CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

Over the years, universities have been saddled with the responsibility of scholarship, innovation, research, knowledge generation, knowledge dissemination, technology adaptation and technology re-engineering (Mutula, 2009). They are expected to guarantee the most efficient utilisation of research results and their possible application to economic life. The pressure on the universities to meet the changing needs of the society have increased with the transition towards a knowledge-based economy and the integration and assimilation of Information Technology (IT) into the academic environment (Mutula, 2009). Moreover, the emergence and use of IT in higher education have led to an increasingly virtual education system, with implications for the dynamics and conduct of university research (Mutula, 2009). With the creation of private and government research institutes, universities no longer remain sole citadels of research activities. For example, we have the National Agency for Science and Engineering Infrastructure (NASENI), in Nigeria with about nine research institutes responsible for research in different branches of Engineering and Technology. These universities and research institutes are faced with the challenges of low state funding, limited equipments / facilities, and geographical distribution, the increased demands from the society notwithstanding. Our human memory is not quite deep and precise enough to remember all the details of a project thus the need for a good and efficient storage system to aid research work. There is the challenge of a less than optimal return on the Government's substantial investments in research due to our inability to track what everyone is doing in a large group as a result of poor documentation, and so risk duplicating or clobbering the work of others. Most of our research institutes end up producing products that are obsolete due to inadequate information on the ongoing research activities in the international research world. Majority of the research outputs from the universities and research institutes end up as prototypes displayed in show glasses for visitors' consumption because of poor awareness creation about their existence.

With the invention of new technologies especially in the information technology sector, universities and research institutes in the developing nations can now re-engineer and reposition themselves in order to meet these ever increasing societal demands effectively (Mutula, 2009).

In the last decades, research collaboration has become one of the focus themes in engineering and technology studies. Research collaboration involves the working together of a group of researchers on a particular idea. It can involve people from the same discipline or it can be inter-disciplinary. It can also be between the industry and researchers aimed at providing societal needs. Technologies are evolving daily at a very high speed. Team work is of great importance in engineering research due to the great complexities involved (Whitehead, 2007). There is need for both local and international collaborations for higher and better productivity. An example is a large organisation such as NASENI with the different parastatals under it, mandated to carry out research in different fields of engineering; one of the research arms may be carrying out a multidisciplinary research which involves other arms which are situated in different geographical locations. This constitutes a very big challenge to the progress of the work. There is also need for the researchers in the developing nations faced with the challenges of inadequate infrastructure to collaborate with those in the developed world for increased efficiency and better research outcomes. This is the major aim of the UNESCO-HP Brain drain initiative ((UNESCO, 2009). Many researchers collaborate / share data and information through telephones, scanners and emails. Some had to travel and pass through various paper documentation processes or use low-end video and/or telephone conferencing equipment, supplemented with print-outs in order to collaborate.

There have been a lot of improvements today in research collaborations with the internet turning the world into a global village. The advent of e-Research and e-collaboration technologies and the proliferation of smart mobile devices have gone a long way in making research collaboration easier and more effective. Distance is no longer a major challenge as the availability of internet facility in almost every part of the world has taken care of that.

The role of international collaborations in advancing knowledge and offering economic opportunities worldwide is growing, thanks to factors such as access to the Internet; globalization; and greater mobility of information, ideas, and people (Sloan and Arrison, 2011). The I-Group established in 2008 by the US government engages in dialogue and discussion to facilitate international collaborations among academic, government, and industrial partners by: identifying policies and operations that enhance the ability to collaborate; identifying barriers to collaboration—policies and operations that could be improved; developing a web-based resource or other compendium of successful strategies and methodologies; and Suggesting how barriers might be addressed (Obama, 2009). This is what developing countries like Nigeria need. Many governments in the world like Australia, UK, and USA are investing heavily in research infrastructure and are developing e-Research skills and tools to facilitate more productive use of the and realise benefits from international collaborative research.`

A lot of software has been developed to aid research work (Whitehead, 2007). Some of them are for easy access to online journals or data repositories, some also have capabilities for real-time data and information exchange, and some have the chat room capabilities of the social networks while some of them can allow for exchange of video and audio information. Previously, the software would typically be a packaged solution installed directly onto each individual personal computer. But over the past few years, with the explosion of cheap bandwidth and server technology as well as the bourgeoning open source movement, companies are gravitating towards a database package that is not managed on an individual machine but hosted upon a server and viewed through an internet browser (Whitehead, 2007). Cloud computing has this capability. It removes the cost of infrastructure and maintenance from the end user and places it on the service providers. Instead of having the software installed in individual computers, it is installed in only one computer and made available to others through the cloud. Cloud computing will help in accessing and manipulating more data, and applying more complex calculations at a cheaper rate. It also provides the storage needed for the large volume of data generated in research. Since cloud computing involves the management of one's data by a third party, the cloud service providers, there comes the challenge of how to ensure the integrity and security of data in the cloud. There is also the challenge of efficient task scheduling since it involves the allocation of computing resources to different clients on demand.

There is therefore need to model an enhanced secure cloud-based collaboration platform with an efficient queuing system that will enhance task scheduling in the cloud environment. This is the major design goal of this work which also incorporates virtualisation of scarce and expensive research equipment.

1.2 Statement of Problem

Research by nature requires team work (either inter- or intra-disciplinary). There is therefore need for collaboration among researchers who may be located in different geographical locations. With large amount of data generated in research, a major challenge arises as to where and how to store them. Cloud computing with its large data bank has this capability. The merits of cloud computing notwithstanding, there is the challenge of how to ensure the integrity and security of data in the cloud since it involves the management of one's data by a third party, the cloud service providers. There is also the challenge of effective task scheduling since it involves the allocation of computing resources to different clients on demand.

Therefore there is need for an enhanced secure cloud-based collaboration platform with an efficient queuing system that will enhance task scheduling in the cloud environment. This work seeks to solve these problems by developing an enhanced cloud-based research collaboration platform, an enhanced job queuing model for efficient task scheduling and a Hybrid Encryption Algorithm which is a combination of a modified form of advanced encryption standards and digital signature to ensure data security and integrity. Hence, in this work, development of a research collaboration platform, effective task scheduling resulting in very low latency, high server utilisation and throughput and implementation of an efficient data security and integrity system form our major design goals in order to solve these problems.

1.3 Aim and Objectives of the Research Work

The aim of this work is to Model a Cloud-based Collaboration Platform with Enhanced Security.

The objectives are:

- To model a conceptual framework for secure research collaboration in engineering and technology with real-time data acquisition from research equipments
- To develop an enhanced queuing model for cloud-based research collaboration platforms aimed at improving task scheduling in the cloud while showing the effect of the system's queuing on response time and server utilization
- To design an improved second tier security system for the platform called Hybrid Encryption Algorithm (HEA) which is a combination of a modified form of advanced encryption standard and digital signature techniques

- To design and implement a prototype cloud-based collaboration platform

1.4 Significance of the Study

The dynamic nature of research demands a robust system that is secure, scalable, cost-effective, and efficient for effective collaboration. A Cloud-based collaboration platform with enhanced security is of great importance in research due to the following reasons:

- Cloud-based collaboration platforms will allow researchers to easily interact, share ideas and expensive equipments, and access the information they need
- The cloud provides a large storage for research data, reduces the cost for infrastructure and general cost of research
- The security system proposed will improve data security and integrity in the platform and also the protection of intellectual property
- The queuing model proposed will help to enhance task scheduling in the cloud environment so as to ensure better server utilization, smaller response time, shorter queue length, and better throughput
- Collaboration leads to increased productivity with reduced cost as time taken to conclude research and send the product to the market is reduced; better approaches are used in the realisation of research outcomes and geographical barriers are broken
- It solves the problem of re-inventing the wheel and also encourages continuity as documentations on previous research works are made easily available.
- The platform serves as a medium for collaboration between the research communities and the industries that will implement research outputs
- The model can be adapted to other research environments

The transformation brought about by information and communication technology also extends and broadens the impact of research, by making its outputs more discoverable and useable by other researchers, and by making its benefits more available to industry, governments and the wider community. The cloud-based research collaboration model proposed in this work will scale efficiently in terms of cost economy, scalability, latency, server utilization, intellectual property protection, data integrity and security and performance as task scheduling will be enhanced.

1.5 Scope of the Work

This dissertation is on a cloud-based collaboration platform with enhanced security for researchers to collaborate, share ideas, knowledge, infrastructures, documents and data no matter their geographic location. A conceptual framework for cloud-based collaboration platform with real-time data acquisition from research equipments will be modelled and simulated. An enhanced queuing model formulation for improved task scheduling in the cloud will be modelled and simulated while showing the effect of the system's queuing on server service utilization and general system efficiency. This is to be implemented at the infrastructure level. A prototype cloud-based collaboration platform, CRCM (Cloud-based Research Collaboration Model) using cloud computing platform as a service model will be designed and implemented. An enhanced security and data integrity system for the application will be designed based on a combination of advanced encryption system and digital signature. This will be deployed at the application level. Validation will be carried out for the proposed queuing model with parameters obtained from Cisco systems, Lagos using Optimised Network Engineering Tool (OPNET) for optimal performance. The application will be tested in terms of collaborators cloud experience, application interactivity, page response and general performance using Syskay, Cisco and some other data centres in order to evaluate its performance. Also the performance in terms of scalability, service availability, latency, throughput, and server utilization will be evaluated.

1.6 **Dissertation Report Organisation**

This dissertation is organized as follows: Chapter one deals with the background of the study, problem statement, aims and Objective of the work, Justification for the project, and Scope of the work and ended with the dissertation report organisation. Chapter two deals with literature review, it also discusses general principles, and available technologies for research collaboration using cloud computing. Chapter three explains the materials and methods used for the work. The chapter also analysed existing systems with a view to highlighting their limitations and then the proposed system was introduced. In Chapter four, the design implementation, results and discussion was dealt with. Conclusions were drawn and recommendations made in Chapter five. Lastly, the work concludes with references and appendices.

1.7 Block Diagram Overview of the Dissertation stages

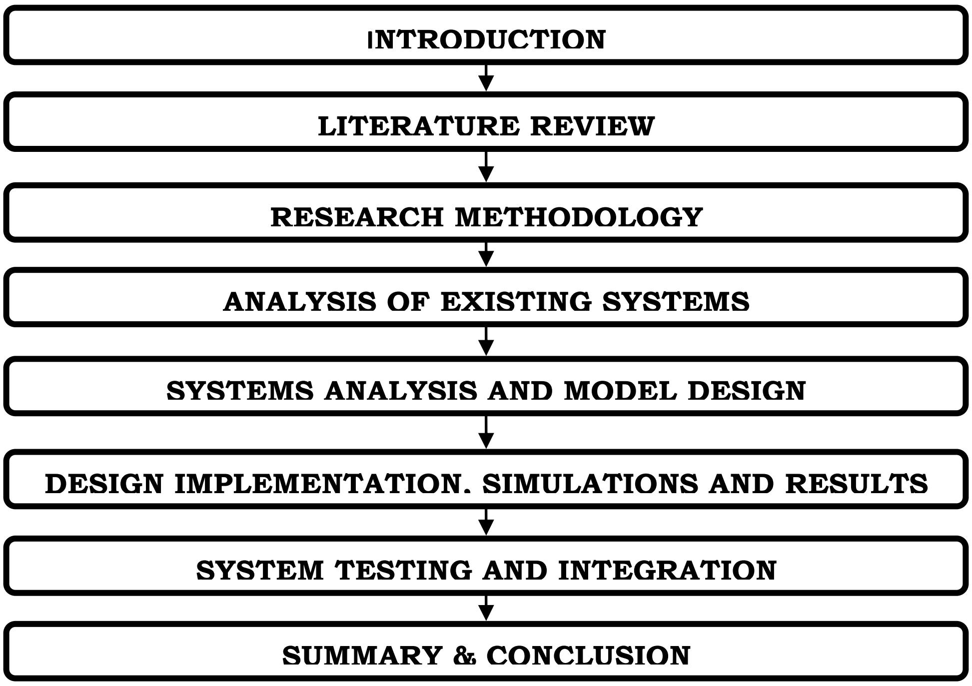


Figure 1: Block diagram overview of the dissertation stages

CHAPTER TWO

LITERATURE REVIEW

This chapter deals with the review of related literatures. It addresses the different basic concepts which are important in grasping what cloud-based collaboration platform entails. It also presents a review of various works done in the recent past on research collaboration platforms by different researchers. Here, the general concept of e-Research, collaboration in research and cloud computing were presented.

2.1 The Basic Concept of Cloud Computing

Cloud computing is a model for enabling network users on-demand access to a shared pool of configurable computing resources that can be rapidly provisioned and released to the client without direct service provider interaction (Subramanian, 2009). It can also be defined as the use of computer technology that harnesses the processing power of many inter-networked computers while concealing the structure that is behind it (Gens, 2008). Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (Wikipedia, n.d.). The name comes from the use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams (U.S. Patent, 1994). It is essentially the use of virtual servers made available over the internet. Cloud computing entrusts remote services with a user's data, software and computation. A slow migration towards this has been going on for several years, mainly due to the infrastructure and support costs that go into standalone hardware. It can simply be broken down to a browser based application that is hosted on a remote server. Cloud computing as it relates to Internet technology is all around us. Accessing mails and searching for information online are all due to the power of processing technology that exists at a distant location without the user knowing about it.

2.1.1 Explanations of common terminologies used in cloud computing

Some of the terms associated with cloud computing include the following (Gens, 2008).

i. **Back office:** Making technology easier for customers saves companies money, by taking the technical issues out of the equation so that businesses can focus their energy on creating a superior product or service. This is commonly known as back office tasks. They are generally rudimentary data parsing procedures that are time consuming as well as tedious. Back office applications are software that an organization uses to administer operations that are not related to any direct sales effort and interfaces that are not seen by consumers. An example of a back office service that is out today is Amazon's Web Services platform.

ii. **Web 2.0:** Web 2.0 cloud computing is a blanket term, but it is usually associated with some type of social networking technology – that is, a large number of social users that are interconnected via their relations with the people and things they find interesting. It describes a second generation of the World Wide Web focused on the ability for people to collaborate and share information online. It basically refers to the transition from static HTML web pages to a more dynamic web that is more organised. One of the biggest Web 2.0 companies today is Facebook.

iii. **On-demand computing:** On demand computing is a business terminology and would refer to back office processing power, for example a remote data processing center that processes payroll functions for a company located thousands of miles away. It is an enterprise-level model of technology and computing in which resources are provided on an as-needed and when-needed basis. This type of model was created to overcome the challenge to enterprises in being able to meet fluctuating demands.

iv. **Thin client:** A thin client is a term used for a terminal that connects to the cloud. This could be a computer, a cell phone or even an mp3 player. It can be referred to as software as well. As long as the device can connect to the cloud, it is known as a thin client for all intents and purposes. The meaning behind its being "thin" is that it does not require much processing power to be a client to the cloud itself.

v. **Workload migration:** Workload migration is the concept of optimizing server farm technology to be data and energy efficient (Gens, 2008). With so much processing ability coupled with an enormous amount of power consumption, companies managing server farm technology are finding that they need IT people who are well versed in workload migration technology to be able to manage all that this entails. Some cloud computing companies tout services to help companies with workload migration, offering services that assist their clients with the "internal cloud" process.

vi. **Server farm:** A server farm is a cluster of computers whose sole purpose is to provide processing power greater than what a single machine would be able to do on its own (Gens, 2008). A perfect example of this would be what companies use for web hosting of individual websites. Even though there is one website the server farm provides failover capability in case something was to happen to any single machine hosting the website. It is ideal for server farms to be located near a reliable source of power.

2.1.2 Key characteristics of cloud computing

The National Institute of Standards and Technology's (NIST) definition of cloud computing identifies five essential characteristics (NIST,2012):

- **On-demand self-service:** A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.

- **Broad network access:** Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).
- **Resource pooling:** The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.
- **Rapid elasticity:** Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.
- **Measured service:** Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

Cloud computing can also be said to exhibit the following characteristics:

- Agility improves with users' ability to re-provision technological infrastructure resources (U.S. Patent, 1994).
- Application programming interface (API) accessibility to software that enables machines to interact with cloud software in the same way the user interface facilitates interaction between humans and computers. Cloud computing systems typically use REST (representational-based transfer)-based APIs (U.S. Patent, 1994).

- Cost is claimed to be reduced and in a public cloud delivery model capital expenditure is converted to operational expenditure (Strachey, 1959). This is purported to lower barriers to entry, as infrastructure is typically provided by a third-party and does not need to be purchased for one-time or infrequent intensive computing tasks.
- Device and location independence enable users to access systems using a web browser regardless of their location or what device they are using (e.g., PC, mobile phone)
- Virtualization technology allows servers and storage devices to be shared and utilization to be increased. Applications can be easily migrated from one physical server to another.
- Multitenancy as shown in the work in (Corbató, Daggett & Daley,1962). enables sharing of resources and costs across a large pool of users thus allowing for:
    - Centralization of infrastructure in locations with lower costs (such as real estate, electricity, etc.)
    - Peak-load capacity increases (users need not engineer for highest possible load-levels)
    - Utilisation and efficiency improvements for systems that are often only 10–20% utilised
- Reliability is improved if multiple redundant sites are used, which makes well-designed cloud computing suitable for business continuity and disaster recovery
- Scalability and elasticity via dynamic ("on-demand") provisioning of resources on a fine-grained, self-service basis near real-time without users having to engineer for peak loads (Farber, 2008).
- Performance is monitored and consistent and loosely coupled architectures are constructed using web services as the system interface according to the work in (Farber, 2008).

- Security could improve due to centralization of data, increased security-focused resources, etc (Cryptoclarity, 2009). But concerns can persist about loss of control over certain sensitive data, and the lack of security for stored kernels. Mills (2009) stated that security is often as good as or better than other traditional systems, in part because providers are able to devote resources to solving security issues that many customers cannot afford. However, the complexity of security is greatly increased when data is distributed over a wider area or greater number of devices and in multi-tenant systems that are being shared by unrelated users. In addition, user access to security audit logs may be difficult or impossible. Private cloud installations are in part motivated by users' desire to retain control over the infrastructure and avoid losing control of information security.
- Maintenance of cloud computing applications is easier, because they do not need to be installed on each user's computer and can be accessed from different places.

Cloud computing was chosen for the deployment of this application due to these characteristics and benefits.

2.1.3 Related Technologies to Cloud Computing

Cloud computing is often compared to the following technologies, each of which shares certain aspects with cloud computing:

- **Grid Computing:** Grid computing is a distributed computing paradigm that coordinates networked resources to achieve a common computational objective (Zhang, Cheng and Boutaba, 2010). It is the collection of loosely coupled, heterogeneous, and geographically dispersed computer resources from multiple locations to reach a common goal. It is a distributed system with non-interactive workloads that involve a large number of files. The development of Grid computing was originally driven by scientific applications which are usually computation-intensive. Cloud computing is similar to Grid computing in that it also employs distributed resources to achieve application-level objectives. However, cloud computing takes one step further by leveraging virtualization technologies at multiple levels (hardware and application platform) to realize resource sharing and dynamic resource provisioning.

- **Utility Computing:** it involves the provision of resources on-demand and customers are charged based on usage rather than a flat rate. Cloud computing can be perceived as a realization of utility computing. It adopts a utility-based pricing scheme entirely for economic reasons. With on-demand resource provisioning and utility based pricing, service providers can truly maximize resource utilization and minimize their operating costs.
- **Virtualization:** Virtualization is a technology that abstracts away the details of physical hardware and provides virtualized resources for high-level applications. A virtualized server is commonly called a virtual machine (VM). Virtualization forms the foundation of cloud computing, as it provides the capability of pooling computing resources from clusters of servers and dynamically assigning or reassigning virtual resources to applications on-demand.
- **Autonomic Computing:** Originally coined by IBM in 2001, autonomic computing aims at building computing systems capable of self-management, i.e. reacting to internal and external observations without human intervention (Zhang et al, 2010). The goal of autonomic computing is to overcome the management complexity of today’s computer systems. Although cloud computing exhibits certain autonomic features such as automatic resource provisioning, its objective is to lower the resource cost rather than to reduce system complexity.

In summary, cloud computing leverages virtualization technology to achieve the goal of providing computing resources as a utility. It shares certain aspects with grid computing and autonomic computing but differs from them in other aspects. Therefore, it offers unique benefits and imposes distinctive challenges to meet its requirements. This work chose cloud computing for the application integration based on these features.

2.1.4 Cloud computing architecture

This section describes the architectural, business and various operation models of cloud computing.

2.1.4.1 A layered model of cloud computing

Generally speaking, the architecture of a cloud computing environment can be divided into 5 layers:

i. **Facility Layer:** Heating, ventilation, air conditioning (HVAC), power, communications, and other aspects of the physical plant comprise the lowest layer, the facility layer.

ii. **The hardware layer:** This layer is responsible for managing the physical resources of the cloud, including physical servers, routers, switches, power and cooling systems. In practice, the hardware layer is typically implemented in data centres. A data center usually contains thousands of servers that are organized in racks and interconnected through switches, routers or other fabrics (Zhang et al, 2010). Typical issues at hardware layer include hardware configuration, fault tolerance, traffic management, power and cooling resource management.

iii. **The infrastructure layer:** Also known as the virtualization layer, the infrastructure layer creates a pool of storage and computing resources by partitioning the physical resources using virtualization technologies such as Xen , KVM and VMware. The infrastructure layer is an essential component of cloud computing, since many key features, such as dynamic resource assignment, are only made available through virtualization technologies.

iv. **The platform layer:** Built on top of the infrastructure layer, the platform layer consists of operating systems and application frameworks. The purpose of the platform layer is to minimize the burden of deploying applications directly into VM containers. For example, Google App Engine operates at the platform layer to provide API support for implementing storage, database and business logic of typical web applications.

v. **The application layer:** At the highest level of the hierarchy, the application layer consists of the actual cloud applications. Different from traditional applications, cloud applications can leverage the automatic-scaling feature to achieve better performance, availability and lower operating cost. Compared to traditional service hosting environments such as dedicated server farms, the architecture of cloud computing is more modular. Each layer is loosely coupled with the layers above and below, allowing each layer to evolve separately. This is similar to the design of the OSI

2.1.4.2`Service Models

a. **Software as a Service (SaaS):** The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface (Gens, 2008). The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings. Examples of SaaS include: Google Apps, Microsoft Office 365, Onlive, GT Nexus, etc.
b. **Platform as a Service (PaaS):** The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or

control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment. Examples of PaaS include: AWS Elastic Beanstalk, Cloud Foundry, Heroku.

c. **Infrastructure as a Service (IaaS):** The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls). Offering virtualized resources (computation, storage, and communication) on demand is known as Infrastructure as a Service (IaaS) (Zhang et al, 2010). Infrastructure services are considered to be the bottom layer of cloud computing systems. Amazon Web Services mainly offers IaaS, which in the case of its EC2 service means offering virtual machines with a software stack that can be customized similar to how an ordinary physical server would be customized. Users are given privileges to perform numerous activities to the server, such as: starting and stopping it, customizing it by installing software packages, attaching virtual disks to it, and configuring access permissions and firewalls rules (Buyya, Broberg, & Goscinski, 2011).

2.1.4.3 Architectural design of data centres

A data center, which is home to the computation power and storage, is central to cloud computing and contains thousands of devices like servers, switches and routers (Cisco, 2008). The data center is home to the computational power, storage, and applications necessary to support an enterprise business (Al-Fare, Loukissas, & Vahdat, 2008). The data center infrastructure is central to the IT architecture, from which all content is sourced or passes through. Proper planning of the data center infrastructure design is critical, and performance, resiliency, and scalability need to be carefully considered.

Another important aspect of data center design is flexibility in quickly deploying and supporting new services so as to be able to support new applications in a short time frame. This can result in a significant competitive advantage. For such a design serious consideration has to be given to areas such as port density, access layer uplink bandwidth, true server capacity, and oversubscription, to name just a few. The data center network design is based on a proven layered approach, which has been tested and improved over the past several years in some of the largest data center implementations in the world. The layered approach is the basic foundation of the data center design that seeks to improve scalability, performance, flexibility, resiliency, and maintenance.

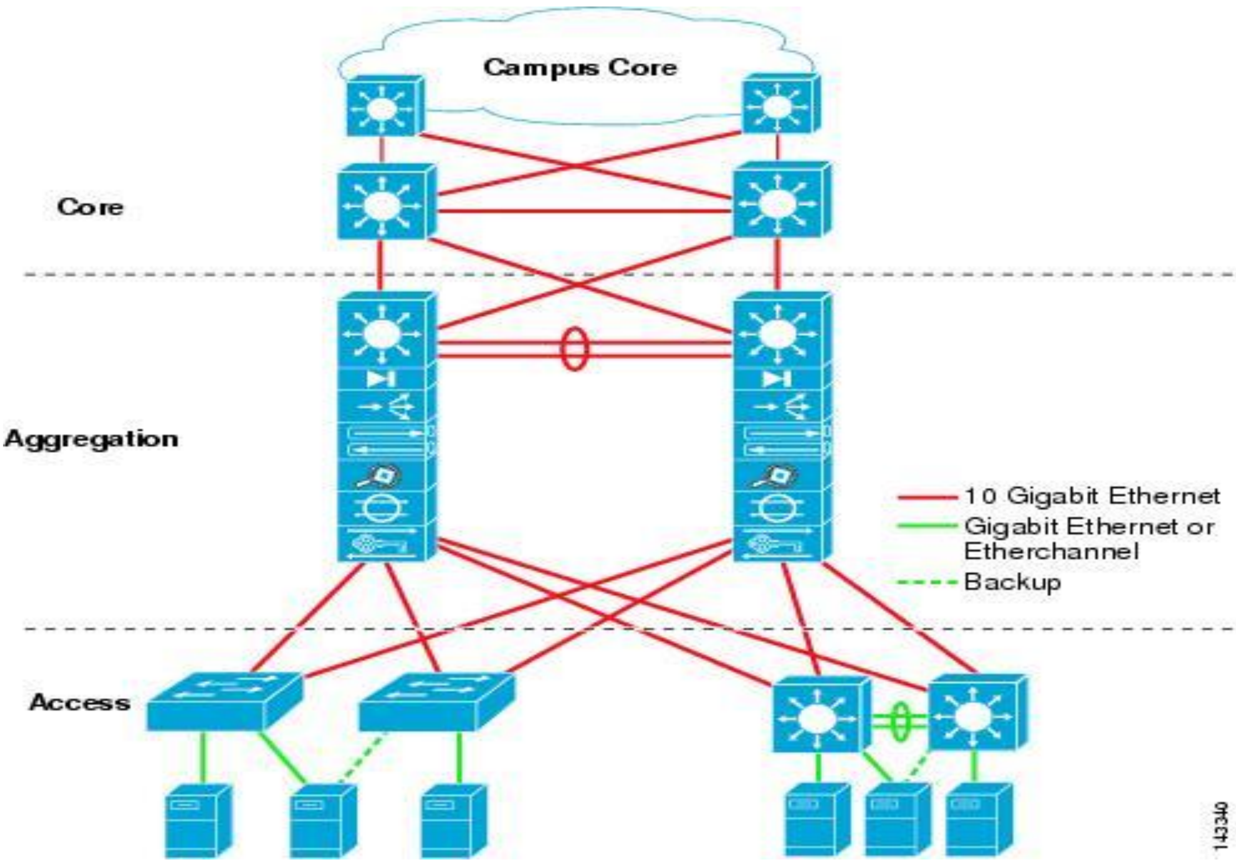


Figure2.1: Basic layered design of Data Center Network Infrastructure business (Al-Fare et al,2008)

The layers of the data center design are the core, aggregation, and access layers.

- **Core layer**: Provides the high-speed packet switching backplane for all flows going in and out of the data center. It also provides connectivity to multiple aggregation modules and provides a resilient Layer 3 routed fabric with no single point of failure. The core layer runs an interior routing protocol, such as OSPF or EIGRP, and load balances traffic between the campus core and aggregation layers
- **Aggregation layer modules**: they provide important functions, such as service module integration, Layer 2 domain definitions, spanning tree processing, and default gateway redundancy. Server-to-server multi-tier traffic flows through the aggregation layer and can use services, such as firewall and server load balancing, to optimize and secure applications. The smaller icons within the aggregation layer switch in figure 2.1 represent the integrated service modules. These modules provide services, such as content switching, firewall, SSL offload, intrusion detection, network analysis, and more.
- **Access layer**: this is where the servers physically attach to the network. The server components consist of 1RU (one-rack unit) servers, blade servers with integral switches, blade servers with pass-through cabling, clustered servers, and mainframes with OSA adapters. The access layer network infrastructure consists of modular switches, fixed configuration 1 or 2RU switches, and integral blade server switches. Switches provide both Layer 2 and Layer 3 topologies, fulfilling the various servers broadcast domain or administrative requirements.

Basically, the design of data center network architecture should meet the following objectives according to Greenberg et al (2009) and Guo, Xiaofeng & Shun-Liang (2008):

- **Uniform high capacity**: The maximum rate of a server-to-server traffic flow should be limited only by the available capacity on the network-interface cards of the sending and receiving servers, and assigning servers to a service should be independent of the network topology. It should be possible for an arbitrary host in the data center to communicate with any other host in the network at the full bandwidth of its local network interface.

- **Free Virtual Machine migration:** Virtualization allows the entire VM state to be transmitted across the network to migrate a VM from one physical machine to another. A cloud computing hosting service may migrate VMs for statistical multiplexing or dynamically changing communication patterns to achieve high bandwidth for tightly coupled hosts or to achieve variable heat distribution and power availability in the data center. The communication topology should be designed so as to support rapid virtual machine migration.
- **Resiliency:** Failures will be common at scale. The network infrastructure must be fault-tolerant against various types of server failures, link outages, or server-rack failures. Existing unicast and multicast communications should not be affected to the extent allowed by the underlying physical connectivity.
- **Scalability:** The network infrastructure must be able to scale to a large number of servers and allow for incremental expansion.
- **Backward compatibility:** The network infrastructure should be backward compatible with switches and routers running Ethernet and IP. Because existing data centres have commonly leveraged commodity Ethernet and IP based devices, they should also be used in the new architecture without major modifications.

The data center design models include:

a. **The multi-tier model:** this is the most common design in the enterprise. The multi-tier data center model is dominated by HTTP-based applications in a multi-tier approach. It is based on the web, application, and database layered design supporting commerce and enterprise business ERP and CRM solutions. This type of design supports many web service architectures, such as those based on Microsoft .NET or Java 2 Enterprise Edition. Today, most web-based applications are built as multi-tier applications. The multi-tier model uses software that runs as separate processes on the same machine using inter-process communication (IPC) or on different machines with communications over the network.

Typically, the following three tiers are used:

- Web-server
- Application
- Database

The multi-tier model relies on security and application optimization services to be provided in the network.

b. **The server cluster model:** this model grew out of the university and scientific community to emerge across enterprise business verticals including financial, manufacturing, and entertainment. The server cluster model is most commonly associated with high-performance computing (HPC), parallel computing, and high-throughput computing (HTC) environments, but can also be associated with grid/utility computing. These designs are typically based on customized, and sometimes proprietary, application architectures that are built to serve particular business objectives.

Another area of rapid innovation in the industry is the design and deployment of shipping-container based, modular data center (MDC). In an MDC, normally up to a few thousands of servers, are interconnected via switches to form the network infrastructure. Highly interactive applications, which are sensitive to response time, are suitable for geo-diverse MDC placed close to major population areas. The MDC also helps with redundancy because not all areas are likely to lose power, experience an earthquake, or suffer riots at the same time. Rather than the three-layered approach discussed above, Guo et al (2008) and Guo, Lu & Li, (2009) proposed server-centric, recursively defined network structures of MDC

2.1.5 Major components of cloud computing

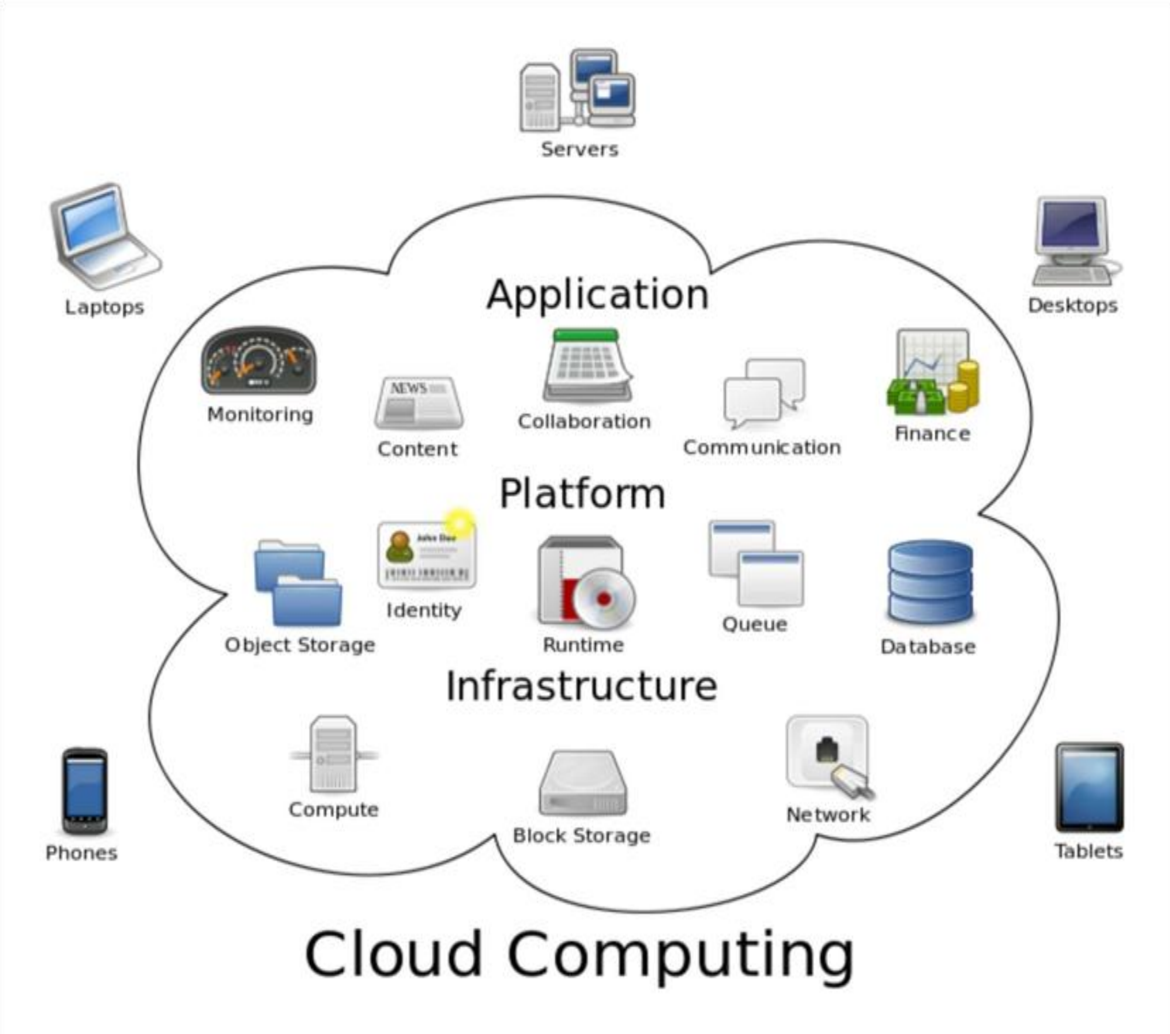


Figure 2.2 Major Components of cloud computing (Gens, 2008)

a. **Application:** The application itself, which is the component that end users will spend most of their time using, is hosted on severs that are remote from the user and can be run in real time from a thin client that hosts the application through a web browser. The majority of applications that are hosted on clouds are run via browsers. This has major benefits in that there is no installation of the application, no maintenance required and support issues are streamlined because the software is hosted on a machine that is dedicated to that software so there is no worry of external influences of the thin client on the software itself. Proponents claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and enables IT to more rapidly adjust resources to meet fluctuating and unpredictable business demand (Baburajan, 2011). Examples of applications are facebook and skype.

b. **Client:** A cloud client is computer hardware and/or software which relies on the Cloud for application delivery, or which is specifically designed for delivery of cloud services, and which is in either case essentially useless without a Cloud. We have the mobile, thin and thick client. Examples are iphones, Mozilla firefox, zonbu e.t.c.

c. **Infrastructure:** Cloud infrastructure (e.g. Infrastructure as a service) is the delivery of computer infrastructure (typically a platform virtualization environment) as a service (Oestreich, 2011). A cloud infrastructure is the collection of hardware and software that enables the five essential characteristics of cloud computing. The cloud infrastructure can be viewed as containing both a physical layer and an abstraction layer. The physical layer consists of the hardware resources that are necessary to support the cloud services being provided, and typically includes server, storage and network components. The abstraction layer consists of the software deployed across the physical layer, which manifests the essential cloud characteristics. Conceptually the abstraction layer sits above the physical layer (Gens, 2008). Some companies use a process known as full virtualization which is a technique where the user is able to completely simulate the hardware that is run on the individual sever. Often this is used to emulate new hardware types e.g. Go Grid and Skytap. Another infrastructure process is called grid computing. It is the usage of sever machines network together for heavier processing power. Grid computing can be implemented via several geographic locations. Example is Sun

Grid. One newer infrastructure process is known as paravirtualization. Paravirtualization utilizes more than one machine to emulate one process. Example is Amazon’s Elastic Compute Cloud.

- **d. Platform:** A cloud platform facilitates deployment of applications without the cost and complexity of buying and managing the underlying hardware and software layers (Guardian, 2008) e.g. Google App Engine. The cloud platform is referring to the way that applications can be deployed.
- **e. Service:** A cloud service (e.g. Web Service) is a software system designed to support interoperable machine-to-machine interaction over a network (Barr, 2008) which may be accessed by other cloud computing components, software or end users directly. It refers to what users can reap from their cloud experience. One of the most popular services in recent years that uses cloud computing would be mapping services e.g. Yahoo Maps.
- **f. Storage:** Cloud storage is the delivery of data storage as a service (including database-like services), often billed on a utility computing basis (e.g. per gigabyte per month) (Cheng, 2007). For example: Database (Amazon SimpleDB), Network attached storage (MobileMe iDisk component), Synchronisation (Live Mesh), Web service (Amazon Simple Storage Service). Traditional storage vendors have recently begun to offer their own flavour of cloud storage, sometimes in conjunction with their existing software products (e.g. Symantec's Online Storage for Backup Exec). Physical storage can be expensive for companies looking to expand their storage needs. It is cheaper to go with the cloud to be able to expand and collapse as the business dictates. Server farms and thus cloud computing offer a great way to keep data safe in case there is an emergency situation.

2.1.6 Deployment Models:

a. **Private cloud:** The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises. Undertaking a private cloud project requires a significant level and degree of engagement to virtualize the business environment, and it will require the organization to re-evaluate decisions about existing resources. When it is done right, it can have a positive impact on a business, but every one of the steps in the project raises security issues that must be addressed in order to avoid serious vulnerabilities. They have attracted criticism because users still have to buy, build, and manage them and thus do not benefit from less hands-on management, (Foley, 2008). essentially lacking the economic model that makes cloud computing such an intriguing concept (Haff, 2009).
b. **Community cloud:** The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises. The costs are spread over fewer users than a public cloud (but more than a private cloud), so only some of the cost savings potential of cloud computing are realized
c. **Public cloud:** The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider. Public cloud applications, storage, and other resources are made available to the general public by a

service provider. These services are free or offered on a pay-per-use model.

d. **Hybrid cloud:** The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds). By utilizing "hybrid cloud" architecture, companies and individuals are able to obtain degrees of fault tolerance combined with locally immediate usability without depending on internet connectivity. Hybrid cloud architecture requires both on-premises resources and off-site (remote) server-based cloud infrastructure (Hogan, Liu, Sokol, &Tong, 2011).

2.1.7 Service Level Agreement

A Service Level Agreement **(**SLA) defines the interaction between a cloud service provider and a cloud service consumer (Ahronovitz, 2010).

An SLA contains several things:

- A set of services the provider will deliver
- A complete, specific definition of each service
- The responsibilities of the provider and the consume
- A set of metrics to determine whether the provider is delivering the service as promised
- An auditing mechanism to monitor the service
- The remedies available to the consumer and provider if the terms of the SLA are not met
- How the SLA will change over time

The marketplace features two types of SLAs: Off-the-shelf agreements and negotiated agreements between a provider and consumer to meet that consumer's specific needs. It is unlikely that any consumer with critical data and applications will be able to use the first type. Therefore the consumer's first step in approaching an SLA (and the cloud in general) is to determine how critical their data and applications are.

Most public cloud services offer a non-negotiable SLA. With these providers, a consumer whose requirements aren't met has two remedies:

1. Accept a credit towards next month's bill (after paying this month's bill in full), or
2. Stop using the service.

Clearly an SLA with these terms is unacceptable for any mission-critical applications or data. On the other hand, an SLA with these terms will be far less expensive than a cloud service provided under a negotiated SLA.

Key factors to be considered in an SLA include the following:

i. Security: The security-related aspects of an SLA should be written. A cloud consumer must understand their security requirements and what controls and federation patterns are necessary to meet those requirements. In turn, a cloud provider must understand what they must deliver to the consumer to enable the appropriate controls and federation patterns.
ii. Data Encryption: If a consumer is storing vital data in the cloud, it is important that the data be encrypted while it is in motion and while it is at rest. The details of the encryption algorithms and access control policies should be specified in the SLA.
iii. Privacy: Basic privacy concerns are addressed by requirements such as data encryption, retention and deletion. In addition, an SLA should make it clear how the cloud provider isolates data and applications in a multi-tenant environment.
iv. Data Retention and Deletion: Many organizations have legal requirements that data must be kept for a certain period of time. Some

organizations also require that data be deleted after a certain period of time. Cloud providers must be able to prove that they are compliant with these policies.

v. Hardware Erasure and Destruction: A common source of data leaks is the improper disposal of hardware. If a cloud provider's hard drive fails, the platters of that disk should be zeroed out before the drive is disposed or recycled. On a similar note, many cloud providers offer the added protection of zeroing out memory space after a consumer powers off a virtual machine.

vi. Regulatory Compliance: Many types of data and applications are subject to regulations. Some of those are laws, while others are industry-specific. If regulations must be enforced, the cloud provider must be able to prove their compliance.

vii. Transparency: Under the SLAs of some cloud providers, the consumer bears the burden of proving that the provider failed to live up to the terms of the SLA. A provider's service might be down for hours, but consumers who are unable to prove that downtime are not eligible for any sort of compensation. For critical data and applications, providers must be proactive in notifying consumers when the terms of the SLA are breached. This includes infrastructure issues such as outages and performance problems as well as security incidents.

viii. Certification: There are many different certifications that apply to certain types of data and applications. For example, consumer might have the requirement that their cloud provider be ISO 27001 certified. The provider would be responsible for proving their certification and keeping it up-to-date.

ix. Terminology for key performance indicators: The term uptime can be defined in many ways. Often that definition is specific to a provider's architecture. If a provider has a data center on six continents, does uptime refer to a particular data center or any data center? If the only available data center is on another continent, that uptime is unlikely to be acceptable. To make matters worse, other cloud providers will use definitions specific to their architectures. This makes it difficult to compare cloud services. A set of industry-defined terms for different key performance indicators would make it much easier to compare SLAs in particular (and cloud services in general).

x. Monitoring: If a failure to meet the terms of an SLA has financial or legal consequences, the question of who should monitor the performance of the provider (and whether the consumer meets its responsibilities as well) becomes crucial. It is in the provider's interest to define uptime in the broadest possible terms, while consumers could be tempted to blame the provider for any system problems that occur. The best solution to this problem is a neutral third-party organization that monitors the performance of the provider. This eliminates the conflicts of interest that might occur if providers report outages at their sole discretion or if consumers are responsible for proving that an outage occurred.

xi. Auditability: Many consumer requirements include adherence to legal regulations or industry standards. Because the consumer is liable for any breaches that occur, it is vital that the consumer be able to audit the provider's systems and procedures. An SLA should make it clear how and when those audits take place. Because audits are disruptive and expensive, the provider will most likely place limits and charges on them.

xii. Metrics: Monitoring and auditing require something tangible that can be monitored as it happens and audited after the fact. The metrics of an SLA must be objectively and unambiguously defined. Cloud consumers will have an endless variety of metrics depending on the nature of their applications and data. Although listing all metrics is impossible, some of the most common are:

- Throughput – How quickly the service responds
- Reliability – How often the service is available
- Load balancing – When elasticity kicks in (new VMs are booted or terminated, for example)
- Durability – How likely the data is to be lost
- Elasticity – The ability for a given resource to grow infinitely, with limits (the maximum amount of storage or bandwidth, for example) clearly stated
- Linearity – How a system performs as the load increases
- Agility – How quickly the provider responds as the consumer's resource load scales up and down
- Automation – What percentage of requests to the provider are handled without any human interaction
- Customer service response times – How quickly the provider responds to a service request. This refers to the human interactions required when something goes wrong with the on-demand, self-service aspects of the cloud.

xiii. Machine-Readable SLAs: A machine-readable language for SLAs would enable an automated cloud broker that could select a cloud provider dynamically. One of the basic characteristics of cloud computing is on-demand self-service; an automated cloud broker would extend this characteristic by selecting the cloud provider on demand as well. The broker could select a cloud provider based on business criteria defined by the consumer. For example, the consumer's policy might state that the broker should use the cheapest possible provider for some tasks, but the most secure provider for others. Although substantial marketplace demand for this requirement will take some time to develop, any work on standardizing SLAs should be done with this in mind.

xiv. Human Interaction: Although on-demand self-service is one of the basic characteristics of cloud computing, the fact remains that there will always be problems that can only be resolved with human interaction. These situations must be rare, but many SLAs will include guarantees about the provider's responsiveness to requests for support. Typical guarantees will cover how many requests the consumer can make, how much they will cost and how soon the provider will respond

2.1.8 Review of Related Literatures on Task Scheduling in Cloud Computing

Task or Job Scheduling in cloud computing refers to dispatch of computing tasks and resource pooling between different resource users according to certain rules of resource use under a given cloud circumstances (Vasani and Sanghani, 2012). At present there is not a uniform standard for job scheduling in cloud computing (Xi, 2014). Resource management and job scheduling are the key technologies of cloud computing that plays a vital role in an efficient cloud resource management. Task Scheduling can also be referred to as a set of policies to control the order of work to be performed by a computer system. A good scheduler adapts its scheduling strategy according to the changing environment and the type of task (Xi, 2014). . Task scheduling can also be called job scheduling. Job-scheduling algorithms are developed to accomplish several goals like expected outcome, efficient use of resources, low makespan, high throughput, better quality of service, maintaining efficiency etc. In job scheduling algorithms, priority of jobs is a challenging issue because some jobs need to be serviced before those other jobs which can stay for a long time. Suitable job scheduling algorithm must consider the priority of a job.

There are different types of scheduling based on different criteria, such as static vs. Dynamic, centralized vs. Distributed, offline vs. Online etc as described below (Patel, and Bhoi, 2013):

- **Static Scheduling:** Pre-Schedule jobs, all information are known about available resources and tasks and a task is assigned once to a resource, so it's easier to adapt based on scheduler's perspective
- **Dynamic Scheduling:** Jobs are dynamically available for scheduling over time by the scheduler (Patel, and Bhoi, 2013). It is more flexible than static scheduling in determining run time in advance. It is more critical to include load balance as a main factor to obtain stable, accurate and efficient scheduler algorithm
- **Centralized Scheduling:** As mentioned in dynamic scheduling, it is the responsibility of centralized / distributed scheduler to make global decision (Patel, and Bhoi, 2013). The main benefits of centralized scheduling are ease of implementation; efficiency and more control and monitoring on resources. On the other hand; such scheduler lacks scalability, fault tolerance and efficient performance as it's not recommended for large-scale grids
- **Distributed / Decentralized Scheduling:** This type of scheduling is more realistic for real cloud despite its weak efficiency when compared to centralized scheduling. There is no central control entity.
- **Pre-Emptive Scheduling:** This type of scheduling allows each job to be interrupted during execution and a job can be migrated to another resource leaving its originally allocated resource, available for other jobs. If constraints such as priority are considered, this type of scheduling is more helpful
- **Non Pre-Emptive Scheduling:** It is a scheduling process, in which resources are not being allowed to be re-allocated until the running and scheduled job finished its execution
- **Co-operative scheduling:** In this type of scheduling, system have already many schedulers, each one is responsible for performing certain

activity in scheduling process towards common system wide range based on the cooperation of procedures, given rules and current system users

- **Immediate / Online Mode:** In this type of scheduling, scheduler schedules any recently arriving job as soon as it arrives with no waiting for next time interval on available resources at that moment
- **Batch / Offline Mode:** The scheduler stores arriving jobs as group of problems to be solved over successive time intervals, so that it is better to map a job for suitable resources depending on its characteristics

Task scheduling is a core and challenging issue in cloud computing. The Task execution time cannot be predicted in cloud computing. Hence the scheduler must be dynamic. The purpose of scheduling is to increase the utilization of resources. The main advantage of job scheduling algorithm is to achieve a high performance computing and the best system throughput. Scheduling manages availability of CPU memory and good scheduling policy gives maximum utilization of resource. It is worthy of note that optimizing the transfer and processing time is crucial to an application when scheduling intensive data or computing an intensive application. As the users are largely concerned about the Quality of Services (QoS), performance optimization of cloud computing has become critical to its successful application. Queuing theory can be used to optimise the performance of multiple requests and services in cloud computing.

2.1.8.1 Job queuing for task scheduling in cloud computing

Queuing theory is a collection of mathematical models of various queuing systems. Queues or waiting lines arise when demand for a service facility exceeds the capacity of that facility i.e. the customers do not get service immediately upon request but must wait or the service facilities stand idle while waiting for customers (Sowjanya, Praveen, Satish, & Rahiman, 2011).

The basic queuing process consists of customers arriving at a queuing system to receive some service. If the servers are busy, they join the queue (i.e., wait in line). They are then served according to a prescribed Queuing model.

Queuing models can be characterised by the following parameters:

- **The arrival process of customers:** In many practical situations customers arrive according to Poisson stream
- **The behaviour of customers:** The customers may be patient or impatient to wait
- **The service discipline:** The customers can be served one by one or in batches. There may be a single server or group of servers helping the customer.

In brief a queuing system is a place where customers arrive according to an ‘arrival process to obtain service from a service facility. The service facility may contain more than one server (or more generally resources) and it is assumed that a server can serve one customer at a time. Basically, a queuing system consists of three major components:

- The input process
- The system structure
- The output process.

a. Characteristics of the Input Process:

   i. The size of arriving population: The size of the arriving customer population may be infinite in the sense that the number of potential customers from external sources is very large compared to those in the system, so that the arrival rate is not affected by the size.

   ii. Arriving patterns: Customers may arrive at a queuing system either in some regular pattern or in a totally random fashion. When customers arrive regularly at a fixed interval, the arriving pattern can be easily described by a single number – the rate of arrival.

iii. Behaviour of arriving customers: Customers arriving at a queuing system may behave differently when the system is free due to a finite waiting time or when all servers are busy.

b. Characteristics of the System Structure

i. Physical number and layout of servers: The service facility may comprise of one or more servers. A customer at the head of the waiting queue can go to any server that is free, and leave the system after receiving his/her service from that server,

ii. The system capacity: The system capacity refers to the maximum number of customers that a queuing system can accommodate, inclusive of those customers at the service facility.

c. Characteristics of the Output Process

i. Queuing discipline or serving discipline: Queuing discipline, sometimes known as serving discipline, refers to the way in which customers in the waiting queue are selected for service. In general, we have:

- First-come-first-served (FCFS)
- Last-come-first-served (LCFS)
- Priority
- Processor sharing
- Random

ii. Service-time distribution: Similar to arrival patterns, if all customers require the same amount of service time then the service pattern can be easily described by a single number. But generally, different customers require different amounts of service times; hence we again use a probability distribution to describe the length of service times the server renders to those customers.

According to Sowjanya, Praveen, Satish, & Rahiman (2011), congestion management is a QoS component that prioritizes traffic using one of the many queuing methods based on the traffic's classification or characteristics. The queuing methods are used to control congestion as it transpires. According to Guo, (2014), to optimize the performance of multiple requesters and services in cloud computing, by means of queuing theory, there is need to analyze and conduct the equation of each parameter of the services in the data center. Then, through analyzing the performance parameters of the queuing system, the synthesis optimization mode, function, and strategy will be proposed after which the simulation based on the synthesis optimization mode will be set up. A comparison of the proposed model with short service time and first in first out method shows that the proposed model can optimize the average wait time, average queue length, and the number of customer. It was also observed that with the increase in the utilization rate of the server, the waiting time also reduces. According to Yang, Tan, Dai, & Guo (2009), the response time is a major constraint in the queuing system as distribution of response time was obtained for a cloud center model as an M/M/m/m + r queuing system where both inter-arrival and service times were assumed to be exponentially distributed, and the system had a finite buffer of size m + r. The response time was broken down into waiting, service, and execution periods, assuming that all three periods are independent which is unrealistic. However, the probability distributions of response time and queue length in M/G/m cannot be obtained in closed form, which necessitated the search for a suitable approximation.

Queuing models can also be classified using Kendall’s notation as follows:

- M/G/1 queue is a queue model where arrivals are Markovian (modulated by a Poisson process), service times have a General distribution and there is a single server (Gittins, 1989). The model name is written in Kendall's notation, and is an extension of the M/M/1 queue, where service times must be exponentially distributed. The length of time required for serving an individual customer has a general distribution function. The lengths of times between arrivals and of service periods are random variables which are assumed to be statistically independent.

- **M/M/1 queue** represents the queue length in a system having a single server, where arrivals are determined by a Poisson process and job service times have an exponential distribution. The model is the most elementary of queuing models (Sturgul, 2000). and an attractive object of study as closed-form expressions can be obtained for many metrics of interest in this model. An extension of this model with more than one server is the M/M/c queue.
- **G/G/1 Model:** The queue has an infinite storage capacity. In the notation, the G stands for a general distribution with a known mean and variance; G/G/1 means that the system's inter-arrival and service times are governed by such a general distribution, and that the system has one server. The variances of the uniform distributions can be changed and the model can be used to examine Little's law. Boxma and Cohen (1999) shows that if the traffic intensity of the G/G/1 system approaches unity and the tail of the service distribution is heavier than that of the arrival distribution, the stationary actual waiting time distribution together with a contraption factor is a function of the traffic load and converges to the Kovalenko distribution. In contrast, if the reverse is the case and all other factors kept constant, the stationary actual waiting time distribution will still depend on the traffic load but converge to the negative exponential distribution. Hence, limiting distributions of queuing systems to a large extent depend on the model constructed. Moreover, a slight variation of significant parameters may shift distribution of systems.

Other queuing models according to Kendall’s notation include: M/D/1, D/M/1, M/M/∞, G1/G/1 etc.

2.1.9 Cloud Computing Standards for Security and Data Integrity

The high-priority concerns and perceived risks related to cloud computing includes ensuring the confidentiality, integrity, and availability of information and information systems. Cloud computing implementations are subject to local physical threats as well as remote, external threats. Consistent with other applications of IT, the threat sources include accidents, natural disasters and external loss of service, hostile governments, criminal organizations, terrorist groups, and intentional and unintentional introduction of vulnerabilities through internal and external authorized and unauthorized human and system access, including but not limited to employees and intruders. The characteristics of cloud computing, significantly multi-tenancy and the implications of the three service models and four deployment models, heighten the need to consider data and systems protection in the context of logical as well as physical boundaries.

Possible types of attacks against cloud computing services include the following (Ahronovitz et al, 2010):

- Compromises to the confidentiality and integrity of data in transit to and from a cloud provider;
- Attacks which take advantage of the homogeneity and power of cloud computing environments to rapidly scale and increase the magnitude of the attack;
- Unauthorized access by a consumer (through improper authentication or authorization, or vulnerabilities introduced during maintenance) to software, data, and resources in use by an authorized cloud service consumer;
- Increased levels of network-based attacks, such as denial of service attacks, which exploit software not designed for an Internet threat model and vulnerabilities in resources which were formerly accessed through private networks;
- Limited ability to encrypt data at rest in a multi-tenancy environment;

- Portability constraints resulting from nonstandard application programming interfaces (APIs) which make it difficult for a cloud consumer to change to a new cloud service provider when availability requirements are not met;
- Attacks which exploit the physical abstraction of cloud resources and exploit a lack of transparency in audit procedures or records;
- Attacks that take advantage of virtual machines that have not recently been patched; and
- Attacks which exploit inconsistencies in global privacy policies and regulations.

