UNIVERSITY OF NAIROBI
DEPARTMENT OF COMPUTING AND INFORMATICS

A VENDOR NEUTRAL QUALITY OF SERVICE MONITORING MODEL FOR SOFTWARE AS A SERVICE CLOUD COMPUTING SOLUTIONS

FRANKLINE MAKOKHA

P80/54513/2019

THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF PHD IN COMPUTER SCIENCE

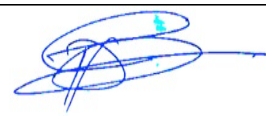
AUGUST 2022

Declaration

Student:

I declare that this Thesis is my original research and to the best of my knowledge, it has not been submitted for a degree award in any institution of learning.

Name : Frankline Makokha

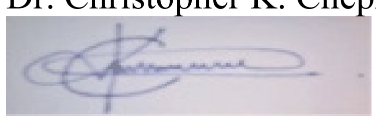
Signature: : 

Date: : 1st August 2022

Supervisors:

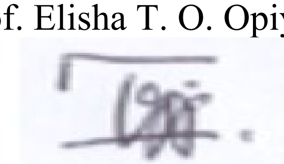
This thesis has been presented for final examination with our endorsement as the University supervisors.

Name : Dr. Christopher K. Chepken, PhD

Signature: : 

Date: : 1st August 2022

Name : Prof. Elisha T. O. Opiyo, PhD

Signature : 

Date : 1st August 2022

Dedication

This work is devoted to my parents, Fredrick Makokha and Stella Awino, for their psychological and spiritual support offered during my entire study period.

Acknowledgments

I wish to acknowledge, with thanks, the academic support and oversight offered by the deceased Prof Okello-Odongo, during the embryonic stages of this study, afore his ill-timed demise. May peace be upon him. I as well thank my supervisors, Prof. Opiyo and Dr. Chepken, for the knowledge, skills, techniques, guidance and insights offered, enabling the success of this research. May the Almighty God bless them abundantly. I thank the entire School of Computing fraternity, the PhD Defense Committees for all the academic criticisms that shaped the trajectory of this research. I finally thank Mr. Dennis Wekesa, the JavaScript go to guru, for the tutorials and insights in JavaScript and Node.js programming that augmented my programming knowledge, leading to the success in prototype development for this project.

Abstract

The large number of providers of cloud services, offering comparable solutions marketed at different prices and at distinctive Quality of Service (QoS) levels, portends a decision challenge to users. The users have to make a selection or a comparison between the available providers of cloud services in so far as performance of their cloud solutions is concerned. Even though there exists computational models for developing QoS measuring tools, they are not vendor agnostic therefore hampering cross vendor performance comparison.

To abate the decision challenge and enable cross cloud performance comparison, various research have been done culminating in probable solutions, like the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), Heterogeneous Similarity Metrics (HSM), Event Based Multi Cloud Service Applications Framework, Multiple-Cloud Monitoring platform, Multicloud Security Applications (MUSA) framework, the PeRformance Evaluation of SErvices on the Cloud (PRECENSE) framework and Cross-Layer MultiCloud Application Monitoring with Benchmarking as a Service (CLAMBS).

Whereas there is existence of research meant to address the cross cloud performance comparison, the shortcoming is that they rely on the use of existing vendor specific tools, customized for the specific cloud providers' infrastructure which are then spread across different cloud providers, while in some instances the use of customized software agents installed in various cloud providers' platform, and use of synthetically generated data.

This research addressed the existing gap by developing a cloud QoS monitoring framework from which a vendor agnostic cloud QoS monitoring model was designed. The focus was on Software as a Service (SaaS) cloud computing solutions. In designing of the model, the research focused on the location of the QoS monitoring tool, the intention of monitoring, and the mode of access to the cloud services.

The QoS parameters monitored by the vendor neutral tool were service stability, service response time and service availability, which are the main quantitative parameters for cloud QoS as far as performance is concerned. The tool was subjected to Google docs and Microsoft 365 cloud services for comparison performance, under the same computing platform and Internet conditions.

From the comparison, the average service response time for Google was 4.47 seconds while for Microsoft was 6.04 seconds. Both platforms had an availability of 100% since at no time during the testing period did any of the platform report a platform failure that would have led to outage of services. Whereas the availability is 100%, the fluctuations in the service response time were higher for Microsoft at 5.966 seconds than for Google at 2.003 seconds, meaning the Google platform was more stable than the Microsoft platform. From the trust evaluation, it was noted that the two compared cloud providers, Google and Microsoft, were both trustable since the results they reported were within the confidence interval of those reported by the vendor neutral model.

Further research could be extended to monitor Infrastructure as a Service and Platform as a Service solutions. Advanced studies could also focus on other common aspects used by all cloud providers at the client side, for example the operating system, where the monitoring capability could be installed as a utility on the operating system.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

API	:	Application Programing Interface
ATM	:	Asynchronous Transfer Mode
CDN	:	Content Distribution Network
CLAMBS		Cross-Layer MultiCloud Application Monitoring as well as Benchmarking as a Service
C-MaaS	:	Cloud Migration as a Service
CPU	:	Central Processing Unit
CRM	:	Customer Relationship Management
CSS	:	Cascading Style Sheets
DaaS	:	Desktop as a Service
DARGOS	:	Distributed Architecture for Resource manaGement and mOnitoring in cloudS
DOM	:	Document Object Model
DPI	:	Deep Packet Inspection
EMR	:	Elastic Map Reducer
ETSI		European Telecommunications Standards Institute
EC2	:	Elastic Computer Cloud
FURPS	:	Functionality, Usability, Reliability, Performance and Supportability
GPD	:	Generalized Pareto Distribution model
GSM	:	Global System for Mobile Communication
GUI	:	Graphical User Interface
HEEM		Heterogeneous Euclidean-Eskin Metric
HSM	:	Heterogeneous Similarity Metrics
HELM	:	Heterogeneous Euclidean-Lin Metric
HEGM	:	Heterogeneous Euclidean-Goodall Metric
HEOM	:	Heterogeneous Euclidean-Overlap Metric
HTML	:	Hypertext Markup Language
HVDM	:	Heterogeneous Value Difference Metric
IaaS	:	Infrastructure as a Service

