisioning and management of cloud computing services for enterprises [9]. OCCS deals with the concept of enterprises taking advantage of cloud computing services to meet their business needs without having to pay or paying a minimal fee for the services. The OCCS network is a social network of enterprises collaborating strategically (possibly selfishly) for the contribution and usage of cloud computing services without entering into any business agreements. Unlike social networking services provide by social networking sites for individual use where users create their own network of friends, in an OCCS network, members do not explicitly create ties with other members but these ties come indirectly through the services and resource contribution and consumption process.

The OCCS network platform is a governing platform that serves as the social networking platform for enterprises and also includes interoperable Cloud management tools with which member enterprises can provision cloud computing services that would be used by other enterprises interested in these services. The OCCS platform thus consists of two main layers – the service layer and the management layer. The service layer consists of all the services contributed by members. These will normally be fundamental cloud computing services such as SaaS, PaaS, and IaaS; but, it can also include value added services normally provided by cloud service brokers. The management layer consists of two main components – the governance component that manages the services from members and cloud services brokerage (CSB) component that serves as an interface between the OCCS network and commercial cloud services providers and cloud service brokers.

The motivation for the OCCS concept is that there are underutilized spare resources available at some companies or organisations that can be useful to others that need them. Additionally the data centre and Cloud management competences that companies develop over time through managing their own resources can be of value to others lacking such competences (e.g. SMEs needs, especially in the developing world). OCCS also has the potential of fostering business collaborations, offering further reduction of cost in IT services and by design is compatible with future cloud computing technologies and solutions. There are currently policy and regulatory challenges for cloud computing and OCCS implementation may bring its unique challenges to these policies and regulations.

## 3 Policy and Regulatory Issues of Cloud Computing

This section presents the various strategic policies being adopted in North America, Europe, Asia Pacific and Africa. Representative countries in these regional blocks with documented publicly available government cloud computing strategic policies are presented and their policies taken as representative for that region. This approach of looking at things from a global perspective has been adopted instead of just at the national level because OCCS needs international scope to flourish. The implementation of national OCCS networks is useful; however, for participating members to find suitable services to meet their business needs, then the broader the scope of the platform the better. The section ends with a summary of the salient overriding goals driving these strategies and the main regulatory environments guiding them as is evident from these strategic policies.

### 3.1 North American Policy

The United States of America government has instituted a “Cloud First” policy to harness the benefits of cloud computing. This policy is intended to accelerate the pace at which the government will realize the value of cloud computing by requiring agencies to evaluate safe, secure cloud computing options before making any new investments [10]. Since Cloud computing can offer benefits in the cost, performance, and delivery of IT services to Federal agencies, the United States Department of Commerce anticipates that the use of Cloud Computing services will grow significantly over the next several years [16]. Cloud computing promises to have far-reaching effects on the systems and networks of federal agencies and other organizations, many of the features that make cloud computing attractive, however, can also be at odds with traditional security models and controls. The policy is therefore to guide federal departments and agencies to carefully plan their Cloud initiatives to meet the organizational and national security and privacy requirements [7].

The Canadian government cloud strategy is in two folds. The first is to position itself as a world leader in cloud computing and the second is its cloud first policy being modeled after that of United States. Due to its geographical characteristics, low-density population, IT expertise, quality construction standards, legislative framework (including the Privacy Act and the Personal Information Protection and Electronic Documents Act) and low-cost green energy, Canada considers itself a prime location for cloud computing. Government of Canada and the provinces and territories are beginning to realize Canada’s advantage and the benefits of positioning Canada as an economical and strategic choice for cloud computing [4].

### 3.2 European Union Policy

The European Union is in the process of finalizing the EU Cloud computing strategy and is yet to publish a formal document on it. The approach of leading EU member nations like Germany and France had suggested home grown national Clouds or at best an Europe first cloud policy mainly in response to the US Patriot Act; but recommendations towards the European Commission’s call for contributions to an EU strategy on Cloud Computing seems to suggest otherwise. DIGITALEUROPE urges the EU to let the EU cloud computing strategy transcend national and regional borders and rather play a leadership role in negotiating global collaboration in addressing jurisdiction and clarifying rules on law enforcements access to data stored in the Cloud; and adopt or implement policies that address actual or potential trade barriers to the evolution of cloud computing [5].

The United Kingdom government has decided to opt for a combination of private and public cloud through its G-Cloud programme after weighing the benefits of both. The Government cloud is not a single, government owned, entity; it is an on-going and iterative programme of work which will enable the use of a range of cloud services and make changes in the way it procure and operate ICT services and solutions throughout the public sector. The vision is for the government to robustly adopt a public cloud first policy; this however will not be possible in every case and there will also be a requirement for a private G-Cloud. The government will push ahead with its agenda for data centre, network, software and asset consolidation and the shift towards cloud computing. It will mandate the reuse of proven, common application solutions and policies. These solutions must balance the need to be open, accessible and usable with the growing cyber-security threat and the need to handle sensitive information with due care [2]. This will be achieved through three main projects - Data Centre Consolidation, the G-Cloud, and the Applications Store for Government. The overriding goals are reduce ICT costs, provide open competition and create a vibrant marketplace enabling the best product at the best price, create flexibility by reducing supplier lock-in to ensure users can readily switch between suppliers for ICT services, reduce time from idea to service, and reduce the carbon footprint of Public Sector ICT services [14].

### 3.3 Asia Pacific Policy

According to a study by Frost & Sullivan in 2010 on the Asia Pacific region, 21 percent of respondents in the government sector have adopted cloud computing in one form or the other. Furthermore, it also revealed that given the governments concerns around security of data and location of data centres, private and hybrid clouds are witnessing significantly higher adoption in the region [3]. Though countries in the region are at different stages of forming a cloud strategy their current initiatives give an indication of their cloud policies to be adopted.

Some of these initiatives are: despite the quite circumspective approach the Australian Government has taken in adopting cloud computing primarily due to their uncertainty over storing data in offshore data centres, given the shrinking ICT budgets certain agencies have gone ahead to try out cloud computing services. The Chinese government besides other projects through its “cloud factory” project is providing adequate computing resources to enterprises which are mainly start-ups without the financial power to acquire the required IT assets. Other countries in the region have similar policies centred on cost reduction, green IT, developing local expertise, research on cloud services, providing subsidies to enterprises to boost industry participation, and perhaps most importantly building national private clouds due to concerns of data security and jurisdiction of location of data centres.

### 3.4 African Policy

The African cloud computing strategic policy as of now seems to be “no policy” and there is no direct effort from governments to create one. The closest documented action towards a cloud computing policy for Africa is an ITU organized forum held in Rwanda which came up with two main recommendations towards a cloud computing strategy for Africa. The first is physical interconnections between African countries via broadband networks are a prerequisite for successful implementation of cloud computing applications and systems. African States must take the necessary action to develop broadband in Africa, improve national, regional and international connectivity. The second is that in order to take advantage of the opportunities afforded by cloud computing while minimizing risks, African States must have a coordinated and coherent approach to their adoption of cloud computing and transition to it. In that regard, they must adopt guidelines on the strategy for the transition to cloud computing, capacity-building programmes, the harmonization of legislative and regulatory reference frames, the adoption of data centre selection criteria, and attracting investment and seizing business opportunities [6].