Major security objectives for a cloud computing implementation include the following:

- Protect customer data from unauthorized access, disclosure, modification or monitoring. This includes supporting identity management such that the customer has the capability to enforce identity and access control policies on authorized users accessing cloud services. This includes the ability of a customer to make access to its data selectively available to other users.
- Protection from supply chain threats. This includes ensuring the trustworthiness and reliability of the service provider as well as the trustworthiness of the hardware and software used.
- Prevent unauthorized access to cloud computing infrastructure resources. This includes implementing security domains that have logical separation between computing resources (e.g. logical separation of customer workloads running on the same physical server by VM monitors [hypervisors] in a multitenant environment) and using secure-by-default configurations.
- Design Web applications deployed in a cloud for an Internet threat model and embed security into the software development process.

- Protect Internet browsers from attacks to mitigate end-user security vulnerabilities. This includes taking measures to protect Internet-connected personal computing devices by applying security software, personal firewalls, and patch maintenance.
- Deploy access control and intrusion detection technologies at the cloud provider, and conduct an independent assessment to verify that they are in place. This includes (but does not rely on) traditional perimeter security measures in combination with the domain security model. Traditional perimeter security includes restricting physical access to network and devices; protecting individual components from exploitation through security patch deployment; setting as default most secure configurations; disabling all unused ports and services; using role-based access control; monitoring audit trails; minimizing the use of privilege; using antivirus software; and encrypting communications
- Define trust boundaries between service provider(s) and consumers to ensure that the responsibility for providing security is clear.
- Support portability such that the customer can take action to change cloud service providers when needed to satisfy availability, confidentiality, and integrity requirements. This includes the ability to close an account on a particular date and time, and to copy data from one service provider to another.

Organizations can protect sensitive user data in the cloud by:

- **Use of Strong Encryption standard:** The goal of encryption is to ensure that data stored in the cloud is protected against unauthorized access. Access to sensitive user data by third parties is a violation of privacy, and should never occur. In the light of PRISM revelations and major data breaches, like the recent Target breach, it is extremely important for enterprises to bolster cloud security. The surveillance programs by the U.S government have raised security concerns among many people (Kalyani, 2014a). One of the things that worry end users the most is possible access to their personal data by parties without their knowledge or permission. Strong encryption technology like the Advanced Encryption Standard (AES), is difficult to break through. AES encryption with a key length of 256 bits should be implemented for better security of users' data. Encryption can be used in data in transit, data at rest, data in processor or memory and on usernames and passwords.

- **Effective Key Management:** Encryption keys should be managed efficiently by using advanced key management technologies. As Network World recommends, enterprises should manage their encryption effectively by efficient key assignment, periodic key rotation, and re-encryption of data with new keys (Kalyani, 2014b).

- **Access Control:** Encrypting data at rest, use and in transit reduces the risk of external attacks. Internal attacks should also be taken care of. By putting the cloud service provider in charge of encrypting sensitive user data, the enterprises are opening doors for new internal attacks. Employees must be given access to the information that is required or relevant to their responsibilities. They should be trained to manage and deal with encrypted data effectively by following the security procedures of the organization.
- **Compliance with Security policies and procedures**: Enterprises should encrypt sensitive user data based on industry compliance guidelines or mandates such as HIPAA, PCI, GLBA, FISMA and so on.

2.1.9.1 Review of Literatures on Cloud Encryption Standard for Data Security

It was discovered from the study in (Ponemon, 2011). that only about 9% of cloud service providers provide security as a service to its customers. Encryption is one of the most effective data protection controls available today. Encryption integrity is based on the technologies and processes governing the cryptographic security services (Cloud Security Alliance, 2012). Encryption in cloud computing is still in a state of flux and infancy (Kalyani, 2014a).Some vendors provide encryption, while others don't. There are different kinds of encryption schemes for securing data in the cloud, sometimes integrated within a system. Encryption is not a foolproof method, and while encryption is the most effective way of data protection, it does come with certain drawbacks. Even if a cloud service provider provides encryption, there is a possibility that the keys can be accessed by them. In order for encryption to work effectively, it is extremely important to manage the encryption keys securely. In other words, when encrypted data is stored in the cloud, the keys used for encryption should be kept separate and should only be accessed by the end user. Another option is for the end users to encrypt their data before sending them to the cloud. That way unauthorised access by cloud service providers and even government spies will be avoided; even in the event of a physical intrusion/damage to cloud service provider's system, the user's data privacy and security is assured since the data available to the unauthorised user is useless since he cannot decrypt it. Key management involves the creation, use, distribution, and destruction of encryption keys. Key management is the toughest part to manage in cryptosystems.

Some of the issues with encryption key management in the cloud include (Kalyani, 2014a).:

- Insider attack: Keys can be accessed or stolen by employees without the knowledge of end users.
- Vulnerability to data breaches: the use of the same key pairs on all the machines of the end user makes all the user's machines vulnerable in the event of a hacker gaining access to just one of those machines
- Proper management of the keys to all users' accounts: The challenge is to index proper accounts with their respective keys, quickly and effectively.

- Availability: there is need for a key cache in order to retrieve keys, even in the event that a system goes offline.
- Weak passphrase: the use of a weak passphrase to protect your account increases the risks of security attacks. If the attacker gets access to the username, then he can carry out a brute force attack in an attempt to guess all possible passwords and access your account.

In order to manage the encryption keys securely, enterprises need to employ encryption in their cloud environment, while maintaining secure off-site storage of their encryption keys. The best way to protect your data in the cloud is to give end users access to the encryption keys to secure their data. This way only the end user, and nobody else, has the access to his personal data. Never store the keys used for encryption in the same place as encrypted data. The keys used for encrypting sensitive customer data should be managed effectively by periodic key rotation and re-encryption of data with new keys. Employees should not be given more access than what is needed to complete their tasks. Cloud storage companies should require strong passwords, longer keys, or complex hash algorithms to make it difficult for anyone to access user data.

2.1.9.2 Data Integrity

Data Integrity refers to data that is considered to be whole in structure and complete, or the level to which that the data meets the criteria. In addition to the information in the data itself, data integrity also encompasses any related business rules. It can also be defined as the trustworthiness of system resources over their entire life cycle. It ensures that data is of high quality, accurate, consistent and accessible and is applied in database operations such as data warehousing and business intelligence. Data integrity is the accuracy and consistency of stored data, indicated by an absence of any alteration in data between two updates of a data record (Business dictionary, n.d.). Data integrity is imposed within a database at its design stage through the use of standard rules and procedures, and is maintained through the use of error checking and validation routines. If the encryption key is stored with the CSP, the use of data encryption alone may not provide sufficient assurance that encrypted information has not been altered. Files placed into the cloud for storage may be subject to tampering or replacement in this case, and encryption alone cannot detect this. Combining data encryption with integrity protections such as digital signatures can ensure that data in the cloud remains both private and authentic. Where available, use of trusted time should be considered by using time-stamped signatures on data.

2.1.9.3 Use of Digital Signatures for Ensuring Data Integrity in the Cloud

A digital signature is a mathematical scheme for demonstrating the authenticity of a digital message or document (Wikipedia, n.d.). A valid digital signature gives a recipient reason to believe that the message was created by a known sender, such that the sender cannot deny having sent the message (authentication and non-repudiation) and that the message was not altered in transit (integrity). Digital signatures are commonly used for software distribution, financial transactions, and in other cases where it is important to detect forgery or tampering.

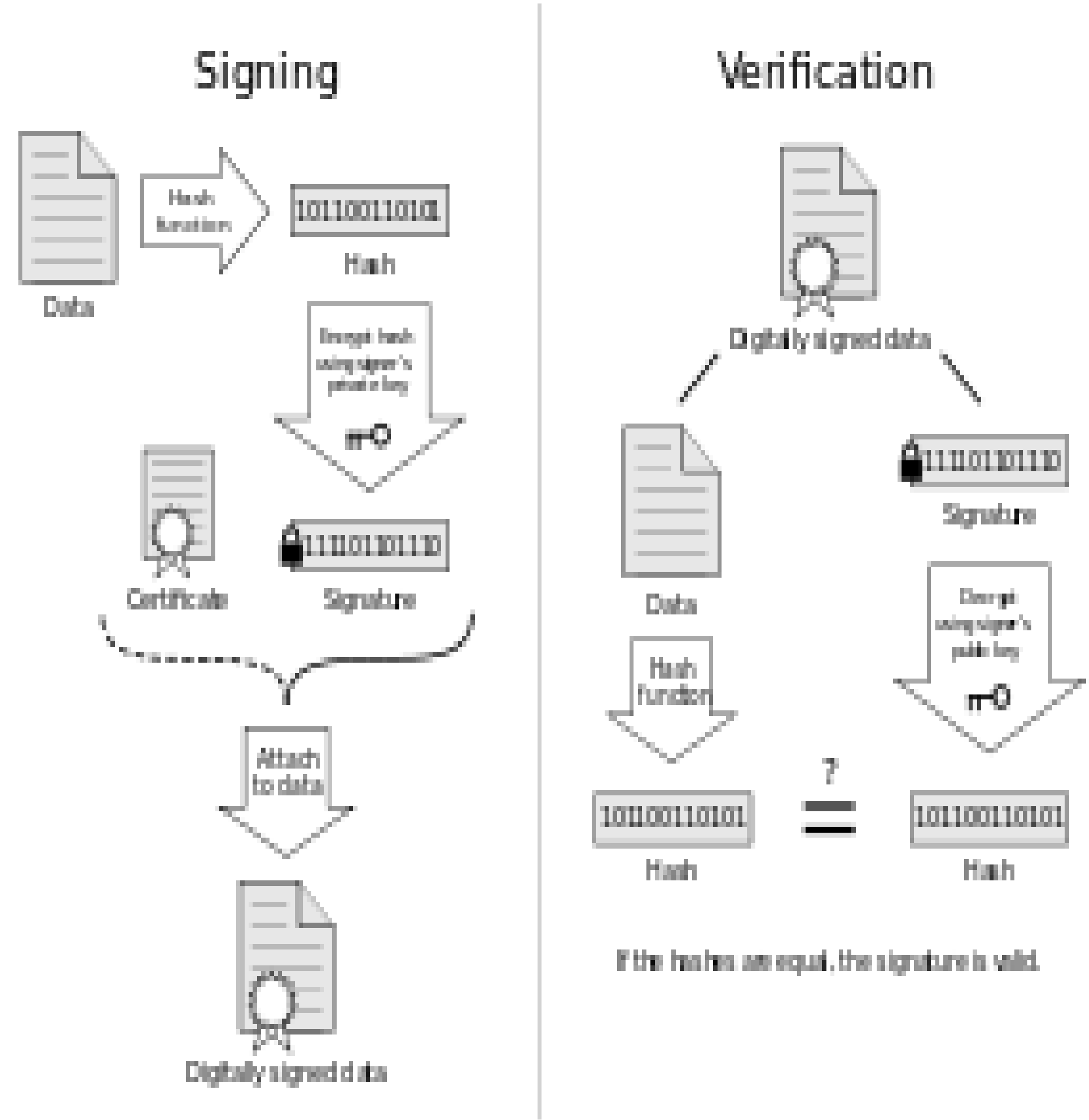


Figure2.3: Application and verification of digital signature(Wikipedia, n.d.).

A digital signature scheme typically consists of three algorithms (Wikipedia, n.d.):

- A key generation algorithm that selects a private key uniformly at random from a set of possible private keys. The algorithm outputs the private key and a corresponding public key.
- A signing algorithm that, given a message and a private key, produces a signature.

- A signature verifying algorithm that, given a message, public key and a signature, either accepts or rejects the message's claim to authenticity.

Two main properties are required. First, the authenticity of a signature generated from a fixed message and fixed private key can be verified by using the corresponding public key. Secondly, it should be computationally infeasible to generate a valid signature for a party without knowing that party's private key. A digital signature is an authentication mechanism that enables the creator of message to attach a code that act as a signature. It is formed by taking the hash of message and encrypting the message with creator's private key.

All digital signature schemes share the following basic prerequisites regardless of cryptographic theory or legal provision:

1. Quality algorithms: Some public-key algorithms are known to be insecure, practical attacks against them having been discovered.
2. Quality implementations: An implementation of a good algorithm (or protocol) with mistake(s) will not work.
3. The private key must remain private: If the private key becomes known to any other party, that party can produce perfect digital signatures of anything whatsoever.
4. The public key owner must be verifiable: there must be a way of verifying that a public key associated with Bob actually came from Bob. This is commonly done using a public key infrastructure (PKI) and the public key user association is attested by the operator of the PKI (called a certificate authority).
5. Users (and their software) must carry out the signature protocol properly.

Only if all of these conditions are met will a digital signature actually be any evidence of who sent the message, and therefore of their assent to its contents. Legal enactment cannot change this reality of the existing engineering possibilities, though some such have not reflected this actuality.

2.1.10 Challenges facing cloud computing

a. Dependability: While unlikely, it is very possible that data could be lost. It is very important to look deeply into a vendor's back up plans should something happen. Ideally knowledge of the geographical location where the data is stored would be helpful but many cloud providers are reluctant to give out this sensitive information. This could be a problem for companies that rely on the cloud to keep critical business functions up and running. One solution around dependability would be to host the most critical functions within the company while much of the non-critical processing can be done through a cloud vendor (Gens, 2008).

- Security: Being able to keep important data secure has always been a priority in IT, but with a technology that takes information outside of the virtual secure walls of most corporations, security becomes an issue. The usage of thin clients could possibly be high-jacked if people are careless with data
- Little or no reference: Because of privacy concerns, cloud vendors for the most part are unable or unwilling to present case studies about companies that are currently using their services. As a matter of fact, there are very few large companies that are publicly reporting their usage of cloud computing at a large scale level. This leaves many organizations feeling shy about using cloud computing resources.
- A bigger issue may lie in the event of a disaster recovery or backup failure issue. Most server farms are located in a well planned and secure location but one cannot be too careful to ponder the fact that something could always go wrong. The possibility is always out there to consider.

2.1.11 Benefits of cloud computing

The challenges of cloud computing notwithstanding, some of the benefits include:

- **Flexibility:** With the idea of a "server rental" model in place, it is easier to become more flexible in terms of technology resources (Menken, 2008). The reason is that businesses are able, with cloud computing, to have lateral options when it comes to technology. They can decide how much storage space to use, and how much processing power is required. While working to update software applications, the process can be pushed out much faster and more efficiently. Administrators can choose when to update an application enterprise-wide all in real time and how much they want to spend on IT with cloud technology.
- **Scalability:** With cloud computing one person can go from small to large quickly. Research organizations would be a great example in that they would be able to process heavy amounts of data at a specified time, and then go back to the norm – all without requiring those heavy servers. It is better for many organizations to simply rent the use of powerful computing as opposed to buying equipment outright. Cyclical and seasonal businesses would be a great fit for the rent-a-server structure that cloud computing avails to them. One cyclical business, like tax preparation, would be able to utilize their resources within the first six months of the year – when they are busy – and then retract their usage instantly when they are not needed.
- **Capital Investment:** IT spending takes a large portion of money out of general funds that companies could use for other pressing business needs such as marketing, research and development and human resources. With cloud computing, many rudimentary IT purchases for things like hardware will no longer be an issue as long as that task or set of tasks can be performed by the cloud.
- **Portability:** In today's global economy organizations need to have people on the ground, far from headquarters, to manage things. With cloud computing technology, organizations are able to use their

computing power wherever their people are as long as users are able to access thin clients. With thin client technology the scale of geography and time variation is flattened somewhat and this allows companies that are trying to globally integrate to be able to be more flexible than ever before.

- The redundancy of cloud computing will help system administrators to ensure constant uptime.
- Cloud computing makes the use of software collaboration tools easier thus increasing productivity. Using this method, projects can be worked on in real time with several people all at once.
- The best part is that when files are saved it is done on the cloud where it is stored for safekeeping.
- The price of energy has increased to a large degree with the fast pace of globalization. As such, cloud computing can also achieve the goal for organizations to have energy efficiency. It is more efficient for power hungry servers to all be in one location so that data center managers can better optimize them using power saving procedures. This can be done through a variety of methods which could include having servers in a timed power saving mode to reduce power usage to even being located near a renewable energy source.
- One element that may indeed have some appeal for organizations would be companies who do research that requires sudden spikes in computing power. Research and development departments, in this case, would be able to employ cloud processing for huge spikes of data testing that would need to be done on a periodic basis without requiring heavy infrastructure investment

2.2 Introduction to E-Research

The term ‘e-Research’ encapsulates research activities that use a spectrum of advanced information and communications technologies (ICT) capabilities and embraces new research methodologies emerging from increasing access to broadband communications networks, research instruments and facilities, sensor networks, data repositories and their associated data standards, management and curation tools, and high performance computing resources (Sargent, 2006). It also involves software and infrastructure services that enable secure connectivity and interoperability between researchers and the wide variety of data repositories, computers, systems and networks on which they depend.

E-Research can be seen as a broader term that includes not only non scientific research but that also refers to large-scale, distributed, national, or global collaboration in research (E-Research co-ordinating committee, 2005). It typically entails harnessing the capacity of information and communication technology (ICT) systems, particularly the power of high capacity distributed computing, and the vast distributed storage capacity fuelled by the reducing cost of memory, to study complex problems across the research landscape. It can also be said to be research conducted in virtual communities across the academic and industrial sectors using specially designed online technical facilities and services (Gabriel, 2010). The virtual community research members have the ability to share, federate and exploit the collective power of global scientific facilities, supported by a technical framework that allows participation regardless of geographic location. Arguably, e-Research has shown its potential to provide greater interactivity among academics and consequently to increase the efficiency and effectiveness of research endeavours across all disciplines. Beyond this, it actually conceives new types of research in new fields with new methodologies. As a result, the UK, Europe, North America and East Asia are investing heavily in a wide range of e-Research developments, such as the study of virtual research environments and collaborative research using grid computing as a basis for computation and data management (Zhang, n.d.).

E-Research can also be said to imply the following (Tsoi, 2004).

- To generate, curate, and, analyse research data from experiments, observations and simulations providing quality management, preservation and reliable evidence

- To develop and explore models and simulations enabling computation and data at extreme scales providing trustworthy, economic, timely and relevant results

- To enable dynamic, distributed virtual organisations facilitating collaboration within the research sector and with industry, with information and resource sharing providing security, reliability, accountability, manageability and flexibility.

E-Research entails Infrastructure support, Personnel support, Collaborative support, and Research support framework.

2.2.1 Benefits of e-Research

E-Research capabilities serve to advance and augment, rather than replace traditional research methodologies. Some of the benefits of e-Research include:

- Better utilisation of existing infrastructure and prolonged life

- Better utilisation of collected databases

- More rapid advancement of knowledge and discoveries as researchers gain greater access to information, ideas, and facilities through international collaboration (Sloan and Arrison, 2011).

- Collaboration of researchers and computer scientists together to solve fundamental problems

- Industry collaboration and technology transfer

- Discovery of knowledge whether held in digital or physical forms

- Access to data as well as the software services that are being made available to manipulate or analyse data;
- Efficient synthesis, organisation and dissemination of new knowledge
- It also opens doors. In many countries where political and economic relations are difficult or complex, scientists can and do work together to find answers and promote human advancements.
- It builds lasting relationships. While science has always transcended borders, the current level of global interaction among scientists is unprecedented. The communications revolution and today's open innovation model allow scientists to partner with colleagues worldwide.

Distributed high-performance computing, digital data resources and high speed communications are just some of the developments improving the capacity of researchers to interact with their colleagues and share data worldwide in ways previously un-heard of (E-Research co-ordinating committee, 2005). In fact, e-Research has the potential to increase the efficiency and effectiveness of research endeavours across all disciplines. Greater interactivity between researchers and an increased ability to access research outputs will benefit industry, governments and the community as a whole. The increasingly intensive use of information and knowledge is driving value creation, productivity, and economic growth.

2.3 Survey of Reported works on Models of Cloud Computing-based Research Collaboration Platform

A research collaboration platform is a category of business software that adds broad social networking capabilities to work processes (Search content management, 2011). The goal of a research collaboration platform is to foster innovation by incorporating knowledge management into research processes so researchers can share information and solve research problems more efficiently.

Vendors are taking different approaches to building collaboration platforms. Some are adding a social layer to legacy business applications while others are embedding collaboration tools into new products. All enterprise successful collaboration platforms share certain attributes- they need to be easily accessible and easy to use, they need to build for integration, and they need to come with a common set of functions that support team collaboration, issue tracking and messaging.

Virtual research environments, (VRE) are set of online tools, systems and processes interoperating to facilitate or enhance the research process within and without institutional boundaries (Lavinia, O'Sullivan, & Cormican, 2006). This means that they will help individual researchers manage the increasingly complex range of tasks involved in doing research. In addition they will facilitate collaboration among communities of researchers, often across disciplinary and national boundaries. The research processes that a VRE will support include: resource discovery, data collection, data analysis, simulation, collaboration, communication, publishing, research administration, and project management. Through the use of common standards, VREs will link with the broad digital context within which they sit, ensuring compatibility with other key systems such as those of research funders. Special attention was given to the structure of virtual teams required to support education-industry collaboration in (Search content management, 2011). It also shows that virtual teams are particularly weak at mistrust, communication and power struggles and cannot totally replace conventional teams (Borda, 2006). The Lambert Report, published in 2003, indicated that research collaborations need to show real benefits to industry in order to attract investment, by bringing academic theory to bear on commercial problems (Ebrahim, 2010). It suggested that publicly funded research must become more responsive to the needs of the economy and public services, and more focused on end-users.

From a firm's perspective, university research appears to offer a potential to enhance national competitiveness in terms of translating university staff knowledge and expertise into new products and services. Laursen and Salter (2004) explored the role of universities in shaping industrial practice. The authors found that only a limited number of firms draw directly from universities as a source of information for their innovative activities. The results imply that only a limited number of firms from specific industrial sectors (e.g. science, technology and medicine), who have certain capabilities in R&D and who have adopted an open innovation approach are keen on developing links with universities. The study, (Saguy, 2011) shows that the conflicts pertaining to university-industry relations span around confidentiality, publishing, intellectual property rights and ownership. In addition, organization, culture and funding have been identified as major constraints that may have a substantial influence on the partnership negotiations, which sometimes may affect the primary purpose of the collaborative relationship

Bezzubtseva and Ignatov (2012) talked about Crowdsourcing platforms which are social networking services used to obtain the necessary services, ideas or content from platform participants. Activities on collaboration platforms often include message (idea or comment) posting, message reading and message evaluation. Some of the platform like Witol-ogy, Imaginatik, BrightIdea, etc usually goes as a certain time-limited project, devoted to some company's problem while some others like laboratree, MyNetResearch were developed for general research.

Some universities and research institutions have also developed platforms for research collaboration with the view of breaking most of the barriers in research and development. Some of them were developed by research institutes in different countries like the network of excellence GARNET, consisting of 42 research institutes from 17 European countries developed a platform in which the core element is a knowledge management system, in which information resources are organized in tree-like structures (Brocke, Riemer, Große-Böckmann, &Richter, 2008). The platform facilitates text discussions which can also be attached to specific objects (e. g. documents), chat rooms, newsgroups and wikis and can be used for communication purposes. There is the iLab in Nigeria which is a virtual laboratory having the same structure as that of MIT being developed by Obafemi Awolowo University in Nigeria in collaboration with MIT and two other African Universities It is an open source platform. Some others cut across academic and research institutions like the Swiss Federal Institute of Technology Zurich (CISCO, 2012) which uses the university's Cisco-based network supporting virtual learning and global collaboration with other academic and research institutions, to help shorten research and development cycles and speed up time-to-benefits realization and knowledge transfer to businesses and the wider community. The UNESCO-HP brain gain project model helps promote embededness and synergetic inter and intra university collaborations (UNESCO, 2009, Palmer, 2001).

Laboratree was launched in 2008 by the Indiana University School of Medicine Bioinformatics (Pincus, Hemmert, Nellis, Peters & Mooney, n.d.) as a secure web-based platform that provides a virtual workspace for dispersed teams to improve communication, data and information sharing, thereby reducing team disconnection that leads to poor project outcomes. It is still being developed. In order to facilitate collaboration and spark innovation in energy research and development, Wiley's new open access journal for Energy science and Engineering was introduced to provide authors with a new platform for the rapid dissemination of the latest energy research (Douglas, 2012). The aim is to support researchers by connecting them to their peers, funding bodies and the wider scientific community. It is mainly for access to journals. It does not have video and audio exchange capabilities. It does not have the ability for real time data capture ability from instruments. MyNetResearch allows you to: manage your thesis and supporting documents online, share your thesis with your committee online, generate ideas through the latest news and feeds from multiple research areas and seek advice and help from thousands of other researchers from across the world (MyNetResearch, 2009). Your project in MyNetResearch is a repository of research papers - those that are shared with your committee and those that are private to you. It is just for managing thesis and for document exchange.

Other research collaboration software includes Alfresco, Confluence, Basecamp, i-Lab by Obafemi Awolowo University Nigeria, SharePoint, HUBzero, and MediaWiki. Examples of some schools that have research collaboration platforms include KTH research platform aimed at complementing the subject-specific research underway within KTH's schools. Other universities that have research portals include: University of Basel, University of Fribourg, University of Witwatersrand, Johannesburg, University of Lucerne, University of Lugano to mention but a few.

A good example of the outcome of research collaboration is SPHERE (Sensor Platform for HEalthcare in a Residential Environment) (University of Bristol, 2013) involving an interdisciplinary research collaboration led by the University of Bristol together with the Universities of Southampton and Reading which will develop home sensor systems to monitor the health and wellbeing of the people living at home.

2.4 Summary of Related Literatures on Research collaboration

The literatures reviewed showed that a wide range of communication and collaboration technologies have been adopted over the years to assist in research coordination. Mainstream communication technology adopted by engineers for project use include telephone, teleconferences, email, voice mail, discussion lists, the Web, instant messaging, voice over IP, and videoconferences. These communication paths are useful at every stage in a project's lifecycle, and support a wide range of unstructured natural language communication.

Additionally, collaborators hold meetings in meeting rooms, and conduct informal conversations in hallways, doorways, and offices. It was also discovered that there are challenges of confidentiality and intellectual property rights and ownerships. There is also the challenge of security and integrity of the data being exchanged. Some were designed for general purpose research while others are just like social networking services. Some of them were created as a time limited project devoted to some company's problem. Some were designed for collaboration among academic researchers. It was discovered that platforms having the capability of research data acquisition from remote/virtual equipments are very rare.

This work comes as a solution to some of the challenges faced in engineering and technology research collaboration by providing a platform for collaboration amongst private, government and academic researchers and the end users of research outputs - the industrialists. It will provide a platform for data acquisition from remote equipments thus making the sharing of very expensive research equipments possible. It also solves the intellectual property right problem and also serves as a platform for researchers to advertise research challenges. The system security proposed in this work will help improve data security and integrity. Also page response and server utilisation will be greatly improved. It can manage a large amount of data because of the cloud infrastructure in addition to other features like the chat room capability, libraries, journals access and other means of information exchange found in most collaboration platforms. It is worthy of note that this platform comes as a means of augmenting not completely replacing the traditional face to face research collaboration method.

2.5 Analysis of Existing Systems

Now, before developing the proposed model, it is imperative to analyse some existing research collaboration systems. Besides some university platforms offering various services like OAU and UNN were visited. The OAU cloud platforms fits better into the design goals of this work than the UNN platform. For this reason, the analysis done in this section will focus more on the information gotten from OAU and other existing platforms. The experience gained from the analysed systems led to the improvement in the design of the system proposed in this dissertation. The analysis will be in two parts covering the three main features of the research work. They are: analysis of some existing research collaboration platforms and the security systems implemented in their design, and analysis of task scheduling algorithms used by some existing cloud environments. The reason behind this is that most of the research collaboration platforms do not have their own private cloud; they depend on a third party for cloud service provision. As such getting the information as to their cloud service providers is difficult due to what they termed security reasons. Also we could only get information on the type of task scheduling algorithm used by some of the cloud service providers. There is also the challenge of getting the type of security system implemented by particular research collaboration platform because the release of such information was also considered a threat to security. Therefore, the analysis done in this work will be based on the information available to us during the period of the research work with a view to finding gaps that will be addressed by this research work.

2.5.1 Analysis of Existing Research Collaboration Platforms

There are many existing software packages that enable research collaboration, and the use of such tools by engineers and technologists would improve the effectiveness of both data sharing and workflows among collaborators. It is important to note, however, that some of these systems do not solve most of the research challenges especially the data deluge problem that the scientific community is currently facing. The storage capacity of many of these systems are scalable, but the transfer of large amounts of data over the internet remains a problem. The analysis done in this sub-section is with regards to the features of the existing research collaboration platforms being considered in this work and the type of security employed by them at the application level. From the observations made during our visit and analysis of existing reports the following findings were made on some of the existing research collaboration platforms:

2.5.1.1 Obafemi Awolowo University (OAU) platform:

OAU has a virtual laboratory called i-Lab which was developed in collaboration with Massachusetts Institute of Technology and is based on the ELVIS platform by National Instrument's Labview (Gikandi, 2006). It enables students and researchers carry out experiments over a network without being in the same geographical location. The i-Labs currently developed are the OpLab, the i-Lab for strength of materials and robotic arm, oplab for android devices, servomotor lab which allows students to carry out experiments in control engineering, and Emona Datex laboratory which enables experiments to be carried out in telecommunications. The platform is hosted by the institution's private cloud environment. A C# form based client that can be accessed from a web browser on any windows based machine was used instead of java applets. The results of the experiments are displayed on a virtual oscilloscope. The main advantage of OpLab is the great interactivity of its client. One of the weaknesses of the i-Lab includes the fact that most of the client software are very experiment specific due to its design. The implication is that a lot of code will have to be rewritten if a new experiment is desired. It did not give the students the ability to manipulate the nature of the input signal used to drive the implemented circuit. It also lacks administrative functionality and experiment data management. The security system implemented in the OAU i-Lab platform is what they called the Authenticate User method. This is used to verify if the person attempting to run the experiment is a registered user. It takes in two string parameters: Username and Password. It checks the mobile Service Broker database for the supplied username; if found, it retrieves its corresponding password and compares with the Password string entered by the potential user. No form of encryption is currently being applied to the user details stored in the database. It is a well known fact that the use of only user login details as a security measure for granting access to a system is a very poor method which is prone to attacks. Unauthorised users can easily gain access into the system and manipulate the results generated from the experiment. The correctness of experimental data being transmitted to users is of utmost importance as any error introduced will definitely affect the result of the experiment being carried out. Some other cloud-based applications developed at OAU implemented RSA (Rivest-Shamir-Adleman) Algorithm. It is an algorithm for public-key cryptography (Leena and Jaiswal, 2013). It involves a public key and a private key. The public key can be known to everyone and is used for encrypting messages but the private key is known only to the owner. The private key does not need to be sent across the internet so it is not prone to the security threat posed by the internet. The algorithm involves multiplying two large prime numbers and through additional operations deriving a set of two numbers that constitutes the public key and another set that is the private key. Both the public and the private keys are needed for encryption /decryption. The private key is used to decrypt text that has been encrypted with the public key. It is called asymmetric cryptography because it is used a pair of keys. It uses a variable size key and a variable size encryption block. It is secure in that the key length is very long, can be up to 4096 bit but has the limitation that the maximum amount of data it can encrypt is equal to its key length. Therefore, it cannot work well for systems involving the transmission of large amounts of data as is the case with engineering and technology research without the introduction of other measures like breaking the large data into smaller chunks which will definitely introduce delay thus defeating another major goal of this work which is to reduce delays. Possible attacks on RSA includes: searching the message space, guessing, cycle attacks, low exponent and factoring the public key. The RSA algorithm has the challenges of very slow key generation, slow signing and decryption which are slightly tricky to implement securely and the two-part key is vulnerable to GCD (Greatest Common Divisor) attack if poorly implemented. Unlike RSA, AES is specifically designed to resist the most sophisticated cryptographic attacks. It has very low memory requirements and is faster than RSA. It can also encrypt very large amount of data. It is therefore more suited for the research collaboration platform being proposed in this work since reduction of the overall delays in the cloud environment is a major design goal.

2.5.1.2 Alfresco:

The Alfresco CMS collaboration system is developed and maintained by the Alfresco Development Group (Alfresco, n.d.) The company offers their collaboration system in both hosted and local versions. The hosted solution is available as a cloud server environment so that deployment does not require any actual local hardware or networking expertise. The locally installed solutions include Alfresco Enterprise Edition and Alfresco Community Edition. The Community Edition is free for use as a download and comes with no support, other than developer support in the forums. An Enterprise subscription entitles users to technical support, knowledge base support, installation and configuration assistance, a warranty, and supported software maintenance. The Alfresco software is open source and developed in Java. The system operates through the Tomcat Java web server, with Alfresco running as a servlet under this system. As such, the packaged installer will operate relatively seamlessly on any operating system. It has customizable collaboration sites, event calendars, site-specific wikis, blogs, document management links, and forum-style (threaded) discussions and lists although its greatest strength is in document management. Access levels include Site Manager, Collaborator, Contributor, and Consumer, with the implied access rights for each. All access levels, except Consumer, can add documents and join in discussions, and collaborators have full access to the sites' functions, short of user administration. All of the described features are accessible via a web-based Java servlet interface that is interactive and logical. It is not user friendly. It is difficult to handle. There is no interface for the acquisition of data from research equipments. As of the time of this research, information on the security system implemented in alfresco could not be accessed.

2.5.1.3 ResearchGate

It is a social network site for scientists and researchers to share papers, ask and answer questions and collaborate. Members can share raw data and failed experimental results as well as successes. It has a scoring system which may not be indicative of true scientific impact. It has profile pages, comments, groups, job listings. The member profile/registration helps in ensuring that collaborators are people with the same research interests and with the requisite qualification. This is lacking in most of the other system. Thus, the reliability of the information generated in the platform is assured to an extent. It has no interface to remote research equipment. The experimental data shared is manually transferred to the platform. As of the time of this research, information on the security system implemented in Researchgate could not be accessed.

2.5.2 Analysis of scheduling algorithms used in some cloud computing environments

The type of scheduling algorithm used for a cloud computing environment determines the speed, efficiency, and utilization of resources in an optimized way. This is usually done to load balance a system effectively or achieve a target quality of service (Lin, 2013). There are certain factors that the scheduler is mainly concerned with. They include throughput, latency, turnaround, response time and fairness. Throughput is the number of processes that complete their execution per time unit. Latency is a measure of time delay experienced in a system. Turnaround is total time between submission of a process and its completion. Response time can be defined as the amount of time it takes from when a request was submitted until the first response produced. Fairness is the ability of the system to allocate equal CPU time to each process generally according to each process' priority. This research work focused more on improving the response time and server utilisation. Many task scheduling algorithms have been proposed for efficient allocation, scheduling and scaling of cloud resources. Information on the scheduling algorithm used by the OAU cloud environment, cloud service providers used by Research Gate and Alfresco could not be gotten as of the time of this research. This informed the decision taken in this work to base the analysis of the scheduling algorithms used in cloud environment as with other analysis done in different literatures on the cloud service provider whose information could be accessed as they also have the capability to host research collaboration applications. Table 3.1 shows some cloud environments and the scheduling algorithm implemented in them. In this section, the task scheduling algorithm used by Eucalyptus, Open Nebula, and Amazon EC2 will be analysed with the view of finding gaps that will be addresses by this work.

Table 2.1: Scheduling Algorithms used in Different Cloud Environment (Backialakshmi and Sathya, 2014)

CLOUD SERVICE PROVIDER	OPEN SOURCE	SCHEDULING ALGORITHM USED
Eucalyptus	yes	Greedy and Round robin
Open Nebula	yes	Rank matchmaker scheduling, Pre-emptive scheduling
Rackspace	yes	round robin, weighted round robin, least connections, weighted least connections
RedHat	yes	BFS ,DFS
Nimbus	yes	Virtual machine Schedulers, Priority Based Scheduling and SGE
Amazon EC2	No	Xen ,swarm, genetic
lunacloud	yes	Round robin

2.5.2.1 Eucalyptus

Eucalyptus implements Greedy algorithm and Round Robin algorithm for task scheduling. Greedy algorithm works in phases in such a way that in each phase, a decision is made that appears to be good (local optimum) at that time without regard for future consequences. It works on the principle of selecting the first fit not best fit task as the local optimum. It is used to generate approximate answers, rather than exact one which need more complicated algorithms. Its advantage is that solutions to smaller instances of a complex problem can be straightforward and easy to understand. The disadvantage is that the local optimum may lead to the worst possible long-term outcome. Pre-emption leads to increased response time and overhead to the cloud providers. It leads to resource waste i.e. low server utilisation and may also lead to disappointment for user on QoS parameters The Round Robin algorithm distributes the load equally to all the resources. Its main focus is on fairness. It uses the ring as its queue to store jobs. The same execution time is allocated to each job in the queue and it will be executed in turn. If a job can’t be completed during its turn, it will be stored back to the queue waiting for the next turn. Its advantage is that each job will be executed in turn and they don’t have to wait for the previous one to get completed. It is more favourable for systems with short time jobs. The challenge with this is that more loads are conceived on servers, thus unbalancing the traffic i.e. if the load is found to be heavy, it will take a long time to complete all the jobs. It does not consider priority and jobs with heavier loads will take longer time to complete. Round Robin is also faced with the challenge of high power consumption as the nodes servicing the various tasks will be kept turned-on for a long time. A combination of the two does not still solve the problem of longer delay for heavier loads as greedy algorithm also works better for short time jobs.

2.5.2.2 Open Nebula

It implements Ranch matchmaker and pre-emptive scheduling algorithms. The ranch matchmaking algorithm works in such a way that the hosts that do not meet the virtual machine requirements and do not have enough resources to run the virtual machine are first filtered out. This is a form of packet drop queuing technique. It prioritises the resources more suitable for the virtual machine. The resources with a higher rank according to how well suited they are to the virtual machines are used first to allocate virtual machines in order to get the best possible mapping of virtual machines to available resources. This is done by considering the physical machines different resources; such as CPU,memory and network. Given that the Match-Making algorithm is both dynamic and tries to find the best fit, it will not be as fast as other algorithms when it comes to deploying new virtual machines (Kollberg, 2014). Pre-emptive algorithm interrupts or removes previously scheduled tasks according to pre-defined criteria and re-allocates the freed resource to a new task. It is used to schedule high priority tasks if there is resource shortage. It fits better for single-capacity resources but is more complex for multi-capacity resources (as is obtainable in cloud computing) in terms of which task to pre-empt. Pre-emptive algorithm allows the system to more rapidly respond to real-time events but has the challenge that care must be taken to ensure that time critical data retains its integrity. The disadvantages of the algorithms used in open nebula also include starvation of low priority or low ranking jobs.

2.5.2.3 Amazon EC2

It implements Xen, swarm and genetic algorithms for task scheduling. In Xen scheduler, a single virtual machine consists of one or more virtual CPUs (VCPUs). The goal of the scheduler is to determine which VCPU to execute on each physical CPU (PCPU) at any instant. To do this it must determine which VCPUs are idle and which are active, and then from the active VCPUs choose one for each PCPU (Zhou, Goel, Desnoyers, & Sundaram, 2011). When there are no qualified VCPUs to run, Xen schedules an idle VCPU. It can preempt processes as needed, and it tries its best to ensure fair allocation. Xen involves tradeoffs between real-time performance and throughput. A large quantum may lead to poor real-time performance due to coarse-grained scheduling. By default, Xen boots one idle VCPU per PCPU (Xi, 2014). . Xen scheduler allowing virtual machines to consume almost all CPU time, in preference to other users. It therefore allows theft of service. Genetic algorithms are stochastic search algorithms based on mechanisms of natural selection strategy. It starts with a set of initial solutions called initial population, and then generates new solutions using genetic operators. Its advantage is that it can handle a large searching space applicable to complex objective function and can avoid trapping by local optimum solutions. It also has better load balancing and resource utilization though not optimal. Its major weakness is that it does not deal with the dynamic behaviour of resource allocation. Particle Swarm Optimization (PSO) as is a self-adaptive global search based optimization. The PSO algorithm is alike to other population-based algorithms like Genetic algorithms (GA) but, there is no direct recombination of individuals of the population. The PSO algorithm focuses on minimizing the total cost of computation of an application workflow. The objective is to minimize the total cost of execution of application workflows on Cloud computing environments. PSO balances the load on compute resources by distributing tasks to available resources. It can be used for any number of tasks and resources by simply increasing the dimension of particles and number of resources respectively. Its disadvantage is that execution time is increased. It also lacks reliability and availability criteria.

Some cloud environment uses packet drop queuing model. This model use token bucket for admission control and traffic shaping. When packets arrives and there are not enough tokens retained in the bucket, the packet can be buffered until new tokens are available or they can be marked and, hence, treated in particular way or they can be discarded depending on the configuration of the token bucket filter. This discarding or dropping of packets increases total delay in the system. There is no feed back to the system on the packets dropped; hence errors are introduced in data transmission. This affects the integrity of the system.

Figure 2.4 shows the comparison of the throughput, page response time and object response time of some of the scheduling algorithms used in the analysed cloud environments.

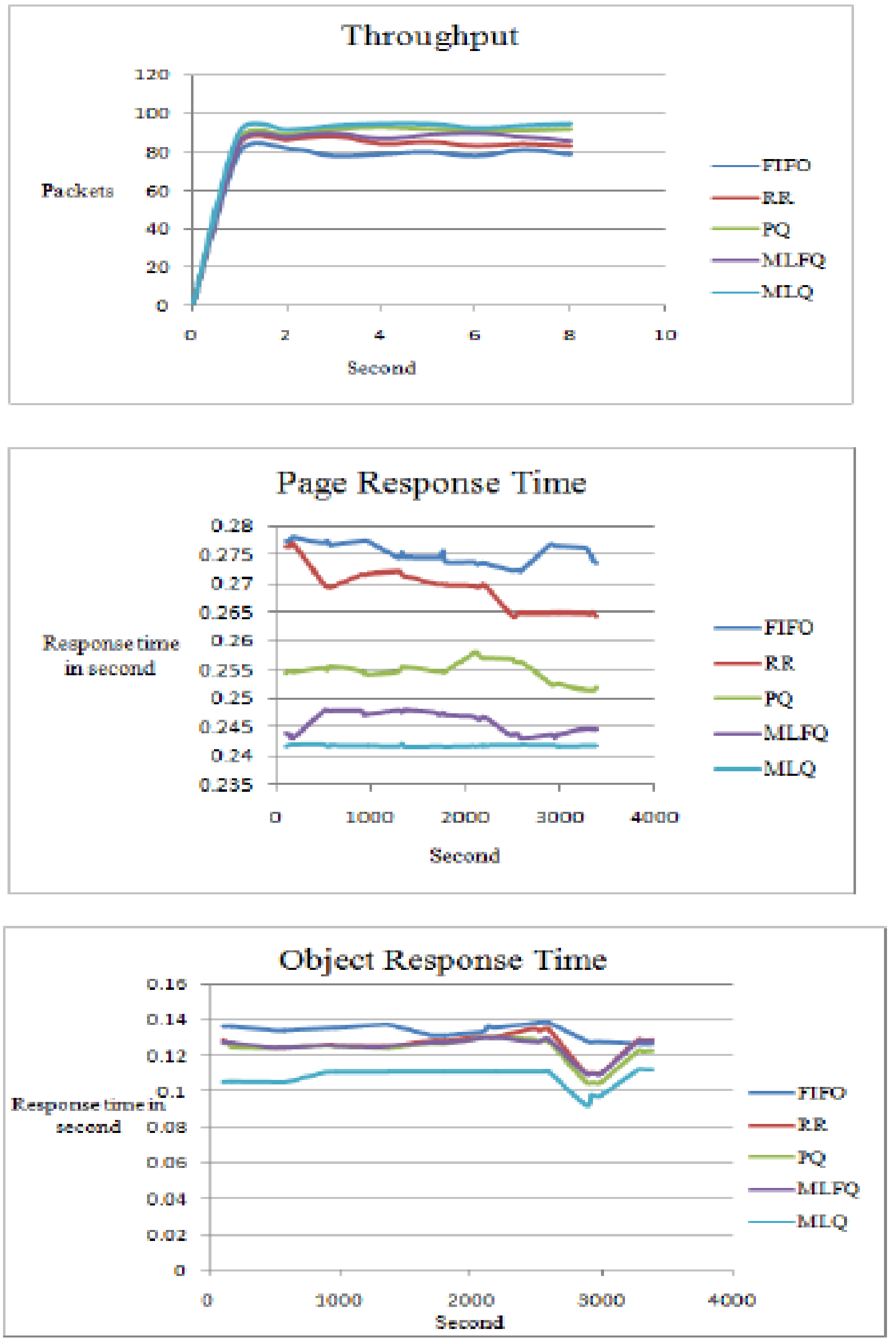


Figure 2.4: Comparison of throughput, page and object response times of some scheduling algorithms (Kumar and Khurana, 2014).

Note:

FIFO – First in First Out

RR – Round Robin

PQ – Priority Queue

MLFQ - Multi Level Feedback Queue

MLQ - Multilevel Queue

2.6 Limitations of Existing Systems

- **a.** Most of the existing systems are platforms for journal access with high density web queuing problems
- **b.** **High Cost of Software licensing:** Some of the systems come as licensed software to be installed on individual computers thus making it very expensive to use in a large organisation.
- **c.** **Application domain Restrictions:** Some are specifically made for different areas of knowledge and only few are made for general purpose research.
- **d.** Some of them lack the research tracking capability which is very essential in research. There is no report on the progress of the research work.
- **e.** **Security Vulnerability Challenge and data integrity issues:** The present security systems are still vulnerable to some form of attacks. There is also the challenge of the delay introduced by security algorithms like RSA Cloud computing involves handing one's data over to a third party. This raises security concerns in the light of the recent prisms revelation where the government had access to users' data without authorisation from the owner. There is therefore need for an improved security system for one's data before committing it to cloud service providers.
- **f.** **Intellectual property agreements:** Most of them like i-Lab, Alfresco and research gate do not provide avenues for collaboration agreements for

easy intellectual property arrangements amongst the members of the research team.

g. **High Infrastructure Cost Index (High Capex):** There is also the issue of the cost of infrastructure for those of them that are not cloud-based. It is plausible to lower the capital expenditure (CAPex) with little increase in the Operational Expenditure (OPex) for profit and resource maximization. Besides, it is important that these systems contain effective file storage and organization systems with lower cost index for the servers and storage systems.

h. **Real time Data acquisition from remote research equipments:** Most of the existing systems like research gate, alfresco etc do not have capability for real-time data acquisition from equipments. The capabilities of transfer of multimedia files such as video, audio, research data and other http services are visibly lacking in some of the existing models.

i. **Reliability of information:** there is the challenge of ensuring that the information generated from the system among team members is reliable. In most of the system, the profiles of the collaborators or team members are not known to others. This is very necessary to ensure that the people involved in a team are people with the right professional qualification and research interests.

j. **High page and object response time and low server utilisation:** Most of the existing cloud computing environments like amazon, OAU cloud, open Nebula and lunacloud reviewed in this work are faced with the challenge of high page response time, low server utilisation and high object response time.

k. Loss of data in cloud environments using packet drop queuing model. This type of model uses token bucket for admission control and traffic shaping. When packets arrives and there are not enough tokens retained in the bucket, the packet can be buffered until new tokens are available or they can be marked and, hence, treated in particular way or they can be discarded depending on the configuration of the token bucket filter. This discarding or dropping of packets increases total delay in the system. There is no feed back to the system on the packets dropped; hence errors are introduced in data transmission. This affects the integrity of the system.

In summary, with abundant information available, the scientific community (researchers, industrialist, etc) seeks to have effective research collaboration via a virtual platform with robust stability and reliable information provisioning. Technology needs to discover the hidden value of data and provide accurate intelligence timely. Collaborative platforms should allow better collaboration and help save precious time that can be put to use in making scientific breakthroughs.

CHAPTER THREE

SYSTEM MODELING

3.1 Methodology

This section briefly explains the methodologies used in this dissertation while presenting the reasons behind the choice of these methodologies for different aspects of the work. Cloud research in general relies either on analytical models or on running actual experiments or simulations and building empirical traffic profiles and signatures (Bhadauria, and McKee, 2010 (Gmach, 2010). The lack of analytical solution for many systems has led to the development of approximations in the literature. These approximations can be very accurate, but, can also be inaccurate under certain conditions. Many approximations have been built upon the behaviour of queues in light or heavy traffic conditions which make them work well in and around the respective traffic intensities (Girish, and Hu, 1997). This dissertation uses modeling and simulations to infer measurements taken from an actual system. This research made use of a composite methodology which includes a modified form of waterfall methodology and discrete events systems methodology in the design and implementation of the platform. The modification is to overcome the weaknesses of waterfall methodology. Discretional mathematics will be used in the mathematical modeling and analysis of the queuing system. Discrete events modeling tools like Matlab Simevent and OPNET IT guru will be used for analysis and validation of network QoS parameters. The collaboration platform will be developed using Platform as a service model and will be deployed as a public cloud application

3.2 Security and Data Integrity System

The first tier security will be provided by the cloud service providers as stipulated in the service level agreement. This dissertation proposes a second tier security system applied to the data by application owners before they are committed to the cloud. This is to enhance system security. The system security will be designed and implemented based on a combination of advanced encryption standard and digital signature which is needed to ensure data integrity. This combination is termed Hybrid Encryption Algorithm (HEA). The AES will be used in designing the system security because of its speed to avoid introducing delays in the system and to ensure the confidentiality, integrity and security of data. It has longer key lengths and works better for large volume of data.

3.3 Modeling Environment and Software Tools Used

The following software applications were used in the work:

- **Matlab SimEvent –** this will be used for the modeling and simulation of the proposed queuing model for enhanced task scheduling in cloud data centres
- **Optimised Network Engineering Tool (OPNET)** – this will be used for the validation of the performance of the queue model proposed. It will also be used. It will also be used for the implementation of network traffic engineering taking advantage of the system's cloud computing architecture
- **SQL Server –** the database for the platform before cloud storage
- **Google App Engine** – the cloud service provider for hosting the platform

3.4 Proposed System's Model Specification

The proposed system will be addressing some of the challenges listed in the previous section in the following context:

- Modeling, Design and implementation of a cloud-based collaboration platform with real time data acquisition from research equipments
- Development of an enhanced queuing model for enhanced task scheduling

- The system will have research tracking capability in which reports on the progress of ongoing research will be given
- An enhanced Security algorithm to improve on second tier security. The proposed security system will be based on a combination of a modified form of advanced encryption standard (AES) cryptographic security scheme and digital signature called Hybrid Encryption Algorithm
- Block diagrams and flowcharts design for the proposed platform will be developed.
- Intellectual property agreement design for collaborators
- Team members registration and profile management to ensure that people of like interests connect in a team
- Simulations using OPnet for the performance evaluation of the proposed queuing model
- Cloud based implementation to reduce the infrastructural needs while providing high data storage capability which is very needful in research**.** This will reduce the overall running cost.

The block diagram overview of the system model design flow is shown in figure 3.1

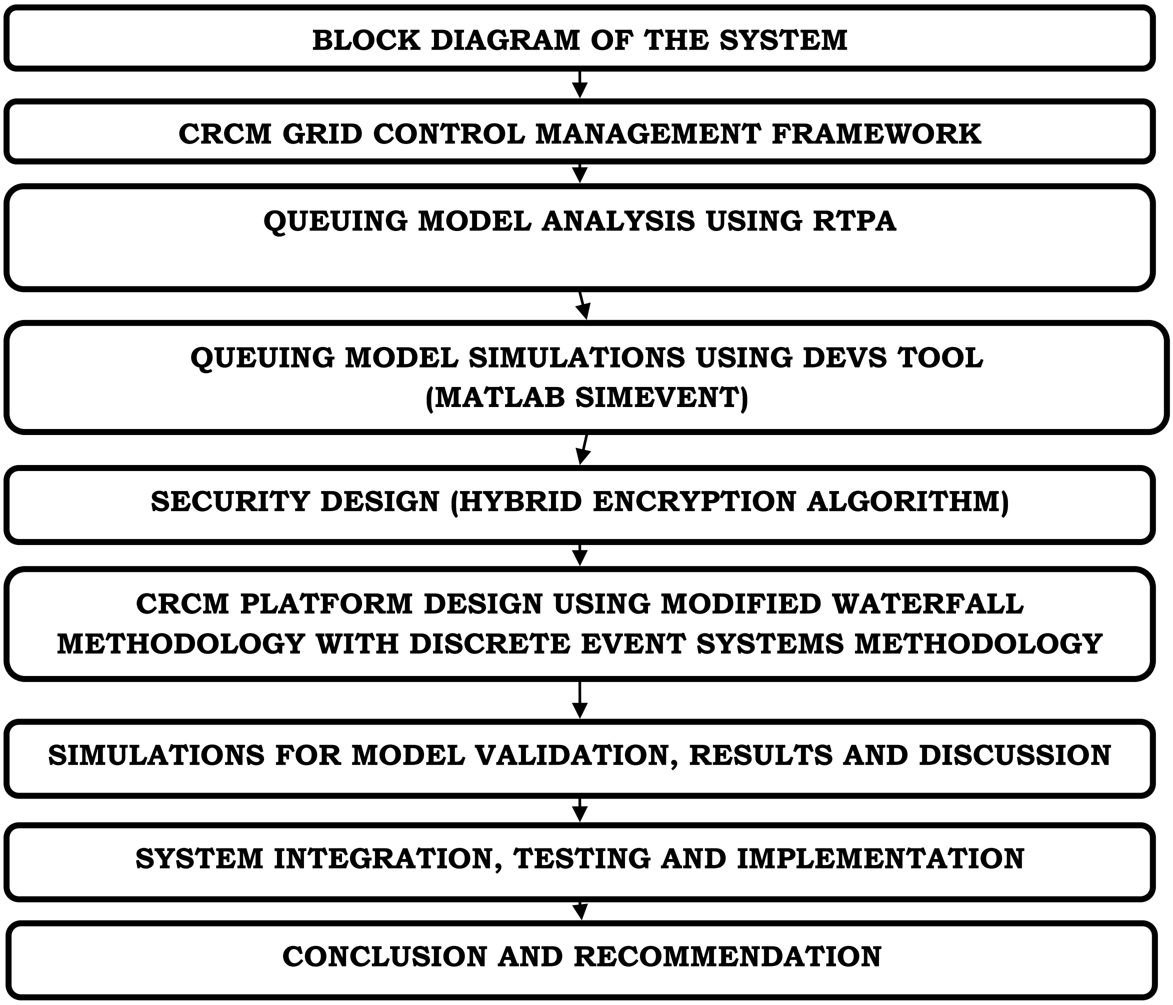


Figure 3.1: Block Diagram of the System Model Design flow

3.5 **Block diagram overview of the CRCM**

The CRCM platform system is made up of five major sub-systems carefully integrated into the internet cloud for remote access. They includes: content acquisition, information management, storage infrastructure, content delivery, and collaboration subsystems as shown in figure 3.3.

i. **Content Acquisition:** The content acquisition sub-system is responsible for collecting data and information used for research. The content may be in form of updates on ongoing research, published works, research data, knowledge etc. It can be collected manually or automatically from virtual equipment representing remote equipments or from the data acquisition system.

ii. **Information Management:** The information management sub-system is responsible for content search, data analysis and organisation.

iii. **Content Storage:** The content storage sub-system takes care of the storage of data and information generated in the cloud.

iv. **Content Delivery:** The content delivery sub-system is responsible for the display and exchange of the acquired information/data through graphical user interface, e-mail, chats, mobile PDAs, personal computers e,t,c.

v. **Collaboration Sub-system:** it consists of the research community communicating through chat rooms, instant messaging, discussion board, community platform and email, exchanging information and data.