IBM	:	International Business Machines Corporation
ICT	:	Information and Communication Technology
IEEE	:	Institute of Electrical and Electronics Engineers
IM	:	Instant Messaging
IP	:	Internet Protocol
ISO	:	International Organization for Standardization
ITU	:	International Telecommunications Union
KPI	:	Key Performance Indicator
MaaS	:	Cloud Migration as a Service
MMT		Montimage Monitoring Tool
MUSA	:	Multicloud Security Applications
PaaS	:	Platform as a Service
PRESENCE	:	PeRformance Evaluation of SErvices on the Cloud
PoP	:	Point of Presence
QoS	:	Quality of Service
QoSMONaaS	:	QoS MONitoring as a Service Model
RAM	:	Random Access Memory
RANTIP	:	Regional Academic Network on IT Policy
SaaS	:	Software as a Service
SAN	:	Storage Area Network
SEO	:	Search Engine Optimization
SIM Card	:	Subscriber Identity Module card
SLA	:	Service Level Agreement
SoA	:	Service Oriented Architecture
SONET	:	Synchronous Optical Networking
SPI Model	:	Software Platform and Infrastructure model
SRT		Subscription Racing Technology
S3	:	Simple Storage Services,
TOPSIS	:	Technique for Order Preference by Similarity to Ideal Solution
U-CaaS	:	Unified Communication as a Service

VMs	:	Virtual Machines
VoIP	:	Voice over IP services
VPC	:	Virtual Private Cloud
XaaS	:	X as a Service
XCP	:	Xen Cloud Platform
XML	:	Extensible Markup Language

CHAPTER ONE: INTRODUCTION

The desire to optimize existing computing resources, and the ever increasing realm of computation problems, coupled with the general automation of various facets of human life, has catalyzed the need for advanced research into the field of computing in a bid to meet the pressure exerted on computing platforms. This research focused on one such technology developed to ease pressure on the computing platforms, namely, cloud computing.

1.1. Background

The current trends in big data and optimizing problems are exerting pressure on current computing platforms in terms of processing speed and storage capacity. Big data is data that is beyond the computing capability of conformist database platforms by virtue of it being voluminous, at high velocity, and varied in formats that it can not be stored in conformist database architectures (Dumbill, 2012). To derive insights from these big datasets, it is imperative to consider alternative processing and storage platforms.

Big data may also be described as a type of data source that has at minimum three common features: huge data Volume, at extreme Velocity and Varied (Hurwitz, Nugent, Halper & Kaufman, 2013). Misra, Sharma, Gulia and Bana (2014) define big data as datasets so large and unwieldy that conformist database platforms strain to capture, store, share and manage.

According to the International Telecommunications Union (ITU, 2015), the term big data has evolved to involve not only the data itself, but also the means available for manipulation of the data. It defines big data as a paradigm for aiding the collecting, storing, managing, analyzing and visualization, in real-time constraints, of extensive data with diverse characteristics.

This has led to development of various computing technologies namely grid computing, distributed computing, utility computing, parallel computing, cluster computing, and now cloud computing.

Parallel computing refers to solving a size n problem through division of its problem areas into $a \geq 2$ (with $a \in N$) parts and solving using k (with $k \in N$) physical processors, at the same time (Navarro, Kahler & Mateu, 2014); Distributed Computing refers to processing different segments of a program at the same time on two or more computers that are collaborating with each other through a network (Kaur, 2015).

Utility computing has been defined as offering of resources needed for computing, such as computation, storage and services, as a service paid on a metered basis (Mondal & Sarddar, 2015); Cluster computing refers to unified but detached computers working in unison as a combined computing resource (Buyya, 1999).

The computational model of sharing computing resources and solving of computational task in a harmonized, dynamic and cross-institutional virtual organization has been termed as grid computing (Foster, Kesselman & Tuecke, 2001).

The focus of this research, cloud computing, is a standard for facilitating universal, appropriate, demand driven usage access to a communal collection of computational solutions that are quickly provisioned and discharged with little effort or solution provider's intervention (National Institute of Standards and Technology, 2011).

According to Vouk (2008), a key distinguishing segment of a fruitful information technology resides in its capability to become a real, treasured, and inexpensive contributor to computing infrastructure. Cloud computing utilizes the cyber platform and capitalizes on decades of studies in utility computing, virtualization, distributed and grid computing, and of late the worldwide web, software services and networking. These essentially are the driving powers for cloud computing.

Further, a formidable core and facilitating concept is computation via Service Oriented Architectures (SOA) – which provides a unified and coordinated set of functions to users through an arrangement of lightly and tightly coupled tasks or services mostly through the network (Vouk, 2008).

Endrei, Ang, Arsanjani, Chua, Comte, Krogdahl, Luo and Newling (2004) define a SOA as a methodology for developing distributed platforms that brings forth software functionalities as services for client applications and related services.

A Service Oriented Architecture (SOA) can be viewed as a model for unifying a suite of capabilities, mostly over the network and under the administration of different domains of ownership, which are used to provision solutions to business needs, which conform to information technology solutions (Laskey & Laskey, 2009).