Cost reduction and efficiency in the delivery of IT services is a central motivation in all the policies discussed above; on the other hand meeting security and privacy requirements for the protection of both national and citizen data are major issues that the policies try to address. It is however also evident that the national and regional strategies being adopted are dependent on the economic environment, their current ICT policy, data protection laws and current infrastructure; the different regions are therefore at different levels of the evaluation and deployment of cloud computing services. For example because the major cloud services vendors and providers are of USA origins coupled with the Patriot Act that compels these companies to release information or data stored on the service providers platforms to their law enforcement agencies irrespective of location where the data is stored makes the USA a little more open to public cloud services than the rest of the world, particularly Europe which has tighter regulations of the protection of privacy of citizen data.

## 4 Impacts of Policy and Regulations on OCCS

Governments' adoption of commercial cloud services reduces potential resources that would have been made available to OCCS. With the current state of highly underutilized resources in government data centres, these would have been perfect resources to have been contributed to the OCCS platform. However, with the governments already adopting commercial cloud computing services these spare resources diminish. A major part of government cloud computing services adoption is in building their private clouds. This has been necessitated by fears of commercial cloud services providers not meeting the security requirements for national information and privacy of citizen data. Even when commercial cloud services providers have been contracted to provide these cloud computing services, governments have insisted that the services be hosted within the borders of their own country.

It is evident from this that most cloud service needs by the public sector of most governments cannot generally be provided by an OCCS network with international scope. Even for those public sector applications that are found to be suitable for utilizing OCCS resources, the OCCS platform will have to provide commercial grade reliability, security and privacy guarantees for it to be useful for government public sector consumption of these services.

The national and regional strategies with focus on investing in cloud computing services to support start-up companies in meeting their IT resources requirements and the promotion of industry participation through subsidies will find OCCS a very useful approach to take. Instead of a direct government investment in providing cloud computing services to these companies, a conducive environment can rather be created for other companies to provide such resources. Government strategies on the promotion of Digital Business Ecosystems, and development of local expertise in cloud computing will also find OCCS a very useful approach. National OCCS networks can be created whereby such cloud computing management competences are developed through the provisioning of cloud services to other companies. The resulting OCCS business ecosystems will also catalyse business growth through new start-up companies and fostering business collaborations.

The promotion of OCCS implementation inherently resolves cloud computing standards, interoperability and vendor lock-in issues. The current cloud computing industry is still dominated by proprietary technologies from the leading cloud service vendors and cloud management tool developers. This makes interoperability a serious issue defeating the purpose of flexibility and freedom of choice that cloud computing is supposed to provide to users and thereby resulting in vendor lock-in that users have to grapple with when it becomes necessary to change their cloud service provider. As has been indicated by [9] a successful implementation of an OCCS network must provide support for the management of fundamental cloud computing services, support for the management of any arbitrary cloud computing service, interoperability with major cloud computing standards and cloud computing management tools, and support for future cloud management technologies. Thus to start with, the OCCS concept must carefully follow cloud computing standards; the situation is however reversed as OCCS network implementations become successful. Thus those standards that are dominant on the OCCS platform will then be followed closely by cloud management tool developers and cloud service providers. This will further promote the success of the OCCS platform; and hence the promotion of cloud computing standardization and promotion of the OCCS implementations will be in a virtuous cycle.

The move by companies to the adoption of cloud computing is a senior management level (the CIO, Head of IT or IT Director) driven strategic technology shift for organizations as they look to lower costs and evolve their computing models to deliver competitive advantage to their businesses. The majority of 46 percent in an AMD sponsored research in 2011 on the adoption trends of cloud computing stated it was a strategic shift in IT policy for the organization with just 19 percent describing it as a cost-saving necessity, and with 35 percent stating it is a tactical move to address a specific need [13]. It is however important to note that even though less than a fifth find it a cost-saving necessity, cost reduction has been a factor in all cases. The further reduction of cost in IT services that OCCS provides is attractive to companies - especially very small companies which normally have lower demands on reliability and with a tighter IT budget.

### 4.1 Regulatory Amendments to Support OCCS

OCCS needs international scope to flourish. The implementation of national OCCS networks is useful; however, for participating members to find suitable services to meet their business needs, then the larger the scope of the platform the better. Secondly users should be indifferent (should not need to worry) about the location or origins of the provider of the service in which they are interested in utilizing. It should also be noted that no SLAs exist between the provider of a service and possible user of that service on the OCCS platform. And the rules of conduct governing the platform should not put undue burden on contributors of resources and services to the platform.

Currently US companies need to use the US-European Union and the US-Switzerland Safe Harbour Frameworks to meet European “adequacy” standards for privacy protection [1]. The recently proposed EU data protection reform [17] which is meant to be cloud computing friendly proposed a “Regulation to replace a Directive: that means a single set of rules for Europe, not 27 different ones. Alongside that, under the new rules you will get a one-stop-shop of enforcement; so that, even if an operator is active in several EU countries, it will only have to deal with one data protection authority – the one where its main base is. Cloud users should not have to guess where their provider is: if a company offers goods or services to people in the EU, or is monitoring them, then it shouldn’t matter where that company’s based – in Madrid, Mumbai or Mountain View. Our rules should apply to the data” [8]. The proposed regulations are expected to make it easier to operate Clouds within and outside the European single market.

Allowing Cloud operations to easily cross borders is a step in the right direction for both cloud computing in general and OCCS. It is not too much of a problem for commercial cloud service providers to go through the necessary trouble of meeting multiple data protection rules; however OCCS should not have to grapple with the “our rules apply” in the formulation of the rules on terms of conduct on the platform since it will be a very daunting task to formulate such rules without putting undue burden on contributors of services to the platform and simultaneously ensuring that participating enterprises on the platform are able to use such free resources in providing services to their customers that meet different national or regional rules on data privacy. Having a single set of globally accepted rules that govern data protection for Cloud operations will therefore be very beneficial for OCCS even though that would not be a sufficient condition for OCCS to be very successful.

## 5 Conclusions

The paper has discussed the impact that public policy and current regulations together with corporate strategies towards cloud computing adoption will have on the implementation of opportunistic cloud computing services. We have looked at the various strategic policies being adopted by North America, Europe, Asia Pacific and Africa. Cost reduction and efficiency in the delivery of IT services is a central motivation in all the policies; on the other hand meeting security and privacy requirements for the protection of both national and citizen data are major issues that the policies try to address. The move by companies to the adoption of cloud computing is a senior management level driven strategic technology shift for organizations as they look to lower costs and evolve their computing models to deliver competitive advantage to their businesses. The further reduction of cost in IT services that OCCS provides is attractive to companies - especially very small companies which normally have lower demands on reliability and with a tighter IT budget.

We conclude that there are regulatory challenges on data protection that raises issues for cloud computing adoption in general; and the lack of a single globally accepted data protection standard poses some challenges for very successful implementation of OCCS for companies. However, the direction of current public and corporate policies on cloud computing make a good case for them to try out opportunistic cloud computing services.