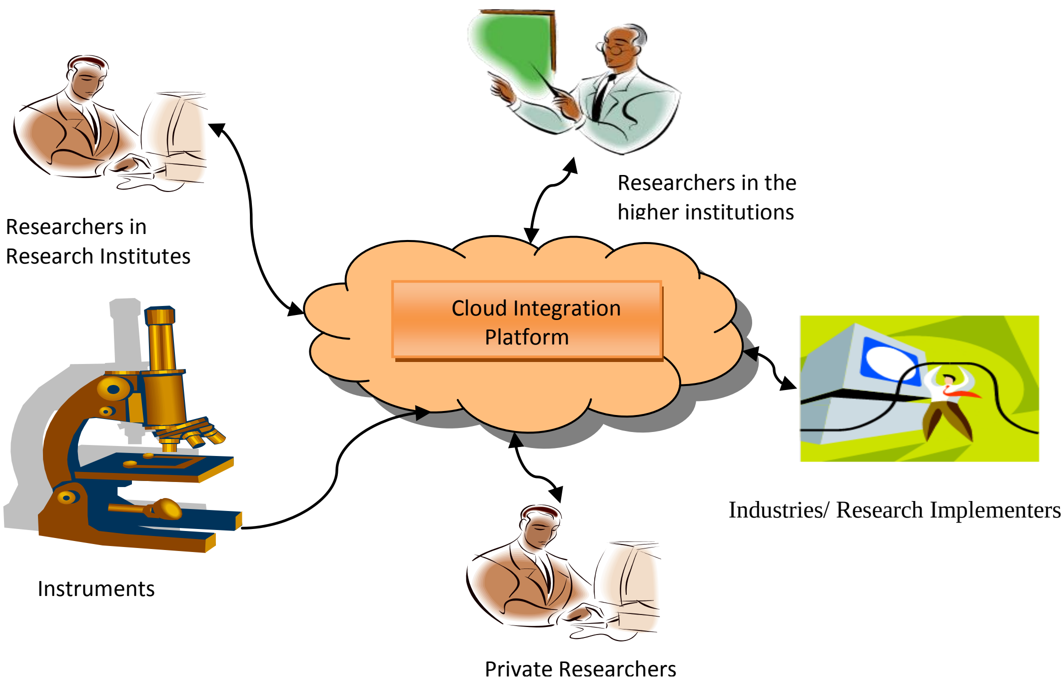


Figure 3.2: Interconnections among collaborators

Figure 3.2 shows different types of researchers in different geographic locations interconnected by the world wide web. They are able to interact with each other, share ideas and information not minding their location. The information exchange is on real time basis. This interaction can be amongst researchers in the higher institutions for an example or between researchers in the research institutes and those in the higher institutions or private researchers carrying out individual research, they can share ideas and tackle challenges facing their different research works more easily. The research implementers / industrialists can also interact with the researchers through the same platform. Real time data acquisition from expensive instruments owned by an institution lacking in another institution or placed in a remote location is made possible by connecting the equipment to the platform. The system is hosted on a cloud infrastructure thereby reducing cost of infrastructure and increasing its storage capacity.

The block diagram of the proposed system is shown in figure 3.3. It will not only solve the problem of research collaboration but will address the challenges of how to effectively allocate system resources to meet Service Level Objectives for users and cloud service providers. From this framework a scheme for autonomous performance control of the cloud collaboration platform application is proposed. It uses a queuing model predictor with an online adaptive modulation loop that enforces moderate queue and admission control of the incoming requests to ensure that the desired response time target is met. The proposed Queuing-Model-Based Adaptive Control approach combines both the modeling power of queuing theory and self-tuning power of adaptive control. Therefore, it can handle both modeling inaccuracies and load disturbances in a better way. To evaluate the proposed approach, the queuing model for the case study cloud application will be built and simulated with MATLAB Simevent components widely used in industry.

Figure 3.3 also incorporates the security system to be implemented in the CRCM. In the design proposal, such intelligent web integration with the CRCM platform demonstrates high efficiency and low cost deployment.

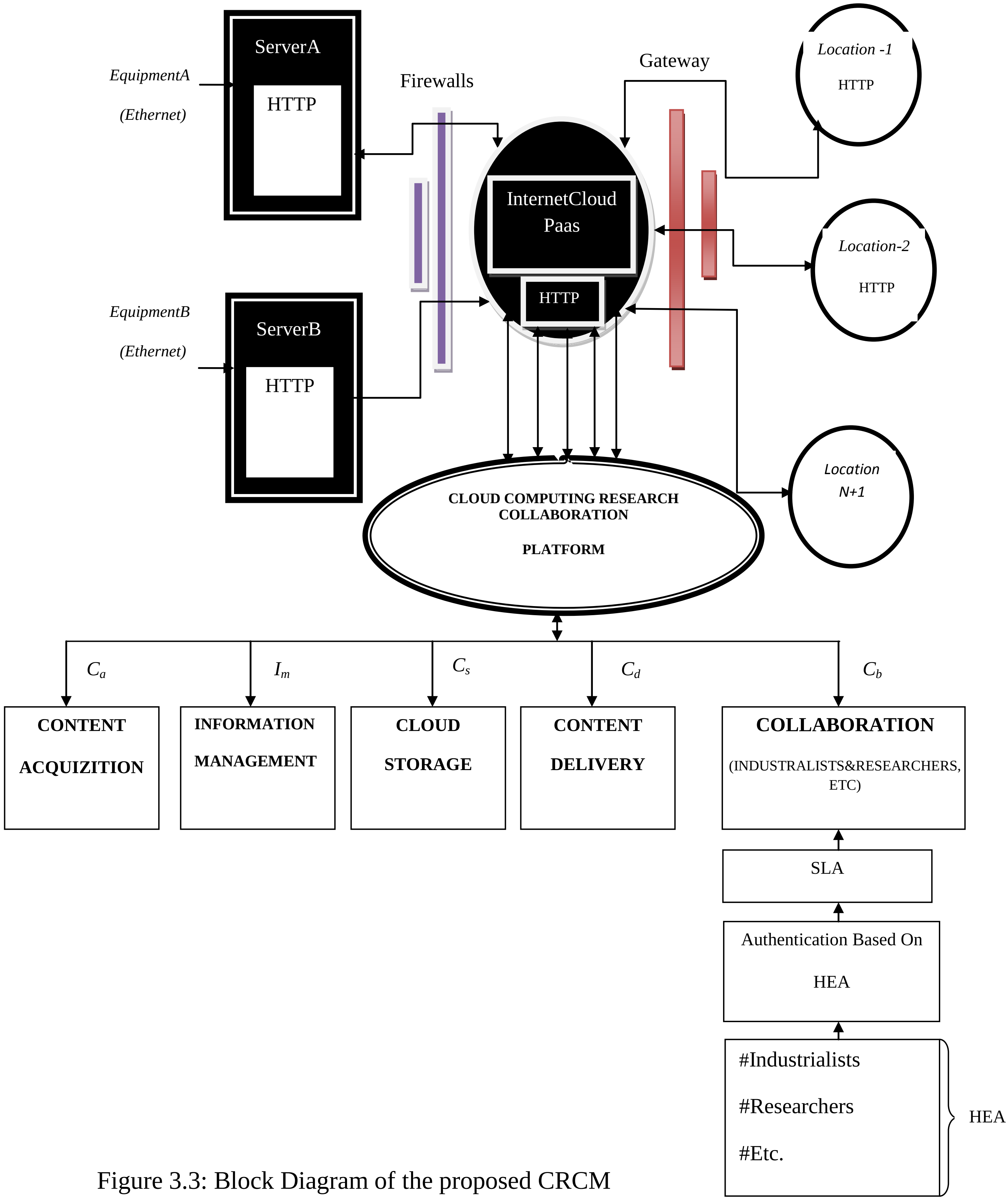


Figure 3.3: Block Diagram of the proposed CRCM

From the block diagram of figure 3.3, the layer-1 authentication utilizes the connection shown in figure 3.4 for security validation. The users are required to supply their login details which are encrypted with the cryptographic algorithm proposed in this work. Afterwards, the grid control or the logic instance then connects the user process to the server process having met all the layer security requirements. This work uses a modified form of advanced encryption algorithm to encrypt user login details from the server process. Firstly, the user process is initialized before the validation by the security system from the server process at the backend. After the validation, to access the collaboration module of the CRCM will require a basic Collaboration Level Agreement (CLA) endorsement. All the passkeys are securely stored in the server. The user's connection to the server is shown in figure 3.4.

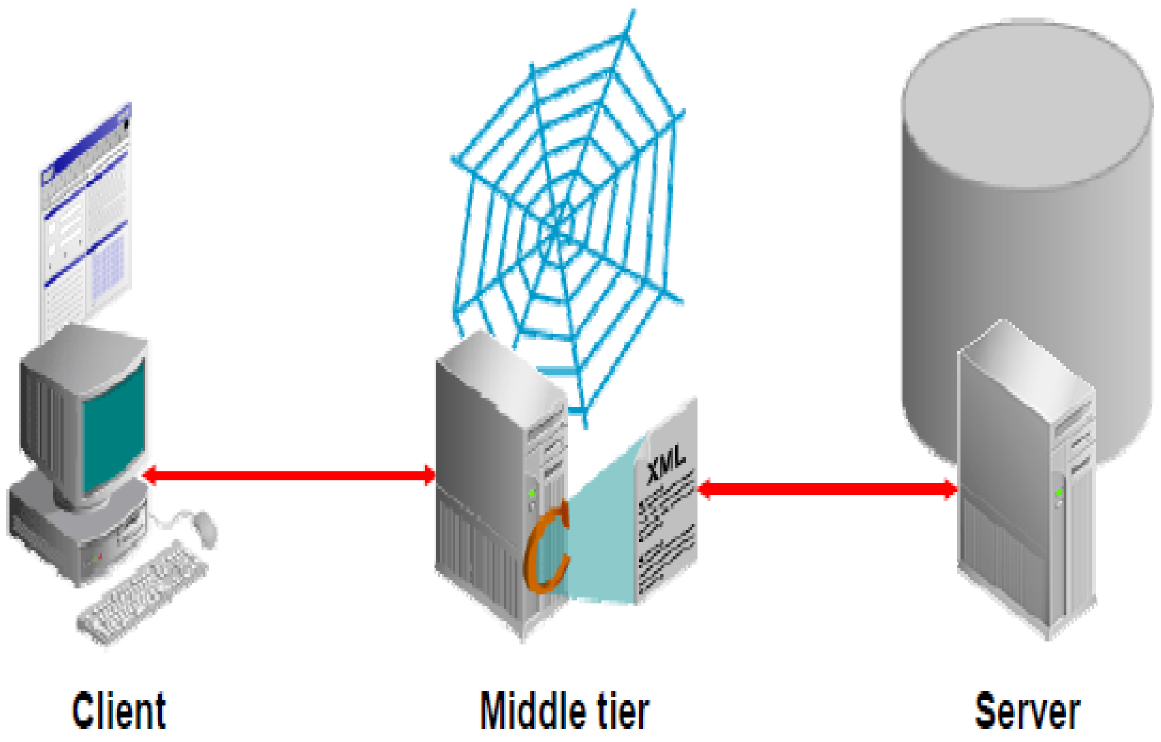


Figure 3.4: User Instance Server Connection

From figure 3.5, a logic instance is the connection bridge between the user process and the server process down to the backend server. The process architecture will be presented later in the work taking cognizance of figure 3.3 to figure 3.6.

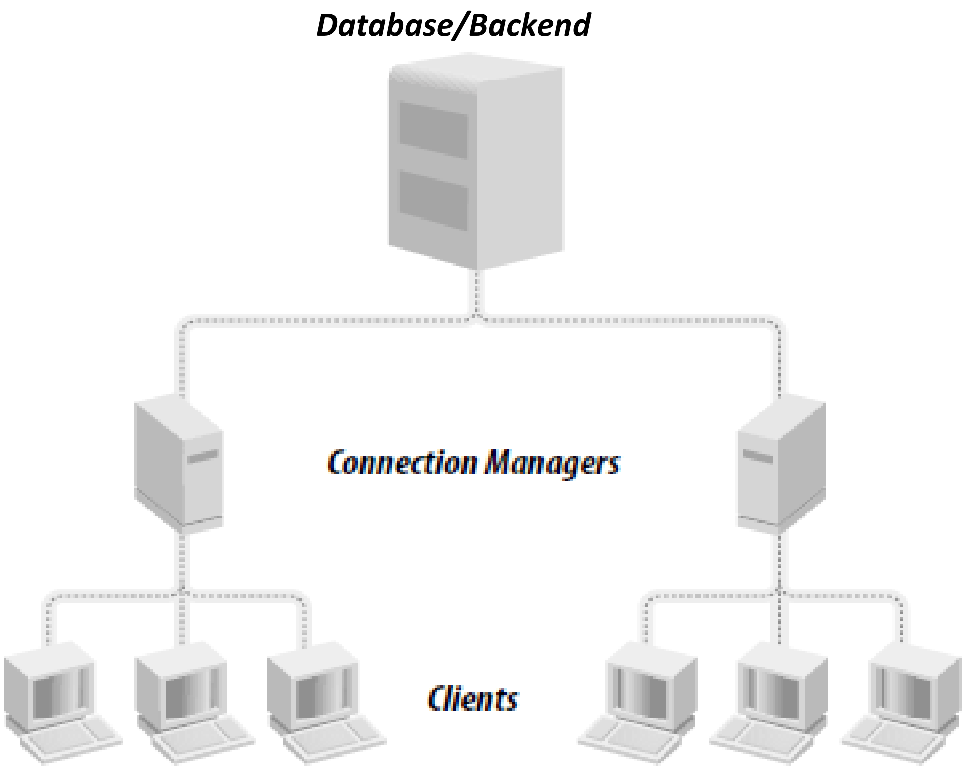


Figure 3.5: Logic Instance Connection Middleware (Connection Manager)

In our design phase, the connection manager is the link between the users and the backend server as depicted in figure 3.5. When a failure occurs in the connection manager, then the CRCM will experience a momentary outage until the background processes restores such failure. Looking at figure 3.6, the CRCM security block is designed to encapsulate the presentation layer which is the graphical user interface, the application layer which is the business logic or the logic instance and the exchange infrastructure layer which communicate with the server for service provisioning.

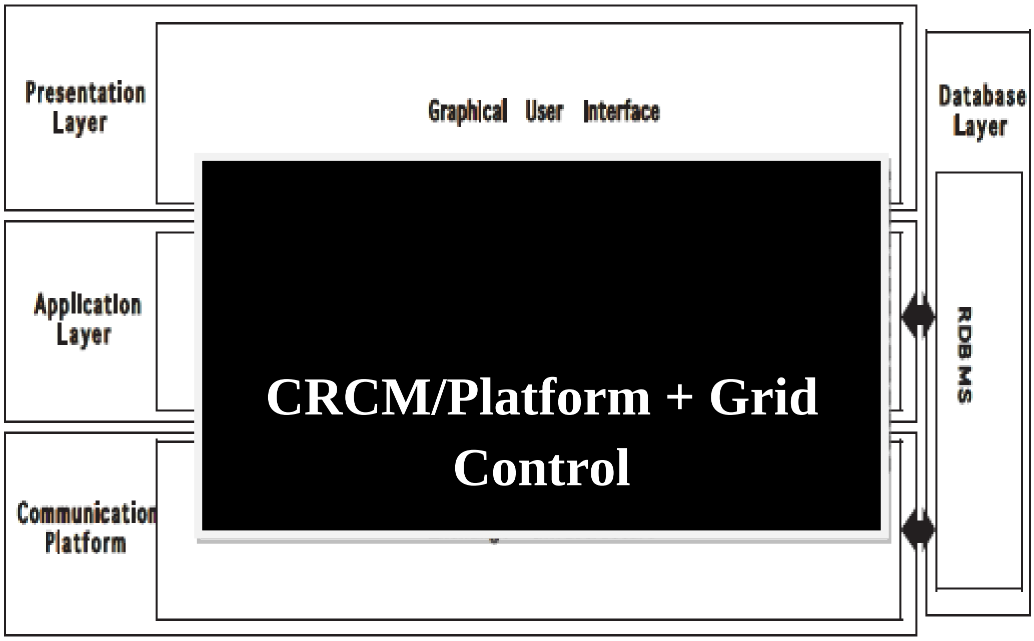


Figure 3.6: Block diagram of CRCM Interfaces with Grid Control

From figure 3.6, the main integration framework uses CRCM grid control to address the issues of software provisioning, maintainability, High availability, performance, Security, safe Information integration and application service monitoring. Performance refers to localising critical operations and minimizing communications while utilizing fine-grain components. Security refers to utilizing a layered architecture with HEA to restrict unauthorized access into the system. Safe Information integration refers to signing the Collaboration Level Agreement (CLA) before accessing critical features in the CRCM subsystem while availability refers to including redundant components and resource allocation mechanisms for fault tolerance and maintainability refers to the use of fine-grain, replaceable components.

3.6.1 Framework for CRCM System Queuing Model

Here, a very simple, high level view of the cloud queuing network framework is presented. The goal is to produce a generally applicable model that abstracts all hardware and software details, but is specific enough to produce significant performance results regarding the relationship between authenticated user processes from the collaborators, internet cloud, backend traffic and cloud server. In this work, this model took cognizance of the high-level details of the HTTP and TCP/IP protocols. The block diagram in figure 3.7 depicts an intelligent framework for CRCM job traffic. Figure 3.8 acts as a simple file server over the collaboration internet as shown.

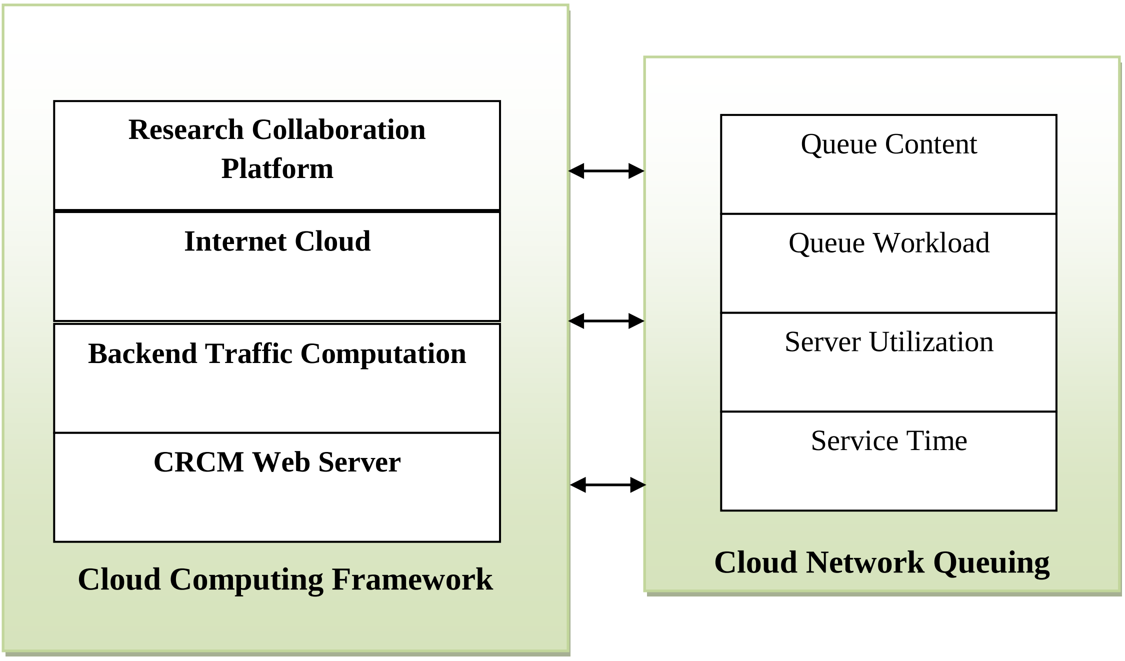


Figure 3.7: Block Diagram of Cloud Queuing Network Model for CRCM

3.6.2 The Proposed CRCM Queuing Design Formulations

Figure 3.7 shows the interaction between the Cloud computing framework and its network queuing metrics which drives figure 3.3. In the design, a web server W_s together with a browser program (i.e., client) constitutes a client/server system. W_s typically process many simultaneous jobs (i.e., file requests), each of which contends for various shared resources: processor time, file access, and bandwidth. Since only one job may use a resource at any time, all other jobs must wait in a queue for their turn at the resource.

As jobs receive service at the resource, they are removed from the queue; and as jobs arrive, they join the queue immediately. The CRCM Queuing model will compute the size of those queues and the time that jobs spend in them. In this context, the concern is with the number of simultaneous HTTP GET file requests handled by a server, and the total time required to service a request.

Now, the CRCM views every service or resource as an abstract system consisting of a single queue feeding one server for various users.

Let the CRCM queue made by the Industrialist be given by Q_{Ind} , that of the researchers by Q_{Res} . Associated with every queue are viz:

i. Total Arrival rate $T\lambda_{r_q}$ (A) - The average rate at which new jobs arrive at the queue.

ii. Service time (T_S) of the server - The average amount of time that it takes a server to process such jobs. $T_S = 1/\mu$

iii. Queuing time (T_Q) - The average amount of time a job spends in the queue.

The average response time (T) is given by equation (3.1)

$$T = \sum (T_S + T_Q) \qquad (3.1)$$

If the total arrival rate $T\lambda_{r_q}$ is less than the service rate ($1/T_S$), then the CRCM queuing system is stable as such all jobs will eventually be serviced, and the average queue size is bounded.

On the other hand, if $T\lambda_{r_q} > (1/T_S)$, then the system is unstable and the queue will grow without bound.

iv. Server utilization U - This is the product of the arrival rate (λ_{r_q}) and service time (T_S)

Hence, $$f(U) = \lambda_{r_q} * T_S \qquad (3.2)$$

This is a dimensionless number between *0* and *1* for all stable systems. *A* utilization of *0* denotes an idle server, while a utilization of *1* denotes a server that is being used at maximum capacity.

At a utilization of *0*, the response time is just the service time; no job has to wait in a queue. As utilization increases, the response time of the queue grows gradually. Only when the utilization approaches 1 does the response time climb sharply toward infinity. The web server queuing model satisfies this behaviour.

Assuming the amount of time between job arrivals (1/A) is random and unpredictable, then the arrivals exhibit an exponential or memoryless distribution. Also, the service rate must be greater than the arrival rate, that is, $1/T_S = \mu > A$. If $\mu \leq A$, then the waiting line (FIFO queue) would eventually grow infinitely large.

On these bases, we shall now characterize the mathematical representation of the CRCM queuing model.

3.6.3 Cloud Queuing Model Design Analysis

Queue Analysis is necessary to provide the service time for each requests to make the architecture free from overflow of data. System requests arrive and depart randomly. This is one of the major challenges in designing a task scheduling algorithm. It is also important to note that the requests submitted to the web server and the responses which are generated either from the application server or database server through application server are termed general distribution because they do not pursue any distribution. The proposed task Scheduling Algorithm is an enhanced GI/G/1 queuing model where GI symbolizes general independent arrivals and G represents general (any distribution) service. Figure 3.8 shows the architecture of the cloud queuing model proposed in this work comprising of the internet and the web server network models. The service efficiency of figure 3.8 was used to develop figure 3.2 and figure 3.3.

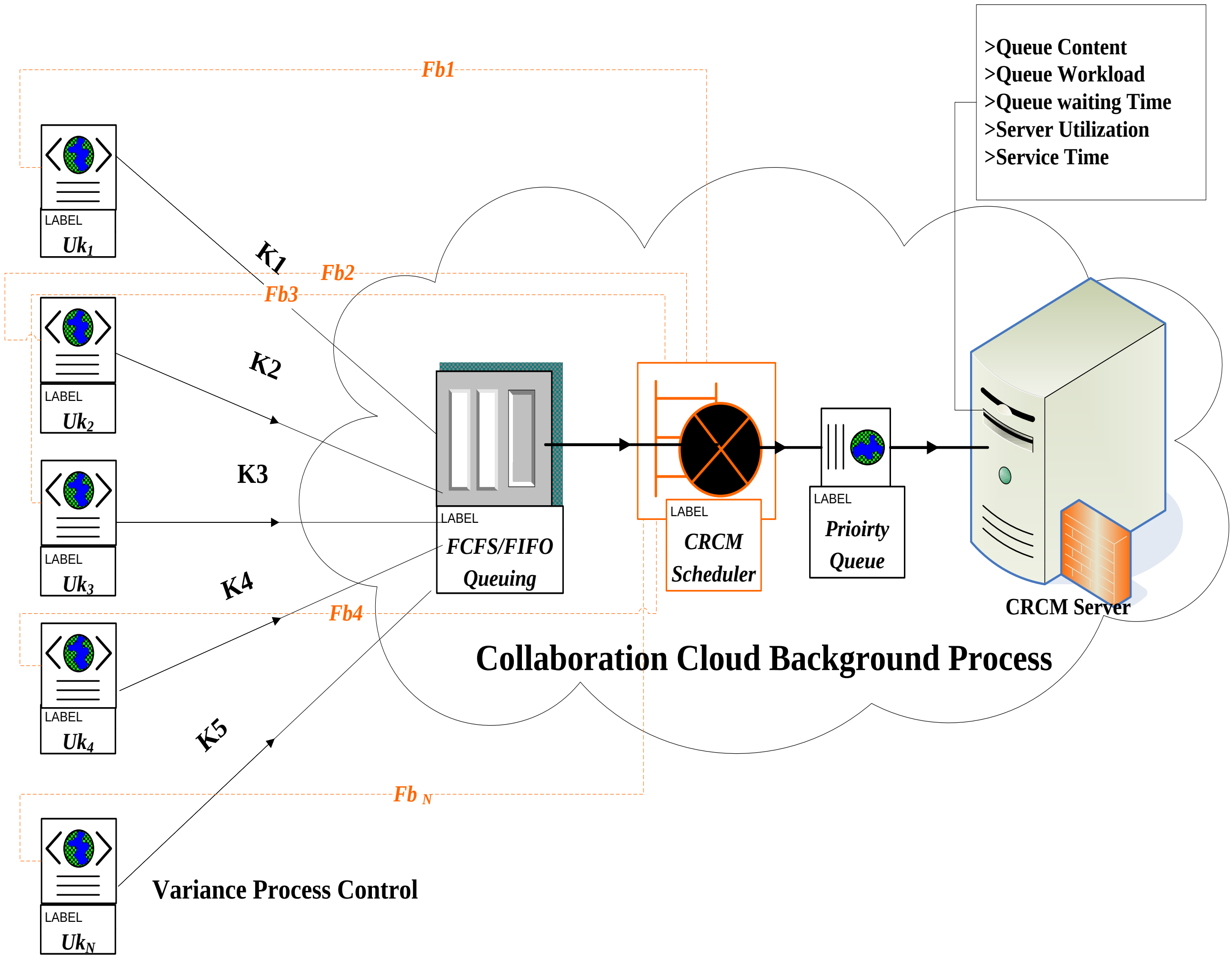


Figure 3.8: Architecture of the proposed Cloud Queuing model for CRCM

In the model, the user task arrival t_a represents the users U_i on the CRCM platform equipped with rate regulator $R_{r.}$ for global queuing which is based on first come first serve.

Service availability and response time form the quality of service (QoS) measures in CRCM, but the dynamic nature of CRCM, diversity of user request (r_{qi}) and QoS compelled the outlining of the main specifications of the queuing analytical model below.

- A binominal Poisson arrival of user request r_{qi} was considered
- Queuing service availability characterization at the CRCM DCN

- Global FCFS (queue F_q) with a task scheduler C_s
- Priority Queue P_q for server provisioning and possible pool management.
- Feedback congestion regulation with respect to user request tasks.

The queuing model as shown in figure 3.8 was considered with the following:

i. There is an infinite wait state at F_q, C_s, P_q

ii. All arrivals are independent of each other

iii. The Poisson process r_{qi} (as users make use of the CRCM platform) is maintained using FCFS probability matrix behaviour.

Recall that CRCM queue model results when the queuing jobs exceeds the service rate capability or capacity of the CRCM server.

Let the system resource in the CRCM server be given by S_t be subdivided into shared compute resources S_{cr} and hibernated or reserved resource h_{cr}.The queue jobs in the model makes use of these system resources.

Hence, $S_t = S_{cr} + h_{cr}$ (3.3)

Where S_{cr} is used to serve the user request jobs r_{qi} of clients such that job request is given by equation 3.4

$$J_r = \sum_{i=0}^{n} r_{q_{i+1}} \qquad (3.4)$$

h_{cr} is used to serve the user request jobs r_{qi} of user when there are so many jobs to be processed. The total shared compute resource is given by equation 3.5

$$TS_{cr} = \sum_{i=0}^{N} S_{cri} \qquad (3.5)$$

While the total hibernated or reserved compute resource is given by equation 3.6

$$Thcr = \sum_{i=0}^{N} h_{cri} \qquad (3.6)$$

S_{cr}, h_{cr} > 0 while $h_{cr} \geq 0$

Equation 3.3 allows prioritized requests originating from different users. Users on CRCM with high priority requests (i.e. users with $\sum_{i=0}^{N} h_{cri} > 0$) are first served provided that $S_t = \sum_{i=0}^{i-1} S_{cri}$ are being processed at the moment.

Other requests from users for which $\sum_{i=0}^{N} h_{cri} = 0$ are served whenever less than S_{cr} resources are used.

From figure 3.8, since, the number of potential users UK_{n+1} that independently generate request r_{qi} are large, we then assumed that the requests arrive the global queue according to a Poisson process.

The rate at which new request jobs from users arrive is denoted by $\lambda_{r_{qi}}$ as previously stated.

Recall that from the established basis previously, the total arrival rate $T\lambda_{r_q}$ is given by $\sum_{i=0}^{N} \lambda_{r_{qi}}$ and the total service rate is given by $T\mu = \sum_{i=0}^{N} \mu_i$. The stability criterion is obtained when $T\lambda_{r_q} / T\mu < 1$.

From figure 3.8, we seek to establish CRCM possible variables ${}^{Q}K_1$, ${}^{W}K_2$, ${}^{U}K_3$, ${}^{D}K_4$, ${}^{Q}K_5$, and τ_{K_6} for describing the behaviour of the CRCM system.

Now, let ${}^{Q}K_1$ = Variable representing the total number of users in the queuing system (ie. waiting in the queues and being served).

${}^{W}K_2$ = Variable representing the total number of users waiting in the queue (ie. excluding the user jobs already served).

${}^{U}K_3$ = Variable representing the total number of users only being served.

${}^{D}K_4$ = Variable representing the total delay in the queuing system (ie. this includes user wait states/time in queues and server).

${}^{Q}K_5$ = Variable representing the time a user waits in the queue (ie. this excludes times spent in service).

τ_{K_6} = Variable representing the service time.

Hence, the possible total number of users in the queue is given by equation 3.7

$$T\,^{Q}K_1 = T[^{W}K_2] + T[^{U}K_3] \quad (3.7)$$

Again, the total delay in the queuing system is given by equation 3.8

$$T\,^{D}K_4 = T[^{Q}K_5] + T[\tau_{K_6}] \quad (3.8)$$

But $T[\tau_{K_6}]$ has been previously defined as $\frac{1}{\mu}$ i.e. 1/service rate.

The utilization rate of the server S_i, U_1 is given by equation (3.9) as

$$U_1 = T\lambda_{r_q}/T\mu \quad (3.9)$$

From figure 3.8, the major discrete delays in which user requests are subjected to include: the total Queuing delays $T\,^{D}K_4$ in the global queue (FCFS/FIFO), look-up delay L_d at the scheduler (Resource Assignment Module, RAM) and the server delay, S_d.

Hence, the response time, R_t is given by equation 3.10

$$R_t = \sum_{i=0}^{n} \left(T^{D}K_{4i} + L_{di} + S_{di} \right) \quad (3.10)$$

The CRCM server is provisioned to service the job tasks of the user arrivals. In the model shown in figure 3.8, the number of user job tasks cannot exceed the server processing capability. The server must service all job tasks scheduled at the same time. User requests (tasks) UK_{n+1} are submitted to the global finite queue and then processed on the FIFO basis.

The CRCM scheduler processes the tasks from the global queue. In the proposed system user requests may be dropped when the FCFS global queue is exhausted. There is therefore need for an enhanced admission control mechanism that will take care of the packet loss challenge. This work then introduces the scheduler feedback, F_b which works with Carrier Sense Multiple

Access feedback scheme to reduce the rate of incoming user and the priority queue then schedules high priority jobs to the server. When the server finishes the job, it becomes available for next job servicing cycle.

The effect of the model figure in figure 3.8 is that resource management in the context of cloud queuing management will be maximized. This is because the system can change its resource requirement and automatically adjust to changing workload conditions. This model captures a cloud application with optional computation that can be activated or de-activated at run time automatically.

With the use of control theory, specific guarantees like user-perceived latency can be provided for. With the variance process control and background processes, efficiency in service time, server utilization, queuing response times, queuing workload and contents will be captured for CRCM system model.

An interesting aspect of the CRCM queuing system that is different from the other models is the feedback system integrated as a rate controller on any user on the cloud. The mathematical characterization of the model is discussed in the next section.

3.6.4 Analytical Model for Scheduler Traffic Control

Recall that figure 3.3 depicts the block diagram of the proposed system. The proposed queuing model for the CRCM platform whose physical architecture was shown in figure 3.8 introduced a traffic control issue which is addressed in this section (see figure 3.9). In this respect, the analytical description for cloud scheduler traffic control is presented. The scheduler component of the architecture identifies the following major components: the users, the global queue, the priority queue and the CRCM server as shown in figure 3.8.

For the purpose of this analytical model, at the core of the scheduler, let the feedback carrier sense multiple access be given as Fb_CSMA. The analytical model is shown in figure 3.9.

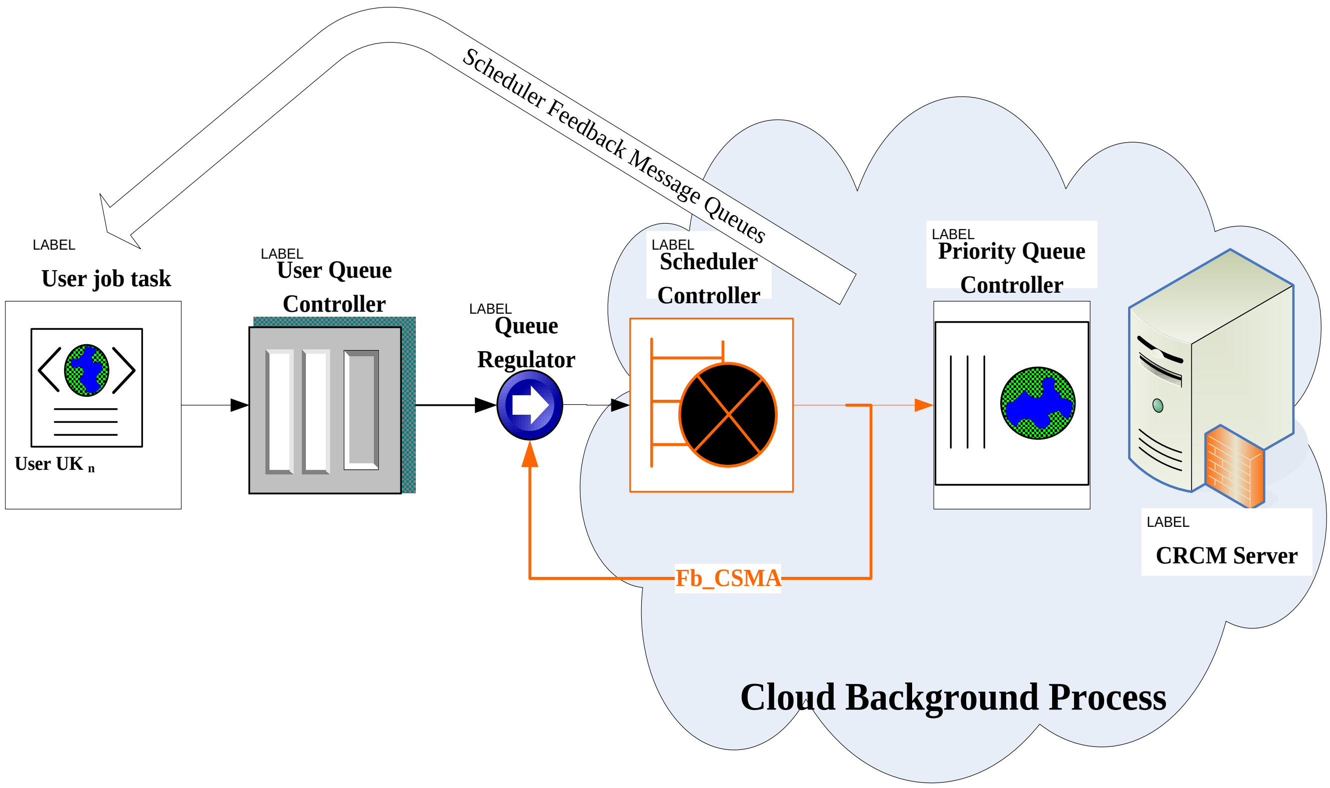


Figure 3.9: Scheduler feedback Control Model

From figure 3.9, scheduler feedback Fb_CSMA is modelled as a rate-based closed-loop feedback control in the cloud. We assume that the user devices are equipped with queue control rate regulator (RR) which modulates traffic queues. Traffic control at the scheduler monitors the length of output queues and makes traffic queue control decision. If traffic poll at scheduler is high leading to queue congestion, the scheduler advertises signals to sources using Fb_CSMA messages. The Fb_CSMA messages contain details for the sources to adjust their flow rates. The sources reset to the received Fb_CSMA and updates the rate of its regulator.

Other assumptions that were made in the scheduler analytical model design include:

- The Fb_CSMA messages are queue jobs / tasks or advertised signals sent through scheduler in the cloud background.
- The messages are tagged with identities (ids) to ensure coexistence and proper identification on the cloud background.
- Owing to the short diameter of the Fb_CSMA, the propagation delay/latency is very small.
- Links are assumed to be 40Gbps in capacity and the scheduler is a combined input/output queued switch having buffers of First-In First-Out (FIFO) output queue.
- Let equilibrium or stability level be Q_{eq} set at the scheduler. This is the optimized queue packet size that should be in queue.
- Let $q_{off\ (t)}$ indicates the instantaneous web load while $q_{delta(t)}$ indicates the rate of change of the web load. The weighted sum is an approximate prediction of the future web load queue.

A threshold is set to indicate tolerable queue congestion levels on the cloud link given by Qsc_{max}. The scheduler basically counts the number of arrivals $T\lambda_{r_q}$ and departure packets (d) and samples the incoming packets with a steady state probability Ps. When the packet is sampled, the scheduler determines the congestion level on the cloud link and may send Fb_CSMA message to the source terminal of the sampled job queue.

If the traffic queue is extreme, the scheduler activates the feedback message. The key details of the Fb_CSMA messages are the source Ethernet type and link queue measure l_q as well as destination point *id* and capacity of the link C.

Now the Ethernet-type informs the scheduler and sources about the Fb_CSMA. The *id* is the id for link queue (MAC address of scheduler interface). l_q is the buffer link queue measure feedback to the source.

From figure 3.9, the key measure of queue presence on a link is l_q. This consists of a weighted sum of the instantaneous queue offset and queue variation over the last sampling interval. This is given by the equation 3.11;

$$l_q = q_{off}(t) - Wq_{delta} \tag{3.11}$$

Where Wq_{delta} = queue weight given by $W(q_a - q_d)$, and instantaneous queue offset given by $q_{off}(t) = q(t) - Q_{eq}$.

Hence, $l_q = (q(t) - Q_{eq}) - W(q_a - q_d)$ (3.12)

q_{delta} is the queue variation over the last sampling interval and is defined as the difference in the number of packets that arrived q_a and the number of packets that were served q_d since the last sampling event. $q(t)$ is the total size of incoming traffic or queue length.

From figure 3.9, the algorithm showing traffic control equilibrium level (Q_{eq}) possible event in scheduler is presented in Algorithm 1.

The first line in the algorithm 1 is the case where the queue length is small and the sources can increase their rates with no effect on the scheduler. The second case is where even though the queue length is slightly greater than the equilibrium threshold; there will be an increasing traffic queue build up. In this case, the sources are signalled to decrease their sending rates via the feedback. In the third case, the large queue indicates that the link is heavily queued, and the sources are signalled to decrease their sending rates.

In general, whenever the queue is large, a feedback message from Fb_CSMA is used to decrease the queuing flows and vice versa.

From the model of figure 3.9, the source adjusts their rates using the enhanced Fb_CSMA which has been proven to be sufficient, efficient and shows fairness under user workload or task scenario. Hence, scheduler queue traffic control with feedback is presented below. In this case, the user on the CRCM interface cards must have a rate regulator.

Algorithm 1: Scheduler queue traffic control for equilibrium Level *(Q_{eq})*.

*Scheduler /*Traffic Queue decision at Q_{eq}*/*

Begin( )

If *$(q(t) < Q_{eq})$ and $(q_a = q_d)$ and $(l_q > 0)$* **Then**

return no_Scheduler (Fb_CSMA);

Endif;

If *$(q(t) < Qeq)$* and *$(q_a > q_d)$, and $(l_q < 0)$* **Then**

return adj_scheduler (Fb_CSMA);

Endif;

If *$(q(t) > Qeq)$ and $(q_a = q_d)$, and $(l_q < 0)$* **Then**

return adj_scheduler (Fb_CSMA);

Endif;

If *$(q(t) > Qeq)$ and $(q_a < q_d)$, and $(l_q > 0)$* **Then**

return adj_scheduler (Fb_CSMA);

Endif;

This work therefore proposes a queuing-based model with scheduler feedback for improved task scheduling in cloud environments. The queuing theory is applied with operational modifications in G1/G/1 queuing model. The flowchart of the proposed CRCM queuing system is shown in figure 3.10.

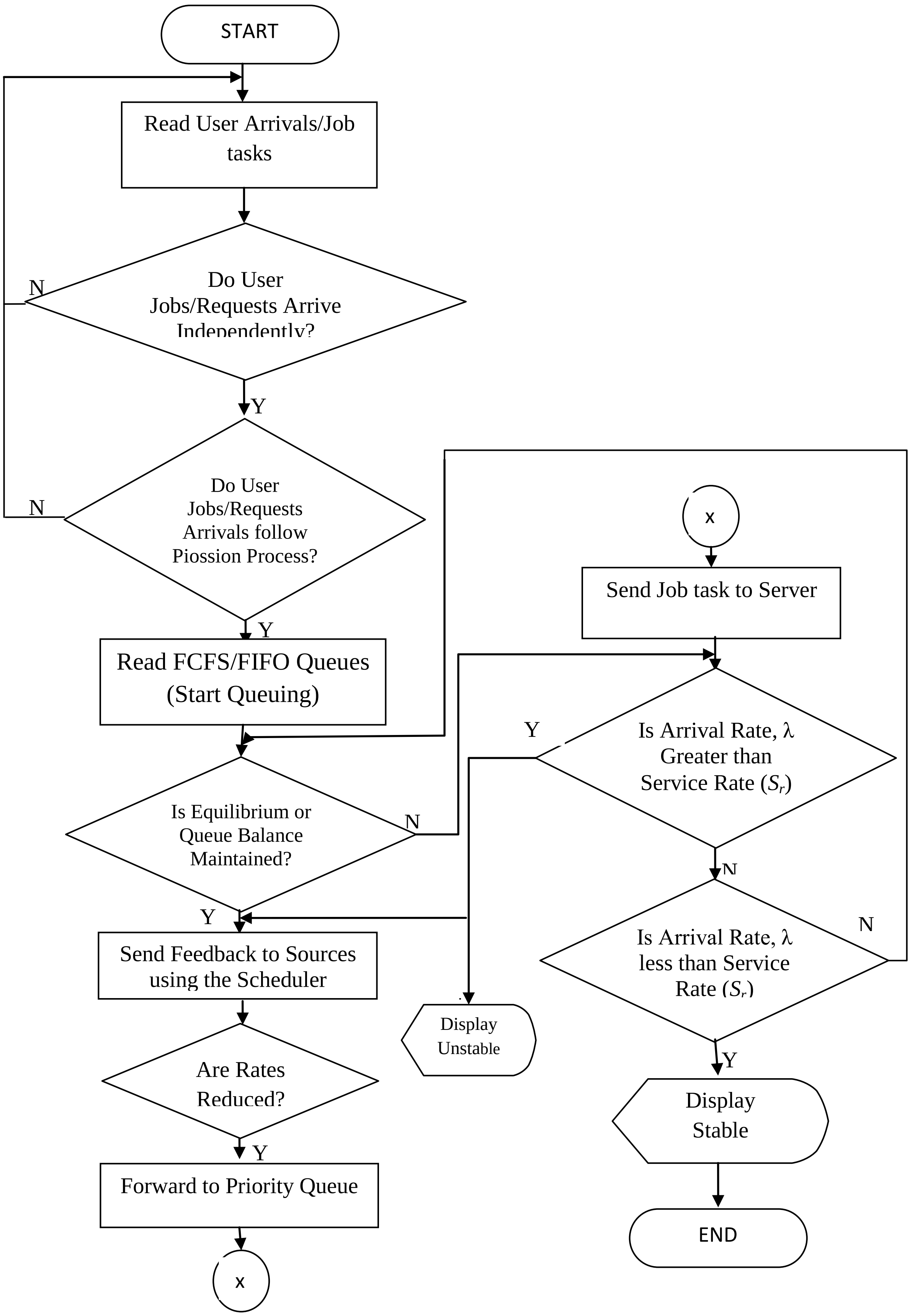


Figure 3.10: Flowchart for the Proposed CRCM Queue Algorithm

In the model, the arrivals are checked to ensure independent arrivals in Poisson fashion. When the queue arrivals are satisfied, they are read by global FCFS with their equilibrium criterion. The scheduler initiates a feedback using Algorithm1. For large queue scenario, the rates of the sources are conditioned and are forwarded to the server for stability verification as discussed previously.

The proposed CRCM task allocation queuing model seeks to dynamically create an improved task scheduling method to enhance server performance. There is no VM (Virtual Machine) live migration involved in this model which makes it much simpler than some existing models.

Recall in figure 3.8, that the model was assumed to have N+1 user nodes in the cloud communication network with a robust web server entity with its QoS profiles. The cloud queue model also has the variant process controls as well as its background processes as depicted in figure 3.11

Let all the file requests or jobs by users Ukn_{+1} (Industrialists and Researchers) arrive at the web server with an arrival rate of r_{qi} and let all the instance initialization processing be performed by the server node $S_{1.}$ The job then proceeds to the global node where a single buffer with data is read from the file, processed and passed into the internet. This block of data (FIFO) is then transmitted to the internet at the server's transfer rate. This data travels via internet and is received by the client browser respectively. If the job request was not completely transmitted, the job returns back to user node UK_{n+1} for further processing, else, the queuing job is complete. Self-adaptation is a first class concern for cloud applications, which should be able to withstand diverse runtime changes (Klein et al, 2014).

Variations are simultaneously happening at the cloud infrastructure level —for example hardware failures — and at the user workload level —flash crowds. However, robustly withstanding extreme variability requires costly hardware over-provisioning. In the cloud queuing formulation, the feedback branching is probabilistic in nature.

Considering figure 3.3, and figure 3.8, the following assumptions are made in the model environment as shown in figure 3.11.

i. The users (Industrialists and Researchers) come from a population that can be considered infinite.

ii. Users (Industrialists and Researchers) arrivals are described by a Poisson distribution with a mean arrival rate . This means that the time between successive user arrivals follows an exponential distribution

iii. The users' service rate is described by a Poisson distribution with a mean service rate of $1/T_S$ or μ. This means that the service time for one customer follows an exponential distribution with an average of $1/\mu$.

iv. The waiting line priority rule in the cloud queue CRCM used is a combination of first-come, first-served and priority.

v. The size of requested files (service times) are exponentially distributed.

vi. The effects of the HTTP GET requests are negligible since requests are moderated for the file processing.

vii. Also, it is assumed that the average queue length = (Mean arrival rate)*(Average waiting time in queue)

This work then presents the formulation discussions for figure 3.8 which will be realized using MATLAB Simevent tool in this section showing the cloud queue metrics for CRCM because it has in built capability for queuing analysis.

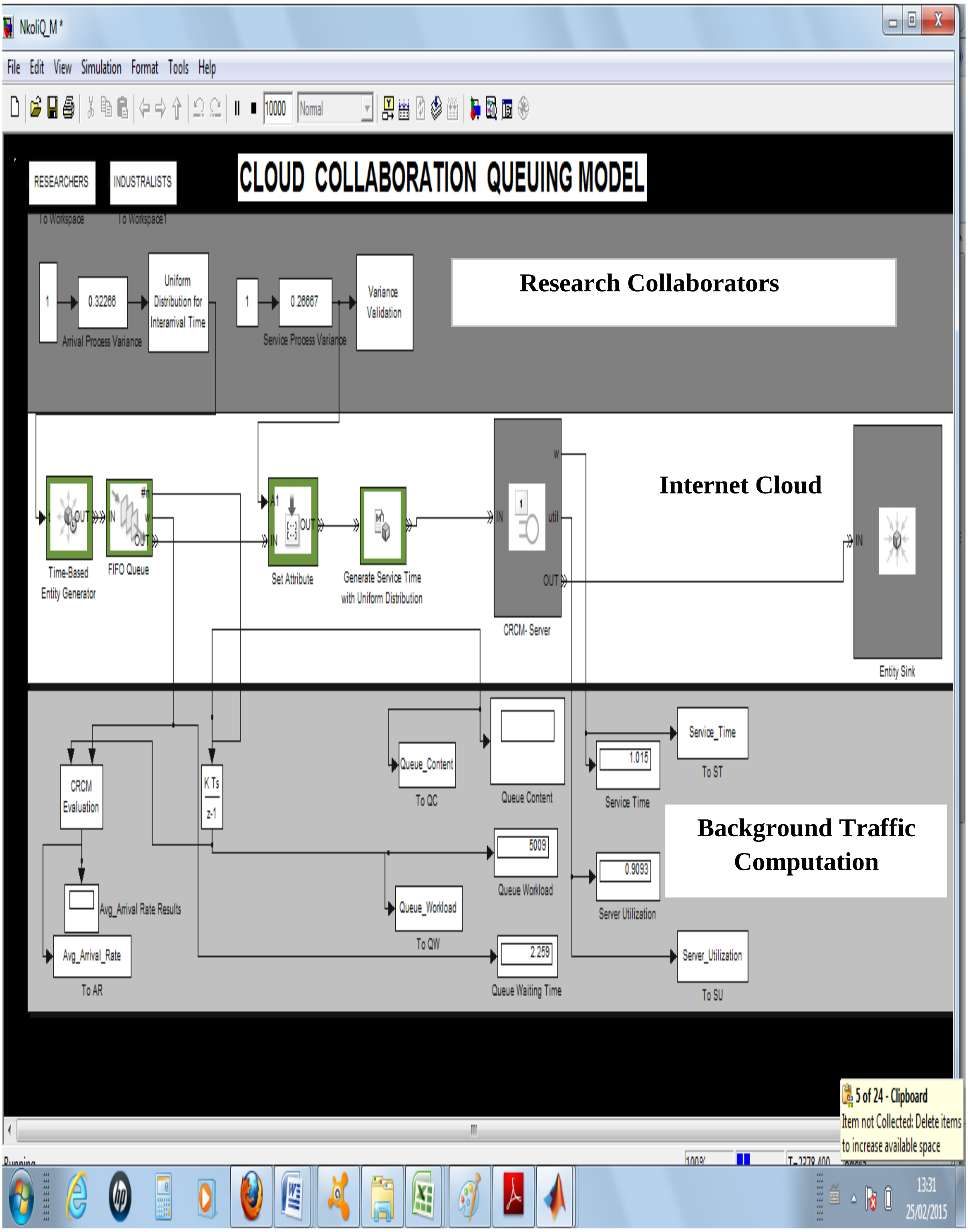


Figure 3.11: Simevent model of Cloud queuing model for CRCM

3.6.5 Description of the Main Blocks of the Cloud Queuing Model

Figure 3.11, shows the process model of the proposed model showing the Researchers/Industrialist background, the Internet background and the traffic background as well.

The model consists of the following main blocks which were configured :

- **Arrival Process Variance:** This captures the arrivals, A comprising of a low and high limits for A. This work used the high margin for A as 0.40333 based on heuristics (Girish and Jian-Qiang, 1997) which is for effective metric studies.
- **Uniform Distribution for Inter-arrival Times:** This output the inter-arrival times according to a uniform distribution with an optimal seed value of 85274. This enables a time-based entity generator for FIFO queue. It creates a signal representing the inter-arrival times for the generated entities. After setting the distribution's variance using the Arrival Process Variance block, the subsystems compute a uniform random variant with the chosen variance and mean.
- **Service Process Variance:** It facilitates the service rate computation by the server comprising of a low and a high limits for TS and was set at 0.3 also based on heuristic. This was configured to drive the set attribute entity for the incoming FIFO queues.
- **FIFO Queue:** This is based on the blocks available in SimEvent/ Simulink library. The buffer queue is controlled by incoming signals from the uniform distribution for inter arrival times which enables the time based entity generator. This activates the set attribute entity which calls a function;

  function out_ServiceTime = GenServiceTime (Variance)

  which generates uniform service time for the server. The FIFO Queue block is initiated by the arrival of a packet generated by the output of a time based entity generator that synchronizes the arrival of the packet (from users) with the timing signals of the entity generator block. It basically stores entities yet to be served by the server.
- **CRCM Evaluation Block:** Sequential simultaneous arrivals are captured by the CRCM evaluation block which transposes the average arrival

results. Here also, the queuing content, workload and waiting time are aggregated.

- **CRCM Server:** These models a server whose service time has a uniform distribution. The operation of this unit is very vital in the model because it processes the service time and the server utilization while outputting the results accordingly in the model. The server block computes the server utilization and average waiting time in the server.
- **Entity Sink:** This accepts or blocks entities and it is found in the SimEvent Library. The block provides a way to terminate an entity path. Selecting Input port available for entity arrivals, the block always accepts entity arrivals. Neglecting to select input port available for entity arrivals, the block never accepts entity arrivals. The model termination is achieved by this unit.

After these configurations, the graphical plots of the simulation results from figure 3.11 were obtained via the Simevent command worksheet and discussed, showing the effectiveness of figure 3.8 and figure 3.3. The corresponding MATLAB scripts are presented in the Appendix section of this work.

From figure 3.11, the system response was ascertained by moving the arrival process variance slider or the Service Process Variance slider during the simulation while observing how the queue content changes. When traffic intensity is high, the average waiting time in the queue is approximately linear in the variances of the inter-arrival time and service time. The larger the variances are, the longer an entity has to wait, and the more entities are waiting in the system. The plots of figure 3.12 to figure 3.16 shows the generated results of figure 3.11.