This progress in the cloud computing technology has lured more companies into adopting the technology because of reduced cost of initial investment as opposed to actual acquisition of hardware and software platforms. This in return has contributed to a sharp rise in the number of cloud service providers, spawning competition for cloud service users.

To help cloud clients during selection of a cloud service provider from among several providers in the market, there is a need for means through which service users can measure the performance levels offered by the different cloud providers.

Further, in instances where a client uses services from more than one cloud provider, especially for redundancy purposes, the client should be in a position to compare the performance levels in terms of QoS of the services being provided by the two providers.

This research aims to explore the existing framework and models used for monitoring the QoS provided by cloud service providers offering Software as a Service solutions, the limitations of the existing QoS monitoring framework, and the QoS monitoring models derived from the framework. Further, this research intends to investigate ways of overcoming the shortcomings of the existing cloud QoS monitoring models.

1.2. Definition of Research Discipline and Sub Discipline

This research concentrates on the advances in the field of cloud computing, namely, performance monitoring, the existing framework under which performance monitoring is done, existing models used in performance monitoring, challenges and shortcomings of the existing framework and models used to monitor performance in cloud computing solutions.

It explores the challenges faced by developers of cloud performance monitoring tools during the development of the tools as well as during integration with the various cloud service providers.

This research also explores the challenges faced by users during monitoring of the performance of the various cloud services as well as the challenges they may encounter when they need to equate the performance of different cloud services as advertised by cloud offering companies.

The sub discipline of this research is Quality of Service experienced in cloud services, focusing specifically on the Software as a Service (SaaS) model of cloud computing, how the existing cloud QoS monitoring models are used, how they are implemented on the cloud platform, the various examples of tools developed using the various models, critical QoS parameters and how they are measured with an aim to overcome the limitation of the existing cloud QoS monitoring models.

1.3. Problem Statement

With the increase in public cloud offerings, it is difficult for cloud service users to determine which cloud operator is able to meet their desired Quality of Service (QoS) demands, since cloud providers propose same services with the only difference being prices and levels of performance with different characteristics (Mamoun & Ibrahim, 2014).

Further, according to Nazir (2012), amongst biggest challenges faced by cloud users is to appraise the Service Level Agreements (SLAs) of cloud providers. This is due to the fact that most vendors design SLAs to make a self-protective buffer against litigations, yet presenting least guarantees to customers.

An SLA refers to a treaty document or an officially negotiated pact based on the commitment and goals between the cloud operators and their customers (Dash, Saini, Panda & Mishra, 2014). This research defines an SLA as terms of engagement between a service providing entity and the service user that stipulate the expectations and responsibilities of each entity in the SLA.

Cloud consumers face the challenge of business responsibility given that most of the cloud provider's SLA states that a client could get a service credit during settling of the bill if the offered service level falls below a given cut-off value, yet SLAs still lack in realizing several parameters related to user's constraints (Jones, 2010).

Thus, in many cases, the information or business harm to the client is not well catered for. Aceto, Botta, Donato and Pescapè (2013) pointed out that some of the emerging issues and future trend of cloud monitoring include new monitoring tools and techniques, cross layer monitoring, monitoring of federated cloud and monitoring of new network platforms based on clouds. This research aims to advance on new cloud monitoring tools and techniques.

According to a survey done by Regional Academic Network on IT Policy (RANTIP) –Cloud Computing Research Case of Kenya (Cloud User Perspectives) in November 2017, one of the barriers and challenges with respect to adoption of Cloud Computing was sighted as poor services from cloud providers (Omwansa & Walubengo, 2017). Further, lack of control over the cloud servers and staff for SLA's enforcement was sighted as a key barrier to migration to the cloud for most organisation (Omwansa & Walubengo , 2017).

From a baseline survey of cloud computing in Kenya by Omwansa, Waema and Omwenga (2014), whose purpose was to examine the present position of cloud computing uptake in Kenya, ascertain the influence of cloud computing and provide a way forward through various channels, among them white papers, academic paper and policy statements, made as one of their recommendations the need to find ways of enforcing Security, SLA's and Privacy in the cloud.

According to Manuel (2014), trust plays a significant part in commercial cloud service ecosystem and is among the major challenges of cloud technology as it facilitates users in selecting the best resources in a heterogeneous cloud infrastructure.

The Manuel (2014) Trust model computes trust value using four parameters, namely, turnaround efficiency, reliability, availability and data integrity. The privacy of data, the confidentiality of data, and trust establishment are deemed to be the major security concerns for any establishment intending to move its data to the cloud platform (Gholami & Ghobaei-Arani, 2015).

Due to the high competitive nature and the service environment being distributed in cloud computing, the assurances (SLA's) are not enough for the cloud clients to recognise reliable and trustworthy cloud service providers. In view of these hindrances, potential clients are not certain on whether they can trust the cloud service providers in so far as offering dependable services is concerned (Habib, Hauke, Ries & Mühlhäuser , 2012).

According to Odun-Ayo, Ajayi, and Falade (2018), the increase in cloud services usage, has made the quality of cloud services to be an increasingly important matter due to many unresolved challenges that have to be addressed, case in point those that relate to trust and availability. QoS is therefore a matter that requires proper addressing to enhance trust in the cloud.

According to Chekfoung, Kecheng and Sun (2013), features that a SaaS cloud consumer should factor prior to embracing a SaaS solution include, Functionality, which addresses whether the offered SaaS service sufficiently supports the existing business model; Availability which establishes whether the SaaS service delivery can exhibit satisfactory and quantifiable uptime in line with the expected operations of the firm.

Chekfoung et al (2013) also postulate that SaaS users should also consider network performance, which is, whether the SaaS provider support enough network capacity and latency to support acceptable performance to all users; Status visibility to gauge the SaaS provider's capability to submit service performance metrics to the SaaS clients; Service Level Agreements (SLA) to gauge whether the SaaS provider provides a detailed SLA which is inclusive of specific security elements and to determine the SaaS provider's past performance alongside this or similar SLAs for other clients.