## 6 Future Work

This work has relied on documented publicly available government and corporate policies on the adoption of cloud computing service and has deduced the impact of these policies on the implementation and adoption of opportunistic cloud computing services. Our future work will include results from interviews that will be conducted on representative organisations of the various sectors of the economy. Secondly based one of the findings of this work that the OCCS platform will have to provide commercial grade reliability, security and privacy guarantees for it to be useful for government public sector consumption of these services, we will also work on trust and security frameworks and their implementations for OCCS.

## References

1. U.S. Department of Commerce, 2012. Safe Harbor. [Online] Available at: http://export.gov/safeharbor/index.asp [Accessed 29 04 2012].
2. Cabinet Office, 2011. Government Cloud Strategy, London SW1A 1AS: Crown.
3. Chandrasekaran, A. & Kapoor, M., 2011. State of Cloud Computing in the Public Sector – A Strategic analysis of the business case and overview of initiatives across Asia Pacific, s.l.: Frost & Sullivan.
4. Danek, J., 2009. Cloud Computing and the Canadian Environment, Ottawa, Ontario: s.n.
5. DIGITALEUROPE, 2011. Cloud Computing, DIGITALEUROPE'S PERSPECTIVE, Brussels: s.n.
6. ITU FTRA-2011, 2011. Cloud computing, development prospects of ICTs: Challenges and opportunities for the policymakers, regulators and ICT operators. [Online] [Accessed 8th March 2012].
7. Jansen, W. & Grance, T., 2011. Guidelines on Security and Privacy in Public Cloud Computing, Gaithersburg: National Institute of Standards and Technology.
8. Kroes, N., 2012. EU Data protection reform and Cloud Computing. [Online] Available at: http://europa.eu/rapid/pressReleasesAction.do?reference=SPEECH/12/40&format=HTML&aged=0&language=EN&guiLanguage=en [Accessed 30 April 2012].
9. Kuada, E. & Olesen, H., 2011. A Social Network Approach to Provisioning and Management of Cloud Computing Services for Enterprises. Rome, Italy, Sep.2011, CLOUD COMPUTING 2011 : The Second International Conference on Cloud Computing, GRIDs, and Virtualization, pp. 98 - 104.
10. Kundra, V., 2011. Federal Cloud Computing Strategy, Washington DC: s.n.
11. Marston, Sean; et al;, 2010. Cloud computing - the business perspective. ELSEVIER, December.Volume Decision Support Systems.
12. Mell, P. & Grance, T., 2009. The NIST Definition of Cloud Computing, USA: s.n.
13. Red Shift Research, 2011. Adoption, Approaches & Attitudes, The Future of Cloud Computing in the Public and Private Sectors, s.l.: AMD.
14. Tait, A., 2010. G-Cloud Founding Principles, London: Cabinet Office.
15. The Economist Intelligence Unit , 2006. Great expectations: The changing role of IT in the business, London: The Economist.

16. U.S. Department of Commerce, 2010. Cloud Computing Policy. [Online] [Accessed 1 March 2012].
17. Viviane Reding, EU Justice Commissioner, 2012. Data protection reform: Frequently asked questions. [Online] Available at: http://europa.eu/rapid/pressReleasesAction.do?reference=MEMO/12/41&format=HTML&aged=0&language=EN&guiLanguage=en [Accessed 30 April 2012].

## Appendix 6: Paper #6

Kuada, Eric, Kwami Adanu, and Henning Olesen. 2013. “Cloud Computing and Information Technology Resource Cost Management for SMEs.” In *Proceedings of IEEE Region 8 Conference EuroCon 2013*, pp. 258 – 265, University of Zagreb, Croatia: IEEE.

# Cloud Computing and Information Technology Resource Cost Management for SMEs

Eric Kuada[#1], Kwami Adanu[*2], Henning Olesen[#3]
*#Center for Communication, Media and Information Technologies, Aalborg University, Copenhagen, Denmark,*
1 kuada@cmi.aau.dk
3 olesen@cmi.aau.dk
** Department of Economics and Finance, GIMPA Business School, Accra, Ghana*

***Abstract* - This paper analyzes the decision-making problem confronting SMEs considering the adoption of cloud computing as an alternative to in-house computing services provision. The economics of choosing between in-house computing and a cloud alternative is analyzed by comparing the total economic costs of the two options assuming the quality of service is identical across the options. The decision-making process was found to require substantial information gathering to identify explicit and implicit costs to inform the final decision. Careful considerations of decision time horizons also matter in determining the relative value of cloud computing.**

*Keywords: cloud computing; SMEs; decision time horizons; cloud economics; cloud adoption*

## I. Introduction

Cloud computing refers to the provision of computing resources as a service rather than as a product. This involves the packaging of traditional Information Technology (IT) infrastructure and software solutions such as storage, central processing unit, network, user and enterprise applications, as virtualized resources. These resources, are then delivered by a service provider to customers as an on-demand, pay-per-use, self-provisioned service through a web portal over a network such as the Internet [1], [2], [3]. This new trend of computing has been driven by major technological advancements as well as social and business demands. The technological factors facilitating cloud computing include the availability and drastic increase in reliable broadband Internet access, and advancements in virtualization technologies. Under cloud computing, the three components of a regular computing environment, namely the hardware infrastructure, the operating system platform, and user application software, have respectively translated into Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) delivered in cloud computing [1], [2], [3], [4]. Under SaaS, application software is provided on the cloud instead of on client computer. Thus, instead of installing and running software, the user can simply access the software over the internet from a cloud service provider's catalogue. Examples of SaaS products include enterprise level applications provided by Salesforce, Microsoft Office 365, and Google Apps customized business email services. Cloud computing brings software solutions within the reach of small businesses by obviating the need for purchasing exorbitant software licences which they usually end up underutilizing. Cloud computing thus provides flexibility in having access without ownership. PaaS provides a supporting platform for the development and deployment of computing applications developed by individuals and businesses without the cost of developing and managing the underlying platform. Good examples of PaaS are the Google App Engine [5], and Microsoft's Windows Azure [6] cloud platform. Several PaaS bundles also come with other supporting services from the cloud service provider. Such ancillary services include automatic accommodation of usage spikes, data storage space, and sales and transaction monitoring. PaaS generally makes it faster and easier to build, test, deploy and scale applications. IaaS, sometimes known as On-demand data centres, involves renting data centre capacity rather than owning and managing the hardware. IaaS normally provides remote access to computing processing power, memory, network, and servers. With increasing development and use of virtualization technology, IaaS is increasingly becoming virtualized. A virtualized service is a software implementation that executes programs just like the real hardware being virtualized, as is in virtual machines (VMs). Unfortunately, regular servers like email servers, web servers, and file servers are often dedicated to single tasks. As a result, such servers tend to be underutilized. Installing appropriate virtualization technologies such as hypervisors and their management tools allows for stacking of the functions of several independent servers on a single physical server to reduce the number of servers in use, and support efficient use of servers deployed. The attendant reduction of power consumption and server maintenance costs also helps to improve the bottom line of small businesses. Some examples of IaaS include Amazon's S3 storage service and EC2 (Elastic Cloud Compute), and Rackspace's cloud server services which are provisioned to users as virtual machines. Other examples include Google Drive, Dropbox [7], and Box [8].