Table 3.2: Cloud Queuing Arrival Rate Response

Arrival Rate	Arrival Job queues
0	0
0.8526	500
0.9164	1000
0.8904	1500
0.8724	2000
0.8771	3000
0.8902	4000
0.8803	5000
0.8715	10000
0.8668	15000
0.8755	20000

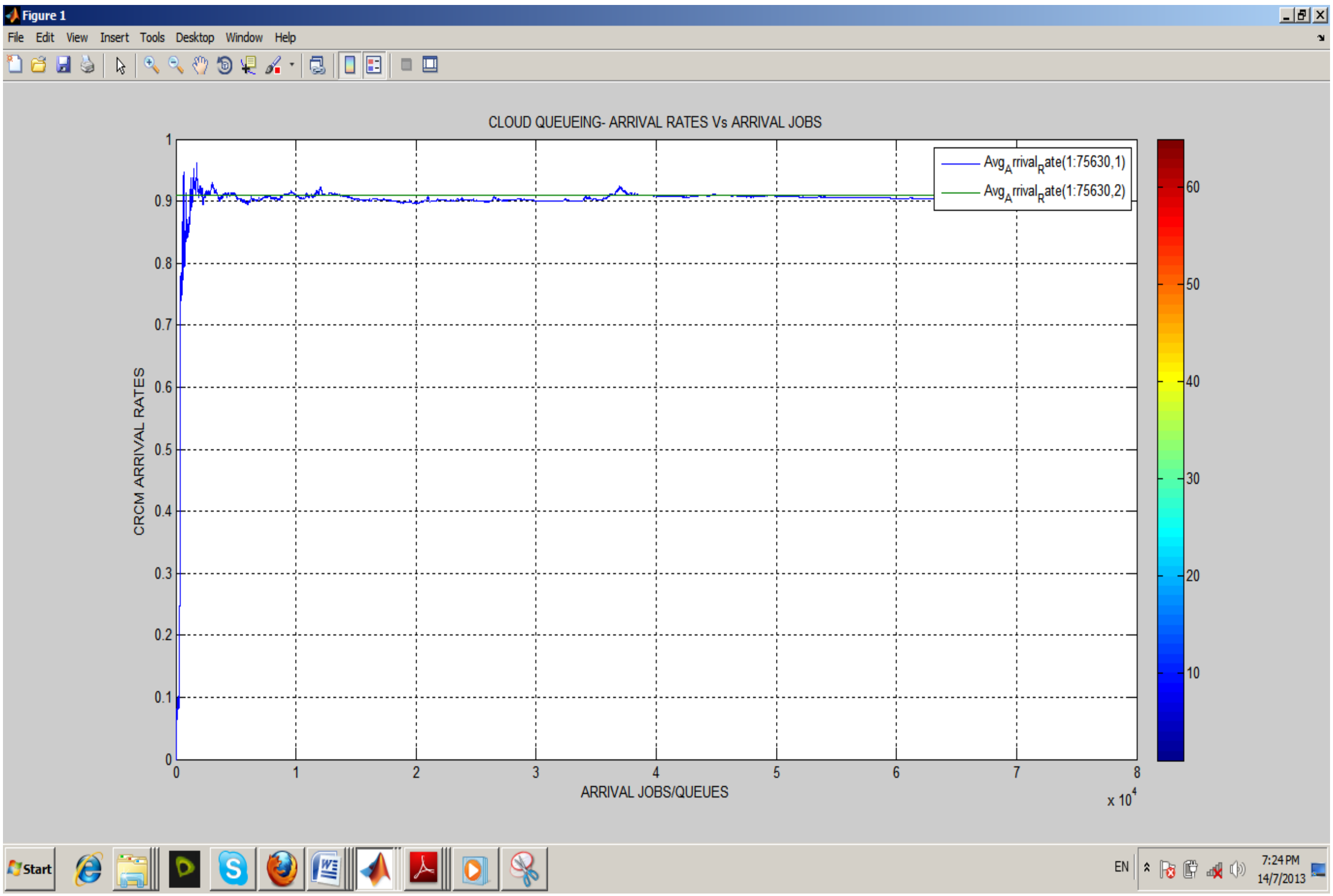


Figure 3.12: Cloud Queuing Arrival Rate Response

Figure 3.12 shows the arrival rate response. From the plot, it was observed that the arrival rate was never greater than the service rate in that there is no buffer overflow. Table 3.2 contains the data for the plot in figure 3.12

Table 3.3: Cloud Queuing Content Response

Cloud Queuing Content	Arrival Jobs
0	0
1	500
2	1000
3	1500
2	2000
1	3000
2	4000
3	5000
0	10000
7	12000

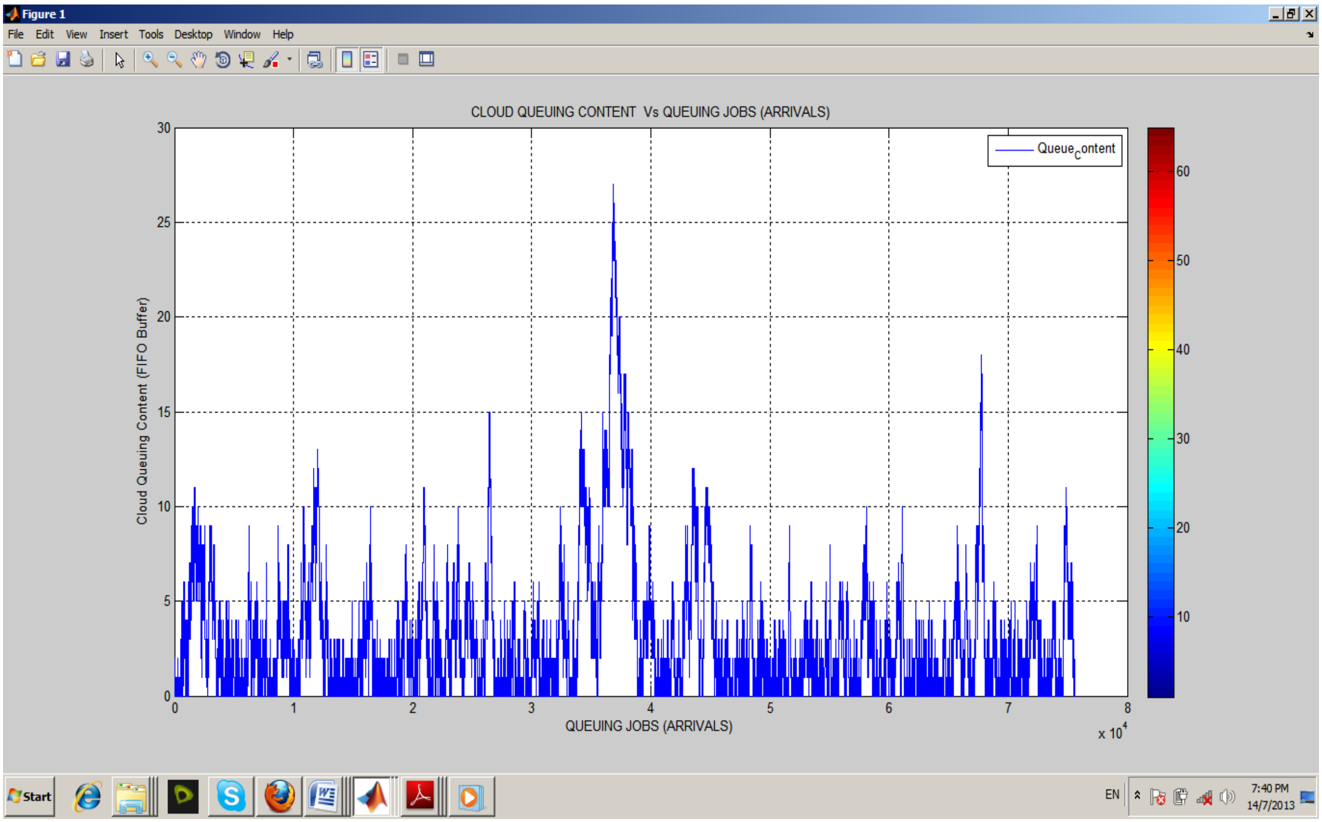


Figure 3.13: Cloud Queuing Content Response

Figure 3.13 shows the cloud queuing content response. The implication of this plot is that jobs will be dispatched as long as the service rate is greater that the arrival rates. The maximum threshold is about 27. Table 3.3 shows the data for the plot of figure 3.13.

Table 3.4: Cloud Queuing Workload

Queuing Workload	Arrival Job queues
0	0
161.2	500
328.6	1000
479.4	1500
654.8	2000
1.1382e03	3000
1.8010e03	4000
2.0760e03	5000
3.6832e03	10000
6.1762e03	15000
8.0850e03	20000

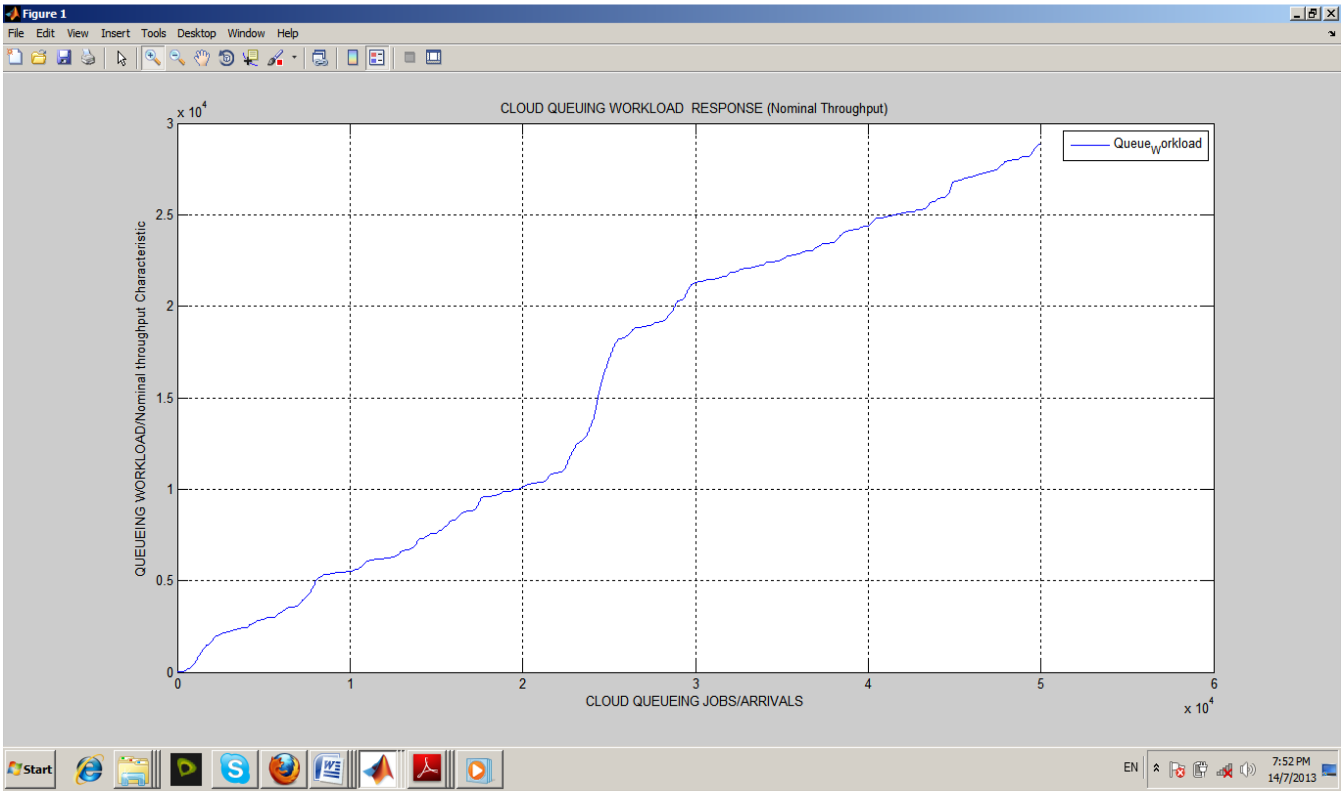


Figure 3.14: Cloud Queuing Workload Response

Figure 3.14 shows the cloud queuing workload response. It was observed in figure 3.12 that the Queuing workload QW which shows a nominal throughput gave a maximum value of $2.88x10^4$ when the arrival rate was $5x10^{4.}$ Table 3.4 shows the data for the plot in figure 3.14

Table 3.5: Server Utilization Response

Server Utilization Index	Arrival Job queues
0	0
0.7764	500
0.8792	1000
0.8770	1500
0.8920	2000
0.9009	3000
0.9181	4000
0.9047	5000
0.8867	10000
0.8863	15000
0.8896	20000

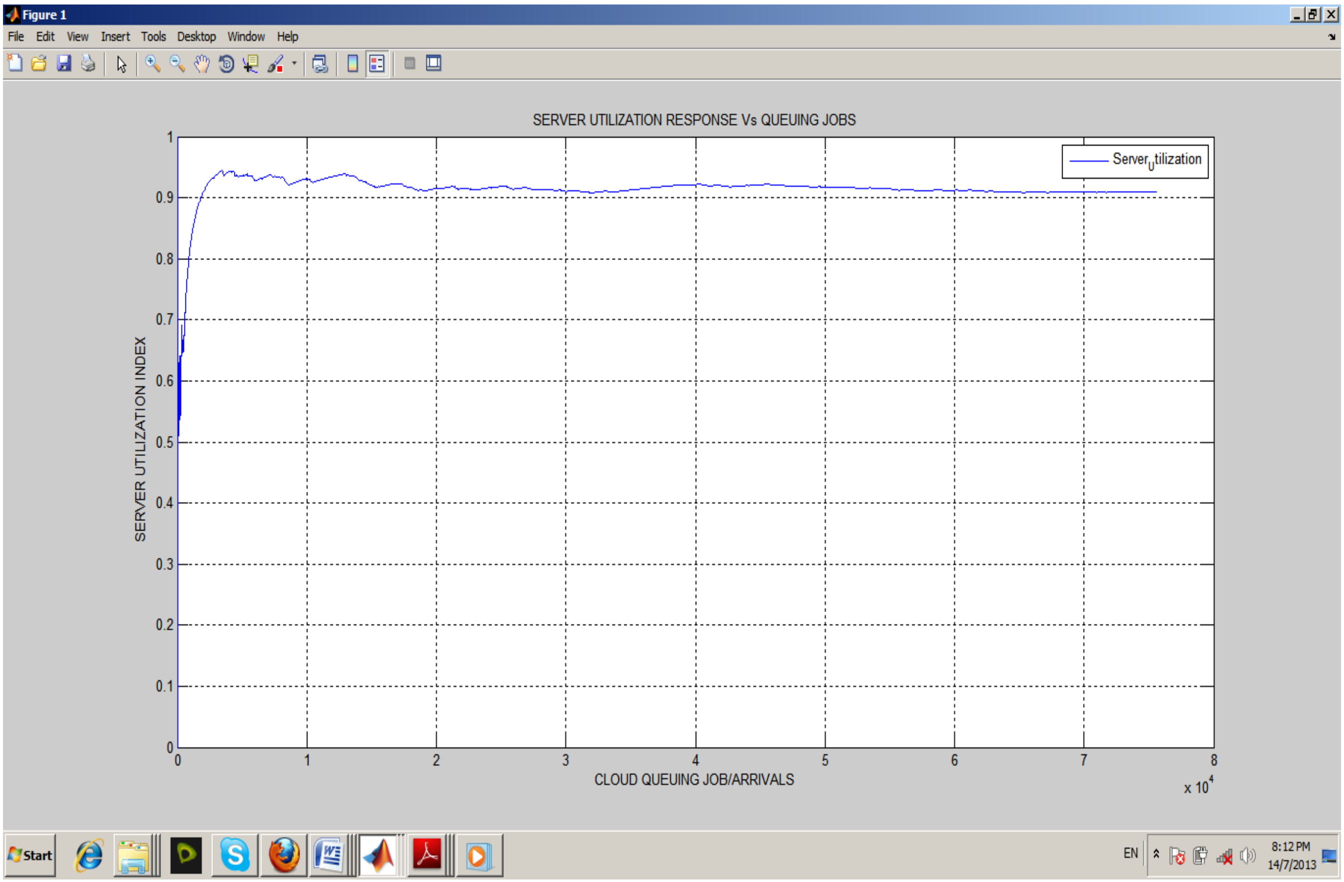


Figure 3.15: Server Utilization Response

Figure 3.15 shows the server utilisation response as obtained from the simulation. At the beginning of the simulation when the number of jobs on the queue was smaller it gave a value that is a little bit above 50% and gradually increased until it stabilised to a value of 0.9094 or 90.94% even with further increase in load. This shows an improved server utilisation. Table 3.5 shows the data for the plot in figure 3.15

Table 3.6: Server Service Time Response

Arrival Rate	Arrival Job queues
0	0
0.9270	500
0.9815	1000
1.0159	1500
1.0254	2000
1.0285	3000
1.0401	4000
1.0294	5000
1.0191	10000
1.0233	15000
1.0182	20000

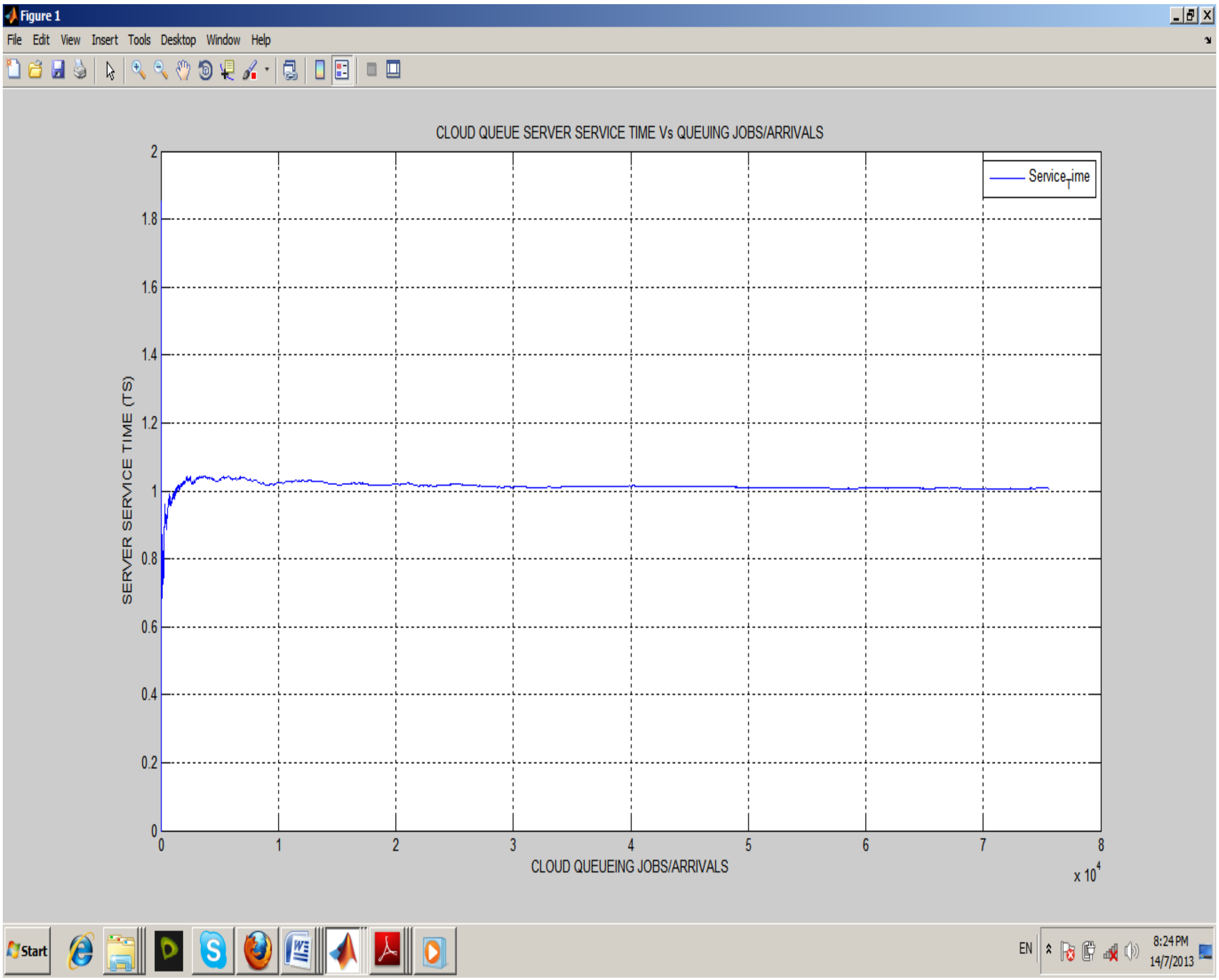


Figure 3.16: Server Service Time Response

Figure 3.16 shows the server service time response. The service time was observed to be 1.007sec. Table 3.6 is the table for the plot in figure 3.16.

Note that these values were obtained by ensuring that the optimal values of the arrival process variance (0.4033) and service process variance (0.3) were kept constant. In the integration phase of figure 3.3, this work now proposes these values for such integration to facilitate service efficiency and optimal performance.

A discrete event simulation with OPNET software will be used to show the operational metrics of the proposed systems.

3.7 Design of the Hybrid Encryption Algorithm (HEA)

Owing to the security challenges faced in cloud computing because of the entrusting of one's data to a third party and the need for the protection of the data generated in the process of research from unauthorised access, an important cryptographic scheme was introduced in the design. Data encryption D_e and security authentication, S_a are critical requirement for securing data files for both cloud service data centre and clients. In this work, the encryption of the data from the client's end will be dealt with i.e. the second level or second tier security arrangement. The first level security will be taken care of by the cloud service providers as explained previously. The reason for this is to ensure that even the cloud service providers will not have unauthorised access to one's data in their care. Also in the event of an infiltration from the service provider's end or a compulsion from say government authorities, whatever is encrypted will remain unreadable to unauthorised viewers as long as the user remains in control of the encryption keys. Also, since the encryption is on the data, the data will remain safe even if hardware vulnerabilities are exploited.

Considering some of the perceived risks related to cloud computing shown in the literature review and the analysis of existing cloud-based research collaboration platforms like confidentiality, integrity and availability of information, some measures were taken in the system design. The issue of physical threats will be taken care of by the cloud service providers as would be specified in the service level agreement. The vendor back up plan in case of unforeseen circumstances was also considered before choosing the cloud vendor for the system deployment. It will be ensured that providers having good physical security for cloud infrastructure in place will be used for the system deployment. The issue of confidentiality and integrity of data in transit to and fro the cloud is taken care of by the data encryption and digital signing applied to the data by client side security system designed in this work and the demand for authentication by the system before access is granted into the database.

The security features of CRCM will encompass identification, authentication, authorization, confidentiality and integrity. A cryptographic primitive which is fundamental in authentication, authorization, and non-repudiation is the Hybrid Encryption Algorithm (HEA). This comprises both the encryption and digital signature algorithm for both data security and integrity. After the user registration, to participate in the CRCM will require a validation and a signing of a Collaboration level Agreement (CLA) so as not to abuse the platform content and also to respect intellectual property rights. This is captured in the design flowchart as well.

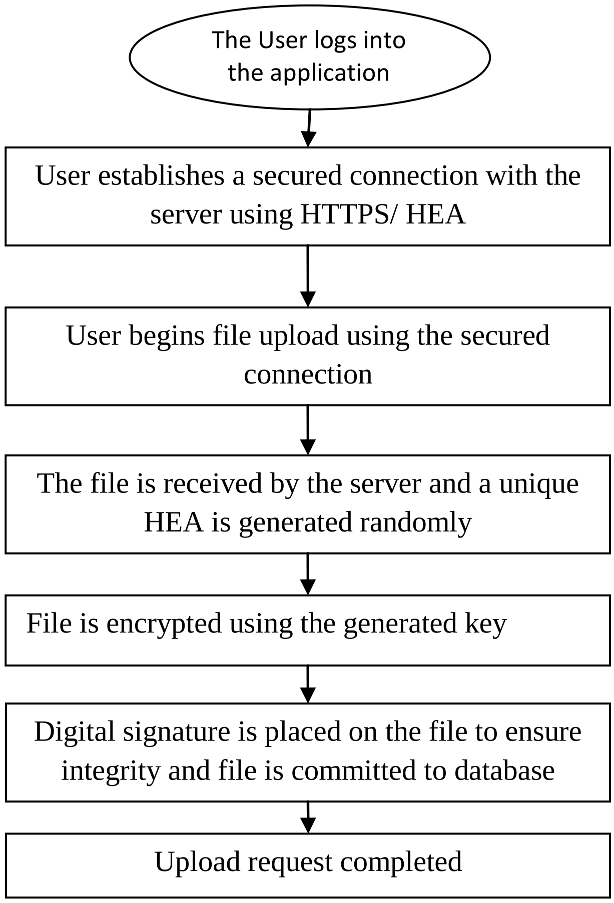


Figure 3.17: Design flow diagram for the security and data integrity system

Figure 3.17 shows the design flow diagram for the security and data integrity system. The data generated from the application through research and other activities are uploaded to the database as shown. Before committing the data to the data base, the file is encrypted using advanced encryption standards after which a digital signature algorithm is used to ensure the integrity of the file.

The Integration of the CRCM using PhP and mySQL database management system forms the application layer integration on the cloud environment. But the CRCM security is given by the background encryption and the background decryption algorithms which are outlined below. The encryption operation in the CRCM cryptosystem is exponentiation to the e^{th} power modulo n.

c = ENCRYPT (m) = $m^e \bmod n$

The input m is the message or plaintext (p_t);

The output *c* is the resulting ciphertext.

In the design, the actual message is encrypted with the shared secret key using HEA. This construction makes it possible to encrypt a message of any length with only one exponentiation. The decryption operation is exponentiation to the d^{th} power modulo *n*:

m = DECRYPT (c) = $c^d \bmod n$

The relationship between the exponents, *e* and *d* ensures that encryption and decryption are inverses, so that the decryption operation recovers the original message *m*.

Consequently, *n* and *e* is made public in CRCM without compromising security, which is the basic requirement for a public-key cryptosystem.

The fact that the encryption and decryption operations are inverses and operate on the same set of inputs also means that the operations can be employed in reverse order to obtain a digital signature. A message/plaintext can be digitally signed by applying the decryption operation to it, i.e., by exponentiation to the d^{th} power:

s = SIGN (m) = $m^d \bmod n$.

The digital signature can then be verified by applying the encryption operation to it and comparing the result with and/or recovering the message given by

m = VERIFY (s) =$s^e \bmod n$.

Hence, the plaintext m is generally some function of the message; this makes it possible to sign a message of any length with only one exponentiation.

Figure 3.18 shows the CRCM encryption system block diagram and depicts how it is applied to the plaintext (P_t) on the CRCM platform.

During the encryption, to initialize n_i: ie. $n_0, n_1, n_2, n_3, \ldots\ldots\ldots.n_{255}$

By setting an index i such that

For i = 0 to 255;

C = Encrypt (m) = m^e Mod 256;

Swap *i;*

End;

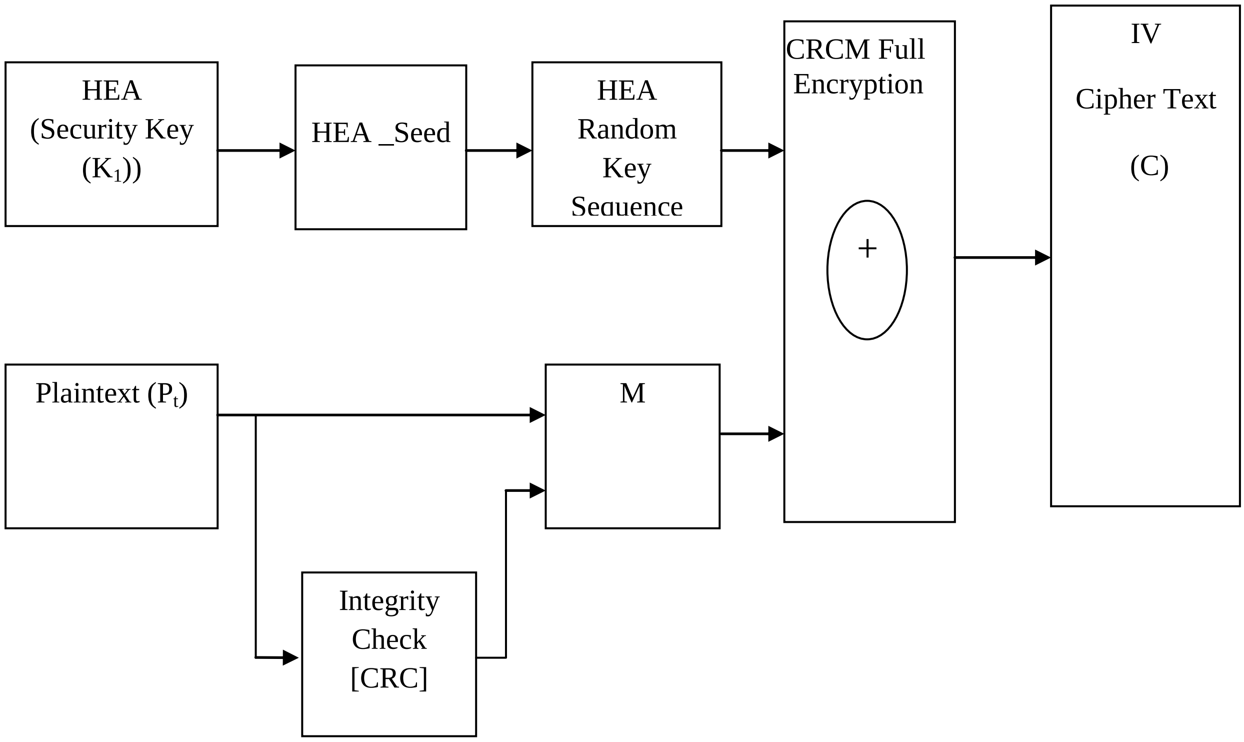


Figure 3.18: CRCM Encryption Model

From figure 3.18, an integrity checksum is calculated on the user plaintext (p_t) using cyclic redundancy check for 256 bits [CRC256]. The p_t concatenates the [CRC256] to form the *M* ie encryption Map.

Hence, M= {P_t, CRC256[p_t]}

Let an initialization vector (IV) be chosen as 2^8 bits random number by the user on the CRCM platform. The seed is processed by a pseudorandom number generator P_{rg} or key sequence generator which outputs a key sequence of P_{rg} bytes equal to length of *M.*

Now, the sequence and the *M* are Xored as shown in figure 3.18 to obtain the Cipher text C. The length of the key sequence and the M are the same so that they can be Xored to get the result with the same length, hence,

$C = \{P_t, CRC256[p_t]\}$ Xor P_{rg}_HEA (IV,K)}

The {IV, P_t , CRC256 [p_t], (p_t)} forms the HEA sent securely to the CRCM backend.

During the decryption phase as shown in figure 3.19, the IV of the incoming authentication data is used to generate the key sequence necessary to decipher the encrypted details.

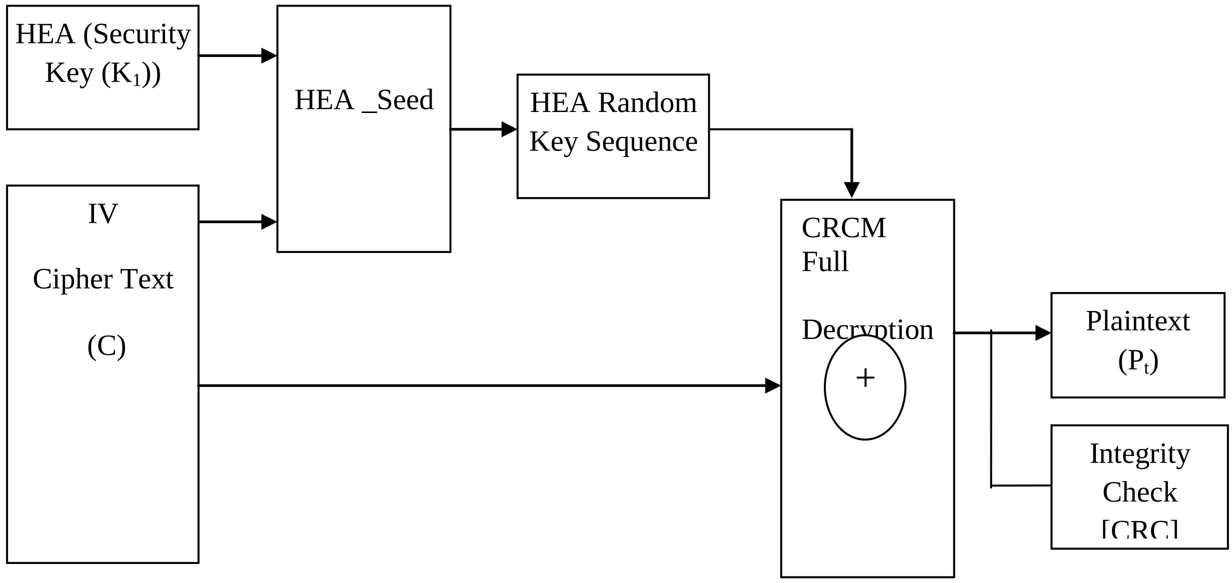


Figure 3.19: CRCM Decryption Model

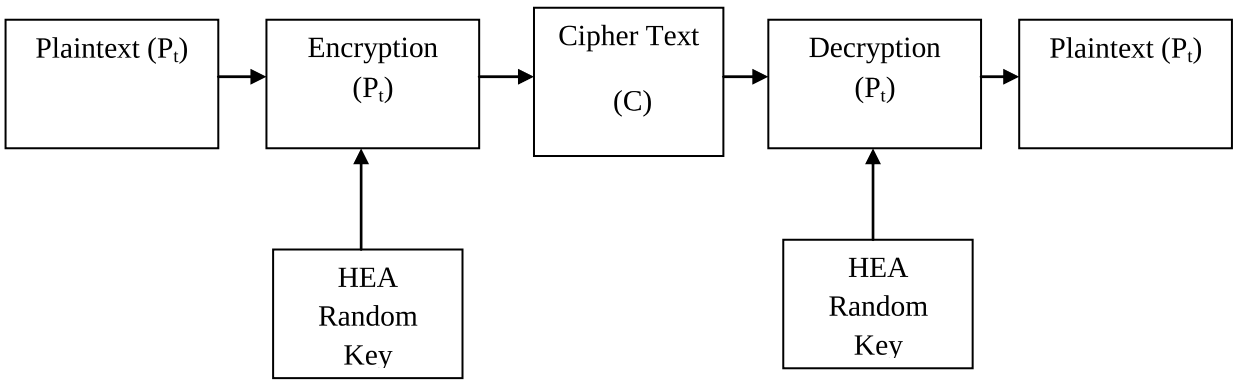


Figure 3.20: Combined Security Model

The HEA pseudorandom generator plays a critical role in the security system as it generates the same key sequence as when used in encryption phase. The reason is because, the same pair of the IV and the security key (K) is passed to it as the input, of the encryption process.

Now, the key sequence is Xored with the Cipher text to extract the plaintext (P_t') and CRC256 bits.

In the system, correct decryption is verified in HEA by performing a CRC256 (P_t') and comparing it with the sent CRC256 (P_t). If CRC256 (P_t') is not equal to CRC256 (P_t), access is denied.

From this analysis, an important aspect of the encryption and decryption processes is the signing and verification phases. This is presented below. The algorithm I describes the analytical signing and verification schemes for the CRCM system which uses 256 bits in its cryptographic process. This is seen as an improved security scheme in that the higher the number of bits used for encryption the stronger the security. In the algorithm, a user that logs in to the CRCM system creates a signature which identifies such user.

Algorithm I: HEA Encryption mode _256bits

Begin ( )

Signing procedure

Entity A (the signer) creates a signature for a login_details $m \in M$, by doing the following:

1. Compute $s = S_A(m)$ //*Signature for entity A*//
2. Transmit the pair (m,s). s is called the signature for login_details m.
3. Store in database as HEA

Verification procedure

To verify that a signature s on a message m was created by A, an entity B (the verifier) performs the following steps:

1. Obtain the verification function V_A of A.
2. Compute $u = V_A (m,s)$.

3. Check for the validity of HEA
4. Accept the encryption signature as having been created by A if u = true, and
5. And Reject the encryption signature if u = false

End ( )

Figure 3.20 is a block diagram showing both the encryption and decryption processes.

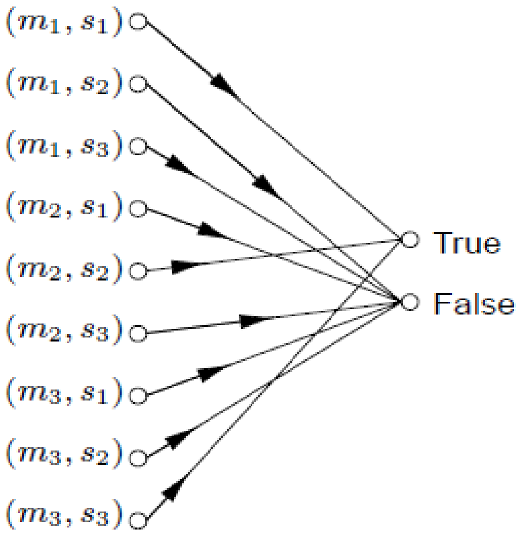
A generic description of the hybrid system security and integrity design follows below.

Consider the set M*S consisting of all pairs (m,s) where $m \in M$, $s \in S$, called the Cartesian product of M and S. Now let,

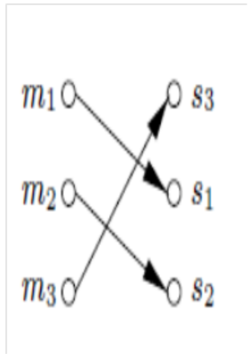
- M be the set of login_details which can be signed by the administrator.
- S be the set of elements called signatures, possibly alpha binary strings of a fixed length.
- S_A is a transformation from the login_details set M to the signature set S, and is called a signing transformation for entity A. The transformation S_A is kept secret by A, and will be used to create signatures for messages from M.
- V_A is a transformation from the set M * S to the set {true; false}. V_A called a verification transformation for A's signatures, is publicly known, and is used by other entities to verify signatures created by A.

Definition- The transformations S_A and V_A provide HEA scheme for A. This is called digital signature mechanism and its formulation is shown below.

Given digital signature scheme M = {*m1*,*m2*, *m3*} and S = {s_1,s_2, s_3}. The right side of Figure 3.21shows a signing function S_A from the set M and, the left side the corresponding verification function, V_A.



V_A



S_A

Figure 3.21: A Signing and verification function for CRCM Encryption Scheme

Figure 3.21 shows a graphical display of CRCM security property in which:

(a) There is an arrowed line in the diagram for V_A from $(m_i;s_j)$ to true provided there is an arrowed line from m_i to s_j in the diagram for S_A.

(b) CRCM Property in figure 3.21 provides the security such that the signature uniquely binds A to the login_details which is signed.

Note that the digital signature is applied on the message digest not on the whole data so as to reduce storage space requirement and increase the speed of computation. The reason for this is that the number of bits for signature is same as that of text if it is to be placed on the text. Doing so will mean using double the amount of storage required for any data to be stored and thus a resultant decrease in the speed of processing. This explains the partial use of the MD5 cryptographic algorithm its weakness as a security algorithm notwithstanding.

The HEA system is a 256-bit checksum that tags the data with identifiers that can be verified cryptographically. These checksums are used to verify the integrity of the data throughout the entire process. If corruption within a particular block is detected, a redundant copy of that block is used to automatically repair that corruption. In addition all data is periodically verified against their checksum to identify and repair select data corruption.

Every data block sent to the cloud is digitally signed by an encryption key and then verified during restore. This provides cryptographic assurance that data sent to the cloud has not been tampered with by anyone and is exactly same as the original copy. For the file download, the file is decrypted on the provision of the correct key and displayed to the user.

The system security deployment was done at the back end to avoid attacks at the application interface by hackers and unauthorised software. Access control was also deployed at the cloud provider's end to avoid the issue of unauthorised access to the data and information sent there. The security design above was integrated into the application implementation.

3.8 CRCM Platform Design using Modified Waterfall Methodology and Discrete Event System Methodology

The proposed model called Cloud-based Research Collaboration Model (CRCM) is designed for efficient performance and security integration. This will address collaboration authentication, efficient modularization and process architecture integrations. The CRCM design involves the building of the new system based on the identified problems. Program design involves solution statement and coding. The design objective in this case is to develop the system flowcharts encompassing: Global Module, Content Acquisition Module, Information Management Module, Cloud Storage Module, Content Delivery Module, Registration Module, industrialist's collaboration module and Collaboration Level Agreement Module. This will be achieved using a modified form of waterfall methodology with XAMP Php and MySQL server database. In this section also, the respective flowcharts of the CRCM design will also be presented leveraging on the previous discussions in this chapter to achieve figure 3.3 completely.

3.8.1 System Requirement

- **Input design:** The input design involves the design of the sub-system through which the users of the system enter data into required fields. The data is then processed and the results are displayed to the user telling him/her the next action to take. An example of this is at the login stage shown in figure 3.22, where the collaborator enters a username and password and on pressing the submit button, the data entered is processed and the result of the login is displayed if it is successful or unsuccessful. The system can also display an error message if the input was wrong. The input subsystem also includes the data upload from virtual equipments.

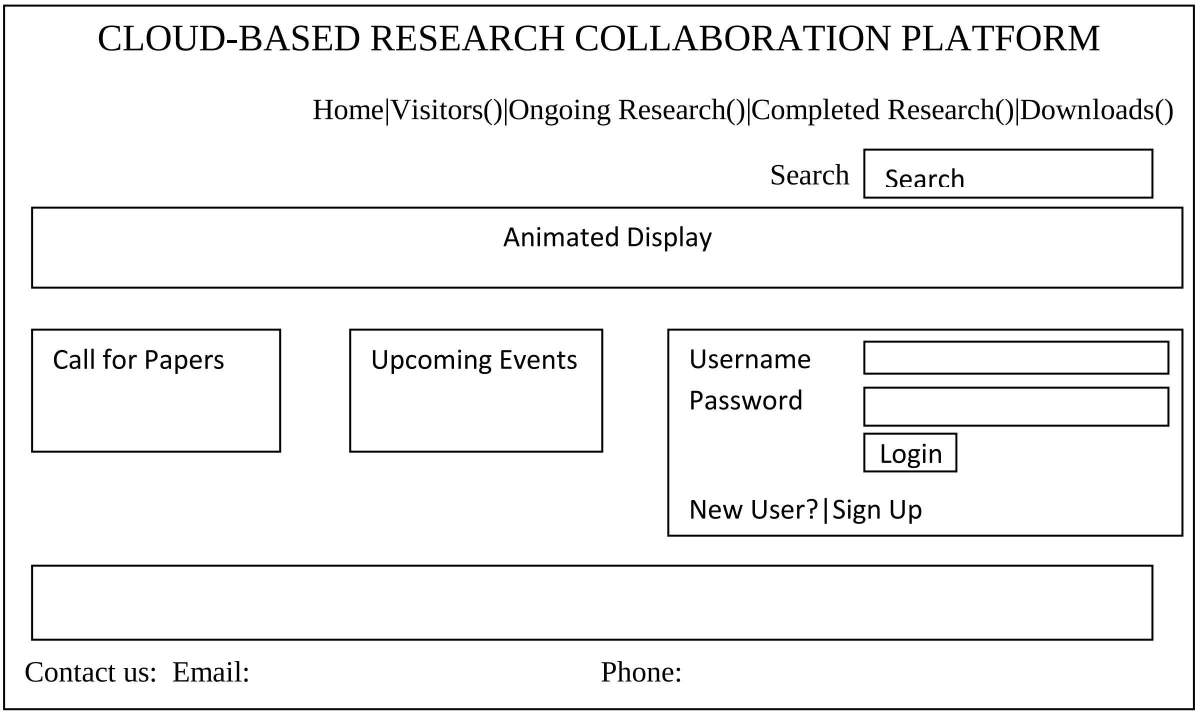


Figure 3.22: Design of Login interface

- **Output design:** The output design shows how information is displayed to the user after a request is made by the user. The user specifies what task is to be done and based on that; the web page displays the necessary information relating to the selected task. The means of output are the web pages displayed by the web browser after the server processing of the users input. Every output is displayed in a web page. The design of the output graphical user interface displayed to a registered user after inputting the login details and clicking ok is shown in figure 3.23.

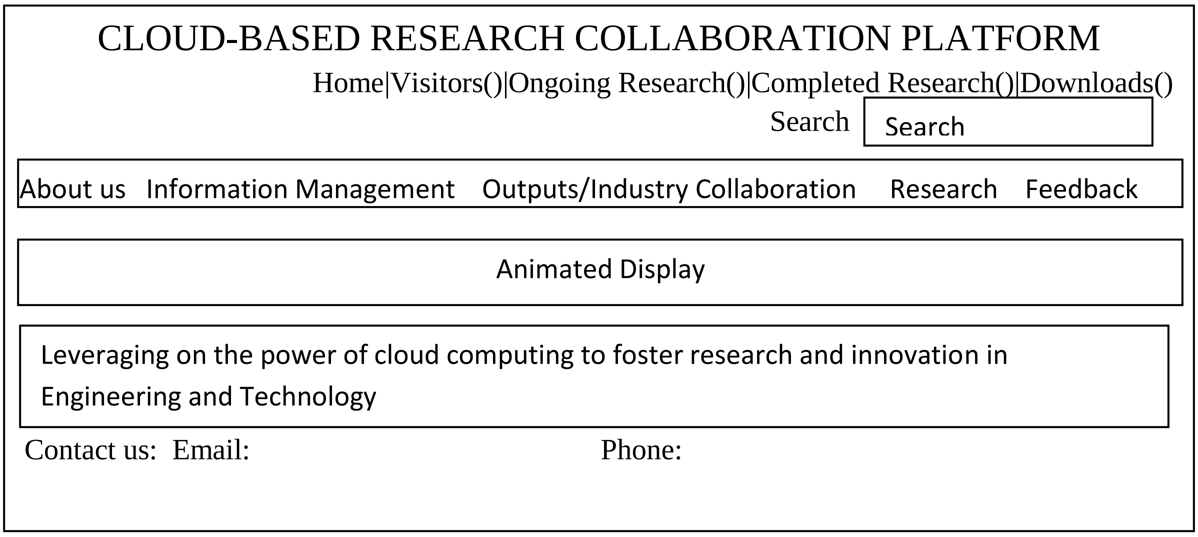


Figure 3.23: Sample GUI display design for output

- **Hardware requirements:** the minimum hardware requirements include:

- Monitor
- 4GHZ or faster processor
  - 4GB of RAM
  - 1TB of available hard-disk space
  - 1280 X 800 display with 64-bit video card
  - Memory requirements:
    - 1 GB for the logic instance (grid control)
  - Disk space requirements:

    – 1.5 GB of swap space

    – 400 MB of disk space in the /tmp directory

    – Between 1.5 GB and 3.5 GB for the CRCM

    – 1.2 GB for the preconfigured database (optional)

    – 2.4 GB for the flash recovery area (optional)

It is worthy of note that the hosted software can be accessed using any browser from lower capacity systems.

3.8.2 Software requirements

The requirements to run the proposed system for optimal performance are listed below and any computer that meets these requirements can install the system without any problem. The platform will be integrated to which is an existing database.

- Microsoft windows vista or windows7 while using Linux Redhat for production deployment.
- Apache server version 1.3.14 & above
- PHP version 4.1.0 & above
- MYSQL version 4.1.0 & above
- All browser compatible
- Operating system requirements
    - Adequate temporary space for paginations to virtual memory
    - 64-bit and 32-bit compatible
    - Windows 7/Server 2007 and Linux Redhart
    - OS patch level 1
    - System and kernel parameters must be enabled
    - Sufficient swapping
    - Nonempty XAMP htdocs_HOME
    - MySQL database

3.8.3 Architectural Design

3.8.3.1 Grid Control Management Framework (GCMF)

From figure 3.3, this work introduced a GCMF whose role is to coordinate all the activities of the CRCM concurrently. The starting point in GCMF is the process architecture that will streamline the interaction between the user and the server backend. Figure 3.24 shows the developed block diagram for the process architecture.

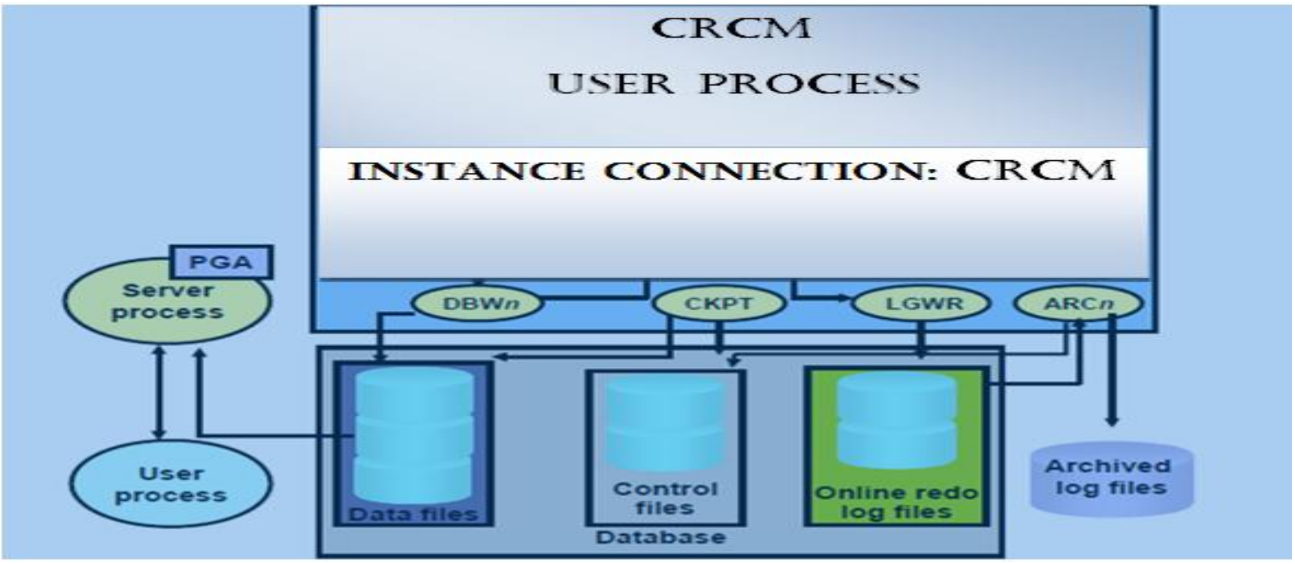


Figure 3.24: CRCM Process Architecture Block diagram

The metrics of Figure 3.11 will be considered in GCMF particularly at the server end where resource utilization and background process management is vital. The CRCM client server architecture shown in figure 3.24 serves as the engine block of figure 3.3. The architecture is made up of the following components: the user process, instance, server, and background processes and the database viz-a-viz the server.

In its operational status, the CRCM server creates server processes that handle the requests of users or client processes that connect to the logic instance. Now, the logic instance is a collection of the background processes and the shared memory allocated by the server during the normal operation. It is also the connection bridge as shown in figure 3.24.

i. **User process**: The user process (which is part of the process structure in the CRCM process architecture) is the connection to the CRCM server having passed the authentication and instance connection as presented in figure 3.4, figure 3.5 and figure 3.6.

ii. **Server process**: This server process is in dedicated server mode. Each user process has its own server process in figure 3.24.

iii. **Background Structures/ Processes:** In CRCM, a set of background processes for the logic instance that interacts with each other and with the server to manage memory structure, and asynchronously perform Input/output to write data to the server are enabled. They provide parallelism for better performance and reliability. They include:

- The System Monitor (SMON) which performs crash recovery when the logic instance fails in the CloudMesh ERP.
- The Process Monitor (PMON) which performs process recovery when a user process fails. It is responsible for cleaning up the cache htdocs when failure occurs from the user perspective.
- Server database (DBWn) which writes to table blocks in the buffer cache of MySQL server on server disk.
- Checkpoint (CKPT) process which is responsible for signalling DBWn for recent updates in the CloudMesh.
- Log writer (LGWR) which writes buffer log entries to the server, while the archive stores log entries of the XAMP database as used in the design.

Essentially, these background processes are taken care of by the server during the installation process. From figure 3.24 also, the server process uses the Program Global Area (PGA) privately to communicate to the background process logically.

iv. **Storage structures:** Data mining capability of the CRCM can be realized in the storage structure of the CRCM server. These are divided into two viz:

- Logical storage which have the database, schema, and table space for logical data storage.
- Physical storage which have the data files, control files, and redo log files for multiplexing of data, security reintegration, and automatic updates on the server. See Figure 3.24.

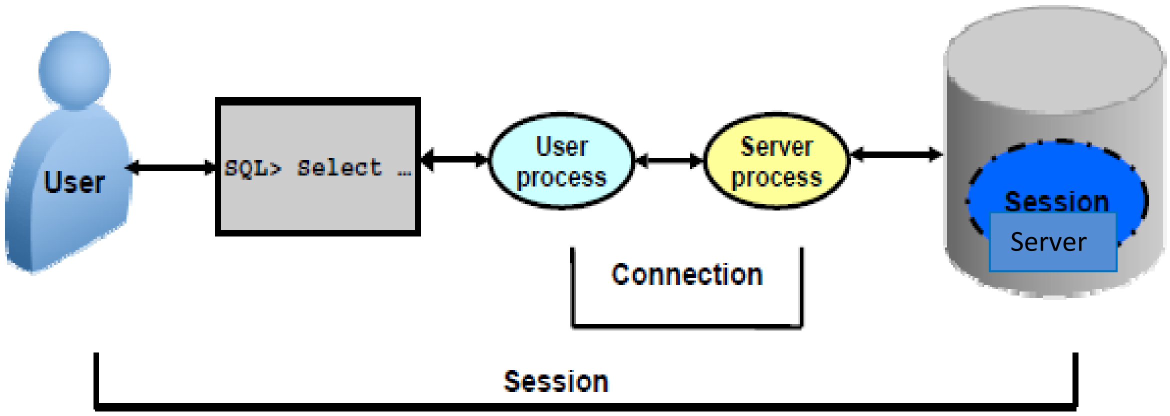


Figure 3.25: Interacting with the CRCM Server

As shown in figure 3.25, every transaction in the CRCM of figure 3.3 forms a session based on structured query language patterning for user connection into the server via the logic instance as discussed above.

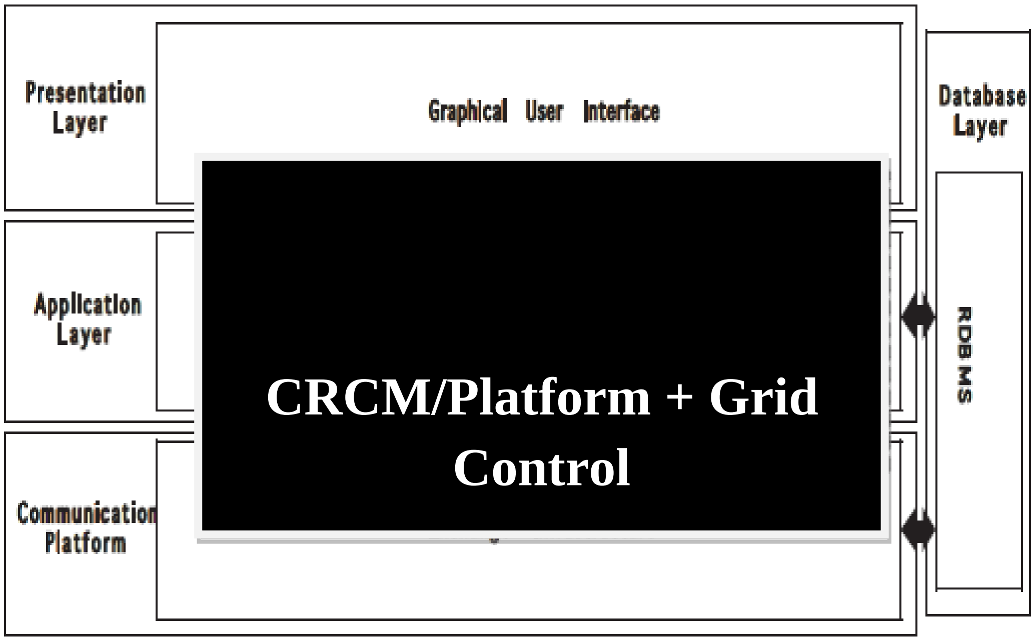


Figure 3.26: Block diagram of CRCM Interfaces with Grid Control

From figure 3.26, the main integration framework uses CRCM grid control to address the issues of software provisioning, maintainability, High availability, performance, Security, safe Information integration and application service monitoring. Performance refers to localising critical operations and minimizing communications while utilizing fine-grain components. Security refers to utilizing a layered architecture with HEA to restrict unauthorized access into the system. Safe Information integration refers to signing the Collaboration Level Agreement (CLA) before accessing critical features in the CRCM subsystem while availability refers to including redundant components and resource allocation mechanisms for fault tolerance and maintainability refers to the use of fine-grain, replaceable components.

3.8.4 CRCM Platform Design using Flowcharts

This work used its development philosophies to formulate the flowcharts for CRCM platform as shown in the system block diagram in figure 3.3. These flowcharts were designed considering the standards of software engineering ethics in principle. The design flowcharts serve as to maintain CRCM software qualities such as functionality, reliability, efficiency, maintainability and portability.

From figure 3.3, a more simplified decomposition of the system yielded nine functional modules that constitute the Http service scripts that are processed in the configuration palette. These subsystems of the CRCM summarily fit into the system model in figure 3.3. As will be shown in the flow charts, users collaborate while sending their queuing jobs to the server for service provisioning and security granularity on the platform. Figure 3.27 is the global model for the CRCM. It displays the log in page where the user logs in and if it is not a registered user is prompted to register before assessing the platform features. Upon successful log in, it displays the home page of the CRCM where the user selects the module to work with. Note that the data encryption and signing runs in the background as data is being packaged and sent to the cloud.