Whereas several tools exists for monitoring the QoS offered by cloud providers, most available tools are developed by cloud platform providers for monitoring the QoS of their own cloud services. The results from the cloud provider's tool is what is presented to the cloud user as the level of QoS of the platform. This arrangement does not offer end-to-end QoS since the measured QoS is up to the cloud platform as opposed to being up to the end user.

To compound the problem, the results are stored in the cloud provider's infrastructure, which the user has no visibility over, except to only query for the QoS values. The existing cloud QoS monitoring tools have this limitation because the underlying models from which the tools are developed are platform dependent, meaning the underlying architecture of this models are designed based on the low level architecture of the cloud providers infrastructure.

This raises an issue of trust with regards to the results from the monitoring tool, since the monitoring tool developer is the same entity whose services are being monitored. In addition, for accurate performance comparison, a single tool should be able to monitor more than one cloud provider, with no modifications on the tool or cloud platform. This is not possible with the existing tools as they are not vendor neutral.

1.4. Research Objectives

i. To Develop a high level Client Trustable QoS Monitoring Framework for Cloud Computing Systems.

ii. To Design a Vendor Neutral Cloud QoS Monitoring Model that implements the developed Framework for SaaS Cloud Computing Solutions.

iii. To Prototype and Evaluate a SaaS Cloud QoS Monitoring Tool which is based on the proposed Vendor Neutral Model.

iv. To Develop Algorithms for implementing the proposed Vendor Neutral SaaS Cloud QoS Monitoring Tool.

1.5. Research Questions

i. Why is there lack of trust in the existing cloud QoS Monitoring Framework?

ii. How can a Vendor Neutral SaaS QoS Monitoring Model be realized?

iii. How does a cloud QoS monitoring tool developed from the new Vendor Neutral SaaS Cloud QoS Monitoring Model compare to other existing tools?

iv. How can the Algorithms needed to realize the proposed Vendor Neutral SaaS Cloud QoS Monitoring Tool be derived?

1.6. Significance of the Study

Since there exists an SLA between cloud users and providers of cloud services in cloud computing, it is imperative to monitor and analyze the services being offered (Qi & Gani, 2012).

This study aims to explore existing cloud QoS monitoring models, highlighting how they are implemented, sample tools that have been developed using each model, the limitations of the identified models and how this limitations can be overcome.

The identified limitations will be profiled according to the challenges they present to the user of the cloud services, as well as the challenges they present to developers of cloud QoS monitoring tools.

The information gained will be used to explore ways of developing a model that addresses the identified challenges to both users and developers of cloud QoS monitoring tools.

The insights gained from this research will be handy during Service Level Agreements (SLA's) evaluation by users and providers of cloud services for settlement purposes. Further, tools developed using the new model will build confidence in use of cloud since the cloud user will be able to authenticate the QoS as experienced against what the cloud solution provider alleges to be their QoS.

1.7. Justification

According to Zia and Khan (2012), all cloud computing services are required to be in accord to role out better QoS i.e., to offer enhanced software functionality, meet the user's requests for their preferred processing power and to use enhanced bandwidth.

Due to undependable internet links, different cloud services may receive different quality levels for same cloud services so there is need to select the optimal services (Subha & Banu 2014). Further, according to the same authors, with the speedy growth of cloud computing, several cloud operators have emerged who provide same services at different performance levels and prices.

According to Saravanan and Kantham (2013) from the user's viewpoint, it is not easy to choose which operator is the best to contract and what is the selection rationale. Further, finding out which is the best service from the cloud for a particular application is very challenging and many times defines the achievement of the core business of the clients.

As there exists a lot of cloud service providing companies, cloud providing companies strive to reduce their fees to the lowest it can get so as to attract as many clients as possible. Further, the cloud providing companies have also to provision as many customers as possible on their cloud platforms to ensure profitability.

The more the cloud users are boarded onto the cloud platform, the high the chances that the QoS of the cloud service will decline. Therefore it becomes essential to monitor, track and quantify the performance level of cloud services in order to provide the correct information to both clients and service providers (Firdhous, Hassan & Ghazali, 2013).

According to Firdhous et al (2013), cloud providers need to win the confidence of customers to enable them use their cloud computing platforms. This can only be done if cloud service providers come up with innovative means to provide the QoS demanded by cloud applications and independent means to verify the claims by service providers of meeting the user's QoS.

According to the same authors, the increase in number of public cloud offerings has made it difficult for users to determine which operator can meet their QoS constraints. Cloud providers provision same services on different performance levels and costs and using different parameters.

Cloud monitoring is important to cloud providers because it assists them and cloud software developers to keep their cloud platforms operating at high proficiency, detecting changes in cloud platform performance, taking note of the Service Level Agreement (SLA) contraventions of some metrics, and following the subscription operations of cloud resources as a result of system fails and configuration changes (Alhamazani, Ranjan, Mitra, Rabhi, Jayaraman, Khan, Guabtni & Bhatnagar, 2014).

According to Ardagna, Casale, Ciavotta, Pérez and Wang (2014), whereas the cloud has to a great extent simplified the provisioning process of cloud capacity, it poses various new challenges in the area of QoS administration.

This is also reinforced by Kashyap and Kashyap (2017), who stated Quality of Service (QoS) management to be among the challenges faced by cloud applications, which is stated as the difficulty of allocating cloud resources to the mobile application to ensure high level of service for performance and availability.