The scalable provisioning of computing under cloud computing offers flexibility so that businesses can scale up their resources when demand increases and scale down when demand for these resources drop. Businesses are therefore able to escape high upfront product acquisition costs since cloud service subscribers only pay for resources used. This implies that for start-ups and businesses entering new markets, cloud computing can reduce firm market entry costs and increase market competition.

978-1-4673-2232-4/13/$31.00 ©2013 IEEE

Deciding to adopt cloud computing services is unfortunately not as simple as it may appear initially. This is because making prudent decisions on the adoption of cloud services requires some technical knowledge to address several choice-related issues. These include understanding all possible business applications of the various service alternatives available on the services menu of cloud computing providers, determining the lowest cloud service user cost that makes adopting a cloud computing service option cheaper than self-managing an in-house IT department, and deriving the time horizon that makes a service user indifferent between signing up for cloud service and managing an in-house IT centre. This study addresses these three issues and provides results that can be used as templates to respond to real-world cloud computing choice problems facing businesses.

The rest of the paper is organized as follows. Section II discusses major computing needs of businesses, particularly small and medium scale enterprises and matches these to cloud computing service alternatives that address these needs. This is followed by Section III on some related studies. The economic analysis of the choice problems associated with cloud computing adoption is presented in Section IV. Section V discusses the factors to consider in deciding on the adoption of cloud computing. The paper ends with some conclusions and future work in Sections VI.

## II. Small and Medium Enterprises (SMEs)

A small scale enterprise is a business enterprise that employs between ten and forty-nine people, while a medium scale enterprise employs between fifty and two hundred and forty-nine people [9]. The definition of SMEs is however diverse. Whereas some refer to the number of employees as the distinctive criteria for SMEs, others use invested capital, and some others, a combination of the number of employees, invested capital, sales, and industry type. Further, SMEs in developing nations including sub-Saharan Africa are normally proprietor managed, and characterized by self or family financing, and lack of a research and development unit[10].

SMEs normally have small ICT departments, if any, and are therefore not likely to have access to skilled IT personnel nor sophisticated IT infrastructure and associated management tools. Furthermore, SMEs normally do not have research and development units to support the performance of market research. These underlying conditions make cloud computing a potential valuable cost reduction tool for SMEs since in addition to providing the well-defined mainstream services, cloud services automatically log and provide other ancillary information like service response times, peak business periods, downtimes, and error rates which can be used to support business decision making.

The IT unit of many SMEs consists simply of a basic Personal Computer (PC), software, and sometimes Local Area Network (LAN). Even without Internet connectivity, SMEs can use PCs for basic word processing, accounting, and other business practices and with the LAN as intranet, be able to share files, documents, and possibly a central printer. Cloud computing opportunities emerge with internet access. With the Internet, SMEs are able to use more advanced communication capabilities such as email, websites, and e-commerce services. Email services with customized domain names for instance may be provided by a business in-house or through a cloud service provider. Customized emails provided within businesses however require the acquisition of relatively expensive email servers. On the other hand, with declining cloud services costs, customized cloud business emails are currently available from webhosting providers, and cloud services providers like Google at very low costs, or even for free in some cases. Other factors that increase the competitiveness of cloud services include increasing storage needs and workforce management problems confronting businesses.

In general, the computing needs of SMEs depend very much on the nature of business of the firm concerned. Of the three major cloud computing services, the computing needs of SMEs in the developing world can be expected to be more concentrated in the SaaS and IaaS cloud service options than PaaS. This is because businesses involved in software development are highly underrepresented in the developing world. Relevant SaaS options for training and research-related SMEs in developing countries for instance include student record management software like Online Student Information System (OSIS)[11], and ArcGIS[12]. The cloud alternative to more basic software like Microsoft office is also available as Office 365 from Microsoft, and Google Docs from Google. Cloud office document alternatives come with other benefits like file sharing and online collaboration, and large external data storage space.

SMEs in manufacturing, transport, food and similar industries may adopt more complex IT tools. Some of these complex IT systems are mainly enterprise application software which may include Supply Chain Management (SCM), Customer Relationship Management (CRM), Enterprise Resource Planning (ERP), students/facilities information systems, and human resource management. Additionally, SMEs may employ advanced communication tools such as VoIP and video conferencing tools. It is evident that cloud service options offer flexibility and cost saving advantages over in-house service options. The size of such advantages is however limited to some extent by the scale of business involved.

## III. Related Studies

A major downside of recent studies discussing decision frameworks to assist management in making decisions among computing service provision options is that the frameworks leave much of the detailed work relating to estimation of cost and benefit parameters undone. Consequently, frameworks developed are often abstract and do little in moving managers any closer to making such decisions relatively easily. Kaisler et al [13] for instance analyzed the concerns of businesses in moving to cloud computing environment and present a decision model outlining the types of issues that small to-medium businesses should consider in deciding to move wholly or partially to a cloud computing environment. The

978-1-4673-2232-4/13/$31.00 ©2013 IEEE

model however does very little in assisting such businesses in their decisions because it does not go into what it takes to answer the defining questions discussed in the model. [14] take a quantitative cost approach to sourcing decisions by developing a model to assist in the selection of appropriate cloud computing services offered by different providers. The model elements were converted into costs, since virtually all sourcing decisions are either directly related to costs or can be converted into costs [15]. The approach adopted by [14] is quite similar to that of [16] which develops a cloud adoption tool-kit that includes a framework for computing costs of service provision. Also see, [17], [18]. These studies however fail to explain how the cost estimates could be obtained. The contribution of these studies to resolving the managerial problem of making informed choices between computing services provision options is therefore very limited.

Reference [17] introduced the generic *CloudGenius* framework that provides a migration process and decision support. *CloudGenius* leverages a well-known multi-criteria decision making technique, called Analytic Hierarchy Process, to automate the selection process based on a model and quality of service parameters related to an application. They have presented an implementation of *CloudGenius* that has been validated through experiments. The framework however is only for web server migration and does not support other application scenarios that enterprises face in the decision of migrating to cloud computing. A discursive approach to tackling the cloud decision problem is presented by [19] . They examined characteristics of applications that are good candidates for cloud computing, and reasonably discussed the risk component of the cloud decision problem as well but again, the more challenging work of providing guidance on how such risks may be modeled in practice, and how cost and/or benefit estimates could be obtained is left out. Overall, all the studies reviewed to provide background for this study fail to account for opportunity costs and do not explain how the cost estimates could be obtained.

## IV. ECONOMICS OF CLOUD COMPUTING ADOPTION

Under frictionless economic conditions the true cost of subscribing to cloud services should be identical to the cost of providing similar services in-house by a business entity. This is because observable pricing discrepancies between these two approaches to providing business computing services should result in businesses leaning towards the cheaper option until prices equalize across the two options. In reality, given that computing service delivery and pricing does not occur under perfect information flow, the cost of accessing computing resources may not be equal across the two options under consideration.