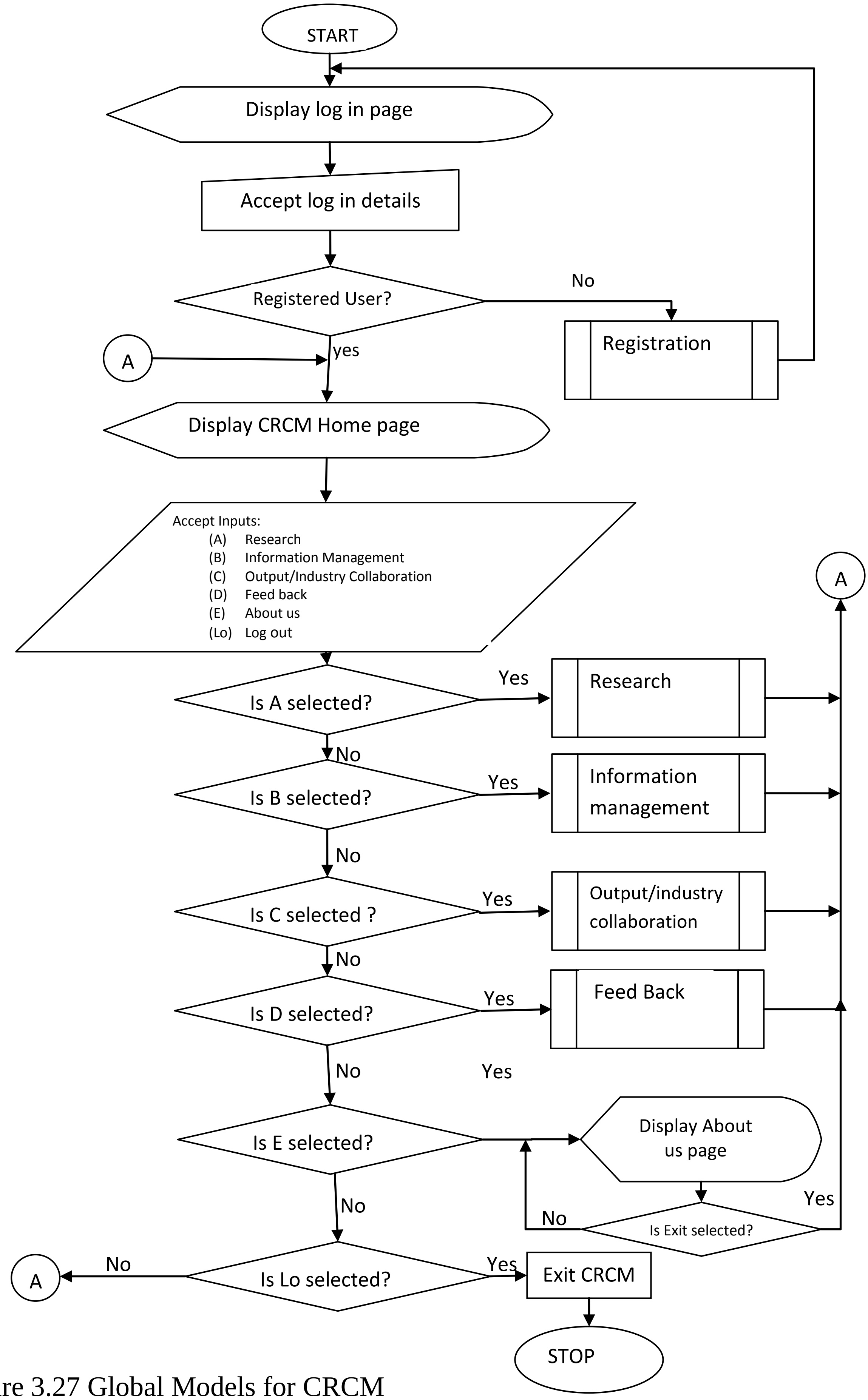


Figure 3.27 Global Models for CRCM

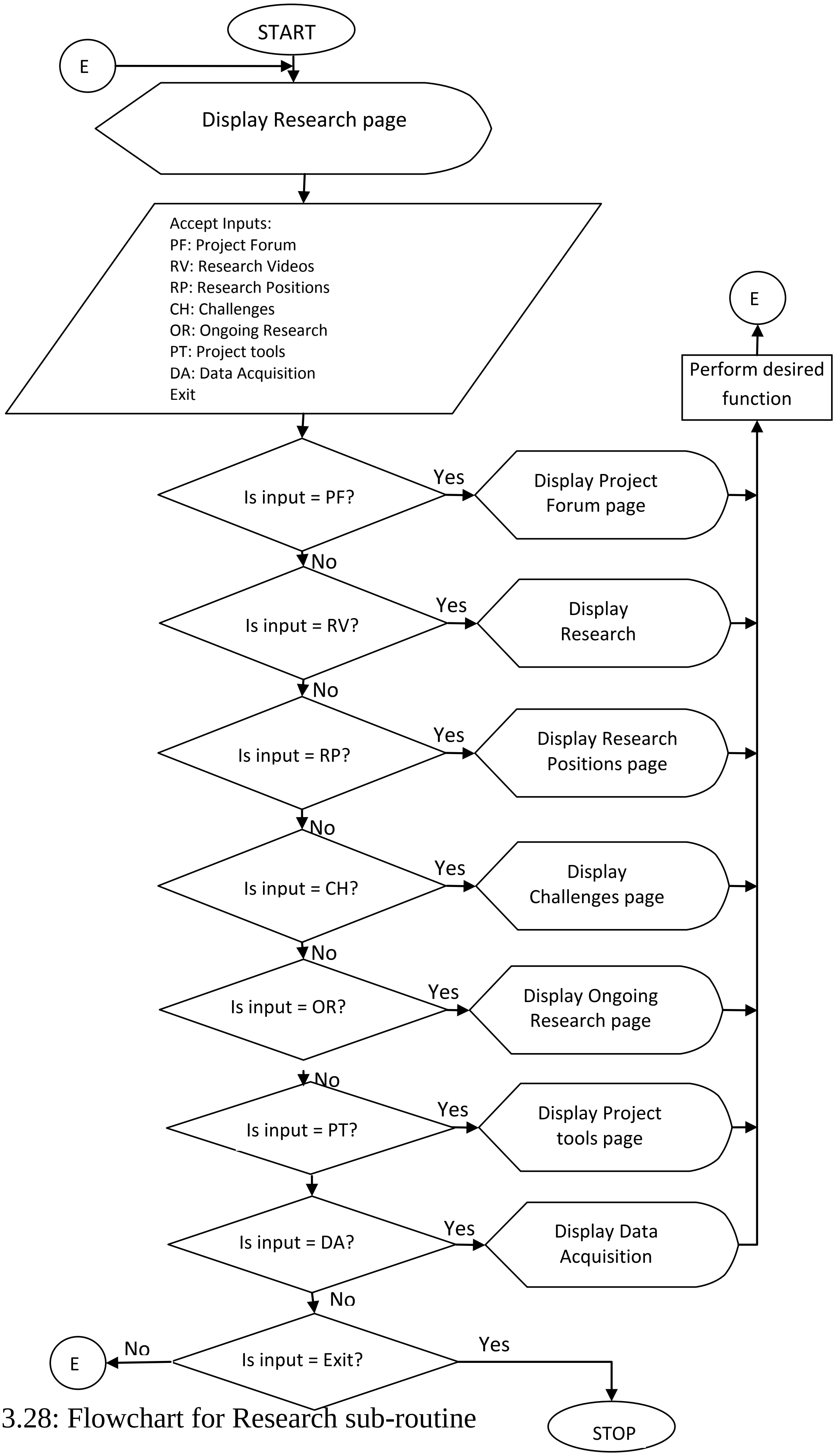


Figure 3.28: Flowchart for Research sub-routine

Figure 3.28 is the flow chart for the Research sub-routine. Here the system displays the subsystem selected like the project forums, ongoing research etc. if none is selected, the system remains at that state. It is on this interface that the collaboration between researchers takes place. It is also this section that controls the real time data acquisition and storage from data acquisition systems.

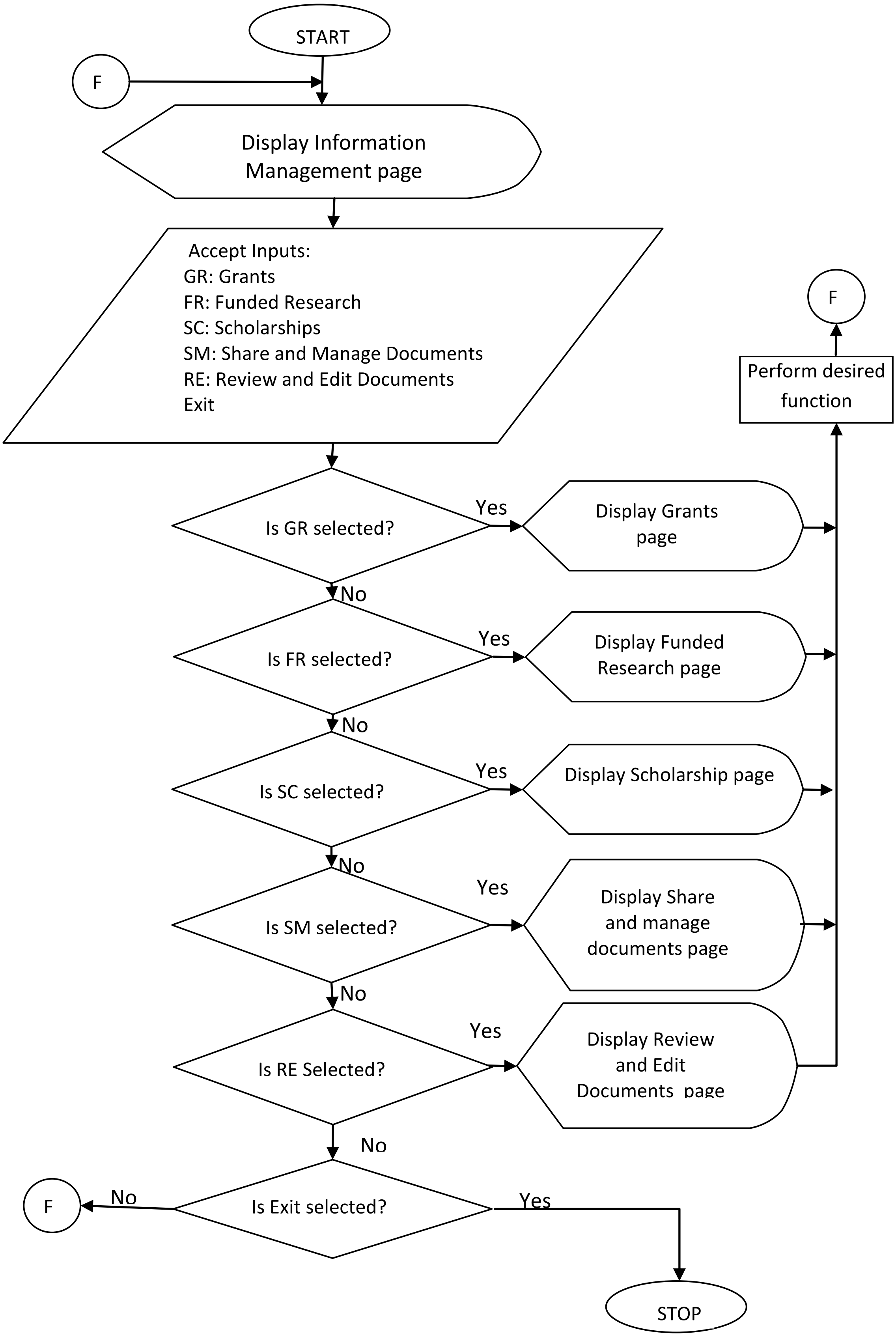


Figure 3.29 Flowchart for Information Management sub-routine

Figure 3.29 is the flow chart for the information management sub-routine. It is in this module that grants, funds and scholarships can be accessed. The system accepts file and document in any format. Here, data/information can be uploaded, downloaded, reviewed, edited, managed and shared.

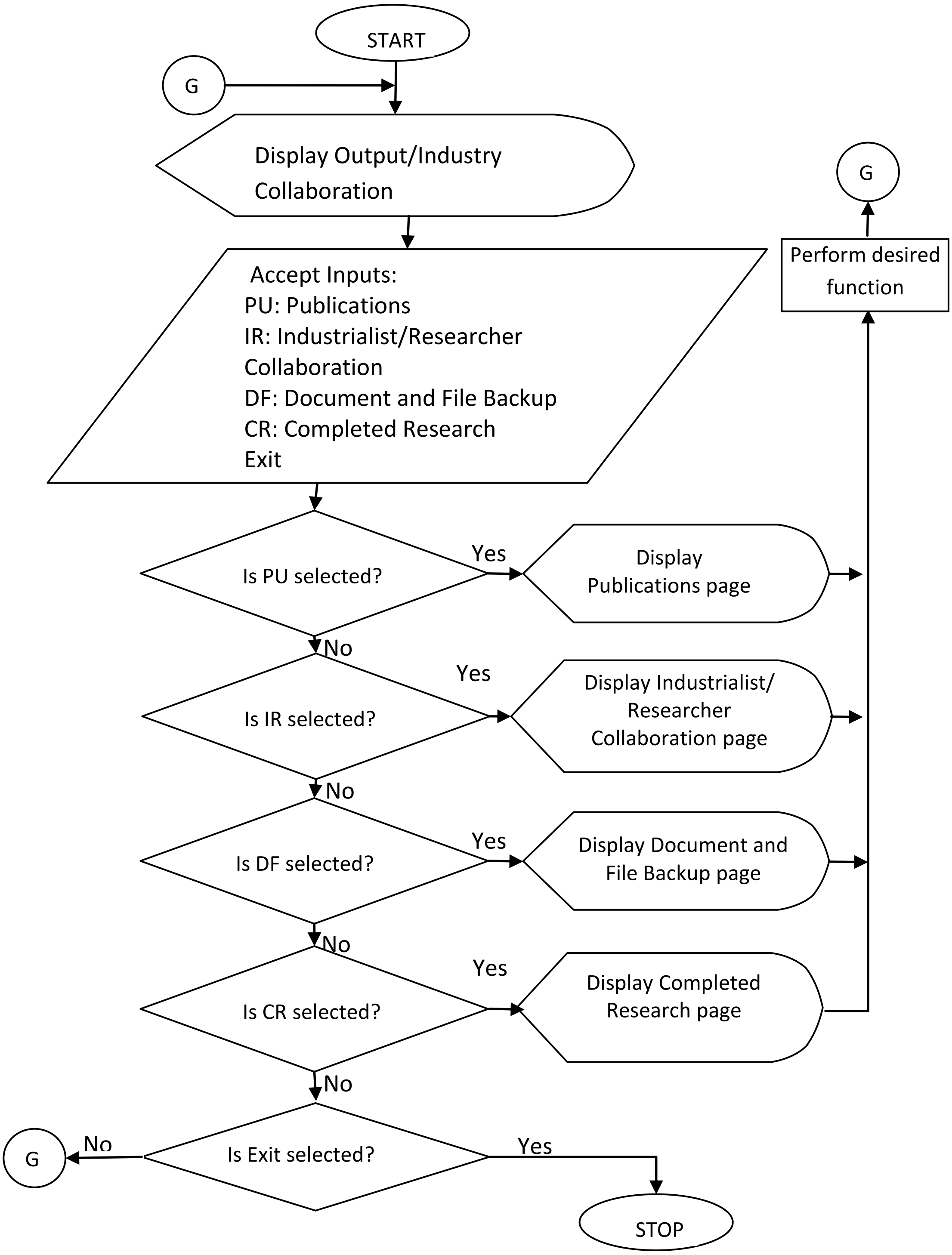


Figure 3.30 Flowchart for Output/Industry Collaboration sub-routine

Figure 3.30 shows the flow chart for the Output/Industry Collaboration sub-routine. Here users can upload and download publications. This section also controls the backup section. This interface also provides the platform for the collaboration between the industrialists and the researchers. Completed research works are also advertised here for the interest of industries that would sponsor their mass production. The cloud storage is a background operation which is done automatically.

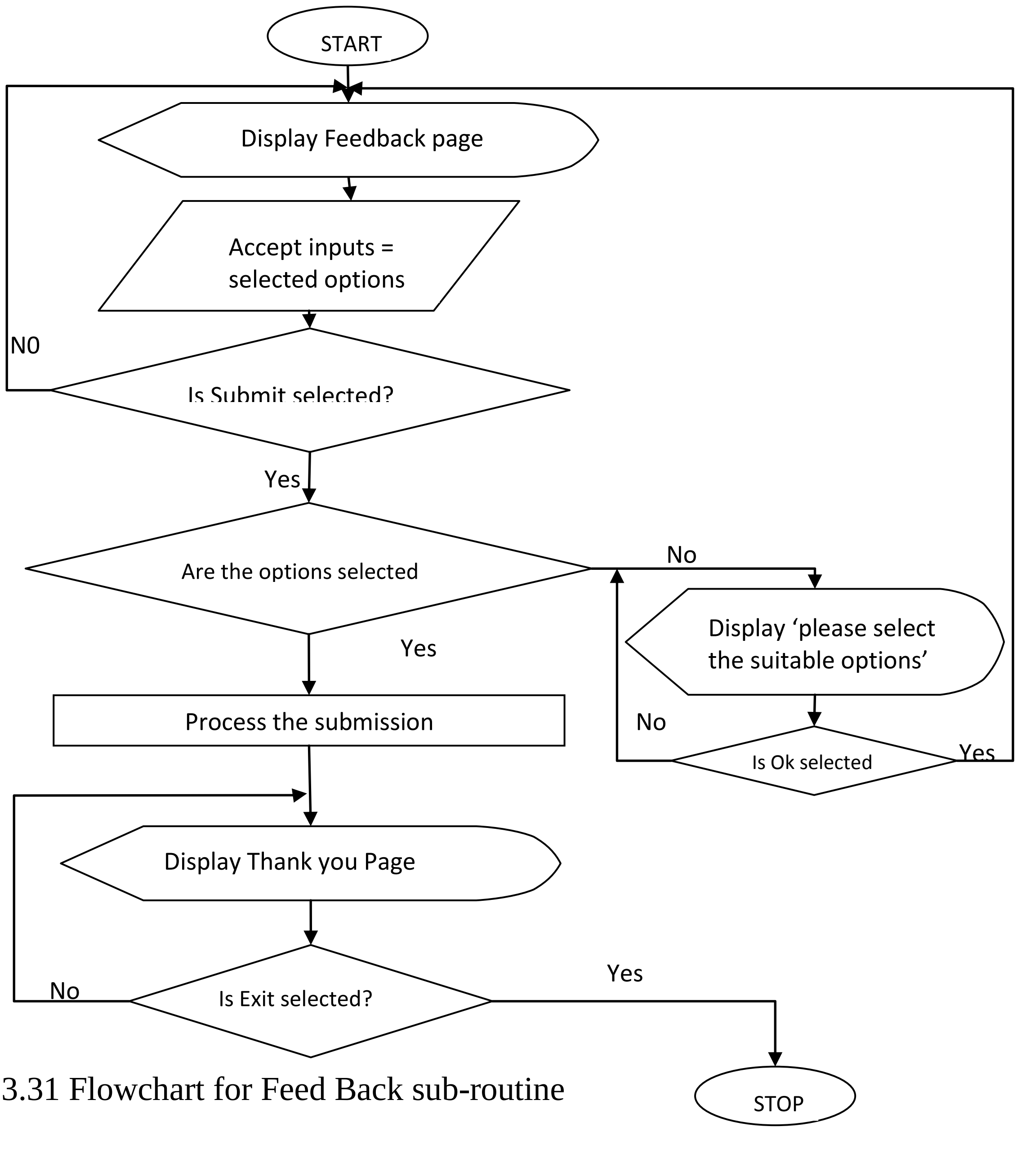


Figure 3.31 Flowchart for Feed Back sub-routine

Figure 3.31 is the flow chart for the feedback sub-routine. The purpose of this section is to check the impact of the application on the research world and its effectiveness in aiding research.

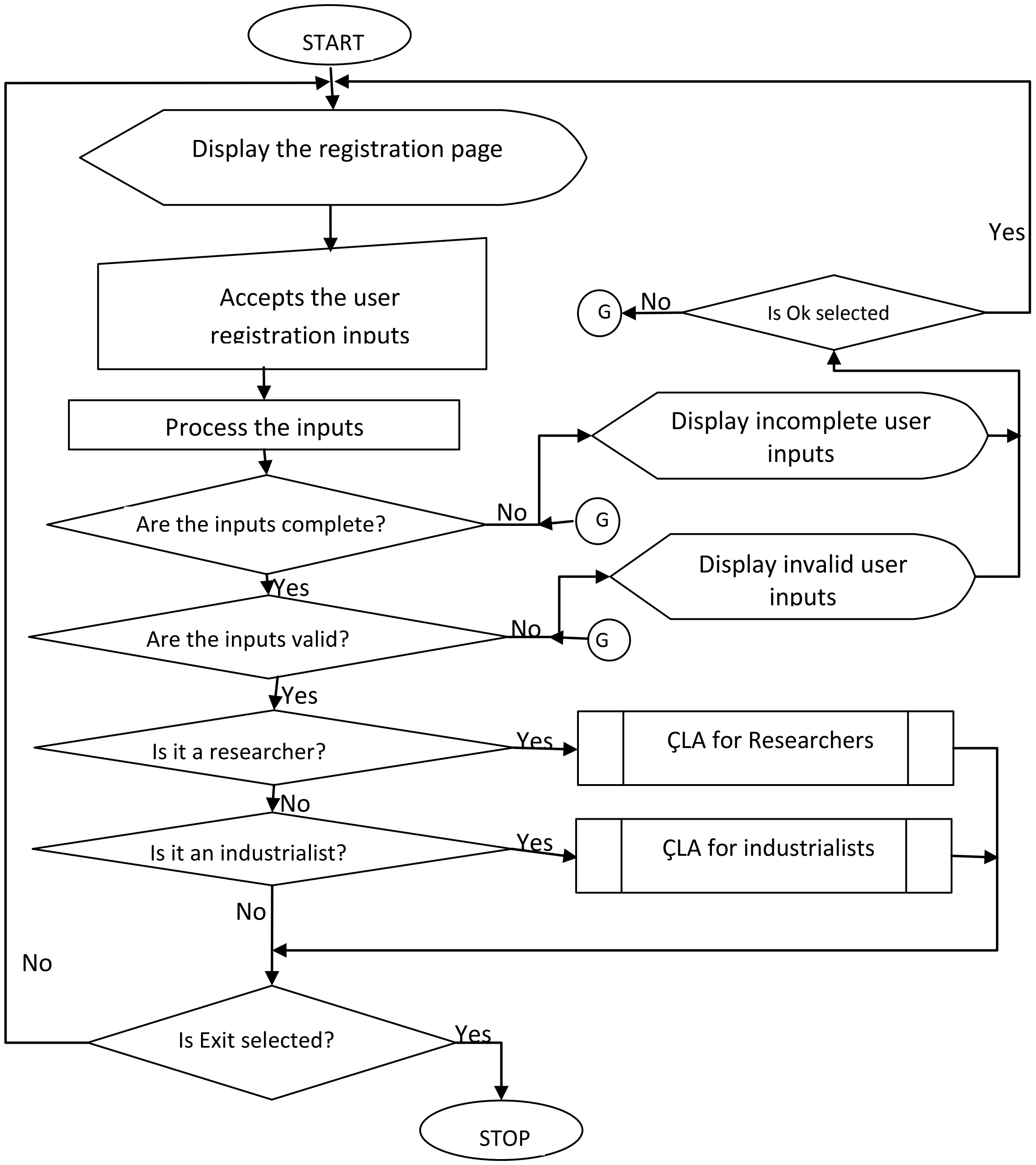


Figure 3.32 Flowchart for Registration sub-routine

Figure 3.32 is the registration sub-routine. Here the user registers either as a researcher or as an industrialist. The registration will not be completed until the user accepts the collaboration level agreement. This is for intellectual property rights protection.

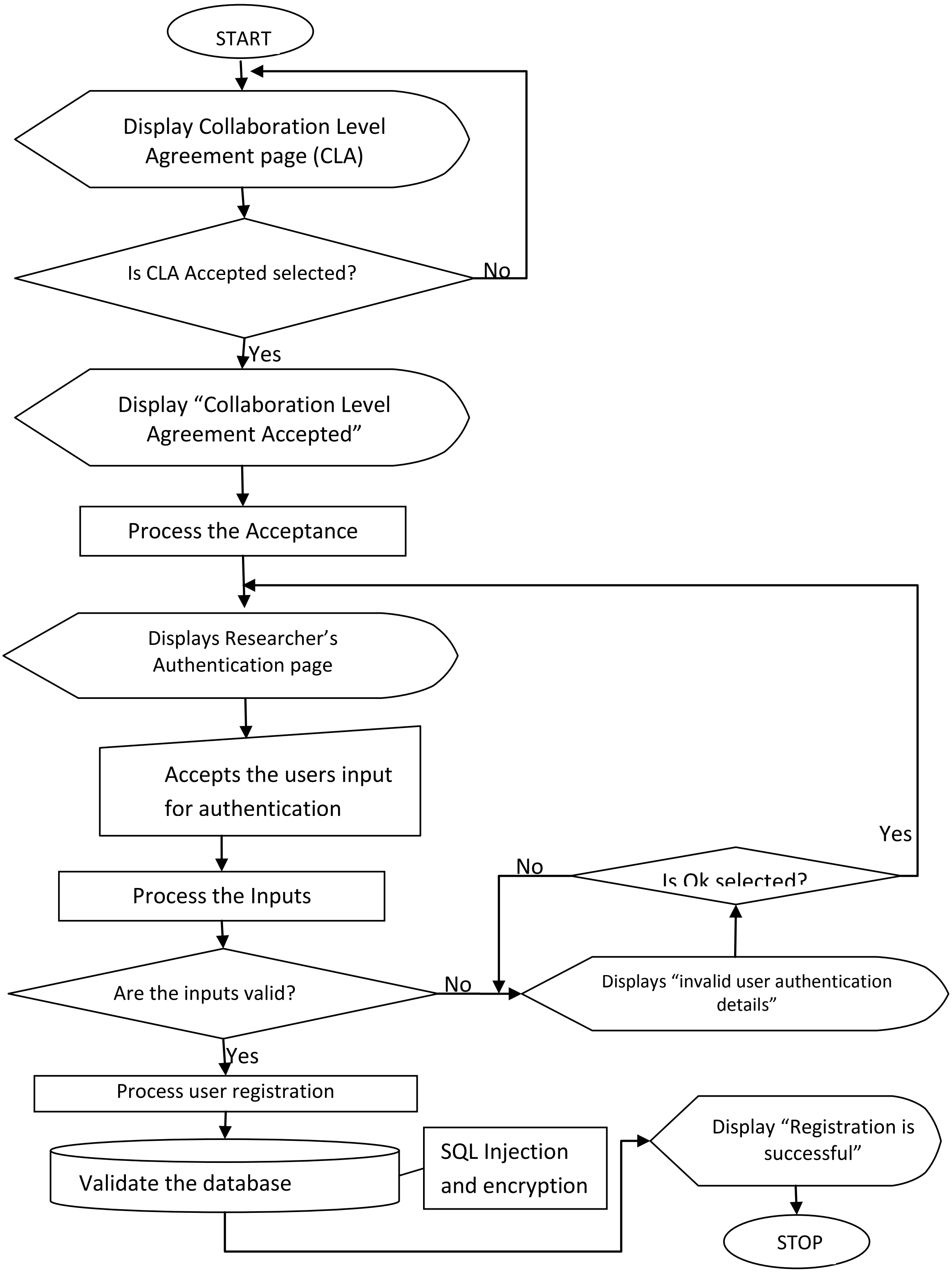


Figure 3.33 Collaboration level Agreement for researchers sub-routine

Figure 3.33 is the flow chart for the collaboration level agreement among researchers sub-routine. The researchers have to accept this agreement before they can collaborate. This is to ensure the protection of intellectual property rights. User registration is completed only when this agreement is accepted.

Security is also introduced in this part of the system. The user password is encrypted with the proposed security system before committing it to the database.

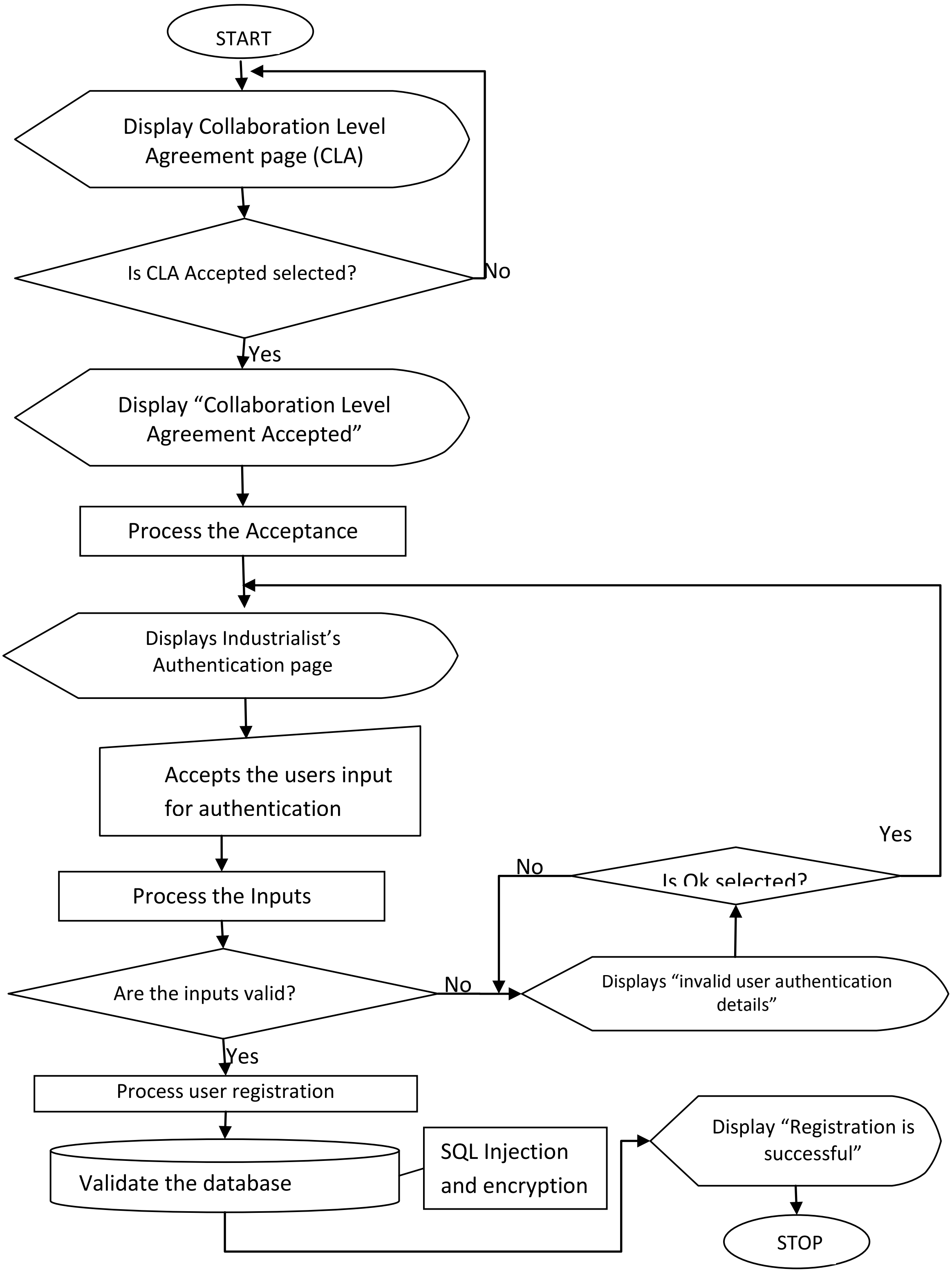


Figure 3.34 Collaboration level Agreement for industrialists sub-routine

Figure 3.34 is the flow chart for the collaboration level agreement for industrialists' sub-routine. This agreement must be accepted before they can collaborate. This is to ensure the protection of intellectual property rights. User registration is completed only when this agreement is accepted. Security is also introduced in this part of the system

3.8.5 Database Design

This work made use of RDBMS in its database design. The data is stored into different tables and relations are established using primary keys or other keys known as foreign keys. This work will make use of the existing database system provided by the cloud service provider for permanent storage of the data. Therefore the database design done here is for the temporary storage before linking it to the cloud. It will be implemented using MySql because it is a fast, easy-to-use RDBMS that uses a standard form of the well-known SQL data language. MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc. The database schema is shown in figure 3.35. It shows the organisation of data and consists of the tables that make up the relational database used by the platform. It serves as a plan showing how the crcmd database will be constructed. It shows the relationships between the different tables that make up the database. It represents the logical view of the database. It also defines how the data is organised and how the relations among them are associated. The entity-relationship model for the database is shown in figure 3.36. It shows the different entities, their attributes and their relationships. It is the graphical representation of the system that shows the relationships between people, objects, places, concepts or events within the system. It has three main components as seen in the diagram: entities which are objects or concepts that can have data stored about them, the relationships between those entities and the cardinality which defines that relationship in terms of numbers. Cardinality notation defines the attributes of the relationship between the entities.

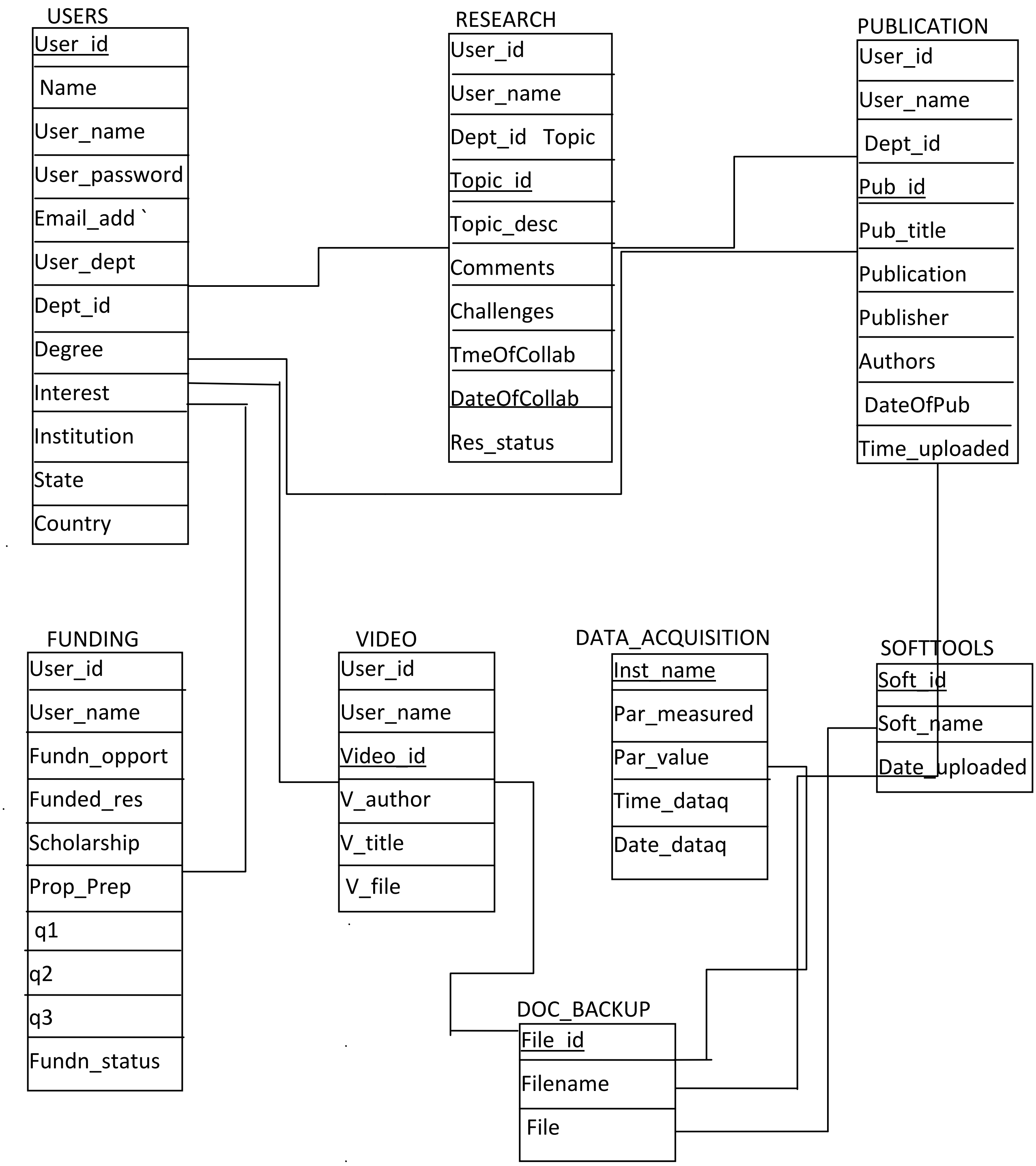


Figure 3.35: Schema diagram for the crcmd database

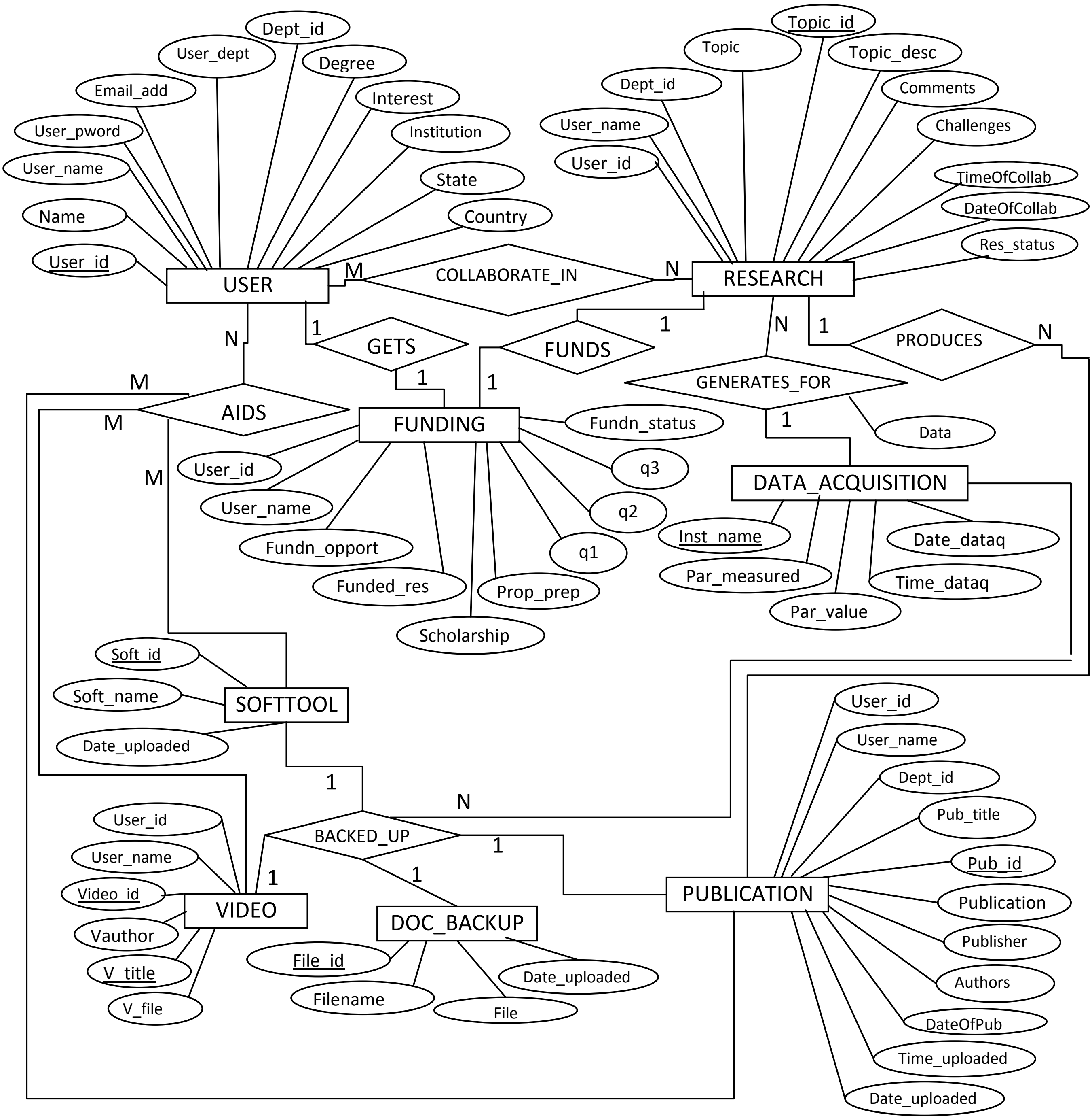


Figure 3.36: Entity-Relationship diagram for crcmd

The database for the proposed CRCM is named crcmd. It consists of 8 tables. It manages the CRCM data before transfer to the cloud. The tables in the crcmd database are:

- Users
- Research
- Publication

- Funding
- Softtool
- Video
- Doc_backup
- Data_acquisition

This section is on the design of the database tables.

3.8.5.1 Design of Users Database Table:

This table is used to store all the captured information about the registered users like their full names, usernames, password, email_add etc. The password collected in the platform is encrypted before storage here. The table contains twelve fields as shown in table 3.7. The table shows the different fields of the database table, the field type and the field length. The primary key is the User_id which is unique for every user.

Table 3.7: Users database table

Field_Name	Field_Type	Field_length
User_id (PK)	int	20
Name	varchar	50
User_name	varchar	50
User_password	varchar	50
Email_add	varchar	320
User_dept	varchar	100
Dept_id	varchar	50
Degree	varchar	10
Interest	memo	2000
Institution	text	200
State	varchar	50
Country	varchar	50

3.8.5.2 Design of Research database table

This table stores information on collaboration among researchers like the research topics, research challenges, discussions etc. It has eleven fields as shown in table 3.8. It also stores the status of the research, whether it is completed or ongoing. The Topic_id is the primary key for this table.

Table 3.8: Research database table

Field_Name	Field_Type	Field_Length
User_id	int	20
User_name	varchar	50
Dept_id	varchar	50
topic	text	200
Topic_id(PK)	int	20
Topic_desc	Text	500
Comments	Memo	20,000
Challenges	Memo	20,000
Time_of_collab	Time	
Date_of_collab	Date	
Research_status	varchar	10

3.8.5.3 Design of Publication database table

This stores the documents like published works, completed research reports etc uploaded to the system for use by collaborators. It has eleven fields as shown in table 3.9. The Pub_id is the primary key for this table.

Table 3.9: Publication database table

Field_Name	Field_Type	Field_Length
User_id	int	20
User_name	varchar	50
Dept_id(PK)	varchar	50
Pub_title	text	200
Publication	OLE object	1,280,000
Pub_id (PK)	int	20
Publisher	Text	200
Authors	Text	400
Date_of_pub	Date	
Time_uploaded	Time	
Date_uploaded	Date	

3.8.5.4 Design of Funding database table

This stores information about research funding, scholarships, proposal preparation, funding status etc. It has ten fields as shown in table 3.10.

Table 3.10: Funding database table

Field_Name	Field_Type	Field_Length
User_id	int	20
User_name	varchar	50
Fundn_opport	text	200
Funded_res	text	200
Scholarship	text	200
Prep_prop	Memo	64,000
q1	Boolean	
q2	Boolean	
q3	Boolean	
Fundn_status	varchar	50

3.8.5.5 Design of Softtools database table

This stores the software tools uploaded to the system for use by researchers. It has three fields as shown in table 3.11. The Soft_id is the primary key for this table.

Table 3.11: Softtools database table

Field_Name	Field_Type	Field_Length
Soft_id (PK)	varchar	50
Soft_name	varchar	200
Date_uploaded	date	

3.8.5.6 Design of Video database table

This stores the different research videos uploaded by researchers on the platform. It has six fields as shown in table 3.12. The video_id is the primary key.

Table 3.12: Video database table

Field_Name	Field_Type	Field_Length
User_id	int	20
User_name	varchar	50
Video_id (PK)	varchar	50
V_author	varchar	50
V_title	text	200
V_file	attachment	

3.8.5.7 Design of Doc_backup database table

This table is for the backup of the data generated in the platform. It has four fields as shown in figure 3.13. The primary key is the File_id.

Table 3.13: Doc_backup database table

Field_Name	Field_Type	Field_Length
File_id (PK)	int	20
File_name	varchar	50
File	attachment	
Date_uploaded	Date	

3.8.5.8 Design of the data_acquisition table

This table stores the data acquired from the research equipments/instruments at particular time intervals. It contains five fields as shown in table 3.14. For the purpose of this work, temperature readings are used as an example. This information can be retrieved by the researchers for use in their research work. The inst_name is the primary key.

Table 3.14: Data_acquisition database table

Field_Name	Field_type	Field_length
Inst_name (PK)	varchar	50
Par_measured	varchar	50
Par_value	varchar	20
Time_dataq	time	
Date_dataq	date	

The data dictionary for the crcmd database is shown in table 3.15.

Table 3.15: Data dictionary for crcmd database

Field_Name	Field_type	Field_length	Field_description
USERS			
User_id (PK)	int	20	Unique number identifying the each user
Name	varchar	50	Full name of the user
User_name	varchar	50	Username assigned to each user
User_password	varchar	50	User password
Email_add	varchar	320	Email of the user
User_dept	varchar	100	User’s discipline
Dept_id	varchar	50	Unique number identifying a discipline
Degree	varchar	10	Highest degree obtained
Interest	memo	2000	Area of interest in research
Institution	text	200	Institution the user is working in
State	varchar	50	User’s state of residence
Country	varchar	50	User’s country of residence
RESEARCH			
User_id	int	20	Unique number identifying the each user
User_name	varchar	50	Username assigned to each user
Dept_id	varchar	50	Unique number identifying a discipline
Topic	text	200	Topic of the research

			being carried out
Topic_id(PK)	int	20	Unique number identifying each research topic
Topic_desc	Text	500	Brief description of the research topic
Comments	Memo	20,000	Collaborators comments on the different research topics
Challenges	Memo	20,000	Challenges faced by researchers
Time_of_collab	Time		Time of collaboration
Date_of_collab	Date		Date of collaboration
Research_status	varchar	10	Status of the research. Whether completed or ongoing
PUBLICATION			
User_id	int	20	Unique number identifying the each user
User_name	varchar	50	Username assigned to each user
Dept_id	varchar	50	Unique number identifying a discipline
Pub_title	text	200	Title of publication
Publication	attachment	1,280,000	Publications uploaded
Pub_id (PK)	int	20	Unique number identifying each publication
Publisher	Text	200	Publisher
Authors	Text	400	Names of the authors of the publication
Date_of_pub	Date		Date of publication
Time_uploaded	Time		Time publication was uploaded
Date_uploaded	Date		Date publication was uploaded
FUNDING			
User_id	int	20	Unique number identifying the each user
User_name	varchar	50	Username assigned to each user
Fundn_opport	text	200	Funding opportunities
Funded_res	text	200	Funded research

Scholarship	text	200	Available scholarships
Prop_prep	Memo	64,000	Proposal preparation
q1	Boolean		Question 1 of survey for funding
q2	Boolean		Question 1 of survey for funding
q3	Boolean		Question 1 of survey for funding
Fundn_status	varchar	50	Status of the funding, whether approved, money released etc
SOFTTOOLS			
Soft_id (PK)	varchar	50	Unique id for each software uploaded
Soft_name	varchar	200	Name of the software
Date_uploaded	date		Date software was uploaded
VIDEO			
User_id	int		Unique number identifying each user
User_name	varchar		Username assigned to each user
Video_id (PK)	varchar		Unique number identifying each video
V_author	varchar		Name of the video author
V_title	text		Title of the video
V_file	attachment		Video file uploaded
DOC_BACKUP			
File_id (PK)	int	20	Unique number identifying each file
File_name	varchar	50	Name of the file
File	attachment		File to be backed up
Date_uploaded	Date		Date uploaded
DATA_ACQUISITION			
Inst_name (PK)	varchar	50	Name of instrument
Par_measured	varchar	50	Parameter being measured
Par_value	varchar	20	Value of the parameter measurement
Time_dataq	time		Time of data acquisition
Date_dataq	date		Date of data acquisition

To retrieve information from the database, the sql select statement is used. If the information about those collaborating on the research topic E-Research for instance is needed, the select statement below will give the required result

SELECT Username, topic,

FROM replies

WHERE topic = E-Research

This will display the names of researchers involved in the E-Research collaboration. To load data into a table, for instance the replies table we use this mysql command

mysql> LOAD DATA LOCAL INFILE '/path/challenges.txt' INTO TABLE replies;

3.8.6 Coding

Hypertext pre-processor (PHP) was chosen for the implementation of the system owing to its light weighted nature which is acceptable in cloud computing context. Also it is widely used for general purpose scripting and can be embedded into HTML. Owing to the presence of the apache server, the CRCM generally runs on a web browser which needs to be configured to process the PHP code. External modules (CMS, Opendoc, Real time chats, backup and recovery for the cloud compute storage, etc) are all unified using the PHP script to realise figure 3.3. Appendix C4 Shows the apache server scripting code listings developed for CRCM.

The design implementation showing the different graphical user interfaces for CRCM generated from the codes will be shown in chapter 4. The testing will also be done in chapter 4 and maintenance stage will be taken care of subsequently as the need arises.

3.9.1 CRCM Data Center Design

Cisco Systems – Nigeria based at Lagos, Nigeria assisted in the design specifications of the data centre model for the developed CRCM. In designing the simulation platform for the developed CRCM, this dissertation leveraged the data centre model of Cisco systems. The main goals are to maintain good resource utilization for the CRCM traffic on the IP cloud. Also, it offers low latency, fair resource allocation with good throughput. The framework in figure 3.37 depicts the final CRCM infrastructural components which will be implemented in this chapter. The implementation was carried out via the simulation in figures 3.38 and 4.1 depicting the characteristics of figure 3.3 and figure 3.7. Figure 2.1 and figure 2.2 fits into this simulation as well.

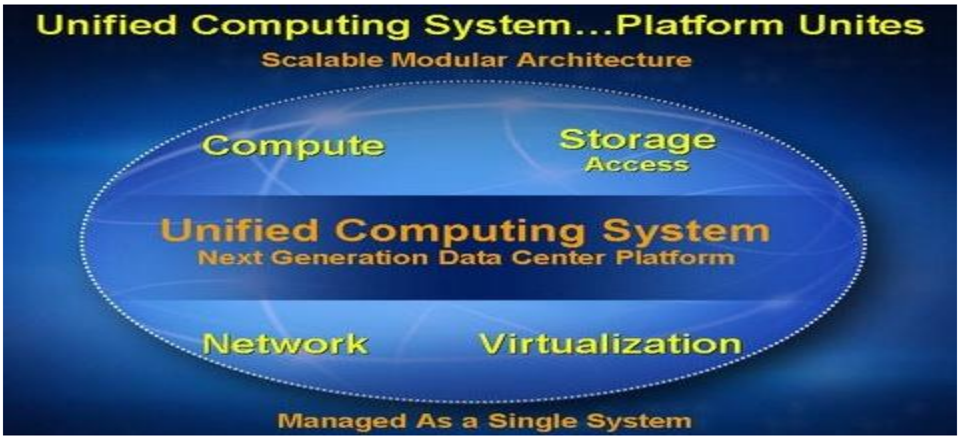


Figure 3.37: CRCM Infrastructural layout (Cisco, 2008).

The layers of the data center design are the core, aggregation, and access layers. This research used a multi-tier model which is an efficient design that runs cloud computing applications. It has the capacity for database layered design supporting various interfaces such as enterprise business ERP and CRM solutions from Siebel and Oracle, etc. It relies on security and application optimization services to be provided in the network.

Also, it facilitates high-performance computing (HPC), parallel computing, and high-throughput computing (HTC) environments, but can also be associated with grid/utility computing. This design is typically based on customized or proprietary application architecture for CRCM. Since, the multi-tier data centre model is dominated by HTTP-based applications, the simulation design runs separate processes on the same machine as a server comprising of the following three tiers:

- Web-server
- Application
- Database

3.9.2 Simulation of Blade Server Systems and Intelligent Services

The Blade server systems in this work are the key component of data centre consolidation that help reduce costs and improve virtualization, automation, and provisioning capabilities. Also, blade system vendors offer high availability, scalability, and security services such as load balancing, firewalls, and intrusion detection to enterprise applications. Depending on the vendor, these services are made available through the blade system integrated network devices or individual blade servers using service specific software.

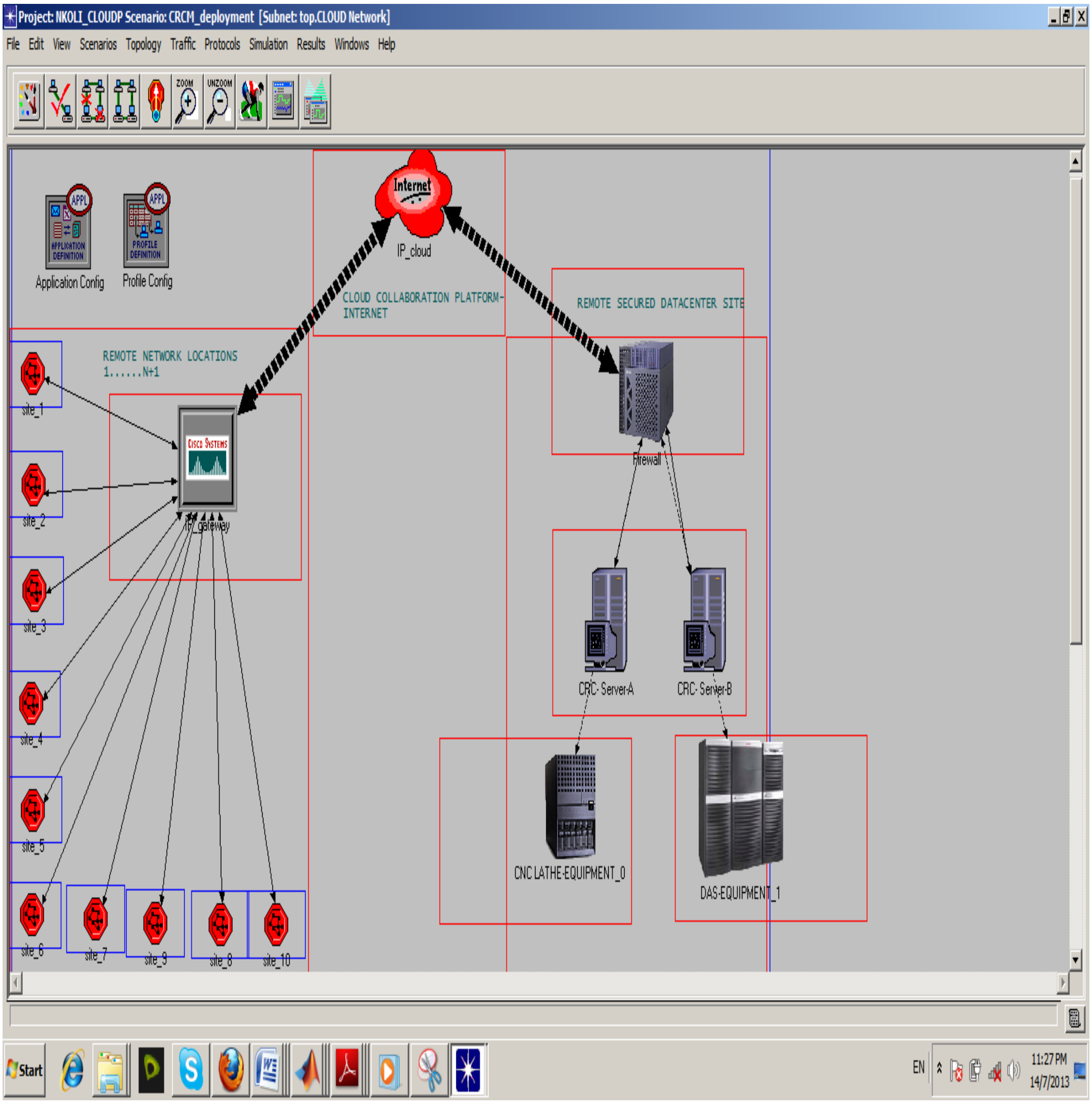


Figure 3.38: CRCM Data centre Simulation Test bed for CRCM HTTP Service Validation

In Figure 3.38 the server shows a service- enabled blade system. In this work, both software and hardware based firewalls are located on two of the blade servers functioning in their chassis. In this work two equipments are integrated to the system, connecting to the server for research/experimental data acquisition.