Due to the importance of cloud QoS monitoring, all cloud providers have their respective tools to monitor QoS on their cloud platform. To raise the confidence of cloud users, it is imperative to have independent means by which the users can measure and validate the level of QoS reported by a given cloud provider.

1.8. Scope of the Research

This research was limited to the QoS monitoring in SaaS cloud computing applications. Further, the research was limited to quantitative cloud QoS metrics. Given that SaaS services can be accessed through browser or vendor application, this study focused on SaaS solutions that can be accessed via a browser.

1.9. Assumptions

The main assumption in this research was that Cloud Service providers offering trial solutions on their platforms have not over provisioned the trial platforms with more resources than the same service or client would receive under ordinary service usage, thus making the platform perform better under trail than under live usage.

This research as well assumes that network congestion, a network performance parameter that may affect cloud application performance, has been taken care of by the network administrator of the user through use of various congestion control techniqiues like TCP/IP window reduction; Fair queuing in network devices such as routers, switches, and other devices; Priority schemes which transmit higher priority packets ahead of other traffic; and Explicit network resource allocation via admission controls toward specific flows.

1.10. Limitations of the Research

This research was not able to factor in its research design the effect of the location of the Service Provider's servers and the associated Point of Presence (PoP) of the Content Distribution Network (CDN) service providers that may have been contracted by the studied cloud service providers.

Throughput, in the context of software systems, which refers to transactions per second that the application can handle, and is measured by subjecting the application to a mix of frequent, critical, and intensive transactions, a process called load testing, to see how many pass successfully in an acceptable time frame governed by the SLAs was not part of this research.

This is because throughput is measured using specialised tools like the Visual Studio Team System which have capability of simulating a mix of the transactions, simulate network latency, user think times and test iterations. However it is imperative to note that response time, a key metric of this research, is inversely related to throughput, in the sense that increasing throughput of the application reduces the response time. Therefore the results of response time from this research tell on the throughput of the SaaS applications studied.

1.11. Knowledge Contribution

This research enhanced the existing domain knowledge in the field of QoS monitoring in cloud computing solutions. It reviewed the limitations of the existing QoS monitoring framework in the cloud with a view of proposing a better framework. It collated the existing cloud QoS monitoring models used in development of cloud QoS monitoring tools and explicitly derived the existing cloud QoS monitoring framework for analysis.

Based on the collated models and the explicit framework, this research identified the shortcoming of this framework and the shortcomings of the existing models and proposed a new cloud QoS monitoring framework and a new model for SaaS cloud QoS monitoring. This research also developed Algorithms for actualising the proposed cloud QoS Model for Software as a Service solution under the new cloud QoS monitoring framework.

1.12. Operational Definition of Terms

Accuracy	:	Refers to the level of correctness of the QoS results from the cloud provider's tool as compared with results from the vendor neutral tool.
Adaptability	:	Refers to the capability and ease with which the cloud provider can amend or enhance the cloud platform features and services based on user's requests.
Availability	:	Refers to ratio of the number of instances that a user requests for a cloud service and gets the service to the number of times the user requests for the cloud service and does not get the requested service
Reliability	:	Refers to the availability of the service throughout the duration that the user has initiated a service therefore enabling atomic completion of a given task.
Service Response Time	:	Refers to the speed with which the requested cloud service loads (Also called service initiation time)
Stability	:	Refers to the degree of variability in the service response times of the cloud service
Trust	:	Refers to the level of confidence the user has in the services provided by a given cloud service provider

1.13. Chapter Summary

This chapter provided a chronological advancement in computing technologies, leading to cloud computing .The chapter also highlighted the computing problems that were the driving force behind the need for improved computing technologies, key among them being the need to process big data sets.

With the widespread adoption of cloud computing, the chapter noted an emerging research issue, quality of service monitoring, in cloud computing platforms. The chapter highlighted the problem of trust in the current framework of cloud quality of service monitoring between the cloud providers and the reported quality of service values during service level agreement evaluation due to vendor centricity of the quality of service monitoring tools.

From the main identified problem, the chapter developed research objectives, assumptions, limitations for the research and the knowledge contribution that resulted from this research.

CHAPTER TWO: LITERATURE REVIEW

To ensure the objectives of this research are comprehensively covered, various published works in the field of cloud computing were synthesized and documented into three segments, namely, cloud computing concepts, current challenges and research trends in the cloud computing sphere, and measuring the Quality of Service derived from cloud computing platforms.

2.1. Cloud Computing Concepts

Cloud computing encompasses computer applications and services executed on a dispersed network platform, by use of virtualized computing resources accessible by mutually agreed network standards and Internet rules (Sosinsky, 2011). It is notable by the virtual and infinite nature of resources and abstraction of physical systems details that run the software.

Other scholars have described it is an extensive and dispersed computing platform driven by economies of scale, where a collection of abstract, virtual, scalable platforms, managed computational power, computing storage, and other computing services are provisioned based on client demands over the Internet (Al-Roomi, Al-Ebrahim, Buqrais & Ahmad, 2013).

Cloud computing therefore provides a platform that supports universal, expedient, on need access to a communal collection of computation resources like storage, applications, servers, networks and services that are quickly configurable and freed with ease in terms of management effort or provider intervention (Mell & Grance, 2011).

The cloud has been defined as hardware, storage space, network devices, software and computing interfaces supporting computing as a service solutions (Zia et al, 2012), with an alternate definition being data center hardware and software that enables computation services to be delivered through the internet (Armbrust, Fox, Griffith, Joseph, Katz, Konwinski, Lee, Patterson, Rabkin, Stoica & Zaharia, 2009).

From the identified definitions, this research defines the cloud as a pool of virtualized computational resources accessible in a multi tenant mode, dynamically and concurrently, while cloud computing is the access to and usage of virtual computational platforms as a service.