The decision to adopt cloud computing as a business management tool hinges on the net benefit derived from use of public cloud service as compared to maintaining a regular IT department. Businesses with no existing IT resources have to compare the net benefit of setting up and running a regular IT department to subscribing to public cloud computing services, while those with well-established IT departments evaluate the net benefit associated with switching from running a regular IT department to cloud computing. Evidently the time horizon over which the service is used is also a major factor affecting decisions to switch to cloud computing service. The shorter the time horizon, the more competitive a cloud computing option becomes relative to a regular IT department since incurring huge upfront costs on computing infrastructure becomes less profitable. Already, research indicates that organizations are increasingly discovering that many companies tend to underutilize the substantial investments made in IT resources. A recent survey of corporate data centers revealed that only between 10% to 30% of the computing power of servers were being used, while desktop computers have an average capacity utilization of less than 5 percent[20]. The choice of computing service provision option may however depend on several other factors other than direct net benefits derived from use of computing services. For instance, service reliability, data security, and service performance levels, are other important factors that could affect a decision to sign up for cloud computing services [21], [22], [23] .

### A. Evaluating the Benefits and Costs of Computing Service Options

As previously noted, an organization considering cloud computing service among several other computing service provision options will normally be interested in evaluating the costs and benefits associated with adopting a given option. For all practical purposes currently, the choice is normally between subscribing to cloud computing services (or outsourcing), and providing such services in-house. If it can reasonably be assumed that the quality of service is identical across the two options, then what matters to the decision-maker is the cost of providing the service under each option. For a cost minimizing manager, the preferred option is the one associated with a lower cost of execution.

When measuring costs, economists always use opportunity cost which is defined as the next best alternative forgone. In other words, cost is measured in terms of the opportunities forgone. Opportunity cost classifications depend on whether the factor of production associated with the cost is already owned by the firm or is rented. For items not already owned by the firm, the opportunity cost incurred is explicit opportunity cost which refers to out-of-pocket expenses incurred to obtain the item. Where the item in question is already owned by the firm, the opportunity cost is implicit. Implicit opportunity costs refer to what the factor could have earned for the firm if put to some alternative use. It is also important to consider both explicit and implicit costs associated with any given switch of computing services provider. For instance, when an institution acquires a server, the cost of acquiring the server is not limited to the out-of-pocket cost incurred to get the server but also foregone investment income given that money spent on buying servers could have been invested to earn some return if payment for the servers could be avoided.

To simplify analyses, the costs of subscribing to public cloud computing services are classified into two categories in this paper. These are, periodic user cost payments, and IT staff

978-1-4673-2232-4/13/$31.00 ©2013 IEEE

and facility maintenance costs. Periodic user cost payments are the regular payments due the cloud service provider for providing on-demand computing services. Staff and facilities maintenance costs cover salary payments for a relatively small IT staff, power bills for the IT unit, and maintenance of servers. Costs associated with in-house computing service provision are also decomposed into two parts; upfront setup and facility maintenance costs, and IT staff maintenance costs. Upfront setup and facility maintenance costs include initial hardware and software acquisition, software updates, power bill payments, and maintenance and replacement cost of servers. Server procurement and maintenance represents a substantial component of computing resource management costs. Generally, businesses having 5 or more computer users on a network need servers [20]. Further, most servers have to be replaced after four to five years of use to avoid expensive downtimes resulting from critical server failures. The cash flow patterns are naturally expected to vary across the two service options under consideration, each with its associated opportunity costs.

The choice problem between in-house computing and cloud computing is identical to the rent or buy decision problem in economics [24], [25] where economic agents collect and analyze cost and benefit flow information to determine the option that maximizes profit or some other objective. The modeling approach adopted in this paper is therefore similar to what is commonly adopted in analyzing the rent or buy decision problem. In particular, the [24] approach to modeling the problem is followed.

Addressing the problem facing a manager choosing between cloud computing services and in-house computing requires outlining the estimated cost of service provision under each option and comparing the results for decision-making purposes. The cost facing a manager subscribing for cloud services can be represented as,

$$C_v = \sum_{j=1}^{n}\left\{\left(P_j^{\,v} + S_j^{\,v}\right)\prod_{i=j-1}^{n-1}(1+k)\right\} \qquad (1)$$

where $P_j$ refers to payment to cloud/virtual *v* service provider in period j

$S_j^{\,v}$ refers to salary payments to IT staff and facility maintenance in period j

$k$ is opportunity cost expressed as a percentage of payment made

In the representation above, $j$ refers to periods in a time horizon while *i* is a counter which equals $j$ at the end of the $j^{th}$ period and $j-1$ at the start of the $j^{th}$ period. For simplicity, it is assumed here that managers are patient about receipt of the net benefit flows from computing resource usage thus do not discount these values. Figure 1 below illustrates the time relationships further.

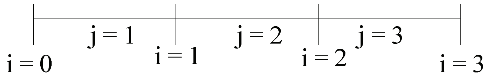


Figure 1: Time horizons

Note that the start of the $j^{th}$ period is same as end of the *j-1* period. Such time distinctions are important here because of the time cost of money. Whether a transaction takes place at the beginning or end of a period can make a difference in the relative costs of computing options under evaluation. The inner arguments of equation 1 give the accumulated value of the sum of periodic payment and opportunity costs beginning from the time of payment to end of the time horizon in the problem. The summation sign sums up all these accumulated payments over the n periods in the problem.

When a business decides to provide its own computing services, there is an upfront set-up cost which comes with facility maintenance charges. Then there is the usual periodic salary payments to IT staff. The total cost of computer service provision for the business can then be represented as,

$$C_r = U^r\prod_{i=0}^{n-1}(1+k) + \sum_{j=1}^{n}\left\{S_j^{\,r}\prod_{i=j-1}^{n-1}(1+k)\right\} \qquad (2)$$

Where $U^r$ refers to upfront cost to set up IT department

$S_j^{\,r}$ refers to salary payments to IT staff in period j

For a decision rule, $C_v < C_r$ indicates that cloud computing is the preferred computing option over a given time horizon, while $C_r < C_v$ suggests that self-provided computing service is cheaper than cloud services. It is easy to search the time horizon to determine the unique time period of use $n^*$ that makes a user indifferent between the two options of business computing service provision. The breakeven period $n^*$ also serves as the minimum period for which self-provision of computing service is optimal. Solving for the breakeven period $n^*$, and obtaining results for other interesting questions to guide decision-making is better done numerically. A computer program can be written to respond to such queries.

### B. Illustrative Numerical Example: impact of implicit costs on cloud computing decisions

Consider a hypothetical educational institution that is interested in equipping its computer laboratory with Geographic Information System (GIS) software for learning and teaching purposes. Suppose this institution is choosing between buying a five-year software license at a cost of $10,000 and, accessing the software through a cloud computing service at a cost of $200 per month for five years. If the annual interest rate is pegged at 10 % then making decisions based on explicit or out-of-pocket payment alone favors the software purchase option rather than obtaining virtual access. As shown in Table 1 below, over the 5 year period, in-house provision of the service costs $10,000 in out-of-pocket expenses while the monthly payments sum up to $12,000. Monthly payments exceed outright purchase of software license thus the obvious decision is to go for a five-year software license.