CHAPTER 4

RESULTS AND DISCUSSION

This chapter incorporates CRCM design implementation and system integration, validation of the queuing model using OPnet, results, discussion of the results and testing. The CRCM implementation also involves the cloud deployment. In this chapter, a scenario for remote access of users on the CRCM is presented in the context of the leveraged internet infrastructure. This is intended to further explain the traffic engineering on the network platform while observing its overall performance. While the model in Chapter 3 explained the cloud queuing behaviour for efficient resource management, the simulation design in this section explains the remote access connectivity into the cloud and the intended data centre for hosting CRCM in future.

4.1 OPNET IT GURU Simulation

OPNET IT Guru Simulation software was used with the http service in the OPNET engine to generate script files for traffic analysis in the validation discussion presented in this work.

The cloud application is collapsed into the CRCM http service in the application configuration palette in figure 3.38 while all the databases run from the blade server in the simulation setup. Only one CRCM scenario was used after configuring the network parameters to setup the environment. Connectivity tests were carried out before executing the simulation runs. This section defines the simulation framework on which the recommended data centre architecture for CRCM is based. The simulation parameters are shown in Table 4.1.

Table 4.1: CRCM OPNET Simulation Parameters

	CRCM Specifications	Attributes/Values
1	Blade Servers	2
2.	Service	HTTP, DBMS, FTP
3	Server CPU (Sun ultra 10 333Hz)	Logical processor / multiprocessor
4	Server modeling	Simple CPU
5	Transmission Control Protocol	Solaris 2.9 (2)
6	Maximum Translation Unit	1500 bytes
7	Firewall gateway	Sun Enterprise 450
8	Resource Reservation Protocol Interface	Enabled
9	Refresh interval	30secs
10	Internet cloud	WAN Default Protocols (OSPF& MPLS)
11	No of Remote clients	200
12	Data center 3-tier	Enabled
13	CRCM-HTTP service	Active

Source: Cisco systems Lagos, Nigeria

4.2 Results and Discussion

In this work, the validation of CRCM HTTP service involved the use of the design parameters obtained from Data centre vendor (Cisco systems) which was then deployed in developing a generic Data centre template in OPNET IT Guru. Based on the measurement parameters configured on the test bed, some selected metrics were then used for performance evaluations of CRCM in comparison with the cloud queuing model developed in chapter 3 (see figure 3.8) . On the generic OPNET template shown in figure 4.1, ten user location sites were created for CRCM, with each site having twenty users. The ten sites connect to the IP cloud via the Cisco IP gateway. On the other remote end, a firewall and the two servers with its external interfaces were configured. For these scenarios, the attributes Data centre architecture were configured on the template while importing the CRCM http service into the template and the simulation was run.

Afterwards, the OPNET engine generates the respective data for each of the QoS investigated in this work.

Essentially, in most of the plots generated, the OPNET based CRCM responses followed similar trend with the plots of figure 3.12 to 3.16, hence justifying the resource management and service efficiency of figure 3.3 in a real life production environment.

The Http traffic condition with respect to queuing jobs employed in figure 4.1 inherits from the discussion in chapter 3. Though the CRCM Data centre have its feedback mechanisms, server virtualization, load-balancing, link state management, and fault tolerance schemes in its design philosophy, the validation experiment seeks to show the effectiveness of the proposed system in terms of throughput, queuing management (network capacity), server utilization, and latency effects of the proposed CRCM Data centre for remote access.

Before the simulation, link consistence tests were carried out randomly to ascertain any possibility of failure in all cases. Randomly selected nodes and servers routes packets in a bidirectional way from the access layer to the core layer and vice versa. In context, an investigation on both the path failure ratio and the average path length for the found paths was carried out and all the links were found satisfactory. In figure 4.1, the plots show path node consistence test based on the animation packet flow. All links and nodes passed the test in all cases. Figure 4.2b shows OPNET screenshot for simulation sequence used in the analysis.

In all the simulations, the essential attributes for each of the data centre network scenarios on the template were enabled beginning the simulation at 08:11:12 Mon Jul 15, 2013. The Simulation completed successfully and the results were collated in the global and object statistics reports.

The simulation gave:

- total events: 1526067,
- Average Speed: 141696 events/sec,
- Time Elapsed:11seconds
- Simulated: 1 hr. 0 min. 0 sec and
- Simulation Log: 3781 entries.

The simulation plots of the CRCM Data centre architecture with Http service under study are shown in the plots from figures 4.2, to figure 4.6 while Tables 4.2 to 4.6 shows the OPNET Data generation for CRCM Qos Metrics.

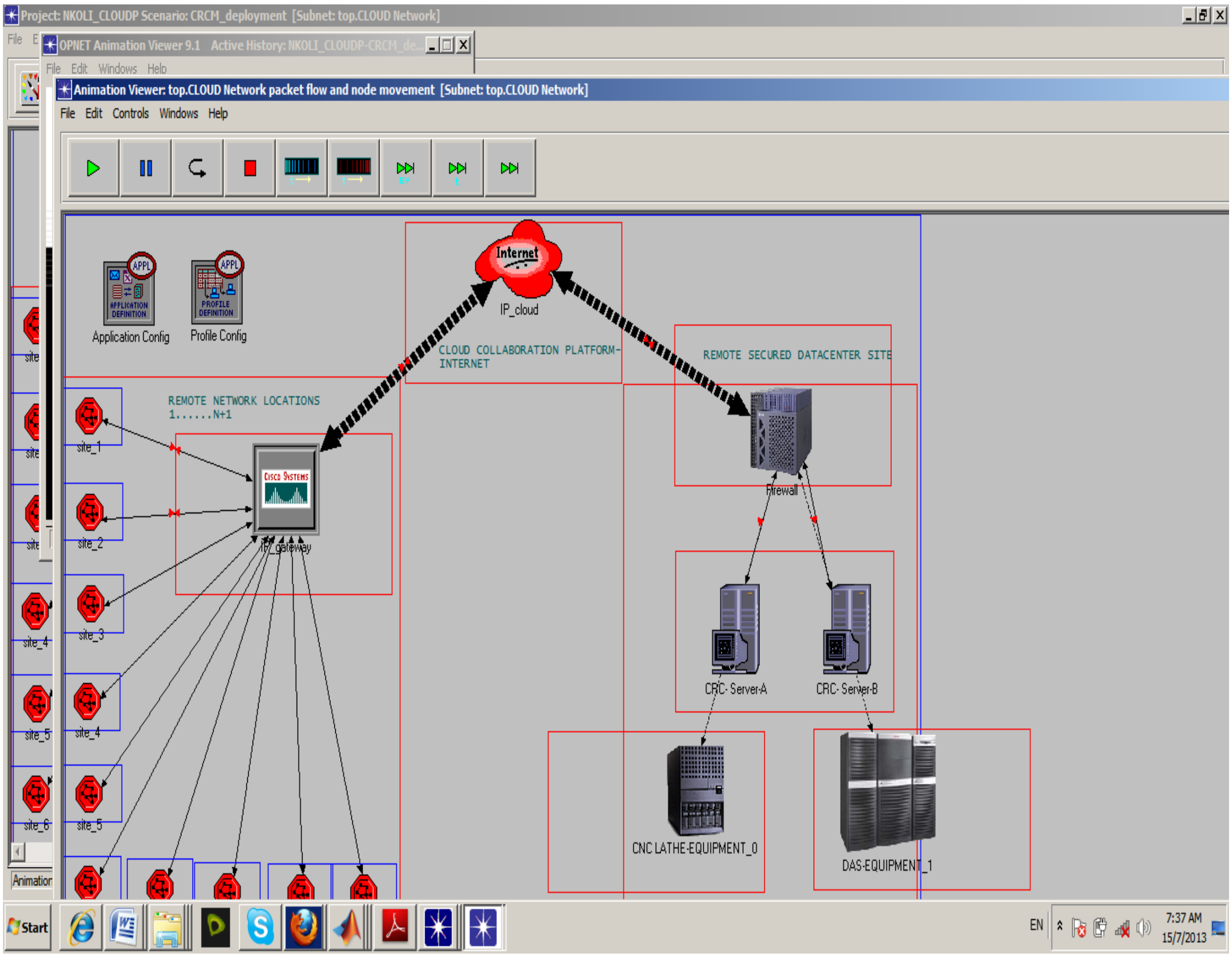


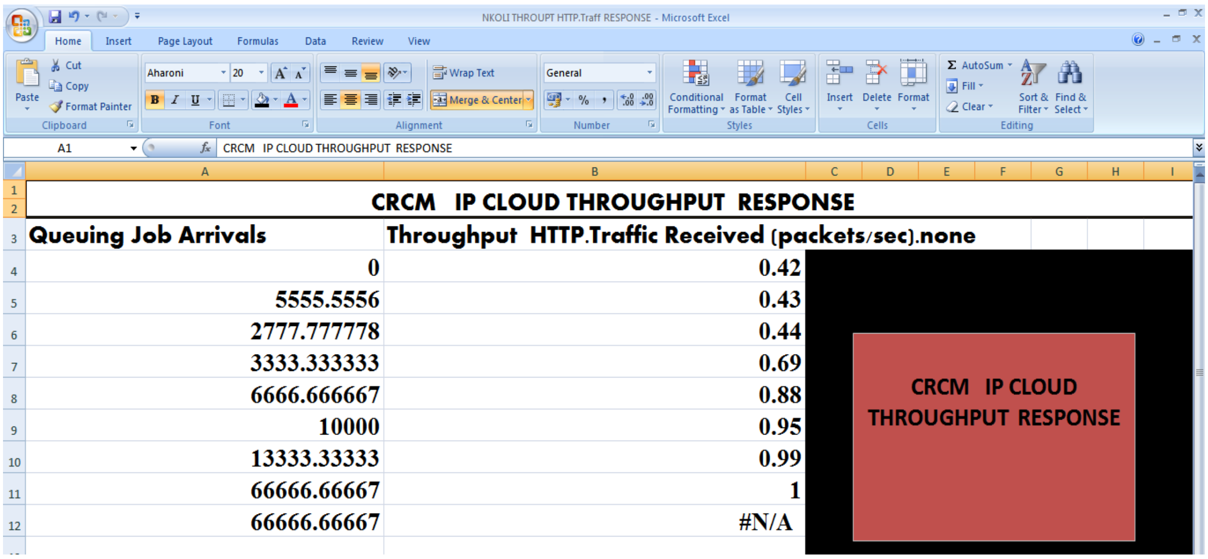
Figure 4.1: OPNET Screenshot for the Simulation Packet Flows (Consistency Test)

4.2.1 Throughput Response of CRCM Data centre

When a traffic condition is present, throughput being the data quantity transmitted correctly starting from the source to the destination within a specified time (seconds) is quantified with the arrival of queuing jobs between nodes/terminal devices, between the access layer and the core layer and more importantly at the core layer of the CRCM data centre. During the simulation, throughput as a global statistics was measured and compared. Table 4.2 is the data for the plot of figure 4.2. Figure 4.2 shows the average throughput as achieved in the simulation. From figure 4.2, the CRCM http traffic in the DCN had an initial interesting throughput response which was sustained by maintaining a very stable throughput response at 99% or 0.99 bytes/secs. This is evidenced as the arrivals traffic pattern follows the same trend with figure 3.12. The main reason for this is that the routing protocols load balances between the user and the server layers in the core. However, the network bandwidth and the topological layout of a DCN usually affect the throughput as well.

Again, this work attributes this observation to the fact that the network topology is communicating on the basis of reduced policy implementation making for efficiency and as such the total http traffic received is about 99% with lesser packet drops which could likely occur.

Table 4.2: CRCM throughput Response for Queuing Arrivals



CRCM IP CLOUD THROUGHPUT RESPONSE	
Queuing Job Arrivals	Throughput HTTP.Traffic Received (packets/sec).none
0	0.42
5555.5556	0.43
2777.777778	0.44
3333.333333	0.69
6666.666667	0.88
10000	0.95
13333.33333	0.99
66666.66667	1
66666.66667	#N/A

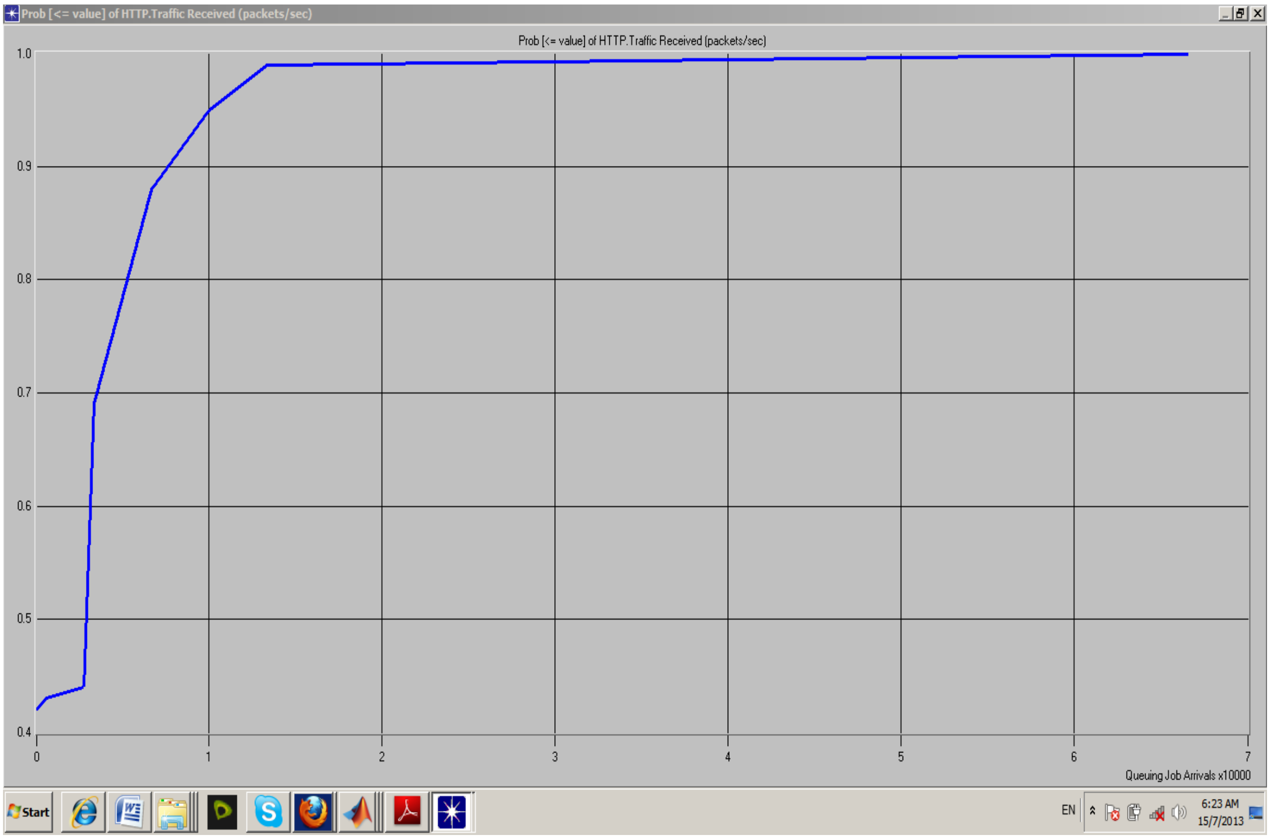


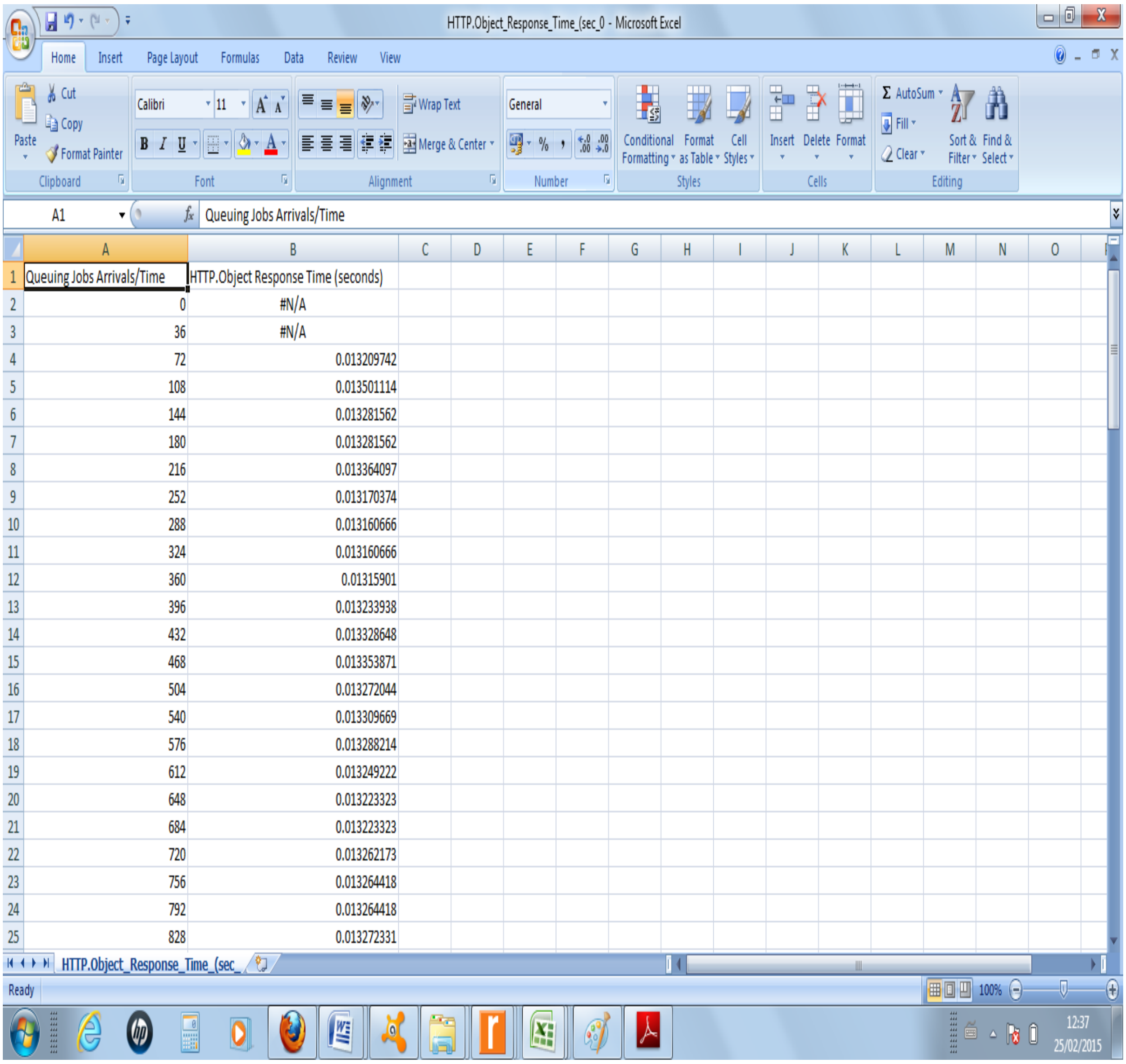
Figure 4.2: CRCM throughput Response for Queuing Arrivals

4.2.2 Latency Effects on Http pages

Page response time is the time it takes each web page to load. The response time is the total amount of time the Http traffic spends in both the queue and in service. An analysis of the network behaviour with respect to the generic scenario was carried out. Table 4.3 and 4.4 shows the data for the plot of figure 4.3. Figure 4.3shows the end-to-end latency result of the Http traffic on the CRCM three-tier topology. As depicted in the plots, the latency or page response time shows that for CRCM Http traffic, the page response time is 0.0323secs while object response time is 0.0128secs, which are much lower compared with existing web applications. For instance, according to Maggio, Klein & Arzen (2014), the average page response is about 0.3secs while Wang et al (2012) proposed an average page response time of 0.068secs. From the work (Kumar and Khurana, 2014). which shows a comparison of the page response and object response time of some of the task scheduling algorithms analysed in this work, we can conclude that the queuing model proposed in this work is better in terms of page and object response times. The reason for this is that in the Data centre model, traffic optimization is enabled by turning most device parameters off, such as routing algorithm which reduces the transmission time between the access and the core layer even when the links are busty and busy.

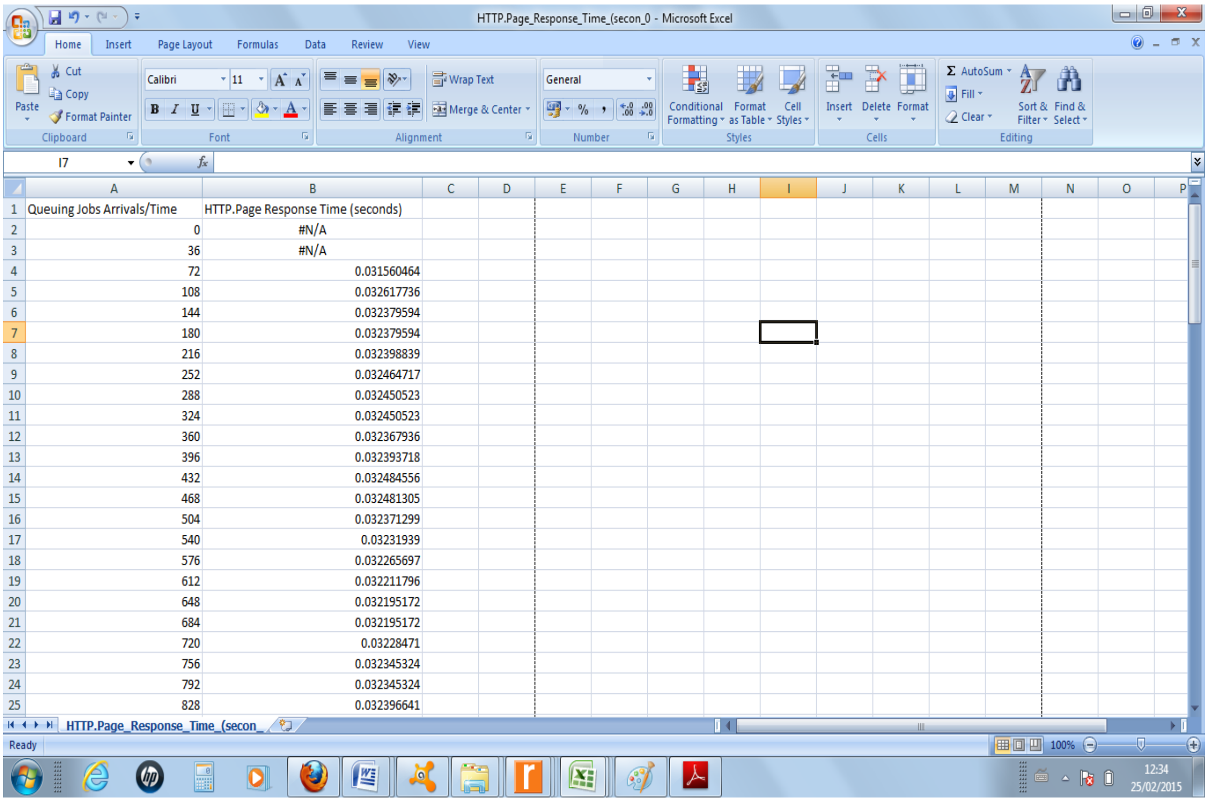
The core layer is highly redundant with little routing policy in the DCN and as such can easily take packets from the access layer with very little wait states

Table 4.3: Object Response Time



Queuing Jobs Arrivals/Time	HTTP.Object Response Time (seconds)
0	#N/A
36	#N/A
72	0.013209742
108	0.013501114
144	0.013281562
180	0.013281562
216	0.013364097
252	0.013170374
288	0.013160666
324	0.013160666
360	0.01315901
396	0.013233938
432	0.013328648
468	0.013353871
504	0.013272044
540	0.013309669
576	0.013288214
612	0.013249222
648	0.013223323
684	0.013223323
720	0.013262173
756	0.013264418
792	0.013264418
828	0.013272331

Table 4.4: Page Response Time



Queuing Jobs Arrivals/Time	HTTP.Page Response Time (seconds)
0	#N/A
36	#N/A
72	0.031560464
108	0.032617736
144	0.032379594
180	0.032379594
216	0.032398839
252	0.032464717
288	0.032450523
324	0.032450523
360	0.032367936
396	0.032393718
432	0.032484556
468	0.032481305
504	0.032371299
540	0.03231939
576	0.032265697
612	0.032211796
648	0.032195172
684	0.032195172
720	0.03228471
756	0.032345324
792	0.032345324
828	0.032396641

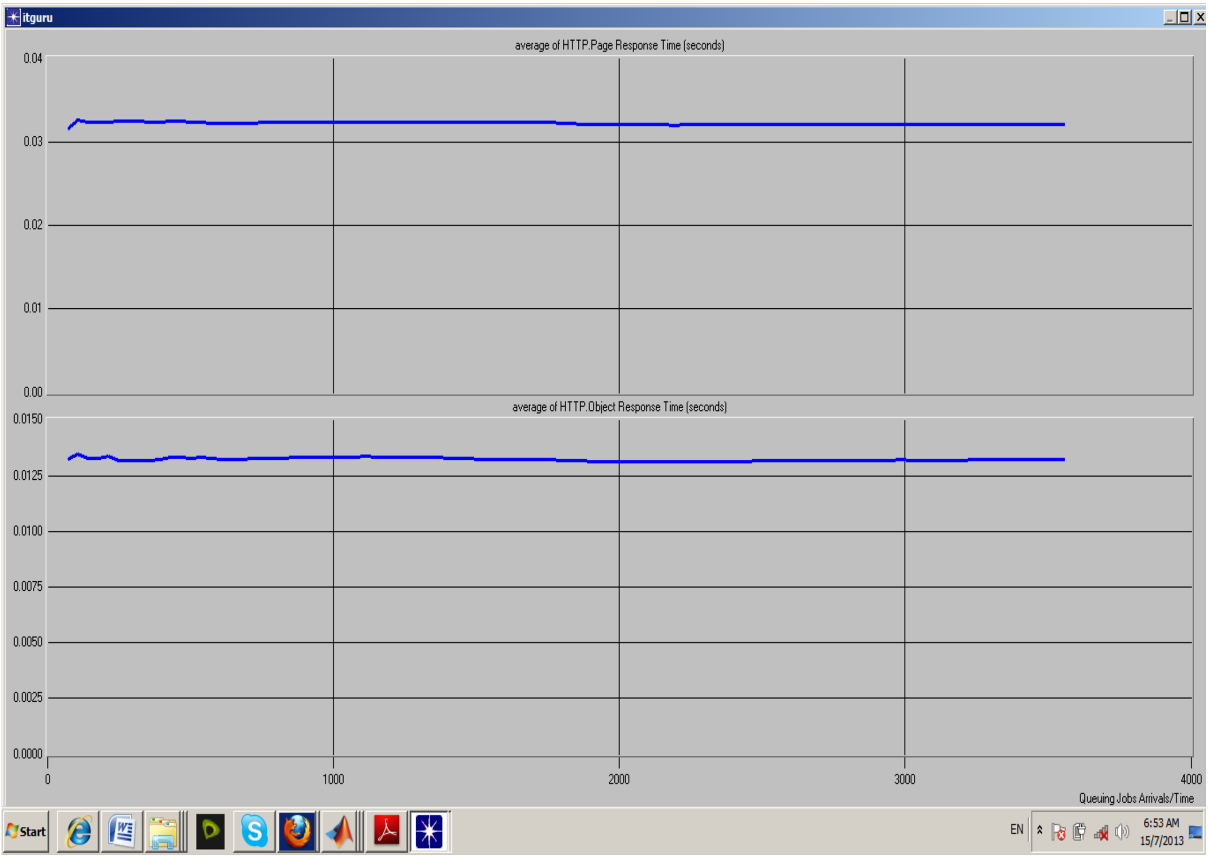


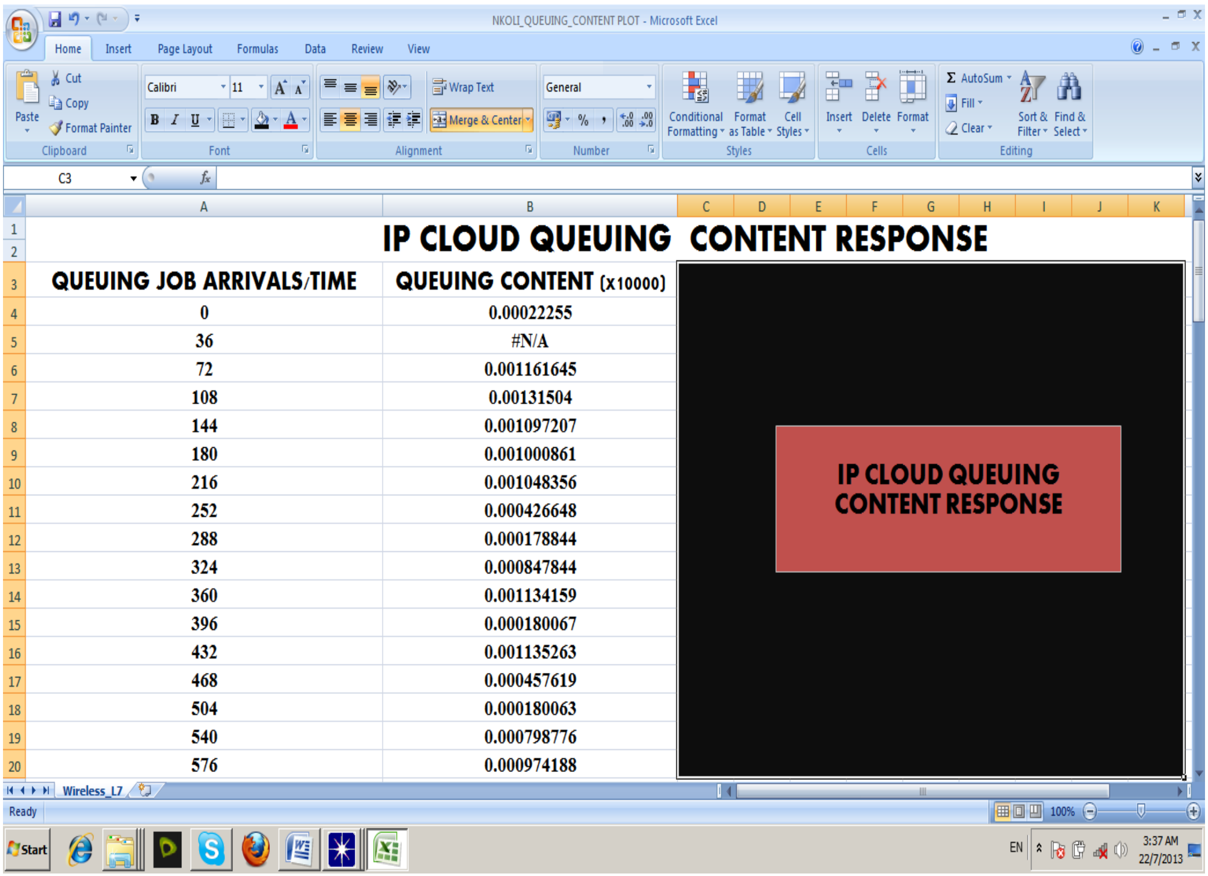
Figure 4.3: Page Response and Object Response Time

4.2.3 CRCM Queuing Content Evaluation

The CRCM Http service was configured in the Data centre platform to ascertain its queuing discipline. The notation G/G/1 queue which is usually referred to as generalized single-server queue with First-in-First-out discipline with a general distribution of the sequences of inter-arrival and service times (which are the driving sequences of the system) was observed.

The arrival jobs are modulated by the admission control module (Resource reservation Protocol-RSVP) making for the oscillation in figure 4.4. The pattern of this plot is similar to the plot in figure 3.10 previously shown. The maximum threshold capacity is about 12.5 which is envisaged to offer efficiency in the collaboration platform. The implication of the figure 4.4 is that jobs will be dispatched as long as the service rate is greater that the arrival rates making for system stability.

Table 4.5: IP Cloud Queuing Content Response



IP CLOUD QUEUING CONTENT RESPONSE

QUEUING JOB ARRIVALS/TIME	QUEUING CONTENT (x10000)
0	0.00022255
36	#N/A
72	0.001161645
108	0.00131504
144	0.001097207
180	0.001000861
216	0.001048356
252	0.000426648
288	0.000178844
324	0.000847844
360	0.001134159
396	0.000180067
432	0.001135263
468	0.000457619
504	0.000180063
540	0.000798776
576	0.000974188

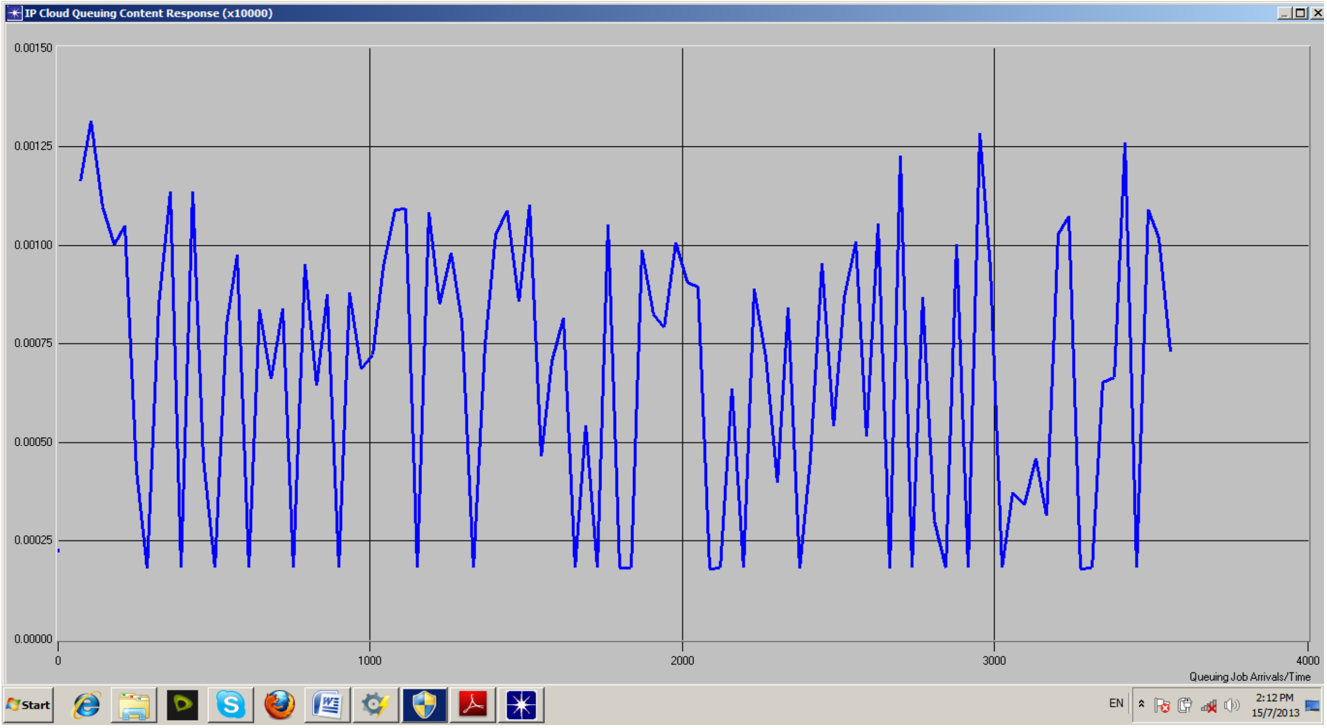
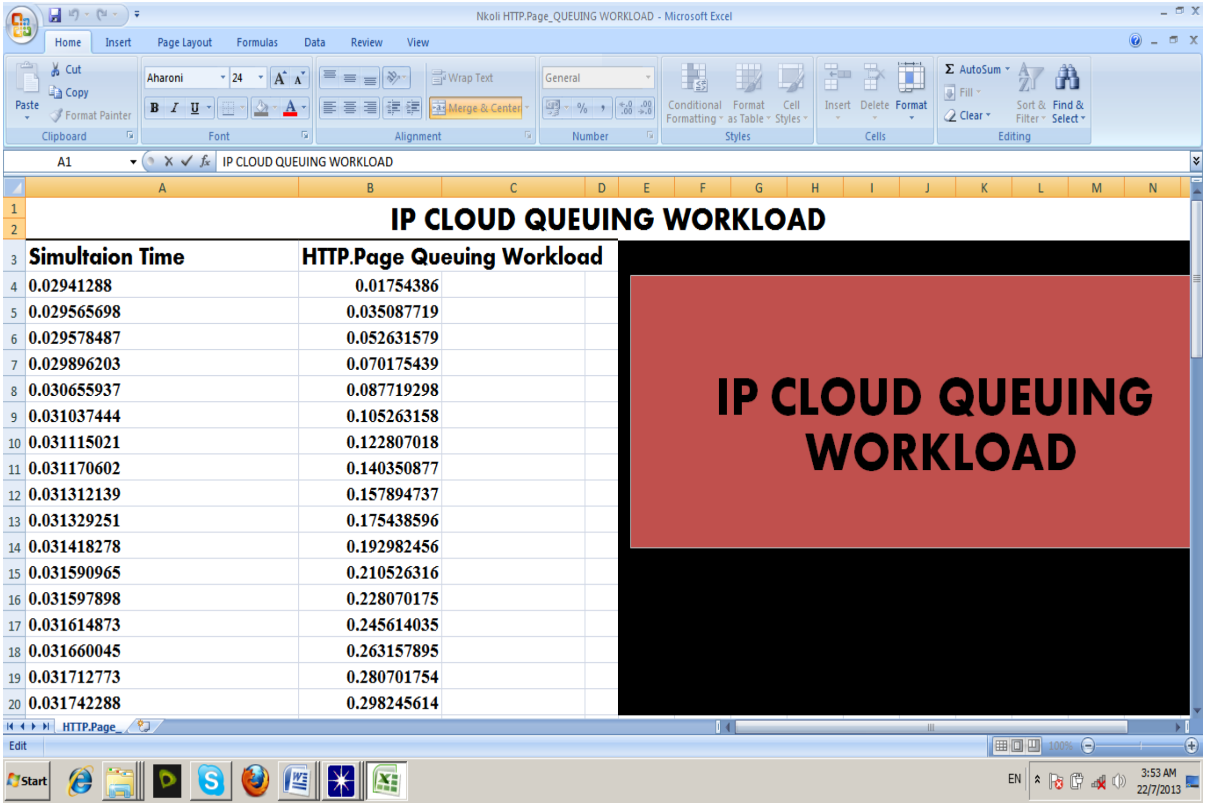


Figure 4.4: IP Cloud Queuing Content Response

4.2.4 CRCM HTTP Queuing Workload

In the CRCM Data centre, Http message queues provide an asynchronous communications protocol, meaning that the sender and receiver of the message do not need to interact with the message queue at the same time. Messages placed onto the queue are stored until the recipient server retrieves them. Message queues have implicit or explicit limits on the size in any workload scenario. It is worthy to note that for the queuing workload, arrivals still occur at rate A according to a Poisson process and move the process from state i to i + 1.The service times have an exponential distribution as depicted in figure 4.5. Since the buffer size can be approximated to infinity in the model assumption, there is therefore no limit to the number of jobs it can contain. The plot shown in figure 4.5 demonstrates that the CRCM will scale efficiently in direct proportion with the incoming job queues or arrivals. This clearly validates figure 3.14 which captures cloud queuing workload or nominal throughput Response.

Table 4.6: Http Queuing Workload



IP CLOUD QUEUING WORKLOAD

Simultaion Time	HTTP.Page Queuing Workload
0.02941288	0.01754386
0.029565698	0.035087719
0.029578487	0.052631579
0.029896203	0.070175439
0.030655937	0.087719298
0.031037444	0.105263158
0.031115021	0.122807018
0.031170602	0.140350877
0.031312139	0.157894737
0.031329251	0.175438596
0.031418278	0.192982456
0.031590965	0.210526316
0.031597898	0.228070175
0.031614873	0.245614035
0.031660045	0.263157895
0.031712773	0.280701754
0.031742288	0.298245614

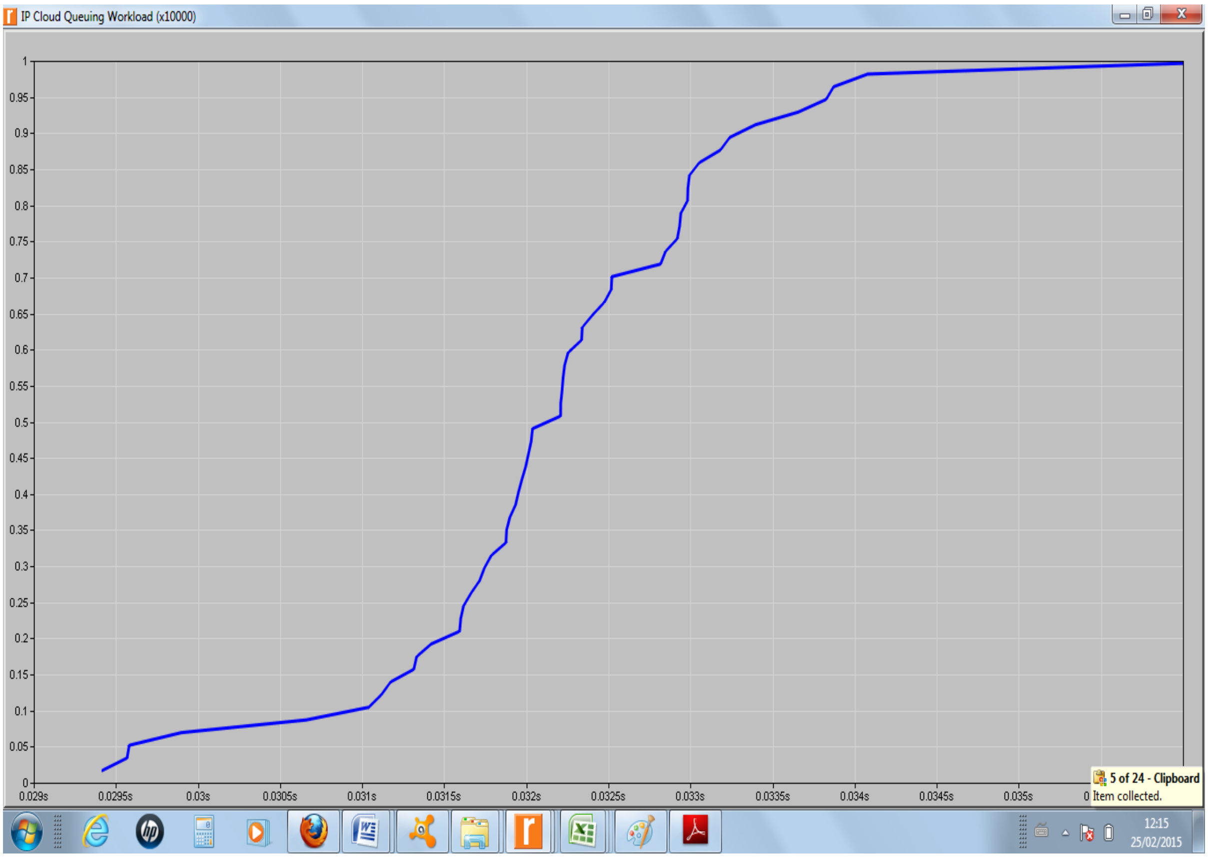


Figure 4.5: Http Queuing Workload

4.2.5 Server Utilization Response

Figure 4.6 shows the CRCM Server Utilization Response plots. The implication of figure 4.6 is that it addresses resource utilization responses elicited from the DCN core in the design. Generally, in DCN environment, latency, throughput, server resources utilization, and link bandwidth are vital resources considered in congestion management for collaboration platforms. With the traffic load sources in the CRCM topology used in this research, an initial gradient was established before resource allocations were fairly distributed. With a connection request, feasible regions of resource allocation are first established. Resource utilization in the CRCM is very low from the beginning while maintaining a relatively stable state till the end. This implies a fairly uniform resource utilisation by the server. The server utilization is 0.99 which shows a similar trend to server utilization shown in figure 3.16 which has 0.9094, hence validating the CRCM queuing design. This is a major improvement compared to the report of the work (Snyder, 2010) which shows that server utilization is about 18% and the work (Silpa, and Basha, 2013) which gave a server utilisation of 83% when the number of requests was 20.

Table 4.7: Server Utilisation Response

Time (Secs)	Ftp Server Utilization Response Time (sec)
0.027012027	0.076923077
0.027601992	0.153846154
0.02777625	0.230769231
0.027995833	0.307692308
0.028058168	0.384615385
0.028268281	0.461538462
0.028274975	0.538461538
0.028314684	0.615384615
0.02869237	0.692307692

0.028883772	0.769230769
0.029759251	0.846153846
0.030208236	0.923076923
0.032433467	1

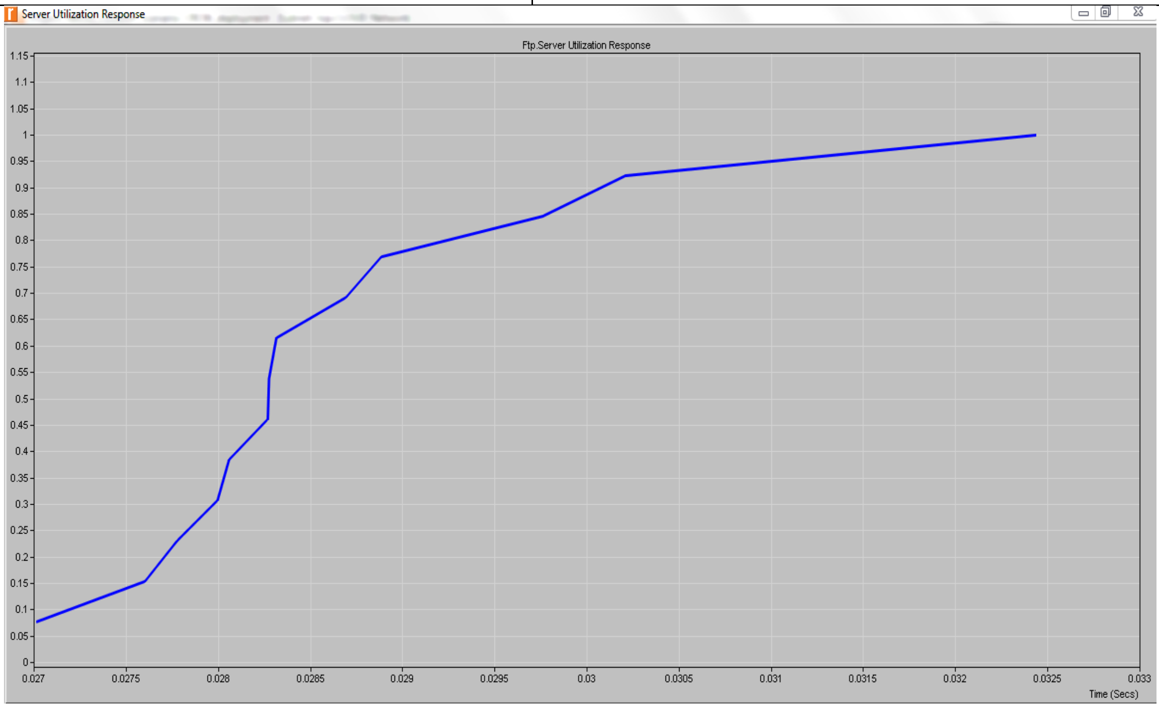


Figure 4.6: CRCM Server Utilization Response plot

4.3 Implication of CRCM on Service Availability

In this research, service availability is a function of network connectivity with fault-tolerance at all layers of the CRCM Data centre design. In the simulation design, Http service deployed via the application configuration palette shown in figure 4.1 establishes a TCP connection between servers at the core layer and the users at access layer in the setup. In addition, other network services including databases, E-mailing, web browsing, FTP, and other TCP & UDP services were introduced in figure 4.1. To study the performance under these configurations, the discrete event simulation was manually initiated in the simulation panel. It was observed that the TCP traffic was maintained while showing the service availability packet trend in figure 4.1.

To further validate the metrics adopted for the CRCM internet Data centre developed in figure 4.1, this work used the trace files and the CRCM http.php scripts in the OPNET environment to present the research findings shown from figures 4.2 to figure 4.6.

4.4 CRCM Prototype Platform Implementation

The prototype cloud-based research collaboration platform was designed to be browser independent. Figures 4.7 to 4.19 are snapshots of some of the pages of the CRCM.

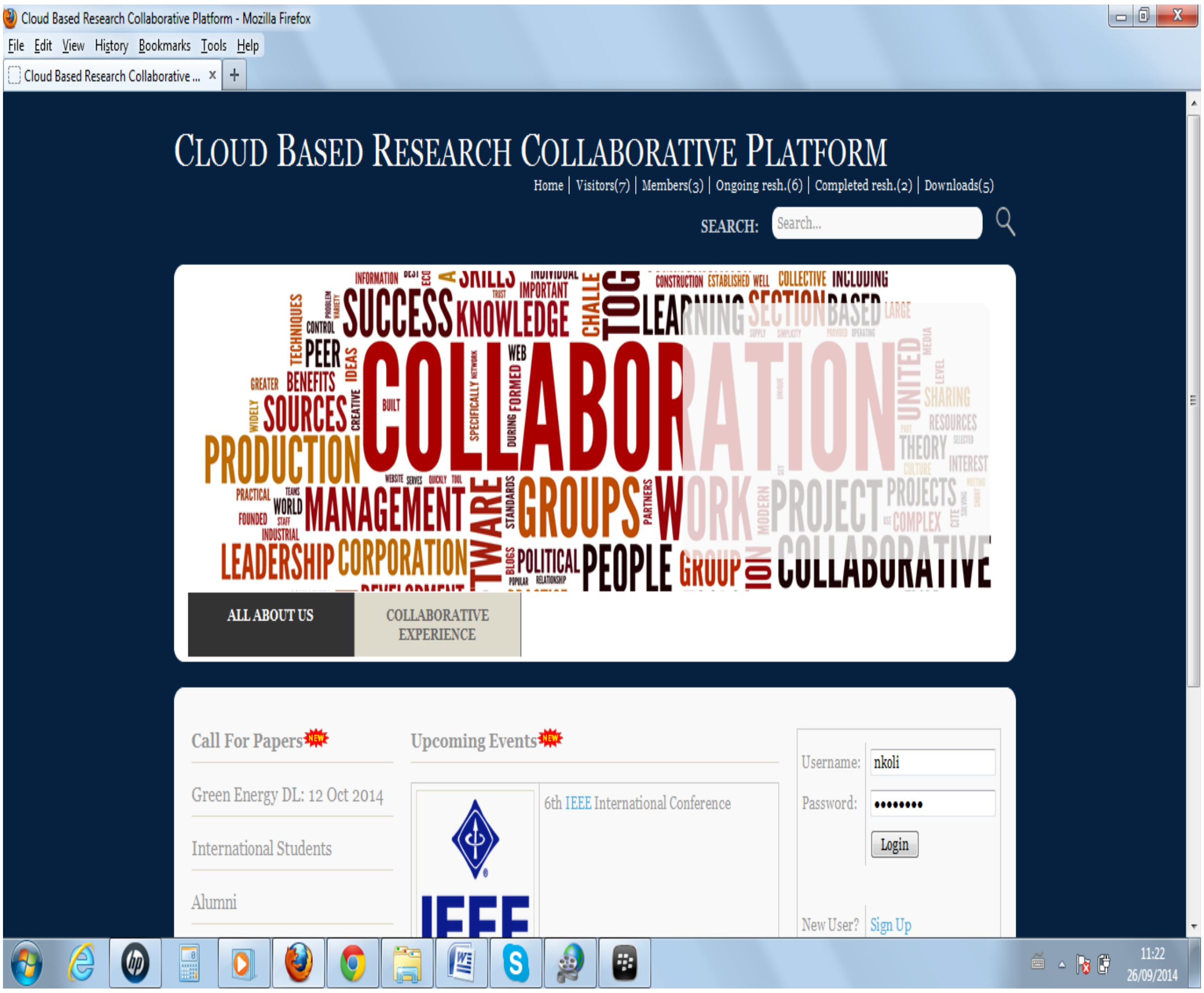


Figure 4.7 Log in interface

Figure 4.7 is the interface for the users' log in. if the user is not registered; it prompts the user to register before access can be granted to him/her. This is to ensure that only authenticated users are given access to the system. This page also features advertisements like call for papers, conferences, news, scholarships and available research topics. If the user successfully logs in to the application, the page shown in figure 4.8 is displayed for the user to select an activity

Figure 4.8: CRCM Activity selection page

Figure 4.9: Researchers' collaboration page

The interface shown in figure 4.9 is for collaboration among researchers. It is from this page that you navigate to the project forums where the researcher can create and close forums on different research interests. Interested researchers can also join and contribute in those forums. Before partaking in the collaboration, a collaboration level agreement must be signed by each participant. This is to ensure proper intellectual property ownership protection. The researchers can also share research videos here. The user can also view research positions, challenges and ongoing research activities from this page. Software Project tools can also be accessed here. Finally, it is from this section that the data acquisition system or data from virtual laboratories can be accessed as shown in figure 4.10. The research data shared here can be in textual form, numerical, audio, graphical or video form. It is through this interface that team connections are done, project team messages are exchanged among team members, and project moderation is also carried out.