The usage of the term cloud refers to two fundamental concepts: Abstraction due to the fact that cloud computing hides the platform realization details from users and software developers i.e. applications are executed on physical infrastructures that have not been specified, data are kept in sites that are anonymous, management of the platform is subcontracted, and access by users is pervasive; Virtualization due to the fact that infinite logical resources are created by merging and sharing resources, provisioning of systems and storage could be as demanded from a federal platform, bills are determined on a rated basis, multiple lease is possible, and resources can be scaled (Sosinsky, 2011).

Various definitions of the term Virtualization exist, among them being the abstraction of a tangible component into a conceptual object, by Portnoy (2012), and a technique that combines or separates computation systems to give more than one execution setting using methods like hardware and software division, machine simulation and emulating, by Naeem, Memon, Siddique and Rauf (2016). The same authors, Naeem et al (2016), have as well defined abstraction as eliminating complexities of a system or process from prominence.

Virtualization enables creation of virtual machines (VMs), which share the same physical server. These VMs are leased to service providers dynamically based on their needs, creating an illusion of infinite resources (Desai, Oza, Sharma & Patel 2013). Figure 1 illustrates a typical high level diagram of a cloud architecture.

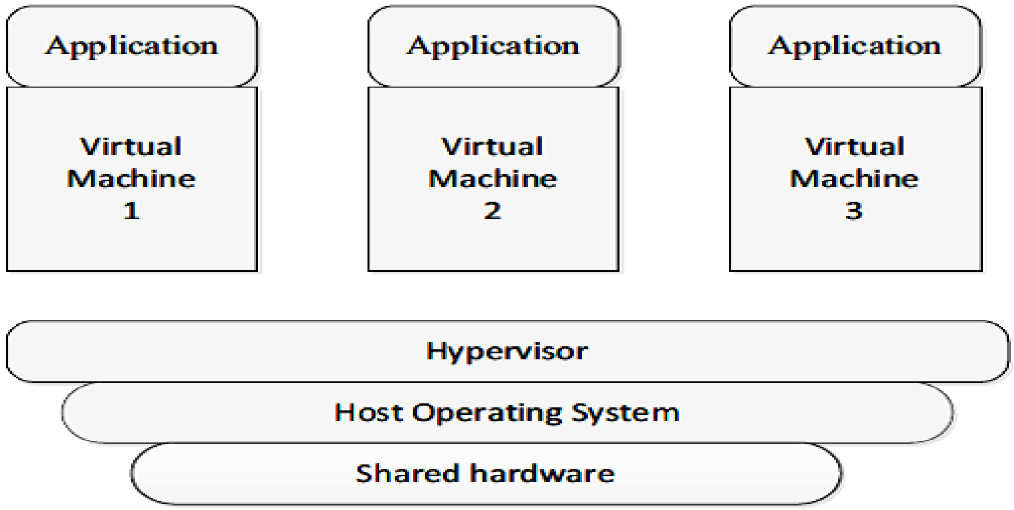


Figure 1: High Level Architecture of Cloud Computing

Source: Nazir (2012)

From Figure 1, the shared hardware could be processors, storage units, networking equipment and servers. The hosting operating system is the application provisioned on the hardware being shared and interacts with hardware's components. The hypervisor creates various execution environments from one shared resource.

Clouds can be categorized based on the model of deployment and the model of service. The model of deployment informs where the cloud is sited and how it is managed, namely, private, community, public, and hybrid.

According to Vyawahare, Bende, Bhajipale, Bharsakle and Salve (2016), cloud services deployed according to user requirements can be classified as Private, Public, Community, Hybrid and Mobile clouds.

Service models define the service type that the provider is offering namely Software, Platform and Infrastructure, all offered as a service. This is usually called *the SPI model* (Sosinsky, 2011). This research focuses on the service models, namely the SPI model.

The various categories of cloud can be summarized as portrayed in Figure 2.

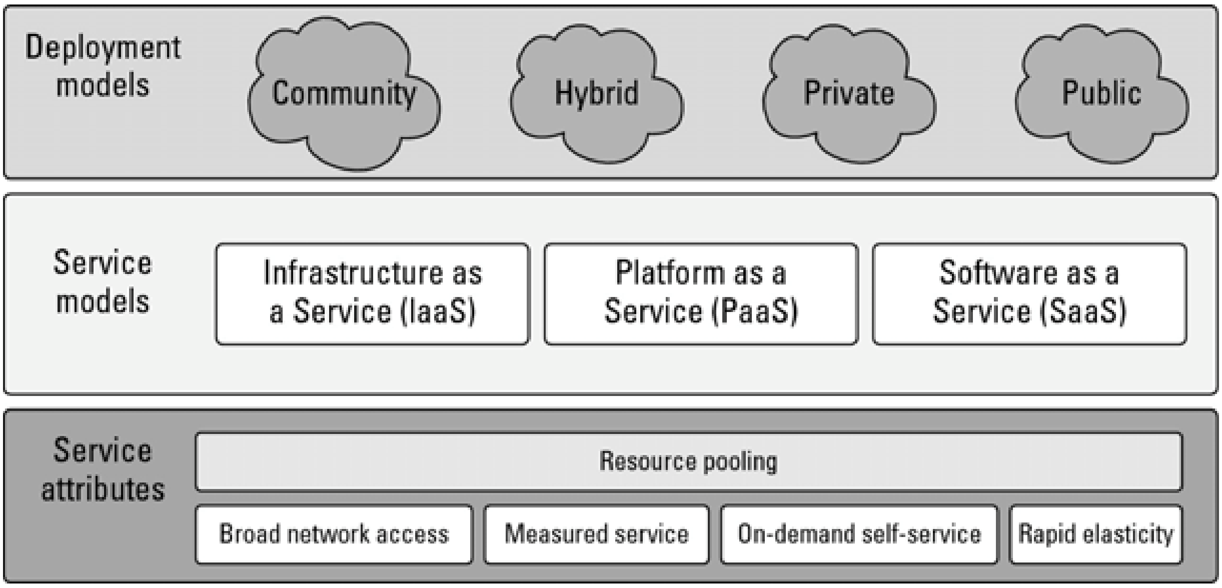


Figure 2: Models of the Cloud

Source: The National Institute of Standards and Technology (NIST, 2011)

2.2. Service Models of Cloud Computing

The various service models of computing on the cloud platform take the form of *"XaaS"* where "*X"* is the service being provided.