TABLE 1: COST OF IN-HOUSE AND CLOUD GIS SERVICE PROVISION

| Costs | Cloud | In-house |
|---|---|---|
| Explicit cost | 12, 000 | 10, 000 |
| Implicit cost | 3,614.77 | 6,453.09 |
| Total cost | 15, 614.77 | 16,453.09 |

Explicit costs however do not represent the total economic costs of this transaction. There are foregone interest receivables on each payment made towards cloud access or outright purchase of the software. Suppose the value of the best alternative use of money to this educational institution is to invest the money in the money market. Then foregone interest receivables associated with such investment constitute implicit costs that must be added to explicit costs to determine the economic cost of the transaction. Once implicit cost is accounted for the result is reversed. The total economic cost of the transaction now indicates that adopting the in-house service provision option results in extra payments of about $ 838 over the cloud service option. In fact, the greater the interest rate the cheaper the cloud option looks. On the other hand, the longer the time horizon the cheaper is the in-house option. It is obvious then that there exist an optimal action or decision that depends on time horizons and interest rates. The challenge of this decision making problem is to find this optimal action. This example does not only illustrate the importance of considering economic costs in such analysis instead of making decisions based explicit costs only, but also shows the potential impact of cloud computing on business profitability.

Obviously, a decision like the one just described cannot be based solely on monetized costs and benefits. Often, there are other relevant costs and benefits that managers find difficult to monetize or lack the required technical skills to monetize. For instance, cost of regular software updates, software underutilization, data insecurity, and slow internet service that affect use of cloud service are other important issues to consider in making a decision to adopt cloud computing service.

### C. A Case Study on GIMPA

A decision problem involving acquisition of ERP software for Ghana Institute of Management and Public Administration (GIMPA) is used to illustrate the application of the decision framework designed in this study. In particular, this section of the study examines cost factors that must be considered in in the ERP purchase decision problem and discusses feasible ways of obtaining the cost estimates and their associated opportunity costs.

GIMPA was established in 1961 to provide civil servants with administrative and professional competence, and to plan and administer national, regional and local services. Currently, GIMPA offers courses leading to the award of certificates, diplomas, and bachelors and masters degrees. The institute is made up of five main schools namely, GIMPA business school, GIMPA School of governance and leadership, GIMPA law school, GIMPA School of technology, and GIMPA Public service school. The school currently has a student population of about 8000.

IT services provided to students and staff include internet access and email services. Until the year 2010, access to email services at GIMPA was restricted to staff and email service was provided in-house by the GIMPA IT department. The main capital asset acquired to provide the service was one email server. In addition to the cost of email server, other related costs include power costs incurred to keep the server room cool and keep the server running at all time as servers could not be shut down without interrupting email service. In 2010, GIMPA migrated its domain to the Google platform. In particular, GIMPA adopted the *Google Educational Apps* which provides educational institutions with free customized email services (for both staff and students), calendar services, and SMS platform. The switch became necessary because the email server in use was old and expensive to maintain, and the MS exchange software in use to provide the service crashed repeatedly. This switch eliminated GIMPA email service downtimes due to server crashes, and provides opportunity for GIMPA to retain student email addresses even after graduation from GIMPA. This will assist GIMPA to track and communicate with alumni. Making the decision to switch to cloud in this case was very trivial because the cost of using the Google email system is zero. On the other hand, the estimated cost of replacing the old email server to continue providing in-house email services is US$ 6000. Further, server acquisition comes with additional energy and server maintenance costs.

GIMPA currently needs Enterprise Resource Planning (ERP) software. This software is needed to manage student resources including student grades, fee payment status and access to school resources, availability and use of classroom and other infrastructural resources, and other resources owned by GIMPA. Apart from helping in managing the institutes resources, a complete ERP with web interface will enable students access learning materials from anywhere around the world. GIMPA's use of ERP software will support plans to expand the GIMPA distance education program by making it easy for students to access student services and learning materials easily from their various stations. GIMPA has received lifetime license price quotes of GH¢1.5 million and GH¢ 420,000 for a complete ERP software with web interface, and ERP software with no web interface respectively. Because of the steep software price, purchase decisions are currently on hold. There are cloud ERP alternatives as well that can be evaluated for comparison and prudent decision-making purposes. Unlike the decision to adopt cloud email services, the decision on ERP software is not a trivial one.

## V. NOTES ON TIME AND COST FACTORS TO CONSIDER IN CLOUD DECISION-MAKING

This section presents information on factors that need consideration in estimating economic costs to support computing service option decisions. Ideas and methods are presented in a very general sense to allow for applications to general cloud-inhouse computing service decision problems. However, the GIMPA ERP acquisition case is used as an

978-1-4673-2232-4/13/$31.00 ©2013 IEEE

example to support the explanatory notes. Table 2 in the appendix presents a summary of important factors to consider and serves as an example of how cost factors may be tallied to guide decisions.

### *A. Choosing the Time Horizon*

One of the most important information requirements for this analysis is the determination of the time horizon for the analysis. Accummulated costs and benefits can be very sensitive to selected time horizons particularly when opportunity costs are included in the analysis. For many cloud computing decision problems, the applicable time horizon may run from zero to infinity. The relevant or useful time horizon however may be much more constrained and defined by several factors in the work environment of the firm. For instance, in computer software service provision problems, many of the software that may be acquired for in-house service provision offer lifetime licenses. Unfortunately, this does not suggest that the software could be used for a lifetime since the software could become obsolete and need to be dispossed off or the firm may close down in the near future. There is no single appropriate time horizon that works in all cases. In general, the appropriate time horizon depends on how far into the future the researcher can reasonably predict costs and benefits. Otherwise, it is instructive to perform the analysis in time bands and examine sensitivity of the results to time. For instance, three sets of results could be prepared for the short-term (1-5 years), medium term (5-15 years), and long-term (15+ years). Alternatively, if the business will exist for some known time period, then this time period could be the appropriate time horizon. To further guide decision-making, it is sometimes useful to find the time horizon at which both computing options provide the same cost value thus making the manager indifferent between the two options.

### *B. Hardware and Software license Cost Virsus Regular Cloud Service Charge*

Depending on the type of computing service being demanded, the firm requesting the service may have to purchase hardware and software to provide the service in-house. The pro forma invoice for such products can easily be obtained from vendor units. The one time purchase price and opportunity costs must be added up to obtain the total economic costs associated with hardware and software purchased. The equivalent economic cost estimates for cloud service option is the sum of total regular payments made to the cloud service provider and the corresponding opportunity costs. Equations 1 and 2 may be used as previously described to obtain these economic cost estimates. In the GIMPA ERP software case, the ERP software price quote of GH¢1.5 million and the associated opportunity cost could be computed using equation 2 over different time bands. A cloud ERP periodic user cost must then be obtained from a cloud ERP service provider and used to estimate its associated opportunity cost as well. The computed values can then be entered in a table as shown in Table 2.