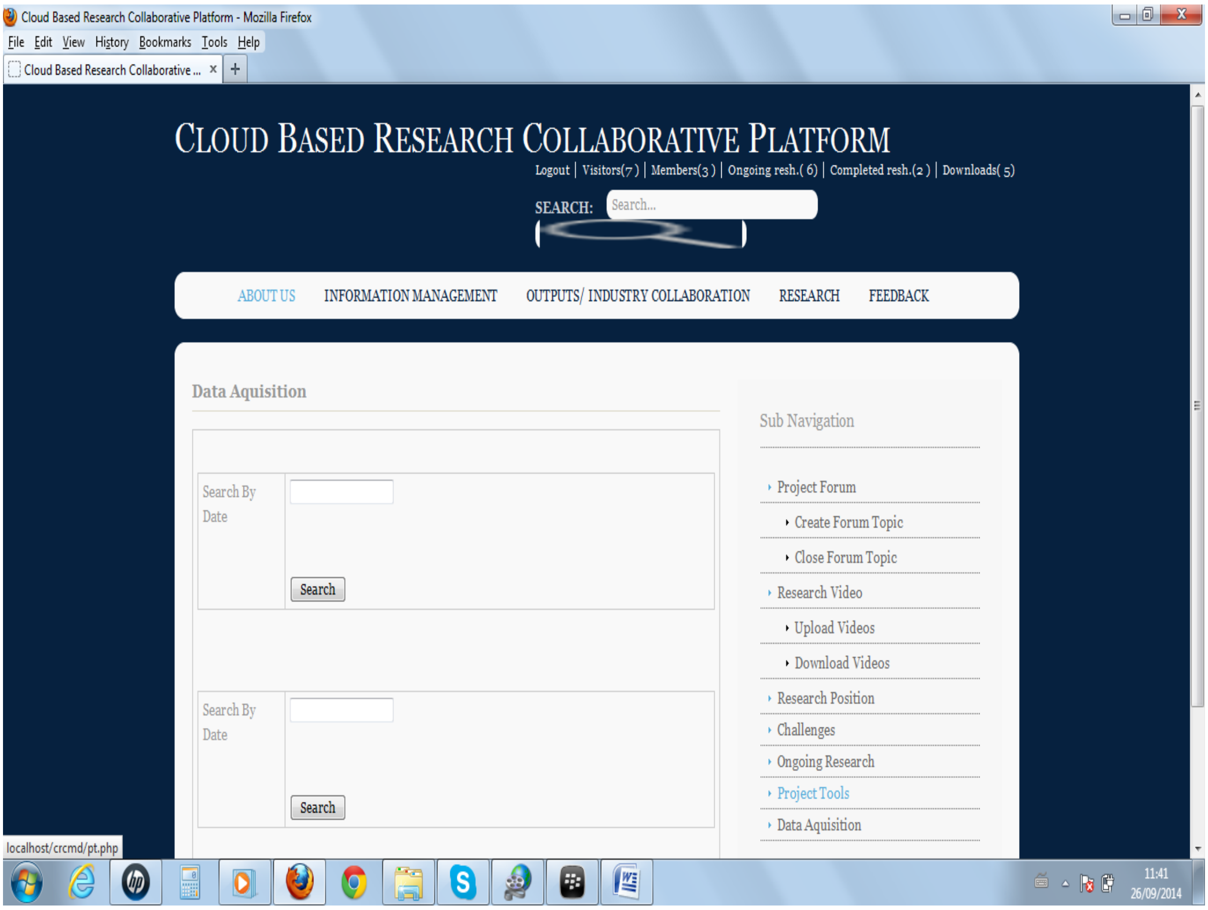


Figure 4.10: Data Acquisition page

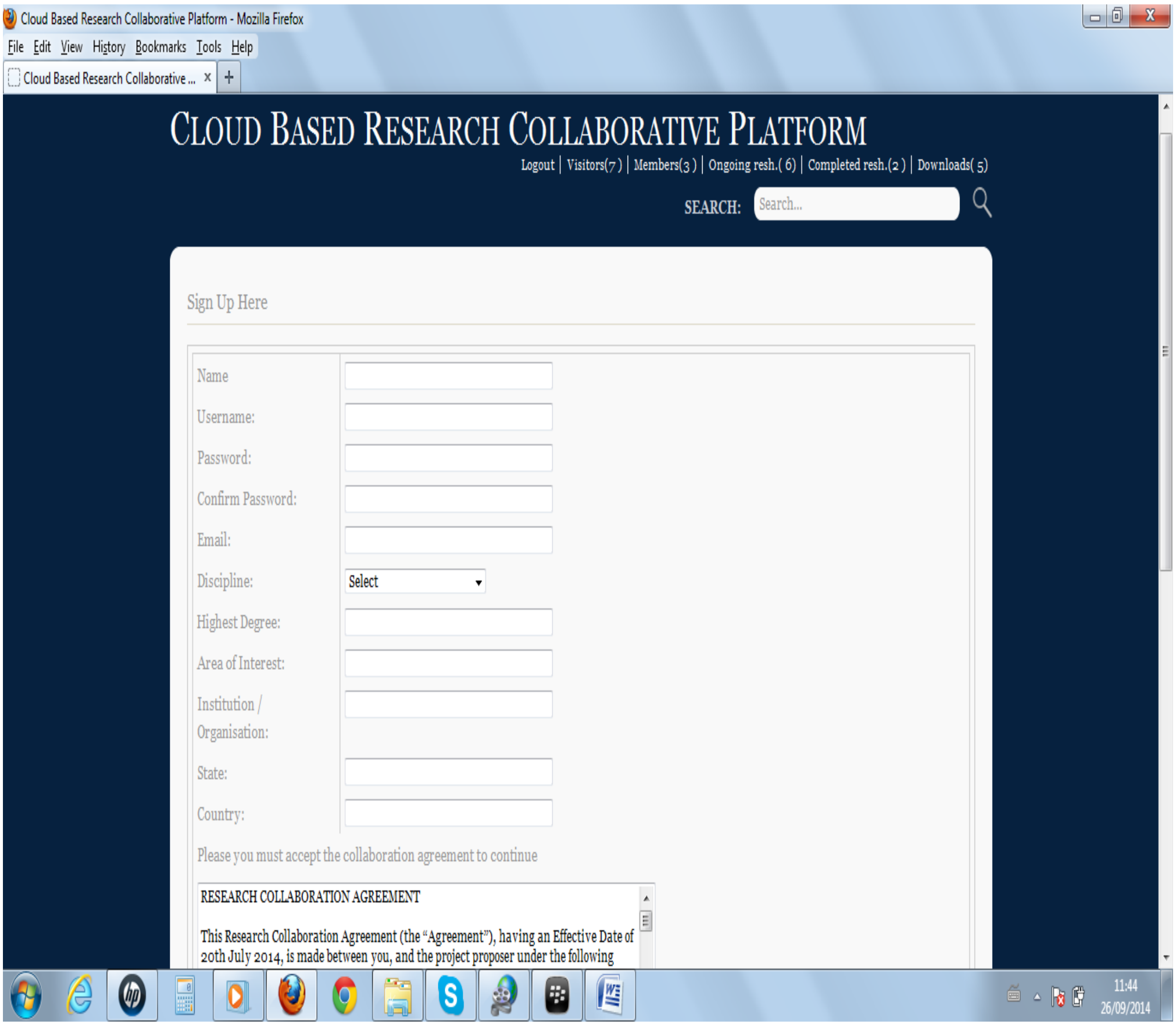


Figure 4.11 CRCM Sign Up page

Figure 4.121is the sign up page for the registration of new users. The users have to supply information on their areas of interest, degrees and discipline. This is to ensure that it is people of like interest and with the right qualification that interact in the research forums. The user also signs a collaboration level agreement here.

Figure 4.12 CRCM Information Management Interface

Figure 4.12 is the information management interface. It is from this interface that the user navigates to available grants/funding opportunities and also accesses information on available scholarship positions. The funded research outputs can also be viewed from here. The users can also share, manage,review and edit documents from here. There is a survey form in the grants section to help track the use of grants acquired through the help of the platform thereby assessing the effectiveness of the platform as a tool for accessing research grants.

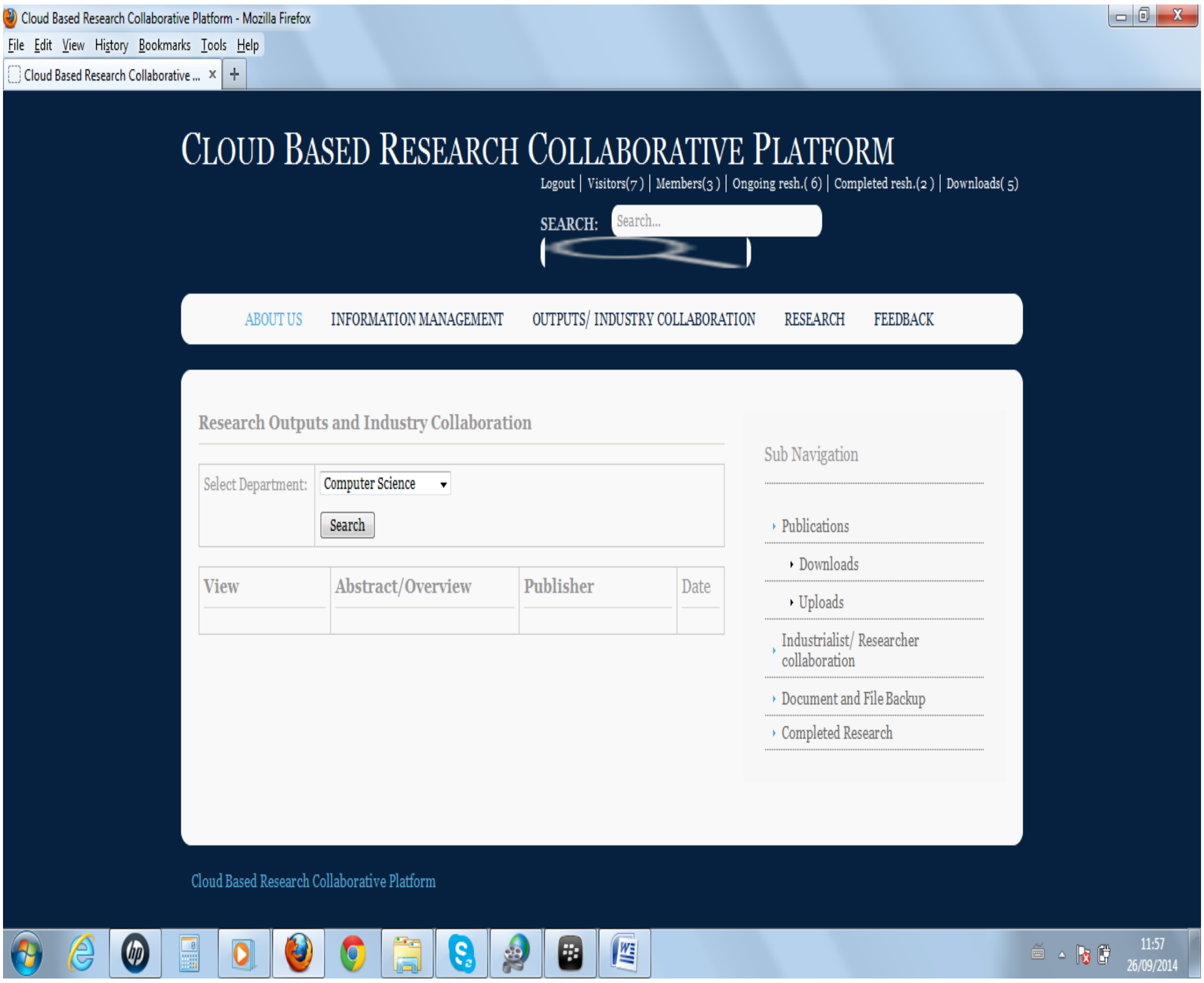


Figure 4.13: Research output and Industry Collaboration interface

Figure 4.13 is the interface for research outputs and industry/researchers collaboration. Publications on research works can be shared here. Completed research works are also shown here. The industrialists have access to abstracts on those works to enable them have a view of what it is all about. They can also sponsor the mass production or further modifications needed on the research works to turn them into products useful for the society at large. It is through this section that files can be backed up to prevent total loss in the advent of a failure. The section has a subsection on the industrialists' and researchers' collaboration as shown in figure 4.14. Before this collaboration an agreement must be signed by the parties involved for the protection orf their rights.

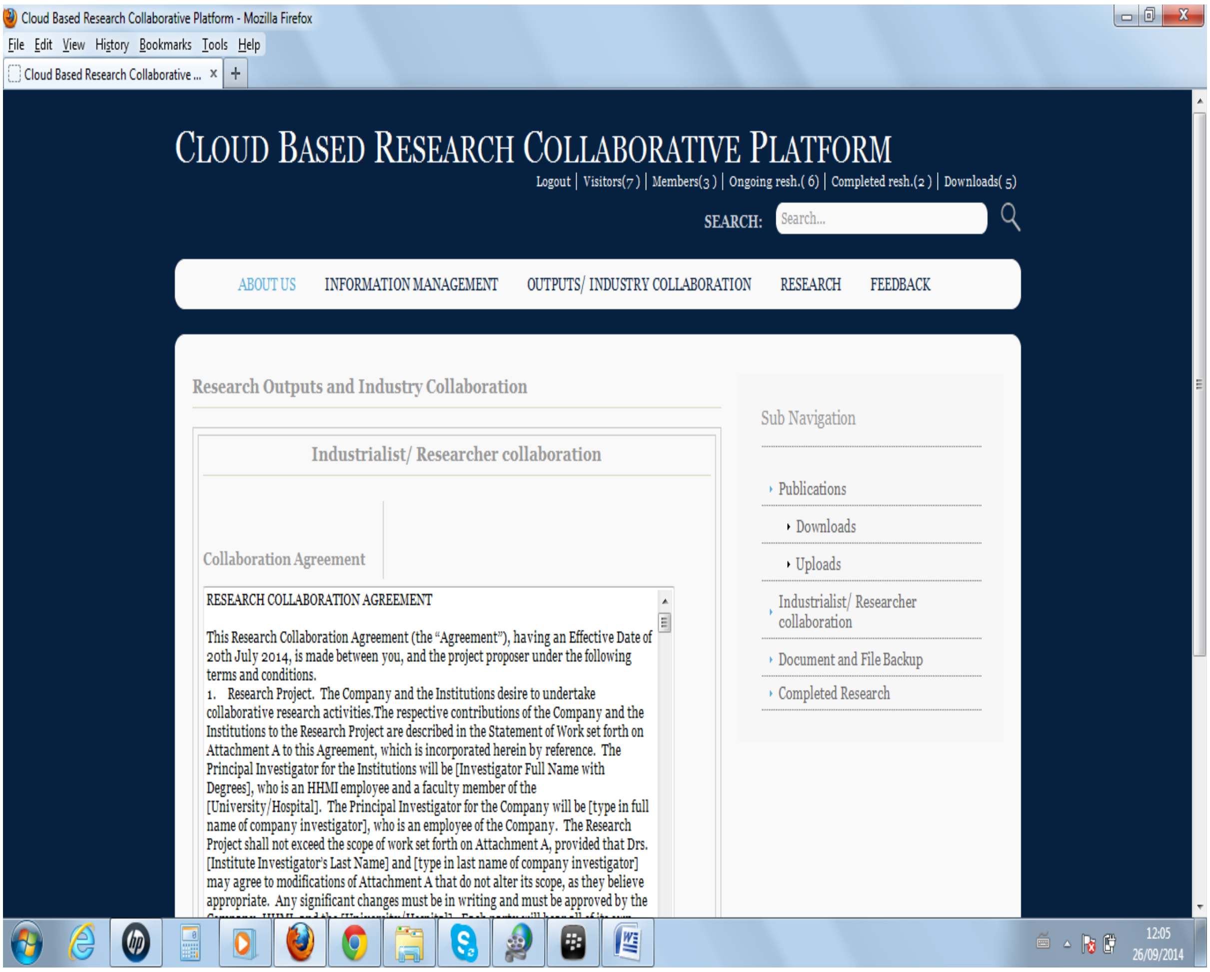


Figure 4.14: Researchers'and Industrialists'Collaboration interface

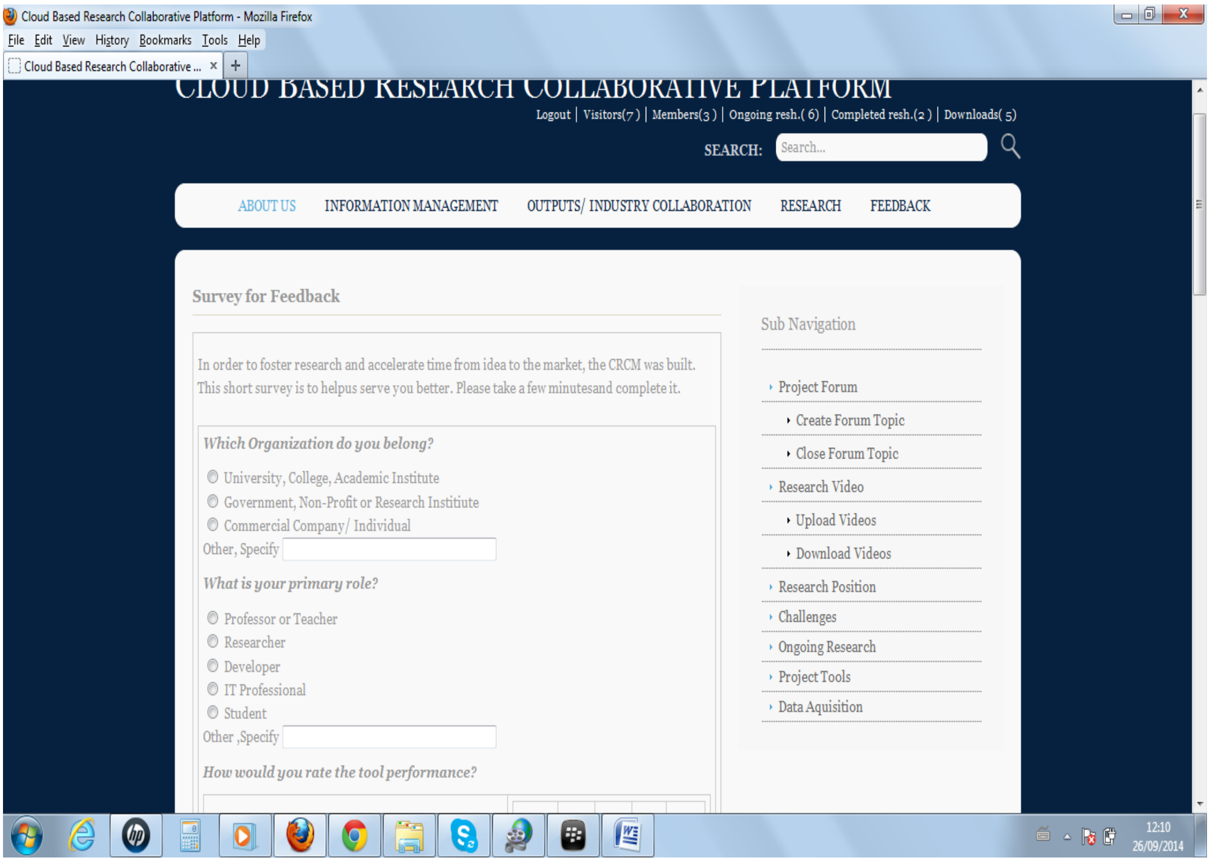


Figure 4.16 CRCM Feedback Interface

Figure 4.16 is the feedback interface used for the collection of data that will be used in evaluating the performance of the CRCM as a tool for aiding collaboration in engineering and technology research.

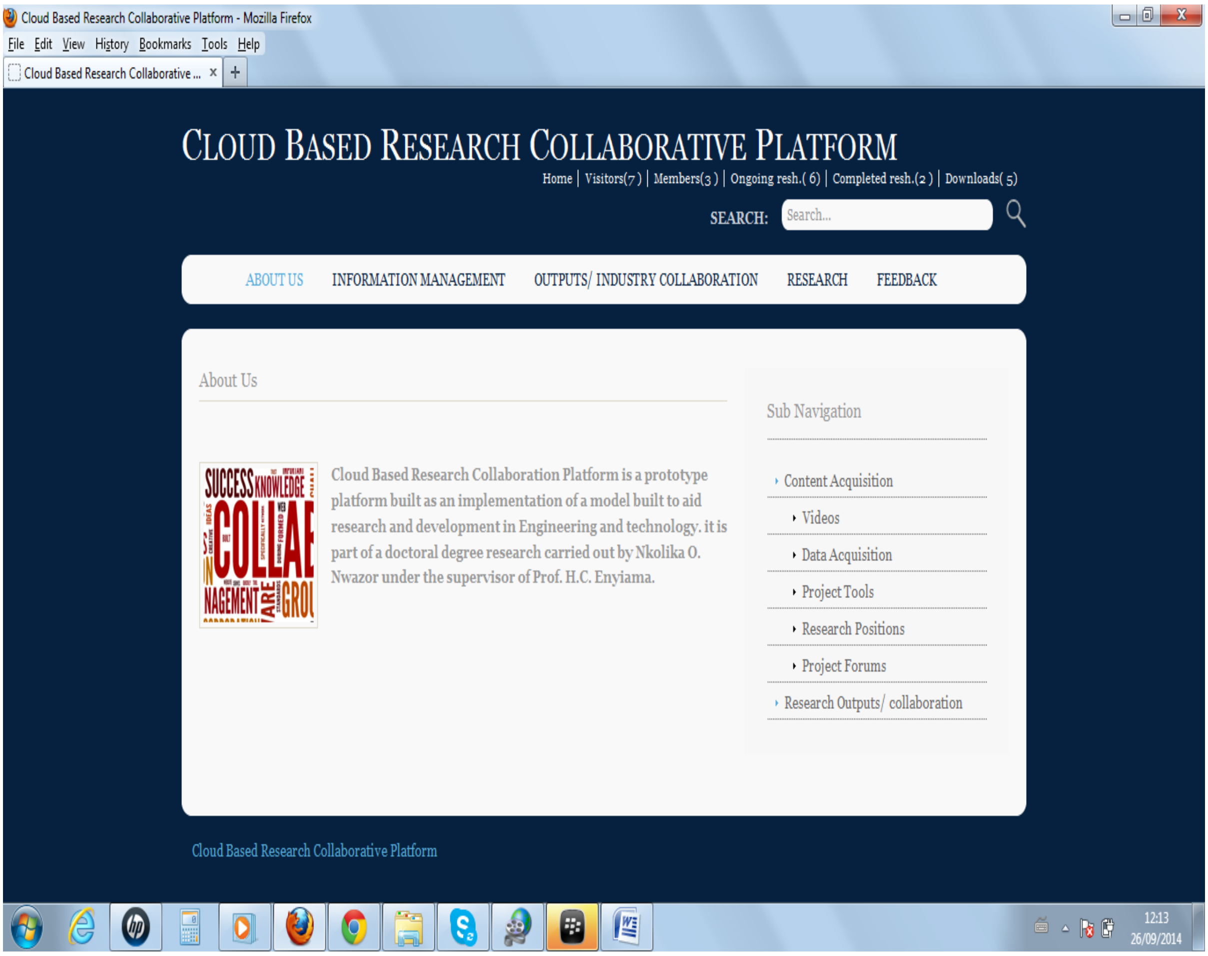


Figure 4.17: About us interface

Figure 4.17 shows the about us interface which gives a brief information on the platform developers.

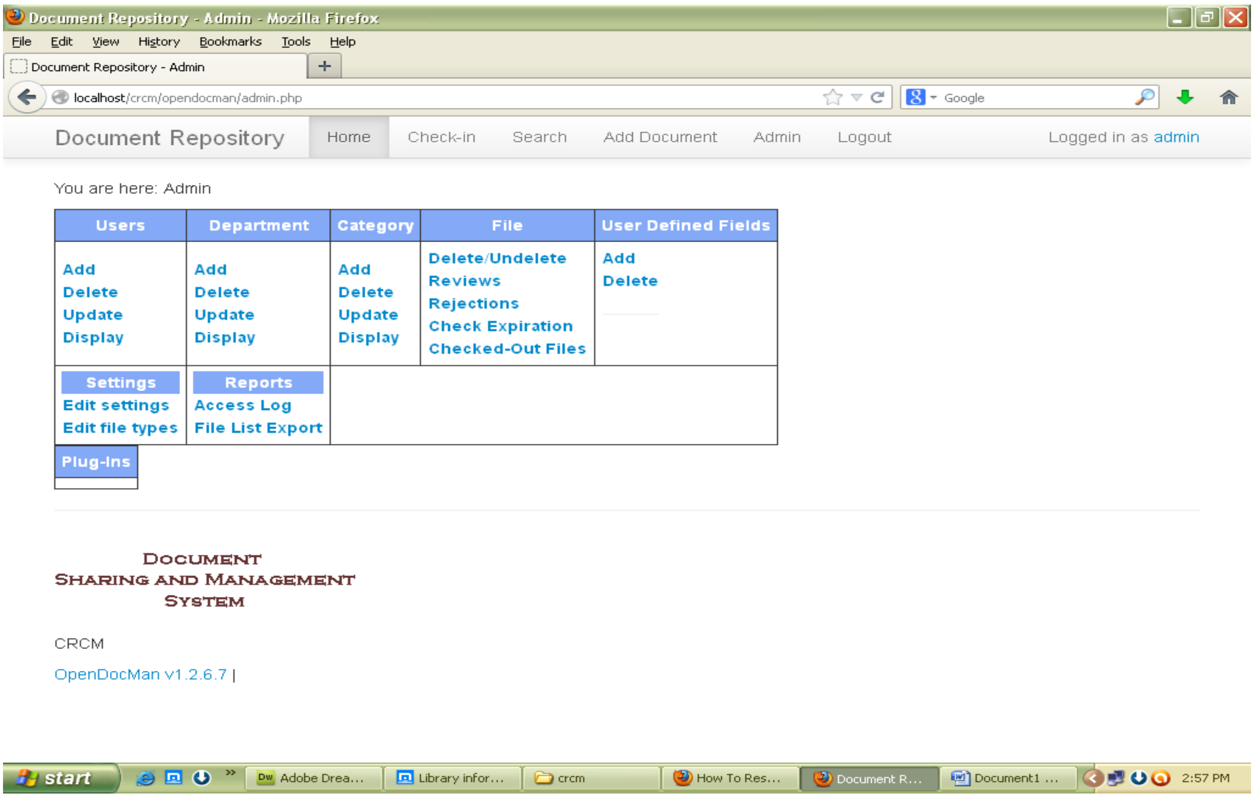


Figure 4.18 CRCM Document Sharing and Management Admin Interface

Figure 4.18 is for administrative access. We also see from this figure that documents are reviewed before being accepted for high data integrity. Rejected documents are removed from the system.

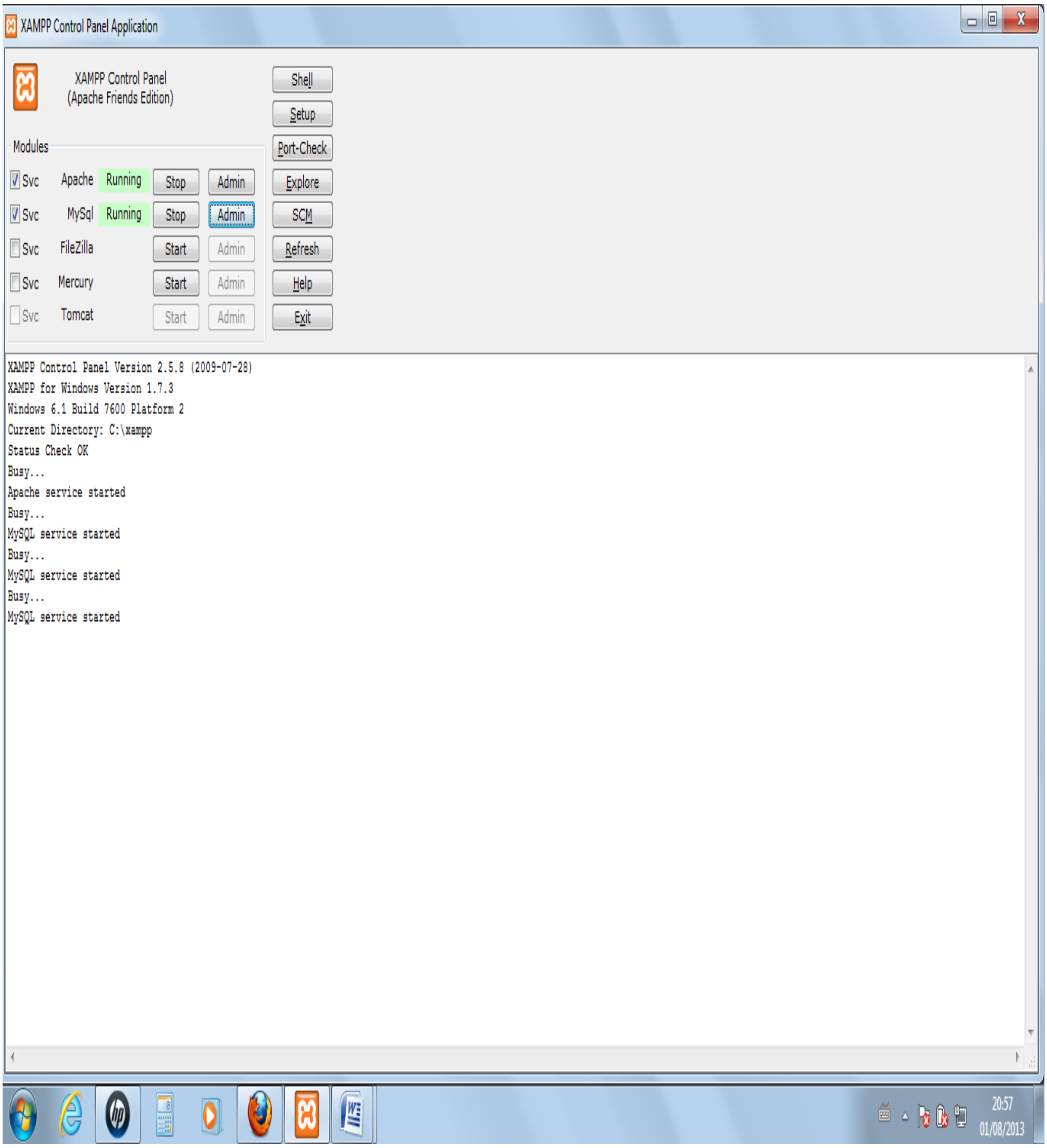


Figure 4.19 CRCM XAMPP Control panel for Modular Integrations

Figure 4.19 is the XAMPP control panel application. The apache is the server responsible for processing the php scripts. Through the MySql admin, the database and data mining configurations are made.

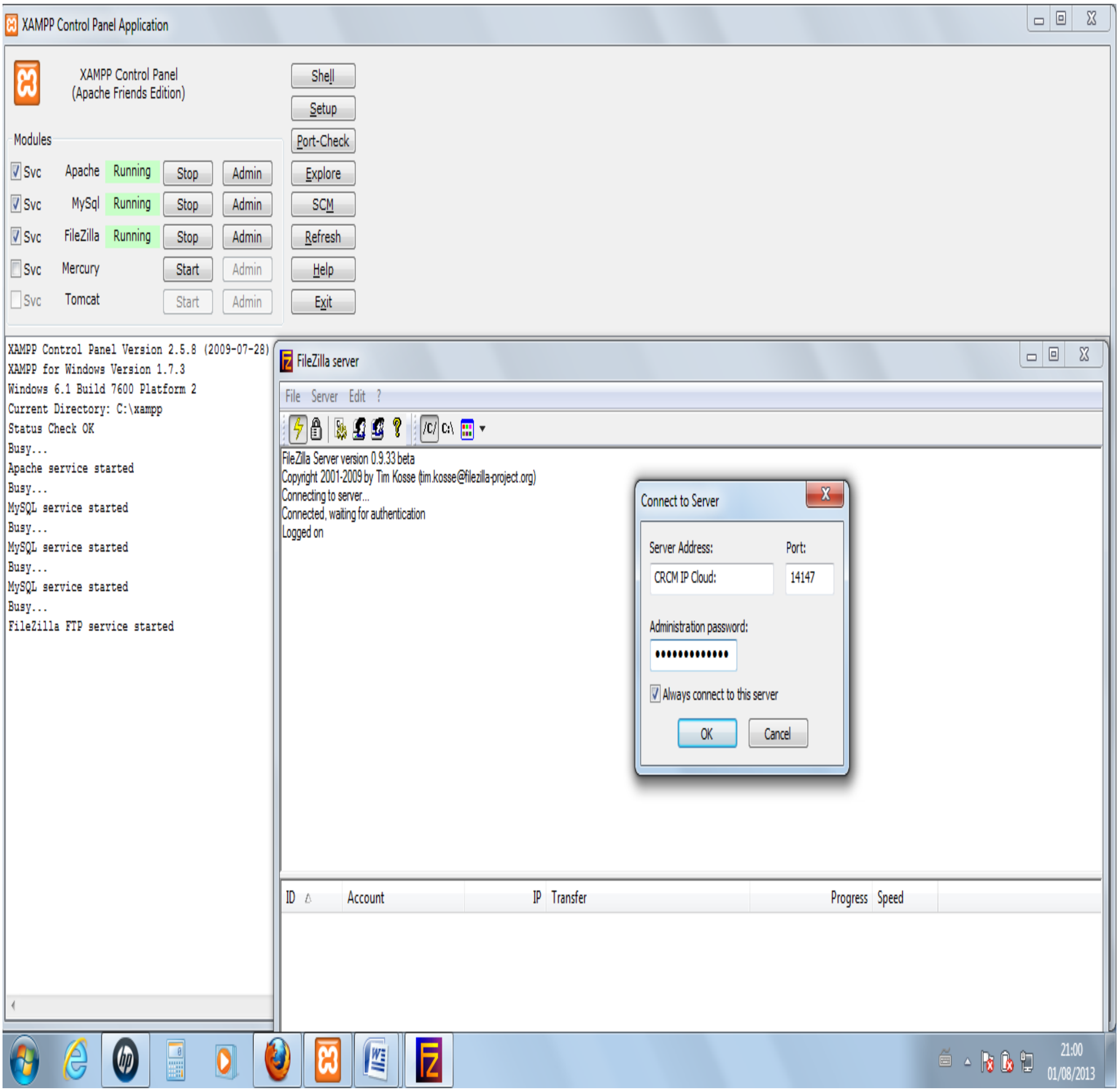


Figure 4.20 CRCM for Cloud Platform deployment on the IP Cloud

The FileZilla is used for connecting to the server as shown in figure 4.20. It is through the filezilla admin that the address/domain name of the IP cloud service provider is input and also the port number required for telneting. Authentication is required for access to these functions.

The implementation of CRCM database is shown in figure 4.21

Server: localhost ▸ Database: crcmd

Structure | SQL | Search | Query | Export | Import | Designer | Operations | Privileges | Drop

	Table ▲	Action	Records [1]	Type	Collation	Size	Overhead
☐	challenges		2	MyISAM	latin1_swedish_ci	2.2 KiB	-
☐	data_acquisition		0	MyISAM	latin1_swedish_ci	1.0 KiB	-
☐	depts		2	MyISAM	latin1_swedish_ci	2.1 KiB	-
☐	docbackup		4	MyISAM	latin1_swedish_ci	2.2 KiB	-
☐	grantsurvey		4	MyISAM	latin1_swedish_ci	2.1 KiB	-
☐	login		9	MyISAM	latin1_swedish_ci	3.0 KiB	-
☐	replies		4	MyISAM	latin1_swedish_ci	2.3 KiB	-
☐	softtools		3	MyISAM	latin1_swedish_ci	2.1 KiB	-
☐	store_file		2	MyISAM	latin1_swedish_ci	2.2 KiB	-
☐	topics		3	MyISAM	latin1_swedish_ci	4.9 KiB	-
☐	videos		3	MyISAM	latin1_swedish_ci	2.2 KiB	-
	11 table(s)	Sum	36	MyISAM	latin1_swedish_ci	26.4 KiB	0 B

Check All / Uncheck All   With selected:

Figure 4.21: Database structure for the CRCM

Figure shows the data base structure for CRCM. It consists of a list of the different tables used in the CRCM database like the challenge table, the data acquisition table and so on. It also shows the records in each table, the size of the table etc.

4.6 CRCM Testing

Before a system is made fully operational, it should be thoroughly tested for vulnerability checks, syntax errors, compatibility checks with data center OS. Here the entire program for the system was tested using different approaches.

Firstly, we tested the CRCM application on a local host server machine (icore 5, 4GB Ram, 4GHz Processor) while observing the application interactivity. Secondly, the CRCM was tested on a Syskay datacenter network, and it was observed that the application scaled very well in terms of collaborators cloud experience, server utilisation, latency. The same behaviour was obtained from various other Data centre networks in Nigeria viz: Swift Networks Lagos, Kswitche Labs, ELDI server farm, Cisco systems Data centre, and OAU Management information System units as shown in figure 4.21

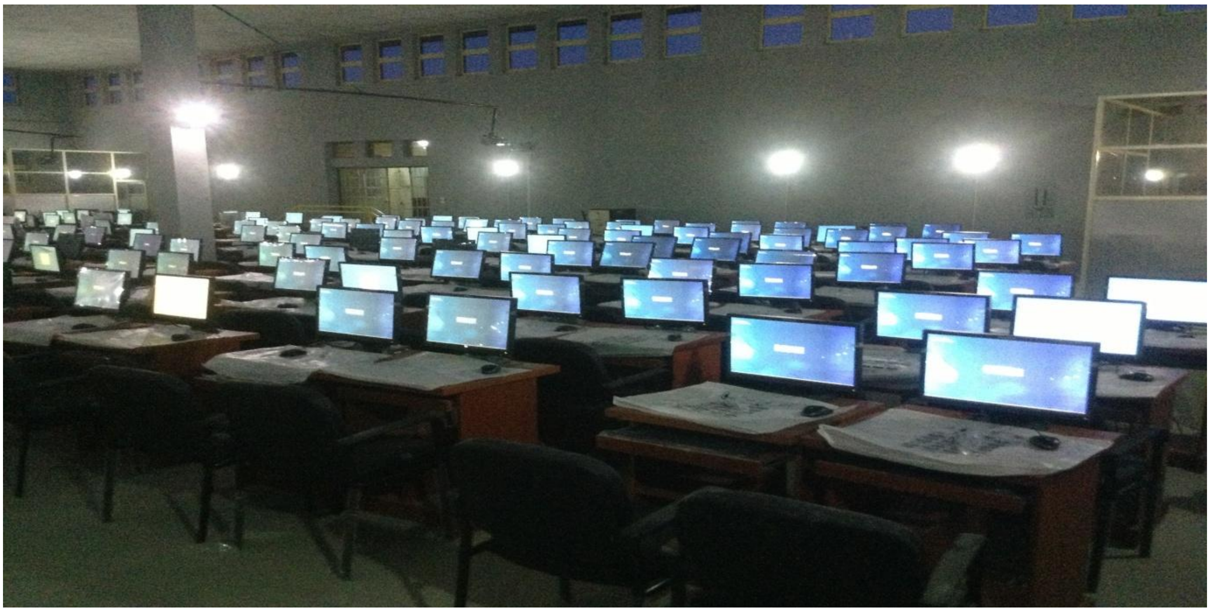


Figure 4.21: Snapshots of CRCM thin clients collaborators at OAU (2013) in site1

4.7 CRCM Cloud Deployment

The requirements to run the proposed system for optimal performance are listed in table 4 with the test server computer having optimal hardware requirements. In this work, Google App Engine was used to run the collaboration web applications on Google's infrastructure to ascertain its performance. In its customization for App Engine, PhP supports made it easy to scale the traffic and data storage as needs grow. With App Engine, there was no need for servers as we just uploaded the application via a filezilla in XAMP application. The summarized features of app Engine is presented in table 4.8.

Table 4.8: Features of Google App Engine (Google, n.d.)

Service	**Free quota per app per day**	**Pricing FOR Each Quota**
Instances	28 instance hours	$0.05 / instance / hour
Cloud Datastore (NoSQL Database)	50k read/write/small 1 GB storage	$0.06 / 100k read or write operation Small operations free* $0.18 / GB / month
Outgoing Network Traffic	1 GB	$0.12 / GB
Incoming Network Traffic	1 GB	Free
Cloud Storage	5 GB	$0.026 / GB / month
Memory Cache	Free Usage of Shared Pool No free quota for Dedicated Pool	Free Usage of Shared Pool Dedicated Pool: $0.06 / GB / hour
Search Supports	1000 basic operations 0.01 GB indexing documents 0.25 GB document storage 100 searches	$0.50 / 10k searches $2.00 / GB indexing documents $0.18 / GB / month Storage
Email API	100 recipients	
Logs API	100 MB	$0.12 per GB
Task Queue and Logs Storage	5 GB 1 GB	$0.026 / GB / month
SSL Virtual IPs	No free quota	$39 / virtual IP / month

In deploying the application, the first step was to Create and set up a Cloud SQL instance. The next step was to create a first project at the Google APIs console. Google Cloud SQL service in the ‘Services’ screen was then enabled. At this point the 'Google Cloud SQL' menu can be seen on the left side. Clicking the 'Google Cloud SQL' menu and click 'New instance...' button displays a dialog box for the user to input the following:

- Name: Name of the Cloud SQL instance
- Authorized applications: the app-id of the App Engine application to which you’re going to deploy this application.

The 'Name' and 'Authorized applications' fields are filled appropriately and 'Create instance' button was then selected. After a few minutes, the first Cloud SQL instance was ready.

After the progress indicator disappeared, the 'SQL prompt' menu was selected. It is at this interface that SQL commands can be executed. As with the local MySQL instance, a database user is created by executing the following commands. This was done by entering one command at a time in the big text area and clicking the 'Execute' button each time.

```
CREATE USER 'CRCM'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON guestbook.* TO 'CRCM'@'localhost'
WITH GRANT OPTION;
```

Then the 'guestbook' database was created with the following command:

```
create database guestbook charset utf8;
```

'guestbook' database was selected in the 'Database' select box, and a table was created for the guestbook.

```
CREATE TABLE entries

    (id int not null auto_increment primary key,
    guest_name varchar(255),
```

content varchar(255),

created_at timestamp)

ENGINE=InnoDB DEFAULT CHARSET=utf8;

The instance was accessed using command line tool.

With the table created in the Cloud SQL instance, deployment of the application in the cloud was done. The cloud SQL instance was accessed with command line tool. App Engine SDK has a command named 'google_sql.py' which enables connection of the Cloud SQL instance from the command line.

Before deploying, the Cloud SQL instance name was added to the file main.py as follows:

CLOUDSQL_INSTANCE = '**cloud_sql_CRCM**

The application was then deployed.

The address of the developed platform is:

www.cbrcpcloud.com.ng

4.8 Performance Evaluation using Users' Feedback

The software was deployed while allowing the thin clients to access and use the application for one month. The work showed that the design architecture of the CRCM matched with the Google infrastructure offering an optimal performance for users. Again, using Syskay infrastructure, the CRCM testing delivered satisfactory performance from operational dimension showing over 95% efficiency.

From the data received through the feedback section, it was discovered that the application was mostly used by those in the academic institutions followed by users in the research institutes as shown in figure 4.22. Ten users came from the industry out of one hundred and fifty users (i.e. 6.7%). This is low but encouraging in a country like Nigeria where there is almost no researchers'/industrialists' collaboration. This shows that with time the number of users from the industry will increase and with the information available to them on the platform, most of the research outputs will eventually be turned into useful products.

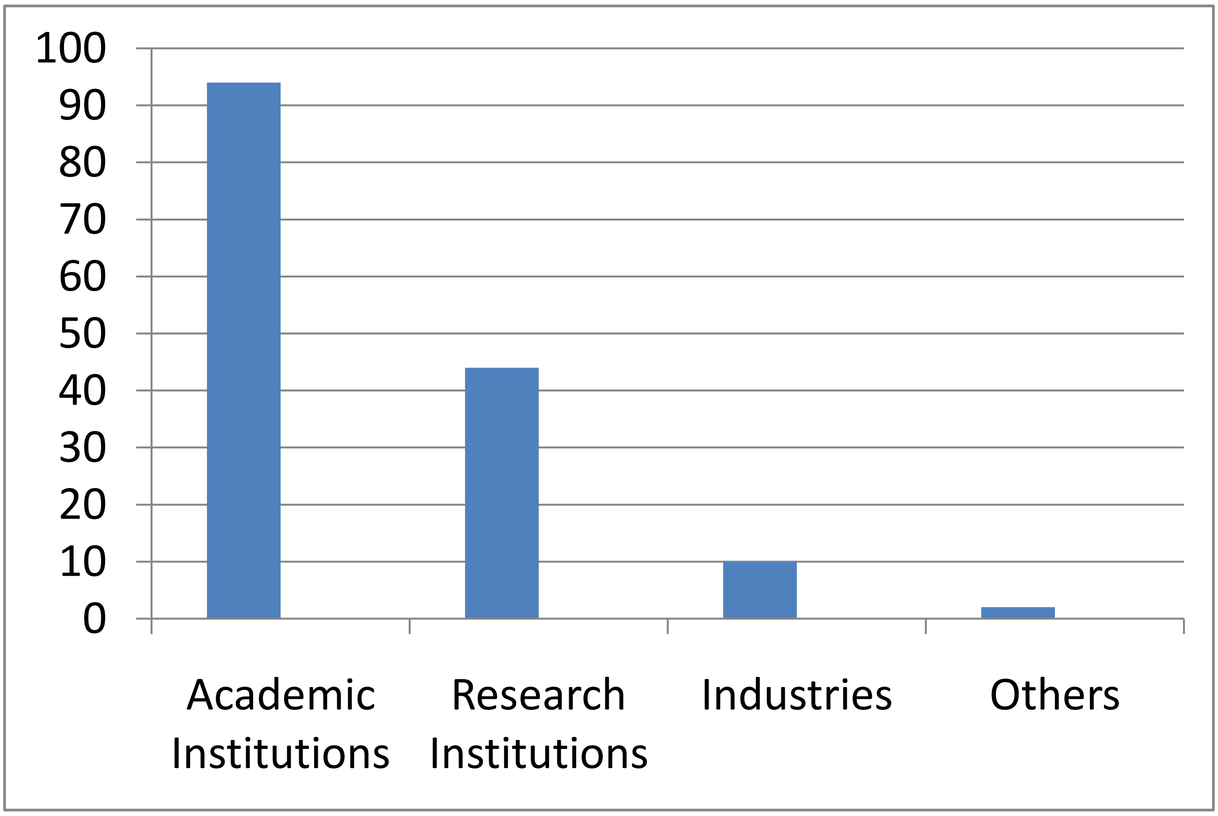


Figure 4.22: Plot showing the user groups

A plot of the users primary roles show that students were greater in population followed by teachers, other researchers and industrialists as shown in figure 4.23

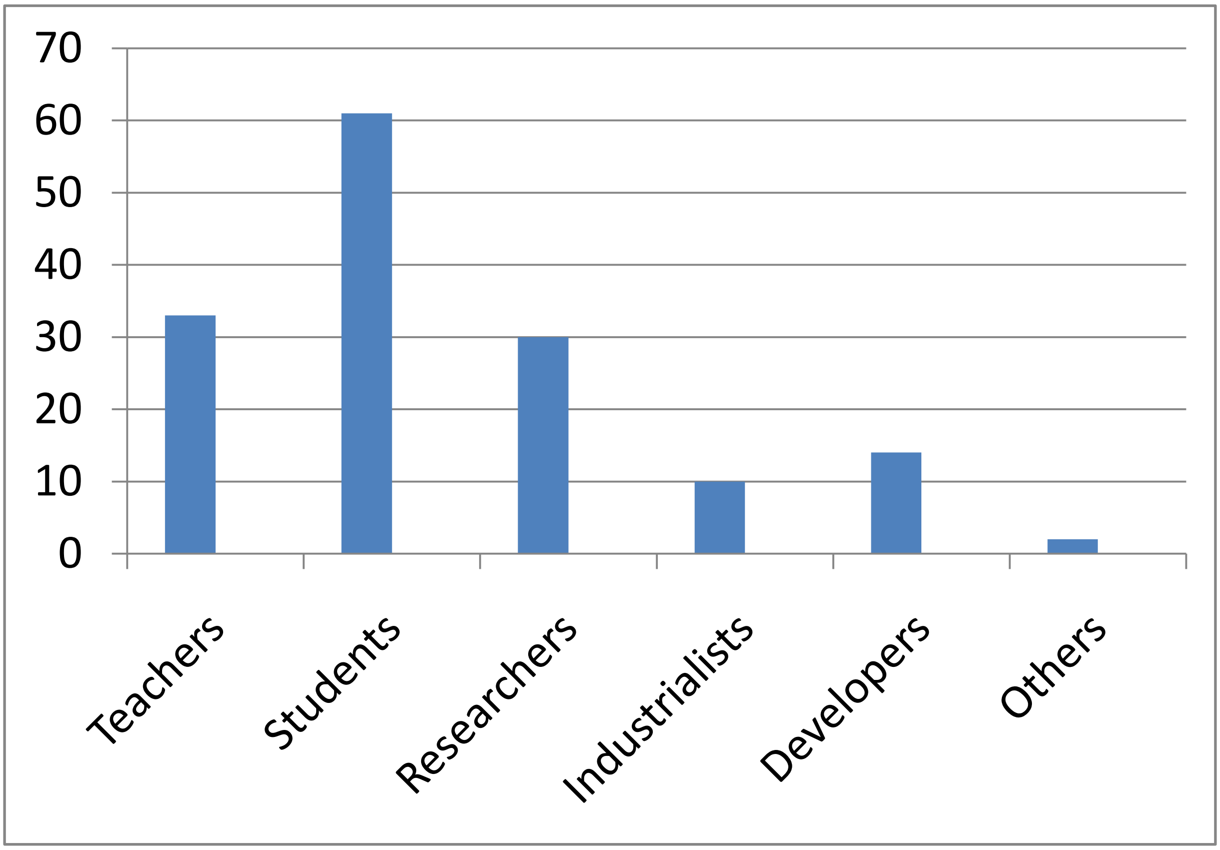


Figure 4.23: Users' Primary role

It was also discovered that most of the users agreed that the application is user friendly, saves research time, makes access to grants and scholarships easier and is secure. Out of the 150 users sampled, 118 strongly agreed that they will recommend the application to other users (i.e. 78.7%) as shown in figure 4.24.

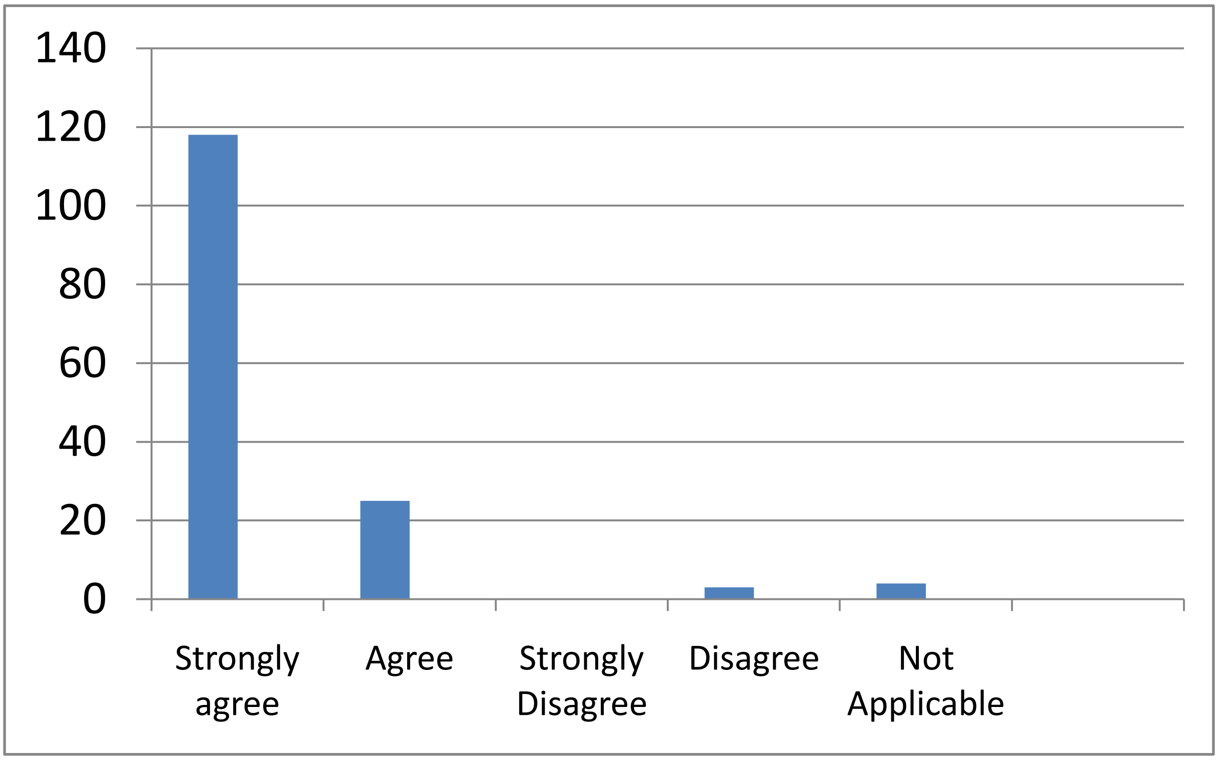


Figure 4.24: A plot showing users' Recommendation agreement of the application

Therefore it can be concluded that the application is indeed a very effective tool for research collaboration in engineering and technology.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Summary of achievements

This dissertation presented a Cloud computing platform for remote research collaboration in Engineering and Technology especially for the Nigerian environment. The CRCM was designed for efficient performance, security integration and data integrity.

A conceptual framework for secure research collaboration in engineering and technology with automatic data acquisition from research equipments was modelled and simulated. A queuing model was formulated for the system which is a modified form of the G1/G/1queuing model showing the effect of the system's queuing on server service utilization and general system efficiency. The proposed cloud queuing model showed a great enhancement in resource management in the context of cloud queuing management. The system flowcharts for the implementation of the prototype CRCM platform were designed consisting of seven major modules which were then implemented using XAMP, php, and mysql server database while taking cognizance of the proposed security system. The CRCM security was based on Hybrid Encryption Algorithm which is a combination of the advanced encryption standards encryption and a digital signature for data integrity. Its encryption operation is an exponentiation to the e^{th} power modulo n.

The Input/output interfaces are GUIs accessible through web browsers. The data center was based on CISCO data center architecture which is a multi-tier data center model. Independent blade server was adopted to help reduce cost, improve virtualisation, atomization and provisioning capabilities.

CRCM http service validation was carried out with OPNET using datacenters parameters from CISCO. It showed the effectiveness of the proposed DCN in terms of throughput, queuing management, server utilization and latency effects of the CRCM data center for remote access. From the evaluations carried out, it was also shown that the system will scale efficiently in direct proportion to the incoming job queues or arrivals. Also the resource utilization by the server was fairly uniform. Finally, the application was tested on a local host server machine. It was then tested on Syskay data center network, Swift technology network, CISCO system and it scaled very well in terms of collaborators cloud experience with page response time of 0.032 secs and server utilisation of 99%.

5.2 Problems Encountered and Solutions

In this research, the following problems were encountered:

1. Transient utility power supply in the course of this work presented a lot of setbacks which was addressed by usage of alternative power supply system (generator), solar power and inverter systems.

2. Getting information for analysis of existing models from the cloud service providers on the QoS metric investigated in this work was also very challenging but the analysis was finally done based on well established models and methods

3. Understanding the process integration of existing collaboration systems and their security standards was a challenge as there are limited documentations on their design processes. However, consultations with industry experts and systems developers facilitated the CRCM research.

5.3 Contributions to Knowledge

The following were the contributions made in this work:

i. A conceptual framework for secure research collaboration in engineering and technology with automatic data acquisition from virtual infrastructures was modelled.

ii. A prototype CRCM based on PaaS cloud computing model which facilitates research collaboration in Engineering and Technology with real-time network data acquisition was designed and developed.

iii. An enhanced adaptive queuing scheme for CRCM service efficiency with a response time of 0.032second and 99% server utilisation was modelled and simulated. This is a model that addressed queue problems and low server utilization for service efficiency, giving rise to effective system resource utilization eg. I/Os, Memory, and storage optimization. This is aimed at improving performance at large

iv. An enhanced security and data integrity system called Hybrid Encryption Algorithm which is an exponentiation to the e^{th} power modulo was designed and implemented

5.4 Conclusion

Research in engineering and technology traversing geographical boundaries can now be carried out with great ease with the realisation of the system proposed in this work. A cloud-based research collaboration platform for engineering and technology with automatic data acquisition from virtual equipments has been modelled. A cloud-based research collaboration platform was designed and implemented. A Hybrid Encryption Algorithm was proposed and developed to ensure second tier data security and integrity. This system will allow easy interaction, exchange of ideas, and access to needed information among researchers and industrialists who are the implementers of research outputs while assuring the users of the security of their data. The system comes readily as a solution to the problems enumerated in the problem statement like breaking geographical barriers, sharing expensive equipments, to mention but a few. The system was evaluated and it competed favourably with existing research collaboration platform by having a very low page response, very good job traffic management and the introduction of automatic data acquisition from remote equipments. The papers published from the dissertation are listed in appendix 1A.

5.5 Recommendations

1. The physical integration of existing remote scientific equipments for real time data acquisition is highly recommended as this will help in speeding up research process and in reducing the limitations placed on engineering research in developing nations due to inadequate infrastructure/ equipments.

2. Ease of access to information on some of the metrics used by the cloud service providers and their security system will greatly aid research in these areas

References

Ahronovitz, M., Amrhein,D., Anderson, P., Andrade, A., Armstrong, J., Arasan, E., ….. , and Zapper, F. (2010). Cloud computing use cases. A white paper produced by the cloud computing use case discussion group, Version 4.0. 2. Retrieved from http://www.cloud-council.org/Cloud_Computing_Use_Cases_Whitepaper-4_0.pdf

Al-Fare, M., Loukissas, L., and Vahdat, A. (2008). A scalable, commodity data center network architecture. *Proc SIGCOMM.*

Alfresco: Open Source Enterprise Content Management System (CMS). Retrieved from http://www.alfresco.com

Baburajan, R. (2011). The rising cloud storage market opportunity strengthens vendors. *InfoTECH*. Retrieved from

http://it.tmcnet.com/channels/cloud-storage/articles/211183-rising-cloud-storage-market-opportunity-strengthens-vendors.htm

Backialakshmi, M. and Sathya, S. (2014). A survey on scheduling algorithms in cloud computing. *International Journal of Engineering Research and General Science, 2*, 6, P. 12-22.