The three universally accepted service models according to the National Institute of Standards (2011) are:

1. Infrastructure as a Service (IaaS): this involves providing storage platforms, machines and various hardware assets as virtual resources to users. In this model, the provider takes care of all the physical and virtual infrastructure, while the user manages the deployment of the virtual services. This can cover the systems software, user software, and any user communications with the cloud platform.

2. Platform as a Service (PaaS): this involves providing systems, machines, software, user software and software creation frameworks as virtual computing resources. In this model, the user can install their user software on the cloud platform or use softwares that were developed using coding environments supported by the PaaS cloud provider. The PaaS provider is in charge of the cloud platform, the systems software, and the enabling environment. The client installs and manages the user softwares they require.

3. Software as a Service (SaaS): refers to the whole working platform consisting of user softwares, managing interface, and the user interface. In this set up, the user system is enabled using a thin client interface (mostly the browser), and the user's obligation is only inputting their data, managing it and user communication.

A diagrammatic presentation of the models and example providers is as depicted in Figure 3.

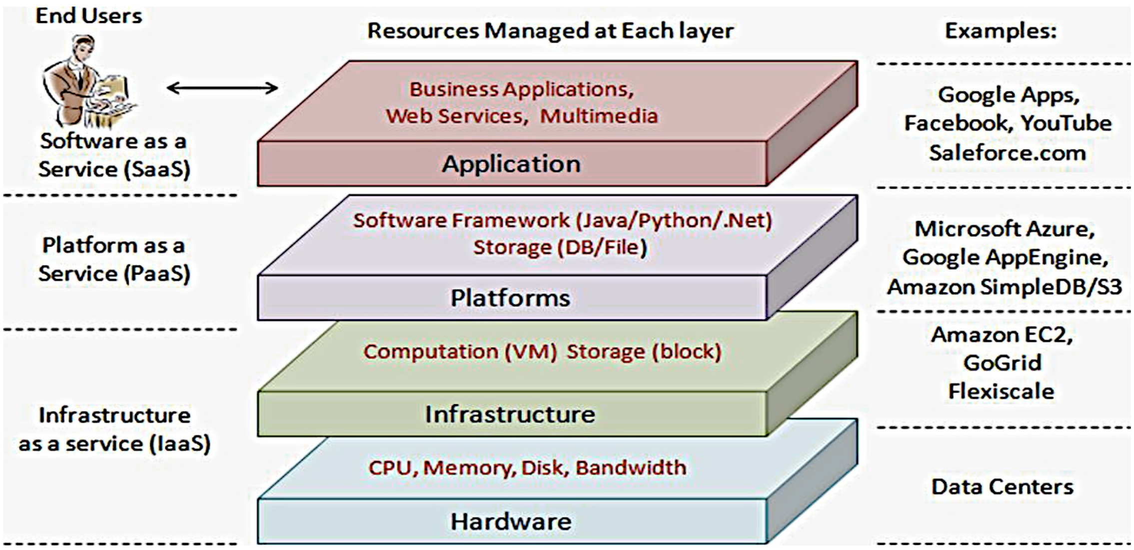


Figure 3: Cloud Service Models

Source: Vidhya (2013)

According to Vidhya (2013), the services currently provided are taking new dimensions. Examples of the new services are Desktop as a Service (DaaS), Cloud Migration as a Service (C-MaaS), Communication as a Service (CaaS), (Monitoring as a Service (MaaS), and Anything as a Service (XaaS).

Monitoring-as-a-Service (MaaS) gives cloud providers the opportunity to amalgamate monitoring requests at different levels (platform, infrastructure, and application) to enable efficient and scalable monitoring (Meng & Liu, 2013). Examples of vendors who offer this service are AppDynamics, Coradiant and NewRelic.

Cloud Migration as a Service (C-MaaS) involves moving the whole or a portion of a company's applications, data, services at user premises into the cloud or transferring them from one cloud provider to another. Migration from in-house resources to the cloud is called cloud migration while moving to a different cloud provider is called cloud service migration (Gouda, Dwivedi, Patro & Bhat, 2014).

One such example of a C-MaaS provider is Rivermeadow enCloud, which allows customers to move to cloud in a cost effective way. It has four steps in moving to the cloud, namely, Collects, Converts, Deploys and Synchronize.

Communication as a Service (CaaS) is a subcontracted business telecommunications solution where operators of the solution (CaaS vendors) are in charge of managing the platform required to convey Voice over IP (VoIP) services, video conferencing capabilities and Instant Messaging (IM) to clients (Gurudatt, Maheshchandra, Sadanand & Hemant, 2013).

XaaS or ‘anything as a service’ is any feature delivered to customers through the cloud rather than depending on in-house technologies. It could be defined as the range of all services that can be delivered via the cloud platform (Esteves, 2011).

Examples of XaaS services include Network as a Service, Storage as a Service, Unified Communications as a Service (U-CaaS) and Desktop as a Service. XaaS can be represented as in Figure 4.