### *C. Interest rates*

Interest charges constitute a critical component of computing cost estimations. Interest charges come in several forms. First, the implicit services cost is easily expressed in the form of forgone interest earnings. Here, it is reasonable to use the risk free rate or the rate payable on short term treasury instruments. Using the rate of return on treasury instruments eliminates or minimizes the problem of assessing and modeling the risk component of interest rates. If acquisition of the computing service is financed with a loan, then the loan interest cost computed using the lending rate represents another cost component that must be considered and added to the total cost of service provision. Discounting has been largely ignored in the analysis thus far. For medium to long-term horizon analyses, results may be sensitive to discounting. Well defined risks inherent in the computing service provision analysis may also be modeled using risk-adjusted discount rate values that is simply the sum of the risk free rate and a risk premium. In summary, there could be three or more interest rates. If estimation of risk premium is a problem, some certainty equivalent estimate may be used [26].

### *D. Computing Downtime Costs*

Computing service downtime costs represent an important source of productivity and loss for firms. Service downtimes also affect the reputation of businesses particularly when a substantial part of a firm's business require online contact with customers. Frustrated customers switch to other firms resulting in revenue and profit loss for the firm in question.

Computing service downtime costs can be easily integrated into cloud service contracts requiring the service provider to compensate service suscribers for service downtimes. For inhouse service provision, such costs must be borne by the firm. In either case however, the first step is to be able to estimate what is lost to the business per hour of service downtime. This may be in the form of profits lost per hour of service downtime or a detailed estimate of service downtime costs. It is simpler and cost effective to approximate damage done by service downtime using profits lost per downtime hour. Historical profit records of the business could be used to estimate a moving average hourly profit figure that is automatically updated over time. Alternatively, the estimated profit lost per downtime hour for any give year could be obtained at the end of that year based on current year financial data. If a detailed downtime cost estimate is required, then two main downtime cost components must be estimated; total downtime hours, and cost per downtime hour. Total service downtime hours is normally always readily available. Obtaining cost per downtime hour information on the other hand calls for aggregation of two major classes of information; costs imputable to service downtime including employee cost per hour and lost revenues per hour of downtime. For banks, such lost revenues may include Automated Teller Machine (ATM) fees that are lost because service downtimes.

### *E. Server Farm Maintenance*

Servers in data centers are typically organized into related groups called server farms. Servers within server farms often

978-1-4673-2232-4/13/$31.00 ©2013 IEEE

contain identical content (referred to as mirrored content) so that if one server becomes inoperative, another server can take its place immediately [27]. In addition to the initial capital expenditure in building and getting a server farm up and running, some other capital expense could be incurred in upgrading and scaling up the server farms with new servers when it becomes necessary. Some of the operating costs incurred in owning a server farm include energy cost  and server farm management costs. Depending on the software used, a server farm can cost a significant amount of money in terms of license fees. All servers need security auditing and patch management. An unaudited, unpatched machine is a danger to a company but rolling out patches for an entire compute farm can be a difficult task. Similarly security auditing requires professional input from security specialists[28]. The above reasons makes it imperative to hire highly skilled staff for the IT department which leads to higher staffing costs for managing server farms. All such costs related to server acquisition and maintenance must be estimated and entered in the estimated cost table to be considered in making a final decision.

### F. Software Maintenace Costs

Computing software represents a central part of almost all computing services. As previously discussed the cost of acquiring  a software is just a fraction of the total cost of using the software. Another dimension to software usage cost is software ownership cost. Ownership costs include installation of software patches, and cost of upgrades. Software maintenance costs could represent close to 30 percent of IT budgets for some organizations [29].  Unfortunately, software maintenance costs can be difficult to incorporate into software cost analysis because software evolution over time can be unpredictable. Estimation models for software ownership cost estimation has however improved over time. [29]  introduced a multi-level approach to create transparency, and estimate maintenance costs realistically based on current spending levels. They present a regression-based statistical cost estimation model that can be followed to estimate reliable software maintenance costs.

### G. Depreciation

For most computing hardware acquisitions,  it is important to account for depreciation of the asset in question. This is mainly because such assets are used to provide services over many periods. Estimated depreciation values make it easy to allocate the value of assets to periods within which they are used to provide service. A simple way of obtaining depreciation cost estimates is by using the straightline method that involves reducing the cost of the asset or hardware by its salvage value and spreading the depreciable value evenly over the useful life of the asset.

### H.  IT Staff Training

The cost of training IT staff to provide ERP services in house must be compared to the corresponding cost of training the same IT staff  to provide support services for cloud ERP services. It is difficult to have apriori ideas about which of these two costs may be greater since only the orientation of the training programs may vary slighltly. In any case, IT training costs must include cost of training materials and utilities, compensation for resource persons, and opportunity cost of time spent training IT staff.

### I. IT Staff Costs

Total IT staff costs depend on staff size, level of education and technical expertise of IT staff, and labour market conditions for IT staff in the economy within which the educational institution operates. IT staff costs are expected to be strictly positive irrespective of whether an institution provides IT services in-house or subscribes to cloud IT services. IT staff costs associated with cloud computing service subscription may be lower than that associated with in-house computing service provision. This is because cloud subscription involves outsourcing at least part of IT staff responsibilities to the cloud service provider.

Estimating IT staff costs first involves estimating the hourly wage rate of each IT staff. For most existing institutions this information is readily available. For in-house service provision, the next stage is to examine the work schedule of the IT staff and determine the total number of hours of work required over a defined period of time. The computed hourly wage rates can then be multiplied by total work hours to obtain total IT staff costs in a month. Identical values must be computed for cloud subscription option. The corresponding opportunity cost values must then be computed and entered in the appropriate columns in the table.

## VI. Conclusion & future work

This paper has analyzed the decision making problem confronting SMEs considering the adoption of public cloud services as an alternative to in-house computing services provision. The economics of choosing between in-house computing and a cloud alternative is analyzed by comparing the total economic costs of the two options given that the quality of service is identical across the options. The importance of accounting for implicit costs is emphasized and illustrated with a numerical example. The choice between in-house and cloud computing may however not always be clear-cut and solely based on numerical estimates.  Several other factors that may be difficult to monetize may prove to be relatively more important in making such decisions. In most cases however, this is the exception rather than the norm. Reliable estimates of the costs and benefits of adopting alternative computing services provision options can and should therefore be produced and utilized to guide such decisions.

As part of our future work, data on the GIMPA ERP decision problem will be collected to estimate economic costs of ERP service provision under Cloud and in-house service provision options. The framework developed in this study will be applied to assist in management decision making. We have also for the past two years been working on Opportunistic Cloud Computing Services (OCCS) [30] because it promises an accelerated adoption of cloud computing services and further reduction in IT cost for small companies. We have therefore been working on the feasibility of its successful implementation in terms of the technical feasibility, impact of public policy and regulations on its implementation [31], development of suitable incentive mechanisms for the OCCS platform [32], and developing suitable trust management systems for OCCS platforms. We will also as part of our future work, investigate how opportunistic cloud services can be leveraged to lighten the IT resource needs of SMEs, particularly in developing economies.

## REFERENCES

[1] N. Sultan, "Cloud computing for education: A new dawn?," *International Journal of Information Management*, vol. 30, no. 2, pp. 109–116, Apr. 2010.

[2] P. Mell and T. Grance, "The NIST definition of cloud computing (draft)," *NIST special publication*, vol. 800, p. 145, 2011.