Barr, J. (2008). The emerging cloud service architecture. Retrieved from http://aws.typepad.com/aws/2008/06/the-forthcoming.html

Bezzubtseva, A. and Ignatov, D. (2012). A typology of collaboration platform users. *Proceedings of International Workshop on Experimental Economics in Machine Learning*, P. 9-19. Retrieved from http://arxiv.org/abs/1312.0162

Bhadauria, M. and McKee, S.A. (2010). An approach to resource-aware co-scheduling for CMPs. *ICS*, P. 189-199.

Borda, A. (2006). Report of the working group on virtual research communities for the OST e-infrastructure steering group. London, UK, Office of Science and Technology, P.46.

Boston: Kluwer school of information and technology (). Research profile. Retrieved from http//www.kth.se/ict/2.1416/var-forskningsprofil-1.7528

Boxma, O.J. and Cohen, J.W. (1999). Heavy traffic analysis of the G1/G/1 queue with heavy tailed distribution queuing systems. *Queuing systems*, *33*, P.177-204. dx.doi.org/10.1023/A:1019124112386

Brocke, J., Riemer, K., Große-Böckmann, S. and Richter D. (2008). Using the internet for collaborative research in Europe – experiences from the Garnet virtual network. Retrieved from http://www2.warwick.ac.uk/fac/soc/garnet/workingpapers/3608.pdf

Business dictionary (n.d.). Data integrity. Retrieved from http://www.businessdictionary.com/definition/dataintegrity.html#ixzz3DtW2cpi0

Buyya, R., Broberg, J., and Goscinski, A.M. (2011). Cloud computing: Principles and paradigms. *John Wiley & Sons, Inc*.

Cheng, J. (2007). Microsoft and apple building online storage havens: You win. Retrieved from http://arstechnica.com/news.ars/post/20070812-google-microsoft-and-apple-building-online-storage-havens-you-win.html

Cisco (2008). Data center architecture overview. Retrieved from http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/DC_Infra2_5/DCInfra_1.html

CISCO (2012). Building collaborative platform for driving innovation. Swiss university. Retrieved from http://www.cisco.com/en/US/solutions/collateral/ns340/ns517/ns224/eth_cs.pdf

Cloud Security Alliance (2012). “SecaaS implementation guidance, category 8 encryption. Retrieved from https://downloads.cloudsecurityalliance.org/initiatives/secaas/SecaaS_Cat_8_Encryption_Implementation_Guidance.pdf

Corbató, F.J., Daggett, M.M., Daley, R.C. (1962). An experimental time-sharing system. *SJCC Proceedings*. MIT. Retrieved from http://larch-www.lcs.mit.edu:8001/~corbato/sjcc62/

Cryptoclarity (2009). Encrypted storage and key management for the cloud. Retrieved from http://www.cryptoclarity.com/CryptoClarityLLC/Welcome/Entries/2009/7/23_Encrypted_Storage_and_Key_Management_for_the_cloud.html

Douglas, J. (2012). Wiley Announces New Open Access Journal: Energy Science & Engineering. Retrieved from http://www.stm-publishing.com/wiley-announces-new-open-access-journal-energy-science-engineering

Ebrahim, N. (2010). Virtual R&D teams: A potential growth of education-industry collaboration. Retrieved from www.academia.edu/472259/Virtual_R&D_Teams:_A_potential_growth_of_education-industry_collaboration

E-Research co-ordinating committee (2005). An e-research strategic framework: Interim report of the e-research coordinating Committee. Retrieved from http://www.dest.gov.au/NR/rdonlyres/B6F765A7-DD2C-432B-9064-2F9CD4E17E66/10518/InterimReport2.doc

Farber, D., (2008). The new geek chic: Data centers. *CNET News*. Retrieved from http://news.cnet.com/8301-13953_3-9977049-80.html

Foley, J. (2008). Private clouds take shape. *InformationWeek*. Retrieved from http://www.informationweek.com/news/services/business/showArticle.jhtml?articleID=209904474

Gabriel, H. (2010). Open source software and research infrastructure. Retrieved from http://wiki.osswatch.ac.uk/OpenSourceAndResearchInfrastructure

Gens, F. (2008). Defining "cloud services" and "cloud computing". Retrieved from http://blogs.idc.com/ie/?p=190

Girish, M.K. and Hu, J. (1997). An interpolation approximation for the GI/G/1 queue based on multipoint padlle approximation. *Baltzer Journals.* Retrieved from http://people.bu.edu/hqiang/papers/gh97_questa.pdf

Gittins, J.C. (1989). Multi-armed Bandit allocation indices. *John Wiley & Sons*.P. 77.

Gmach, D. (2010). Selling T-shirts and Time Shares in the Cloud. *In: CCGrid*, P. 539–546.

Google (n.d.). Google Cloud Platform. Retrieved from https://cloud.google.com/appengine/

Greenberg, A., Hamilton, J., Jain, N., Kandula, S., Kim, C., Lahiri, P., …. and Sengupta, S. (2009). VL2: A scalable and flexible data center network. *Proc SIGCOMM*.

Guardian (2008). Google angles for business users with platform as a service. Retrieved from http://www.guardian.co.uk/technology/2008/apr/17/google.software

Guo, C., Lu, G., and Li, D. (2009). BCube: A high performance, server-centric network architecture for modular data centres. *Proc SIGCOMM*.

Guo, C., Xiaofeng, Z., and Shun-Liang, M. (2008). DCell: A scalable and fault-tolerant network structure for data centres. *Proc SIGCOMM*.

Guo, L. (2014). Dynamic performance optimization for cloud computing using M/M/m queuing system. *Journal of Applied Mathematics, Hindawi*

Publishing Corporation,2014, Article ID 756592. dx.doi.org/10.1155/2014/756592

Haff, G. (2009). Just don’t call them public clouds. *CNET News*. Retrieved from http://news.cnet.com/8301-13556_3-10150841-61.html

Hogan, M., Liu, F., Sokol, A. and Tong, J, (2011). NIST cloud computing standards roadmap – Version 1.0. Retrieved from http://www.nist.gov/itl/cloud/upload/NIST_SP-500-291_Jul5A.pdf

Kalyani, M. (2014). Cloud security: Efficient and reliable encryption key management crucial for data protection. Retrieved from https://spideroak.com/privacypost/cloud-security/secure-encryption-key-management-in-the-cloud/

Kayliani, M. (2014). The importance of encryption for companies in the cloud. Retrieved from https://spideroak.com/privacypost/business-the-cloud/the-importance-of-encryption-for-companies-in-the-cloud/

Kim, S. Devs-suite: A simulator supporting visual experimentation design and behavior monitoring. (Unpublished).

Kollberg, S. (2014). Proposed scheduling algorithm for deployment of fail-safe cloud services. Retrieved from http://www8.cs.umu.se/education/examina/Rapporter/SimonKollberg.pdf

Kumar, S. and Khurana, S. (2014). Analysis of different scheduling algorithms under cloud computing. *International Journal of Computer Science and Information Technologies, 5*, 2, P. 2592-2595. Retrieved from http://www.ijcsit.com/docs/Volume%205/vol5issue02/ijcsit20140502367.pdf

Laursen, K. and Salter, A. (2004). Searching high and low: What types of firms use universities as a source of innovation. *Research Policy, 33*, 8, P. 1201–1215.

Lavinia, P., O'Sullivan, D. and Cormican, K. (2006). Virtual team environment for collaborative research projects. *International Journal of Innovation and Learning, Inderscience Publishers, 3*, 1, P. 77-94.

Leena, K. and Jaiswal, A. (2013). Cloud computing: Security issues and description of encryption based algorithms to overcome them. *IJARCSSE, 3*, 3, P.279-283.

Lin, C.T. (2013). Comparative Based Analysis of Scheduling Algorithms for Resource Management in Cloud Computing Environment. *IJCSE, 1*, 1. Retrieved from http://ijcseonline.org/pub_paper/4.pdf

Maggio, M., Klein, C. and Arzen, K. (2014). Control strategies for predictable brownouts in cloud computing. *Preprints of the 19th World Congress.* The International Federation of Automatic Control Cape Town, South Africa

Menken, I. (2008). Cloud computing- The complete cornerstone guide to cloud computing best practices. *The art of service.*

Mills, E. (2009). Cloud computing security forecast: Clear skies. *CNET News*. Retrieved from http://news.cnet.com/8301-1009_3-10150569-83.html

Mutula, S. (2009). Challenges of doing research in sub-Saharan African universities: Digital scholarship opportunities. *Inkanyinso: Journal of Humanities and Social Sciences, 1,* 1. Retrieved from http://www.ajol.info/index.php/ijhss/article/view/62101/50140

MyNetResearch (2009). Retrieved from http//www.mynetresearch.com/Default.aspx

National Institute of Standards and Technology (2012). *The NIST definition of cloud computing*. Retrieved from http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

Obama B. (2009). Remarks by the president at the National Academy of Sciences annual meeting. Retrieved from http://www.whitehouse.gov/the_press_office/Remarksby-the-President-at-the-National-Academy-of-Sciences-Annual-Meeting/

Oestreich, K. (2011). Converged infrastructure. CTO Forum. Retrieved from http://www.thectoforum.com/content/converged-infrastructure-0

Palmer, C.L. (2001). Work at the boundaries of science: Information and the interdisciplinary research process. *Springer Science & Business Media.*

Parsons, R. (2004). VoIP congestion management - basic queuing methods. Retrieved from http://searchnetworking.techtarget.com/tip/VoIP-congestion-management-basic-queuing-methods.

Patel, S. and Bhoi, U. (2013). Priority based job scheduling techniques in cloud computing: A systematic review. *International Journal Of Scientific & Technology Research*, 2, 11. Retrieved from www.Ijstr.Org.

Pincus, J., Hemmert, J.R., Nellis, J.A., Peters, B.J. and Mooney, S.D. (n.d.). Laboratree: A web-based platform for team research collaboration. Retrieved from vivoweb.org/files/Laboratree.pdf

Ponemon Institute LLC, (2011). Security of Cloud Computing. Sponsored by CA Technologies Providers Study. Retrieved from http://the-forexjournal.rhcloud.com/security-of-cloud-computing-providers-final-11/

Saguy, S.I. (2011). Paradigm shifts in academia and the food industry required to meet innovation challenges. *Trends in Food Science & Technology, 22*, 9, P. 467–475.. DOI:10.1016/j.tifs.2011.04.003.

Sargent, M. (2006). AM final report of the e-research coordinating committee: An Australian e-research strategy and implementation framework. A joint initiative of the Department of Education, Science and Training and the Department of Communications, Information Technology and the Arts. P.1.

Search content management (2011). Collaboration platform definition. Retrieved from http://searchcontentmanagement.techtarget.com/definition/collaboration-platform.

Sharma, V. (n.d.). Waterfall model and Spiral model. *MIS Models (MCA).* Retrieved from http://docslide.us/documents/miswaterfall-and-spiral-model.html

Silpa, C.S. and Basha, S.M. (2013). A Comparative Analysis of Scheduling Policies in Cloud Computing Environment. *International Journal of Computer Applications, 67,* 20. Retrieved from http://research.ijcaonline.org/volume67/number20/pxc3887228.pdf

Sloan,S. and Arrison T. (2011). Examining core elements of international research collaboration: Summary of a workshop. *National Academy press,* P.3-8. Retrieved from http://www.nap.edu/catalog/13192/examining-core-elements-of-international-research-collaboration-summary-of-a

Snyder, B. (2010). Server virtualization has stalled, despite the hype. *InfoWorld*. Retrieved from http://www.infoworld.com/article/2624771/server-virtualization/server-virtualization-has-stalled--despite-the-hype.html

Sowjanya,T., Praveen, D., Satish, K. and Rahiman, A. (2011). The queuing theory in cloud computing to reduce the waiting time. *International Journal of Computer Science, Engineering and Technology, 1,* 3, P.110-112.

Strachey C. (1959). Time sharing in large fast computers. *Proceedings of the International Conference on Information processing, UNESCO* paper B.2.19, P. 336–341.

Sturgul, J.R. (2000). Mine design: Examples using simulation”, *SME*, P. vi.

Subramanian K. (2009). Recession is good for cloud computing – Microsoft Agrees. Retrieved from http://www.cloudave.com/link/recession-is-good-for-cloud-computing-microsoft-agrees

Tsoi, A. (2004). Capital City Workshops. Retrieved from www.arc.gov.au/pdf/e-Research_slides3.pdf

U.S. Patent (1994). A network […] is shown schematically as a cloud", U.S. Patent 5,485,455, column 17, line 22, filed Jan 28, 1994. Retrieved from http://www. google. com/ patents?vid=5790548

UNESCO (2009). A digital infrastructure linking African and Arab region universities to global knowledge. Retrieved from http://unesdoc.unesco.org/images/0018/001832/183218e.pdf

University of Bristol (2013). The high-tech future of healthcare: A digital health assistant in your home. Retrieved from http://www.bristol.ac.uk/news/2013/9358.html

Vasani, P.P. and Sanghani, N.S. (2012). Literature review: Various priority based task scheduling algorithms in cloud computing. *Journal of Information, Knowledge and Research in Computer Engineering, 02,*02, 298-302. Retrieved from http://www.ejournal.aessangli.in/ASEEJournals/CE60.pdf

Wang, Q., Kanemasa , Y., Li, J., Jayasinghe, D., Kawaba, M. and Pu, C. (2012). Response time reliability in cloud environments: An empirical study of n-tier applications at high resource utilization**.** Retrieved from www.cc.gatech.edu/~qywang/papers/SRDS12Wang.pdf

Wang Y. and Ngolah, C.F. (2008). An operational semantics of real-time process algebra (RTPA). *Int’l Journal of Cognitive Informatics and Natural Intelligence, 2,* 3, P.71-89

Whitehead, J. (2007). Collaboration in software engineering: A roadmap. University of California, Santa Cruz, USA. Retrieved from Users.soe.ucsc.edu/~ejw/papers/whitehead

Wikipedia (n.d.). Digital signature. Retrieved from http://en.wikipedia.org/wiki/Digital_signature

Wikipedia (n.d.). Cloud Computing . Retrieved from https://en.wikipedia.org/wiki/Cloud_computing

Xi, S. (2014). *Real-time virtualization and cloud computing*. All Theses and Dissertations (ETDs), Paper 1366. Retrieved from http://openscholarship.wustl.edu/cgi/viewcontent.cgi?article=2366&context=etd

Yang, B., Tan, F., Dai, Y. and Guo, S. (2009). Performance evaluation of cloud service considering fault recovery. *Proc. First Int'l Conf. Cloud Computing (CloudCom '09),* P. 571-576.

Zhang, J. (n.d.). International research collaboration between different universities at a distance: Analysing e-reseach networks. Retrieved from http://portal.unesco.org/education/en/files/58023/12245997955Zhang.pdf /Zhang.pdf

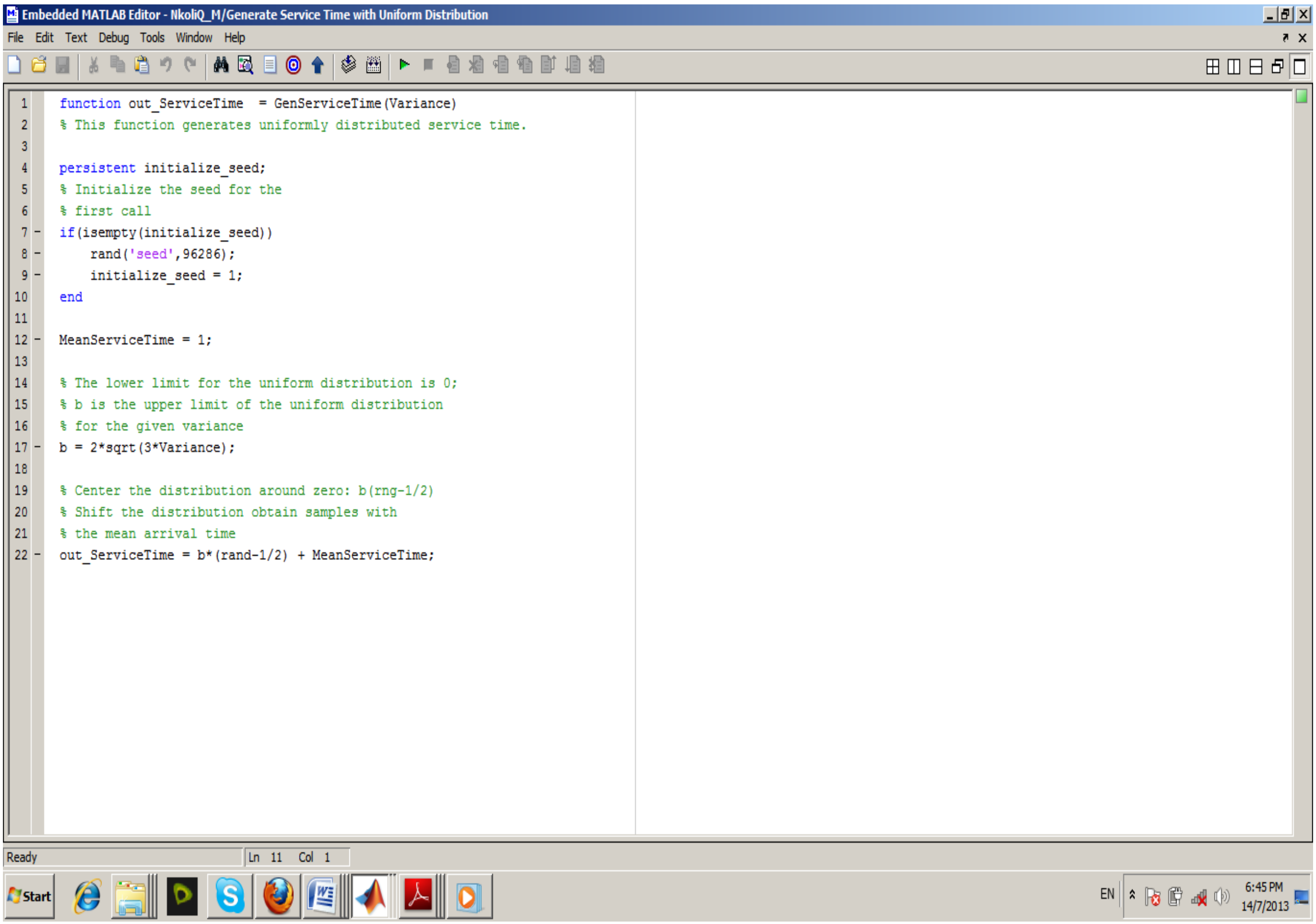
Zhang, Q., Cheng, L., and Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges". *J Internet Serv Appl 1. The Brazilian Computer Society*, p. 7–18

Zhou, F., Goel, M., Desnoyers, P. and Sundaram, R. (2011). Scheduler vulnerabilities and coordinated attacks in cloud computing. *Journal of Computer Security*, Retrieved from http://www.ccs.neu.edu/home/pjd/papers/nca11-xen.pdf

Appendix 1A: Publications from the Research Work

1. Modeling a Conceptual Framework for Secure Research Collaboration in Engineering and Technology by Nwazor N.O. and Inyiama H.C. IJCET Vol.4 No 5, pp 3161-3166
2. Model of a Job Traffic Queue of a Cloud-based Research Collaboration Platform by Inyiama H.C. AND Nwazor N.O. IJCET Vol. 4, No 5, pp3167-3171

Appendix A: MATLAB Scripts for CRCM Queue Model



```
function out_ServiceTime  = GenServiceTime(Variance)
% This function generates uniformly distributed service time.

persistent initialize_seed;
% Initialize the seed for the
% first call
if(isempty(initialize_seed))
    rand('seed',96286);
    initialize_seed = 1;
end

MeanServiceTime = 1;

% The lower limit for the uniform distribution is 0;
% b is the upper limit of the uniform distribution
% for the given variance
b = 2*sqrt(3*Variance);

% Center the distribution around zero: b(rng-1/2)
% Shift the distribution obtain samples with
% the mean arrival time
out_ServiceTime = b*(rand-1/2) + MeanServiceTime;
```

Service Time Plot

```
function out_ServiceTime  = GenServiceTime(Variance)
% This function generates uniformly distributed service time.
 persistent initialize_seed;
% Initialize the seed for the
% first call
if(isempty(initialize_seed))
    rand('seed',96286);
    initialize_seed = 1;
end
 MeanServiceTime = 1;
 % The lower limit for the uniform distribution is 0;
% b is the upper limit of the uniform distribution
```

```
% for the given variance
b = 2*sqrt(3*Variance);
 % Center the distribution around zero: b(rng-1/2)
% Shift the distribution obtain samples with
% the mean arrival time
out_ServiceTime = b*(rand-1/2) + MeanServiceTime;
        'CLOUD QUEUEING- ARRIVAL RATES Vs ARRIVAL JOBS'
function createfigure(YMatrix1)
%CREATEFIGURE(YMATRIX1)
%  YMATRIX1:  matrix of y data
 %  Auto-generated by MATLAB on 14-Jul-2013 19:59:34
 % Create figure
figure1 = figure;
 % Create axes
axes1 = axes('Parent',figure1);
box(axes1,'on');
grid(axes1,'on');
hold(axes1,'all');
 % Create multiple lines using matrix input to plot
plot1 = plot(YMatrix1,'Parent',axes1);
set(plot1(1),'DisplayName','Avg_Arrival_Rate(1:75630,1)');
set(plot1(2),'DisplayName','Avg_Arrival_Rate(1:75630,2)');
 % Create xlabel
xlabel({'ARRIVAL JOBS/QUEUES'});
 % Create ylabel
ylabel({'CRCM ARRIVAL RATES'});
 % Create title
title({'CLOUD QUEUEING- ARRIVAL RATES Vs ARRIVAL JOBS'});
 % Create colorbar
```

```
colorbar('peer',axes1);
 % Create legend
legend(axes1,'show');

'CLOUD QUEUING CONTENT  Vs QUEUING JOBS (ARRIVALS)'
function createfigure(Queue_Content1)
%CREATEFIGURE(QUEUE_CONTENT1)
%  QUEUE_CONTENT1:  vector of y data
 %  Auto-generated by MATLAB on 14-Jul-2013 20:02:27
 % Create figure
figure1 = figure;
 % Create axes
axes1 = axes('Parent',figure1);
box(axes1,'on');
grid(axes1,'on');
hold(axes1,'all');
 % Create plot
plot(Queue_Content1,'Parent',axes1,'DisplayName','Queue_Content');
 % Create xlabel
xlabel({'QUEUING JOBS (ARRIVALS)'});
 % Create ylabel
ylabel({'Cloud Queuing Content (FIFO Buffer)'});
 % Create title
title({'CLOUD QUEUING CONTENT  Vs QUEUING JOBS (ARRIVALS)'});
 % Create legend
legend(axes1,'show');
 % Create colorbar
colorbar('peer',axes1);
'CLOUD QUEUING WORKLOAD RESPONSE (Nominal Throughput)
```

```
function createfigure(Queue_Workload1)
%CREATEFIGURE(QUEUE_WORKLOAD1)
%  QUEUE_WORKLOAD1:  vector of y data
%  Auto-generated by MATLAB on 14-Jul-2013 19:56:15
% Create figure
figure1 = figure;
% Create axes
axes1 = axes('Parent',figure1);
box(axes1,'on');
grid(axes1,'on');
hold(axes1,'all');
 % Create plot
plot(Queue_Workload1,'Parent',axes1,'DisplayName','Queue_Workload');
 % Create xlabel
xlabel({'CLOUD QUEUEING JOBS/ARRIVALS'});
% Create ylabel
ylabel({'QUEUEING WORKLOAD/Nominal throughput Characteristic'});
% Create title
title({'CLOUD QUEUING WORKLOAD RESPONSE (Nominal Throughput)'});
% Create legend
legend(axes1,'show');
'SERVER UTILIZATION RESPONSE Vs QUEUING JOBS'
function createfigure(Server_Utilization1)
%CREATEFIGURE(SERVER_UTILIZATION1)
%  SERVER_UTILIZATION1:  vector of y data
 %  Auto-generated by MATLAB on 14-Jul-2013 20:14:15
 % Create figure
figure1 = figure;
```

```
 % Create axes
axes1 = axes('Parent',figure1);
box(axes1,'on');
grid(axes1,'on');
hold(axes1,'all');
 % Create plot
plot(Server_Utilization1,'Parent',axes1,'DisplayName','Server_Utilization');
 % Create xlabel
xlabel({'CLOUD QUEUING JOB/ARRIVALS'});
 % Create ylabel
ylabel({'SERVER UTILIZATION INDEX'});
 % Create title
title({'SERVER UTILIZATION RESPONSE Vs QUEUING JOBS'});
 % Create legend
legend(axes1,'show');
```

CLOUD QUEUE SERVER SERVICE TIME Vs QUEUING JOBS/ARRIVALS

```
function createfigure(Service_Time1)
%CREATEFIGURE(SERVICE_TIME1)
%  SERVICE_TIME1:  vector of y data
%  Auto-generated by MATLAB on 14-Jul-2013 20:26:13
% Create figure
figure1 = figure;
 % Create axes
axes1 = axes('Parent',figure1);
box(axes1,'on');
grid(axes1,'on');
hold(axes1,'all');
 % Create plot
```

```
plot(Service_Time1,'Parent',axes1,'DisplayName','Service_Time');
 % Create xlabel
xlabel({'CLOUD QUEUEING JOBS/ARRIVALS'});
 % Create ylabel
ylabel({'SERVER SERVICE TIME (TS)'});
 % Create title
title({'CLOUD QUEUE SERVER SERVICE TIME Vs QUEUING JOBS/ARRIVALS'});
 % Create legend
legend(axes1,'show');
```

Appendix B: CRCM Summary Codelisting for Apache Scripting Server

Chat.php

```
<link rel="stylesheet" type="text/css" href="chatfiles/chatstyle.css" />
<div id="chatarea">
 <div id="chatrooms">
<?php
include('chatfiles/setchat.php');
echo $chatS->chatRooms();         // add the chat rooms
?>
 </div>
 <div id="chatwindow"><div id="chats"></div><div
id="chatusers"></div></div>
<div id="playchatbeep"><img src="chatex/playbeep2.png" width="25"
height="25" alt="chat beep" id="playbeep" onclick="setPlayBeep(this)"
/><span id="chatbeep"></span></div>
<?php echo $chatS->chatForm().jsTexts($lsite); ?>
<script type="text/javascript"
src="chatfiles/chatfunctions.js"></script><noscript><a
href="http://coursesweb.net/php-mysql/" title="PHP-MySQL Course">PHP-
MySQL Course</a></noscript>
</div>
```

Test.php

```
<?php if(!isset($_SESSION)) session_start(); ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>CRCM - Team Connections</title>
```

```
<meta name="description" content="Script for chat simple, online, made with
php and javascript, ajax" />
<meta name="keywords" content="chat script, chat simple, chat online" />
<link rel="stylesheet" type="text/css" href="chatfiles/chatstyle.css" />
</head>
<body>
<?php include('chat.php'); ?>
<a href="../index.php"><< Back</a>
</body>
</html>
```

Chat_form.php

```
<form action="" method="post" id="form_chat" onsubmit="return
addChatS(this)">
<div id="name_code">
<?php
echo '<input type="hidden" name="chatroom" id="chatroom" value="'. $this-
>chatrooms[0]. '" />';
// if not set for logged users ($chatadd is 1) add field to set the name
// else, if $chatuser not empty, sets the field with the name, hidded
if($this->chatadd === 1) {
 echo $this->lsite['addnmcd'].':   <em id="code_ch">'. substr(md5(date("
j-F-Y, g:i a ")), 3, 4). '</em><br />'.
  $this->lsite['name'].': <input type="text" name="chatuser" id="chatuser"
size="12" maxlength="12" /> '.
  $this->lsite['code'].': <input type="text" name="cod" id="cod" size="4"
maxlength="4" />  
  <input type="button" name="set" id="set" value="Set"
onclick="setNameC(this.form)" />';
}
```

```
else if(defined('CHATUSER')) {
 echo '<input type="hidden" name="chatuser" id="chatuser" value="'.
CHATUSER. '" />
  <span id="enterchat" onclick="enterChat()">'.sprintf($this->lsite['enterchat'],
CHATUSER).'</span>';
}
// Add or not button for links
$alink = (CHATLINK === 1) ? '<img src="chatex/url.png" alt="URL"
title="Link" onclick="setUrl(\'adchat\');" />' : '';
?>
</div>
 <div id="chatadd">
  <div id="chatex">
  <img src="chatex/bold.png" alt="B" onclick="addChatBIU('[b]','[/b]',
'adchat');" />
  <img src="chatex/italic.png" alt="I" onclick="addChatBIU('[i]','[/i]', 'adchat');"
/>
  <img src="chatex/underline.png" alt="U" onclick="addChatBIU('[u]','[/u]',
'adchat');" />
  <?php echo $alink; ?>
   
  <img src="chatex/0.gif" alt=":)" title=":)" onclick="addSmile(':)', 'adchat');" />
  <img src="chatex/1.gif" alt=":(" title=":(" onclick="addSmile(':(', 'adchat');" />
  <img src="chatex/2.gif" alt=":P" title=":P" onclick="addSmile(':P', 'adchat');"
/>
  <img src="chatex/3.gif" alt=":D" title=":D" onclick="addSmile(':D', 'adchat');"
/>
  <img src="chatex/4.gif" alt=":S" title=":S" onclick="addSmile(':S', 'adchat');"
/>
```

```
  <img src="chatex/5.gif" alt=":O" title=":O" onclick="addSmile(':O', 'adchat');" />
  <img src="chatex/6.gif" alt=":=)" title=":=)" onclick="addSmile(':=)', 'adchat');" />
  <img src="chatex/7.gif" alt=":|H" title=":|H" onclick="addSmile(':|H', 'adchat');" />
  <img src="chatex/8.gif" alt=":X" title=":X" onclick="addSmile(':X', 'adchat');" />
  <img src="chatex/9.gif" alt=":-*" title=":-*" onclick="addSmile(':-*', 'adchat');" /></div>
  <input type="text" name="adchat" id="adchat" size="88" maxlength="200" />  
  <input type="submit" value="<?php echo $this->lsite['chat']; ?>" id="submit"/>
  <div id="logoutchat" onclick="delCookie('name_c')"><?php echo $this->lsite['logoutchat']; ?></div>
  <a href="http://coursesweb.net/" title="Web Programming Development Courses" target="_blank" id="mp">CoursesWeb.net</a>
 </div>
</form>
```

Users.php

```
include("connect.inc.php");
include("config.inc.php");
include("utils.inc.php");
include("lang.inc.php");
if (!isset($username))
	$username = "";
if (!isset($password))
	$password = "";
```

```
/* security checks */
$username = stripslashes(urldecode($username));
header("Expires: Sun, 28 Dec 1997 09:32:45 GMT");
header("Cache-Control: no-cache, must-revalidate");
header("Pragma: no-cache");
header("Refresh: " . $refresh_users_every); /* refresh users' list every
$refresh_users_every seconds */
echo "<!DOCTYPE html PUBLIC \"-//W3C//DTD HTML 4.0
Transitional//EN\">\r\n";
echo "<HTML>\r\n";
echo "<HEAD>\r\n";
echo "<LINK REL=\"stylesheet\" HREF=\"style.css.php\"
TYPE=\"text/css\">\r\n";
echo "</HEAD>\r\n\r\n";
echo "<BODY CLASS=\"users\">\r\n";
/* delete users inactive for 2 minutes */
$sent_on = date("YmdHis", time() - 300);    /* 5 minutes ago*/
$query = "UPDATE users SET active = 'n' WHERE sent_on < '$sent_on'";
do_the_query($chat_db, $query);
echo "<DIV CLASS=\"center\">\r\n";
echo "<A HREF=\"logoff.php\" CLASS=\"chat\"
TARGET=\"_top\">$lang[logoff]</A><BR> <BR>\r\n";
$query = "SELECT * FROM users WHERE active = 'y'";
$result = do_the_query($chat_db, $query);
echo "<SPAN CLASS=\"notes\">\r\n";
echo "<B>" . mysql_num_rows($result) . " $lang[users_in_chat]
:</B><BR>\r\n";

while ($row = mysql_fetch_array($result)) {
```

```
        echo strip_tags(stripslashes($row["username"])) . "<BR>\r\n";
}
if (mysql_num_rows($result) == 0) {
        echo "No user in the chat<BR>\r\n";
}
mysql_free_result($result);

echo "<p ALIGN=\"left\"><B><img src=\"images/sad.gif\" alt=\":(\"/>= :( or
=(<BR />\r\n";
echo "<img src=\"images/cry.gif\" alt=\";(\"/>= ;( <BR />\r\n";
echo "<img src=\"images/mad.gif\" alt=\":@\"/>= :@ <BR />\r\n";
echo "<img src=\"images/smile.gif\" alt=\":)\"/>= :) or =) <BR />\r\n";
echo "<img src=\"images/laugh.gif\" alt=\":D\"/>= :D or :d <BR />\r\n";
echo "<img src=\"images/tongue.gif\" alt=\":P\"/>= :P or :p <BR />\r\n";
echo "<img src=\"images/shocked.gif\" alt=\":O\"/>= :O or :o <BR />\r\n";
echo "<img src=\"images/wink.gif\" alt=\";(\"/>= ;) <BR />\r\n";
echo "<img src=\"images/cry.gif\" alt=\";(\"/>= ;( <BR />\r\n";
echo "<img src=\"images/sick.gif\" alt=\":S\"/>= :S or :s <BR />\r\n";
echo "<img src=\"images/love.gif\" alt=\"8)\"/>= 8)<BR />\r\n";
echo "<img src=\"images/half-frown.gif\" alt=\":/\"/>= :/<BR />\r\n";
echo "<img src=\"images/roll.gif\" alt=\":roll:\"/>= :roll: <BR />\r\n";
echo "</B></P>\r\n";

echo "</SPAN>\r\n";

echo "<SPAN CLASS=\"smallwhite\">$lang[comments] <A
HREF=\"mailto:$mail_address\"
CLASS=\"chat\">$mail_address</A></SPAN>\r\n";
echo "</DIV>\r\n";
```

```
echo "</BODY>\r\n";
echo "</HTML>\r\n";
?>
```

Talk.php

```
include("connect.inc.php");
include("config.inc.php");
include("utils.inc.php");
include("lang.inc.php");
if (!isset($username))
	$username = "";
if (!isset($password))
	$password = "";

/* security checks */
$username = stripslashes(urldecode($username));
header("Expires: Sun, 28 Dec 1997 09:32:45 GMT");
header("Cache-Control: no-cache, must-revalidate");
header("Pragma: no-cache");
header("Refresh: $talk_refresh");	/* update this page every talk_refresh seconds */
/* delete old messages from our db */
$sent_on = date("YmdHis", time() - 60 * 60 * 24 * $msg_delete);
$query = "DELETE FROM msg WHERE sent_on < '$sent_on'";
do_the_query($chat_db, $query);
echo "<HTML>";
echo "<HEAD>";
echo "<LINK REL=\"stylesheet\" HREF=\"style.css.php\" TYPE=\"text/css\">\r\n";
echo "</HEAD>";
```

```
echo "<BODY CLASS=\"talk\">\r\n";
echo "<FONT FACE=\"verdana, arial, helvetica, sans-serif\">\r\n";
$query = "SELECT username, color, msg, HOUR(sent_on) AS hour, MINUTE(sent_on) AS minutes, SECOND(sent_on) AS seconds FROM msg ORDER BY sent_on desc LIMIT $display_limit";
$result = do_the_query($chat_db, $query);
while ($row = mysql_fetch_array($result)) {
        switch ($row["color"]) {
                case "1":
                        $colore = $white;   /* white */
                        break;
                case "2":
                        $colore = $yellow; /* yellow */
                        break;
                case "3":
                        $colore = $red;       /* red */
                        break;
                case "4":
                        $colore = $green;/* green */
                        break;
                case "5":
                        $colore = $blue;     /* blue */
                        break;
                case "6":
                        $colore = $brown;/* brown */
                        break;
                case "7":
                        $colore = $violet;/* violet */
                        break;
```

```
            case "8":
                $colore = $darkyellow;    /* light red */
                break;
            case "9":
                $colore = $black;/* black */
                break;
            default:
                $colore = $white;   /* white (by default) */
                break;
        }
        $hour = ($row["hour"] < 10 ? "0" . $row["hour"] : $row["hour"]);
        $minutes = ($row["minutes"] < 10 ? "0" . $row["minutes"] :
$row["minutes"]);
        $seconds = ($row["seconds"] < 10 ? "0" . $row["seconds"] :
$row["seconds"]);

        /* which kind of message is this? */
        if ($row["username"] == "") {
            echo "<FONT COLOR=$gray SIZE=1>[" .
strip_tags(stripslashes($row["msg"])) . " $lang[chat_enter] -
$hour:$minutes:$seconds]</FONT><BR>\r\n";
        }
        else {
            $row["msg"] = strip_tags($row["msg"]);
            $row["msg"] = str_replace("=(","<img src=\"images/sad.gif\"
alt=\"=(\"/>", $row["msg"]);
            $row["msg"] = str_replace(":(","<img src=\"images/sad.gif\"
alt=\":(\"/>", $row["msg"]);
```

```
$row["msg"] = str_replace(";(","<img src=\"images/cry.gif\" alt=\";(\"/>", $row["msg"]);
$row["msg"] = str_replace(":@","<img src=\"images/mad.gif\" alt=\":@\"/>", $row["msg"]);
$row["msg"] = ereg_replace(":)","<img src=\"images/smile.gif\" alt=\":)\"/>", $row["msg"]);
$row["msg"] = ereg_replace("=)","<img src=\"images/smile.gif\" alt=\"=)\"/>", $row["msg"]);
$row["msg"] = ereg_replace(":D","<img src=\"images/laugh.gif\" alt=\":D\"/>", $row["msg"]);
$row["msg"] = ereg_replace(":d","<img src=\"images/laugh.gif\" alt=\":d\"/>", $row["msg"]);
$row["msg"] = ereg_replace(":p","<img src=\"images/tongue.gif\" alt=\":p\"/>", $row["msg"]);
$row["msg"] = ereg_replace(":P","<img src=\"images/tongue.gif\" alt=\":P\"/>", $row["msg"]);
$row["msg"] = ereg_replace(":O","<img src=\"images/shocked.gif\" alt=\":O\"/>", $row["msg"]);
$row["msg"] = ereg_replace(":o","<img src=\"images/shocked.gif\" alt=\":o\"/>", $row["msg"]);
$row["msg"] = ereg_replace(";)","<img src=\"images/wink.gif\" alt=\";)\"/>", $row["msg"]);
$row["msg"] = ereg_replace(":S","<img src=\"images/sick.gif\" alt=\":S\"/>", $row["msg"]);
$row["msg"] = ereg_replace(":s","<img src=\"images/sick.gif\" alt=\":s\"/>", $row["msg"]);
$row["msg"] = ereg_replace("8)","<img src=\"images/love.gif\" alt=\":s\"/>", $row["msg"]);
```

```
		$row["msg"] = ereg_replace(":/","<img src=\"images/half-frown.gif\" alt=\":s\"/>", $row["msg"]);
		$row["msg"] = ereg_replace(":roll:","<img src=\"images/roll.gif\" alt=\":roll:\"/>", $row["msg"]);
		echo "<FONT COLOR=$gray SIZE=1>($hour:$minutes:$seconds)</FONT>\r\n";
		echo "<FONT COLOR=$colore SIZE=4><B>[" . strip_tags(stripslashes($row["username"])) . "] :</B> " . $row["msg"] . "</FONT><BR>\r\n";
	}
}
mysql_free_result($result);
/* this user is active: save it! */
$query = "UPDATE users SET active = 'y', sent_on = DATE_ADD(NOW(), INTERVAL $diff_timezone HOUR) WHERE username = '$username'";
do_the_query($chat_db, $query);
echo "</FONT>";
echo "</BODY>";
echo "</HTML>\r\n";
?>
```

Sendmsg.php

```
include("connect.inc.php");
include("config.inc.php");
include("utils.inc.php");
include("lang.inc.php");
ignore_user_abort();
if (!isset($username))
	$username = "";
if (!isset($password))
```

```
	$password = "";
if (!isset($color))
	$color = "1";
/* security checks */
$username = stripslashes(urldecode($username));
$password = stripslashes(urldecode($password));
/* is the login valid? */
$query = "SELECT * FROM users WHERE username = '$username' AND
password = '$password'";
$result = do_the_query($chat_db, $query);
if (mysql_num_rows($result) != 0) {   /* login ok! */
	$failed_data = false;
}
else
	$failed_data = true;

if ($failed_data) {  /* sorry, there is an error*/
	header("Location: index.php?username=$username");
	exit;
}
if (!isset($color)) {
	$color = random(6);
}
if (isset($msg)) {
	$query = "INSERT INTO msg(username, msg, color, sent_on) VALUES
('$username', '" . htmlspecialchars(addslashes($msg)) . "', '$color',
DATE_ADD(NOW(), INTERVAL $diff_timezone HOUR))";
	do_the_query($chat_db, $query);
```

```
	/* update the user so that he is not thrown out of the chat */
	$query = "UPDATE users SET active = 'y', sent_on = DATE_ADD(NOW(), INTERVAL $diff_timezone HOUR) WHERE username = '$username'";
	do_the_query($chat_db, $query);
}
echo "<!DOCTYPE html PUBLIC \"-//W3C//DTD HTML 4.0 Transitional//EN\">\r\n";

echo "<HTML>\r\n";
echo "<HEAD>\r\n";
echo "<LINK REL=\"stylesheet\" HREF=\"style.css.php\" TYPE=\"text/css\">\r\n";
echo "</HEAD>\r\n\r\n";

echo "<BODY CLASS=\"snd_msg\">\r\n";

echo "<FORM ACTION=\"$PHP_SELF\" METHOD=\"get\">\r\n";
echo "<INPUT TYPE=\"hidden\" NAME=\"username\" VALUE=\"" . urlencode(addslashes($username)) . "\">\r\n";
echo "<INPUT TYPE=\"hidden\" NAME=\"password\" VALUE=\"$password\">\r\n";
echo "<SPAN CLASS=\"white\">$lang[your_msg] :</SPAN><BR>\r\n";
echo "<INPUT TYPE=\"text\" NAME=\"msg\" VALUE=\"\" SIZE=\"40\" MAXLENGTH=\"255\">\r\n";
echo "<SELECT NAME=\"color\" SIZE=\"1\">\r\n";
echo "<OPTION " . ($color == "1" ? "SELECTED" : "") . " VALUE=\"1\">$lang[white]</OPTION>\r\n";
```

```
echo "<OPTION " . ($color == "2" ? "SELECTED" : "") . " VALUE=\"2\">$lang[yellow]</OPTION>\r\n";
echo "<OPTION " . ($color == "3" ? "SELECTED" : "") . " VALUE=\"3\">$lang[red]</OPTION>\r\n";
echo "<OPTION " . ($color == "4" ? "SELECTED" : "") . " VALUE=\"4\">$lang[green]</OPTION>\r\n";
echo "<OPTION " . ($color == "5" ? "SELECTED" : "") . " VALUE=\"5\">$lang[blue]</OPTION>\r\n";
echo "<OPTION " . ($color == "6" ? "SELECTED" : "") . " VALUE=\"6\">$lang[brown]</OPTION>\r\n";
echo "<OPTION " . ($color == "7" ? "SELECTED" : "") . " VALUE=\"7\">$lang[violet]</OPTION>\r\n";
echo "<OPTION " . ($color == "8" ? "SELECTED" : "") . " VALUE=\"8\">$lang[light_red]</OPTION>\r\n";
echo "<OPTION " . ($color == "9" ? "SELECTED" : "") . " VALUE=\"9\">$lang[black]</OPTION>\r\n";

echo "</SELECT>\r\n";
echo "<INPUT TYPE=\"submit\" VALUE=\"$lang[send]\">\r\n";
echo "</FORM>\r\n";
echo "</BODY>\r\n";
echo "</HTML>\r\n";
?>
```

Admin.php

```
session_start();
// admin.php - administration functions for admin users
// check for valid session
// includes
include('odm-load.php');
```

```
include('udf_functions.php');
if (!isset($_SESSION['uid']))
{
   header('Location:index.php?redirection=' . urlencode( $_SERVER['PHP_SELF'] . '?' . $_SERVER['QUERY_STRING'] ) );
   exit;
}
// open a connection to the database
$user_obj = new User($_SESSION['uid'], $GLOBALS['connection'], DB_NAME);
$secureurl = new phpsecureurl;
// Check to see if user is admin
if(!$user_obj->isAdmin())
{
   header('Location:error.php?ec=4');
   exit;
}
$last_message = (isset($_REQUEST['last_message']) ? $_REQUEST['last_message'] : '');
draw_header(msg('label_admin'), $last_message);
?>
   <table border="1" cellspacing="5" cellpadding="5" >
      <th bgcolor ="#83a9f7"><font color="#FFFFFF"><?php echo msg('users')?></font></th><th bgcolor ="#83a9f7"><font color="#FFFFFF"><?php echo msg('label_department')?></font></th><th bgcolor ="#83a9f7"><font color="#FFFFFF"><?php echo msg('category')?></font></th><?php if($user_obj->isRoot()) echo '<th bgcolor ="#83a9f7"><font color="#FFFFFF">' . msg('file') . '</th></font>'; ?>
      <?php
```

```
if($user_obj->isRoot())
    udf_admin_header();
?>
<tr>
    <td>
        <!-- User Admin -->
        <table border="0">
            <tr>
                <td><b><a href="<?php echo $secureurl->encode('user.php?submit=adduser&state=' . ($_REQUEST['state']+1)); ?>"><?php echo msg('label_add')?></a></b></td>
            </tr>
            <tr>
                <td><b><a href="<?php echo $secureurl->encode('user.php?submit=deletepick&state=' . ($_REQUEST['state']+1)); ?>"><?php echo msg('label_delete')?></a></b></td>
            </tr>
            <tr>
                <td><b><a href="<?php echo $secureurl->encode('user.php?submit=updatepick&state=' . ($_REQUEST['state']+1)); ?>"><?php echo msg('label_update')?></a></b></td>
            </tr>
            <tr>
```

```
<td><b><a href="<?php echo $secureurl->encode('user.php?submit=showpick&state=' . ($_REQUEST['state']+1)); ?>"><?php echo msg('label_display')?></a></b></td>
          </tr>
        </table>
      </td>
      <td>
        <!-- Department Admin -->
        <table border="0">
          <tr>
            <td><b><a href="<?php echo $secureurl->encode('department.php?submit=add&state=' . ($_REQUEST['state']+1)); ?>"><?php echo msg('label_add')?></a></b></td>
          </tr>
          <tr>
            <td><b><a href="<?php echo $secureurl->encode('department.php?submit=deletepick&state=' . ($_REQUEST['state']+1)); ?>"><?php echo msg('label_delete')?></a></b></td>
          </tr>
          <tr>
            <td><b><a href="<?php echo $secureurl->encode('department.php?submit=updatepick&state=' .
```

```
($_REQUEST['state']+1));                    ?>"><?php                    echo
msg('label_update')?></a></b></td>

                </tr>

                <tr>

                    <td><b><a        href="<?php        echo        $securl-
>encode('department.php?submit=showpick&state='                                .
($_REQUEST['state']+1));                    ?>"><?php                    echo
msg('label_display')?></a></b></td>

                </tr>

            </table>

        </td>

<td>

    <!-- Category Admin -->

    <table border="0">

        <tr>

            <td><b><a        href="<?php        echo        $secureurl-
>encode('category.php?submit=add&state='    .    ($_REQUEST['state']+1));
?>"><?php echo msg('label_add')?></a></b></td>

        </tr>

        <tr>

            <td><b><a        href="<?php        echo        $secureurl-
>encode('category.php?submit=deletepick&state=' . ($_REQUEST['state']+1));
?>"><?php echo msg('label_delete')?></a></b></td>
```

```
        </tr>

        <tr>

          <td><b><a href="<?php echo $secureurl->encode('category.php?submit=updatepick&state=' . ($_REQUEST['state']+1)); ?>"><?php echo msg('label_update')?></a></b></td>

        </tr>

        <tr>

          <td><b><a href="<?php echo $secureurl->encode('category.php?submit=showpick&state=' . ($_REQUEST['state']+1)); ?>"><?php echo msg('label_display')?></a></b></td>

        </tr>

    </table>

</td>

<?php if ( $user_obj->isRoot()  ) { ?>

<td>

    <!-- Root-Only Section -->

    <table border="0" valign="top">

        <tr>

          <td ><b><a href="<?php echo $secureurl->encode('delete.php?mode=view_del_archive&state=' . ($_REQUEST['state']+1)); ?>"><?php echo msg('label_delete_undelete')?></a></b></td>

        </tr>
```

```
        <tr>
          <td><b><a href="<?php echo $secureurl->encode('toBePublished.php?mode=root&state=' . ($_REQUEST['state']+1)); ?>"><?php echo msg('label_reviews')?></a></b></td>
        </tr>
        <tr>
          <td><b><a href="<?php echo $secureurl->encode('rejects.php?mode=root&state=' . ($_REQUEST['state']+1)); ?>"><?php echo msg('label_rejections')?></a></b></td>
        </tr>
        <tr>
          <td><b><a href="<?php echo $secureurl->encode('check_exp.php?&state=' . ($_REQUEST['state']+1)); ?>"><?php echo msg('label_check_expiration')?></a></b></td>
        </tr>
        <tr>
          <td><b><a href="<?php echo $secureurl->encode('file_ops.php?&state=' . ($_REQUEST['state']+1)); ?>&submit=view_checkedout"><?php echo msg('label_checked_out_files')?></a></b></td>
        </tr>
    </table>
</td>
    <?php udf_admin_menu($secureurl); ?>
```

```
</tr>

<tr>

   <td>

      <table>

         <tr>

            <th bgcolor ="#83a9f7"><font color="#FFFFFF"><?php echo
msg('label_settings')?></font></th>

         </tr>

         <tr>

            <td><b><a href="<?php echo $secureurl-
>encode('settings.php?submit=update&state=' . ($_REQUEST['state']+1));
?>"><?php echo msg('adminpage_edit_settings'); ?></a></b></td>

         </tr>

         <tr>

            <td><b><a href="<?php echo $secureurl-
>encode('filetypes.php?submit=update&state=' . ($_REQUEST['state']+1));
?>"><?php echo msg('adminpage_edit_filetypes'); ?></a></b></td>

         </tr>

      </table>

   </td>

    <td>

       <table>
```

```
<tr>
  <th bgcolor ="#83a9f7"><font color="#FFFFFF"><?php echo msg('adminpage_reports');?></font></th>
</tr>
<tr>
  <td><b><a href="<?php echo $secureurl->encode('access_log.php?submit=update&state=' . ($_REQUEST['state']+1)); ?>"><?php echo msg('adminpage_access_log');?></a></b></td>
</tr>
<tr>
  <td><b><a href="reports/file_list.php"><?php echo msg('adminpage_reports_file_list');?></a></b></td>
</tr>
</table>
</td>
</tr>
<?php } ?>
</table>
<?php
if(is_array($GLOBALS['plugin']->getPluginsList()) && $user_obj->isRoot())
{
?>
<table border="1" cellspacing="5" cellpadding="5" >
```

```
<th bgcolor ="#83a9f7"><font color="#FFFFFF"><?php echo msg('label_plugins')?></font></th>
        <tr>
          <td>
      <?php
    //Perform the admin loop section to add plugin menu items
    callPluginMethod('onAdminMenu');
      ?>
          </td>
        </tr>
          </table>
  <?php
}
  ?>
  <?php
draw_footer();
```

Appenddix C

PSEUDOCODES FOR THE CRCM FLOWCHARTS

Pseudocode for Global Model for CRCM

```
Begin()
Display log in page
Accept inputs
        If registered user then
        Display CRCM Home page
        Accept inputs
                If A then
                Go to Research sub-routine
                Return to home page
                Else if B then
                Go to Information management sub-routine
                Return to home page
                Else if C then
                Go to output/industry collaboration sub-routine
                Return to home page
                Else if D then
                Go to Feedback sub-routine
                Return to home page
                Else if E then
                 Display About us page
                Else if log out then
                Exit CRCM
                Else NOP
                End if
        Else go to Registration sub-routine
        Return to log in page
        End if
```

Else NOP

End if

Pseudocode for Research Sub-routine:

Begin ()

Display Research page

Accept inputs

If PF then

Display Project Forum page

Perform desired function

Return to Research page

Else If RV then

Display Research Videos page

Perform desired function

Return to Research page

Else if RP then

Display Research Positions page

Perform desired function

Return to Research page

Else if CH then

Display Challenges page

Perform desired function

Return to Research page

Else if OR then

Display Ongoing Research page

Perform desired function

Return to Research page

Else if PT then

Display Project tools page

Perform desired function

Return to Research page

Else if DA then

Display Data acquisition page

Perform desired function

Return to Research page

Else if Exit then

Exit the sub-routine and return to main routine

Else NOP

End if

Pseudocode for Information Sub-routine:

Begin ()

Display Information management page

Accept inputs

If GR then

Display Grants page

Perform desired function

Return to Information management page

Else If FR then

Display Funded Research page

Perform desired function

Return to Information management page

Else if SC then

Display Scholarship page

Perform desired function

Return to Information management page

Else if SM then

Display Share and manage documents page

Perform desired function

Return to Information management page

Else if RE then

Display Review and Edit Documents page

Perform desired function

Return to Information management page

Else if Exit then

Exit the sub-routine and return to main routine

Else NOP

End if

Pseudocode for Output/Industry Collaboration sub-routine:

Begin ()

Display Output/Industry Collaboration page

Accept inputs

If PU then

Display Publications page

Perform desired function

Return to Output/Industry Collaboration page

Else If IR then

Display Publications page

Perform desired function

Return to Output/Industry Collaboration page

Else if DF then

Display Document and File Backup page

Perform desired function

Return to Output/Industry Collaboration page

Else if CR then

Display Completed Research page

Perform desired function

Return to Output/Industry Collaboration page

Else if Exit then

Exit the sub-routine and return to main routine

Else NOP

End if

Pseudocode Feed Back sub-routine:

Begin ()

Display Feed back page

Accept inputs

If Submit is selected then

If the options are selected then

Process the submission

Display “thank you page”

Else Display “Please select suitable options” then

If Ok is selected then

Return to feedback page

Else NOP

End if

Return to Feedback page

Else if Exit is selected then

Exit the sub-routine and return to main routine

Else NOP

End if

Pseudocode for Registration sub-routine

Begin ()

Display Registration page

Accept the user inputs

Process the inputs

If the inputs are complete then

Check for valid inputs

Else display “incomplete user inputs”

If Ok is selected then

Return to registration page

Else NOP

If the inputs are valid then

If it is a researcher then

Go to researcher’s collaboration level agreement sub-routine

Else if it is an industrialist

Go to industrialist’s collaboration level agreement sub-routine

Else display “invalid user input”

If Ok is selected then

Return to registration page

Else NOP

Else If Exit is selected then

Exit the sub-routine and return to main routine

Else NOP

End if

Pseudocode for Collaboration level Agreement for researchers’ sub-routine:

Begin ()

Display CLA page

If CLA is accepted then

Process the acceptance

Display researcher’s authentication page

Accept user inputs for authentication

Else if CLA is rejected then

Display log in page

If the inputs are valid then

Register the users

Validate the data base

Display “Registration is successful”

Exit sub-routine and return to Registration sub-routine

Else display “invalid user authentication details” then

If Ok is selected then

Return to researcher’s authentication page

Else NOP

end if

End if

Pseudocode for Collaboration level Agreement for industrialist’s sub-routine

Begin ()

Display CLA page

If CLA is accepted then

Process the acceptance

Display industrialist’s authentication page

Accept user inputs for authentication

Else if CLA is rejected then

Display log in page

If the inputs are valid then

Register the users

Validate the data base

Display “Registration is successful”

Exit sub-routine and return to Registration sub-routine

Else display “invalid user authentication details” then

If Ok is selected then

Return to industrialist’s authentication page

Else NOP

end if

End if