[3] D. Beimborn, T. Miletzki, and S. Wenzel, "Platform as a Service (PaaS)," *Business & Information Systems Engineering*, vol. 3, no. 6, pp. 381–384, Dec. 2011.

[4] X. Xu, "From cloud computing to cloud manufacturing," *Robot. Comput.-Integr. Manuf.*, vol. 28, no. 1, pp. 75–86, Feb. 2012.

[5] Google App Engine, "Google App Engine," Feb-2013. [Online]. Available: https://developers.google.com/appengine/. [Accessed: 09-Apr-2013].

[6] M. Windows Azure, "Microsoft Windows Azure," 2013. [Online]. Available: http://www.windowsazure.com/en-us/documentation/. [Accessed: 09-Apr-2013].

[7] Dropbox, "Get more space - Dropbox," *Dropbox*, 2013. [Online]. Available: https://www.dropbox.com/getspace. [Accessed: 07-Apr-2013].

[8] Box, "Box for Business: What Can Box Do For Your Business? | Box," *Box*, 2013. [Online]. Available: https://www.box.com/business/. [Accessed: 07-Apr-2013].

[9] W. Luetkenhorst, "Private Sector Development: The Support Programmes of the Small and Medium Enterprises Branch," United Nations Industrial Development Organization (UNIDO), Vienna, Technical Working Papers Series Working Paper No. 15, 2006.

[10] R. Dababneh and T. Farah, "Booklet of Standardized Small and Medium Enterprises Definition," USAID/Jordan Economic Opporturnity Office, United States Agency for International Development, BEARINGPOINT, INC., Deliverable, Aug. 2007.

[11] OSIS, "Online Student Information System (OSIS)," *Edhub Incorporated*, 2010. [Online]. Available: http://www.edhub.net/main/index.php?option=com_content&view=article&id=23&Itemid=72. [Accessed: 09-Apr-2013].

[12] E. ArcGIS, "ArcGIS: The Mapping Platform for Your Organization," *About ArcGIS*, 2013. [Online]. Available: http://www.arcgis.com/about/. [Accessed: 09-Apr-2013].

[13] S. Kaisler, W. H. Money, and S. J. Cohen, "A Decision Framework for Cloud Computing," in *2013 46th Hawaii International Conference on System Sciences*, Los Alamitos, CA, USA, 2012, vol. 0, pp. 1553–1562.

[14] B. Martens and F. Teuteberg, "Decision-making in cloud computing environments: A cost and risk based approach," *Inf Syst Front*, vol. 14, no. 4, pp. 871–893, Sep. 2012.

[15] M. J. Schniederjans and K. M. Zuckweiler, "A quantitative approach to the outsourcing-insourcing decision in an international context," *Management Decision*, vol. 42, no. 8, pp. 974–986, Sep. 2004.

[16] A. Khajeh-Hosseini, D. Greenwood, J. W. Smith, and I. Sommerville, "The cloud adoption toolkit: supporting cloud adoption decisions in the enterprise," *Software: Practice and Experience*, vol. 42, no. 4, pp. 447–465, 2012.

[17] M. Menzel and R. Ranjan, "CloudGenius: decision support for web server cloud migration," in *Proceedings of the 21st international conference on World Wide Web*, New York, NY, USA, 2012, pp. 979–988.

[18] M. Menzel, M. Schönherr, J. Nimis, and S. Tai, "(MC2) 2: A Generic Decision-Making Framework and its Application to Cloud Computing," *arXiv preprint arXiv:1112.1851*, 2011.

[19] L. S. Lee and R. D. Mautz Jr, "Using cloud computing to manage costs," *Journal of Corporate Accounting & Finance*, vol. 23, no. 3, pp. 11–15, 2012.

[20] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalsasi, "Cloud computing—The business perspective," *Decision Support Systems*, vol. 51, no. 1, pp. 176–189, 2011.

[21] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1–11, 2011.

[22] W. Jansen and T. Grance, "Guidelines on security and privacy in public cloud computing," *NIST Special Publication*, pp. 800–144, 2011.

[23] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," *Future Generation Computer Systems*, vol. 28, no. 3, pp. 583–592, 2012.

[24] R. Mojena, G. G. Booth, and J. P. Bowman, "A deterministic model for the housing decision," *Omega*, vol. 4, no. 1, pp. 85–91, 1976.

[25] T. Van Hecke, "Rent or Buy," *PRIMUS*, vol. 19, no. 6, pp. 541–547, 2009.

[26] J. C. Hull, "A NOTE ON THE RISK-ADJUSTED DISCOUNT RATE METHOD," *Journal of Business Finance & Accounting*, vol. 13, no. 3, pp. 445–450, 1986.

[27] A. N. M. Cisco, *User Guide for the Cisco Application Networking Manager 5.2.2 - Configuring Real Servers and Server Farms*, Cisco Systems, Inc. San Jose, CA: , 2012.

[28] SKP & Associates, "Five reasons to outsource compute farm and server management," *SKP and Associates*, 27-Jan-2012. [Online]. Available: http://www.spkaa.com/five-reasons-to-outsource-compute-farm-and-server-management. [Accessed: 10-Apr-2013].

[29] I. Buchmann, S. Frischbier, and D. Putz, "Towards an estimation model for software maintenance costs," in *Software Maintenance and Reengineering (CSMR), 2011 15th European Conference on*, 2011, pp. 313–316.

[30] E. Kuada and H. Olesen, "A Social Network Approach to Provisioning and Management of Cloud Computing Services for Enterprises," presented at the CLOUD COMPUTING 2011, The Second International Conference on Cloud Computing, GRIDs, and Virtualization, 2011, pp. 98–104.

[31] E. Kuada, H. Olesen, and A. Henten, "Public Policy and Regulatory Implications for the Implementation of Opportunistic Cloud Computing Services for Enterprises," in *Workshop on Security in Information Systems*, Wroclav, 2012.

[32] E. Kuada and H. Olesen, "Incentive mechanisms for Opportunistic Cloud Computing Services," in *2012 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, 2012, pp. 127 –136.

## APPENDIX

TABLE 2: COST OF IN-HOUSE AND CLOUD ACADEMIC ERP SERVICE PROVISION AT GIMPA

| **Costs Items** | **Cloud ERP** | | | **In-house ERP** | | |
|---|---|---|---|---|---|---|
| | Explicit GH¢ | Implicit GH¢ | Total GH¢ | Explicit GH¢ | Implicit GH¢ | Total GH¢ |
| Software license/Cloud Service Charge | XXX | XXX | XXX | XXX | XXX | XXX |
| Interest charges | XXX | XXX | XXX | XXX | XXX | XXX |
| Downtime costs | XXX | XXX | XXX | XXX | XXX | XXX |
| Server Farm Maintenance | XXX | XXX | XXX | XXX | XXX | XXX |
| Software upgrades | XXX | XXX | XXX | XXX | XXX | XXX |
| Depreciation | XXX | XXX | XXX | XXX | XXX | XXX |
| IT Staff Training | XXX | XXX | XXX | XXX | XXX | XXX |
| IT Staff Costs | XXX | XXX | XXX | XXX | XXX | |
| **Total Costs** | **XXX** | **XXX** | **XXX** | **XXX** | **XXX** | **XXX** |