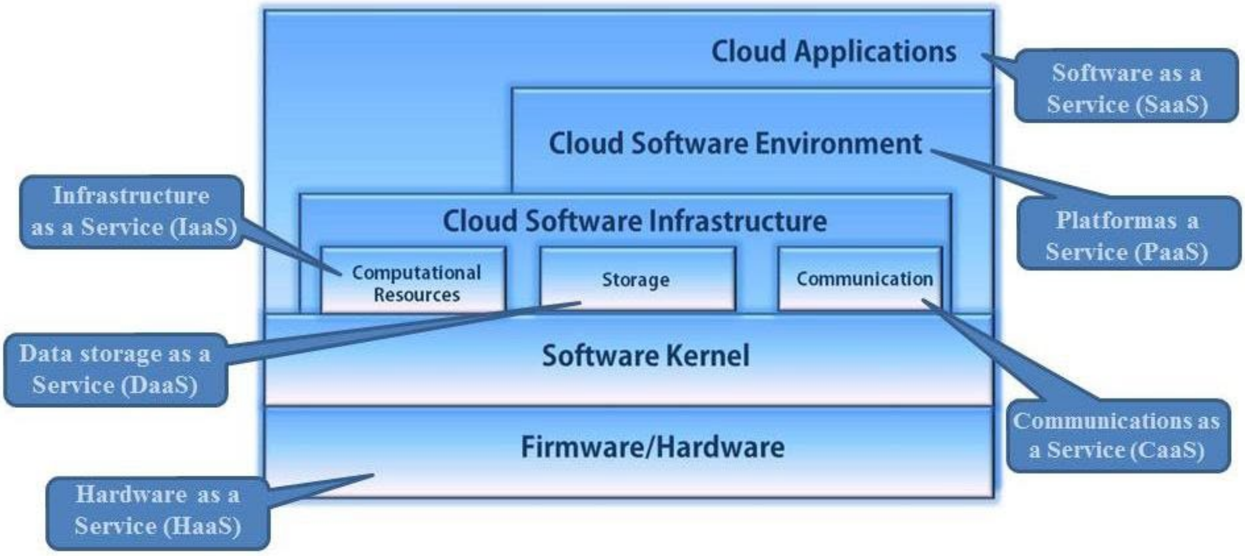


Figure 4: XaaS Architecture

Source: Gouda, Dwivedi, Patro and Bhat (2014).

A list of global providers for the stated cloud service models is as shown in Table 1.

Table 1: Examples of International Cloud Service Providers

Source: Dash, Saini , Panda and Mishra (2014)

No	Provider	Contribution	Services	Platform	Infrastructure Details	Interfaces
1.	Amazon	EC2 Elastic Computer Cloud, EMR Elastic Map Reducer, S3 Simple Storage Services, VPC Virtual Private Cloud	IaaS PaaS SaaS	Enterprise Linux by Red Hat Windows R2 Servers of 2003,2008 as well as 2008.	RAM: 1.7GB, Local storage :160GB Compute Unit: 1 EC2	API Command Line GUI Web Based Application/Control Panel.
2.	IBM	Dynamic Infrastructure Smart Cloud Blue Cloud	IaaS PaaS SaaS	IBM Web sphere and DB2.	Virtual CPUs of 32 bit with 1.25GHz; Virtual memory of 2 GB; Instance Storage (60 GB)	API Web Based Application/Control Panel.
3.	Google	Platform for creation of gaming and mobile applications development Google Drive	PaaS	Windows Mac OS X, Linux	Based on Requirements and existing environment	API Web Based Application /Control Panel.

No	Provider	Contribution	Services	Platform	Infrastructure Details	Interfaces
4.	Microsoft	Windows Azure	PaaS	Managed code Languages Supported by . NET	CPU of 1.6 GHz, RAM of 1.75 GB, Instance Storage 225 GB Moderate I/O Performance	Use of Web Based Application/Control Panel Use of API Use of Command Line
5.	AT&T	Synaptic Hosting Synaptic Storage	PaaS	Synaptic Platform for Hosting Virtual Solution for Hosting	Based on requirements and existing environment	Use of Web Based Application/Control Panel
6.	Salesforce	Heroku	PaaS IaaS SaaS	Development Environments including .NET, Java, PHP	CPU: 1.6 GHz, 1.75 GB RAM, Instance Storage: 225 GB I/O Performance: Moderate	Use of of API Use of Web Based Application/Control Panel
7.	Rackspace	Provides Infrastructure requirements for the cloud implementation	IaaS	All Main development platforms	RAM: 512 MB, 1 vCPU, local storage: 20GB, public network throughput: 20 Mbps internal network throughput: 40 Mbps	Use of API Use of Web Based Use of Application /Control Panel

No	Provider	Contribution	Services	Platform	Infrastructure Details	Interfaces
8.	OrangeScape	Delivers a platform as Orange scape using Cent OS	PaaS	all major Development platforms are supported	RAM: 0.5 GB, 1/2vCPU, Storage: 20 GB SATA SAN, Inclusive of 1 TB of data transfer	API Web Based Application/Control Panel GUI
9.	Cisco	Provides Infrastructural requirements for cloud applications	IaaS	Based on Requirements	Based on requirements and current platforms	Web Based Application/Control Panel
10.	Enki Consulating	PaaS operator of personalized cloud services based on ENKI enabled platform.	PaaS	Java, .NET , PHP as well as major development environments	Firewall Backup Storage Data Encryption Frequent Data Back-Up	API Web Based Application/Control Panel GUI

According to NIST (2011), the five vital features the cloud computing platform must provide are: On-Demand Self-Service by the user so that users can self allocate computation resources without having to liaise with cloud platform owner; Broad Network Access for accessing the cloud platform is offered via the system through defined processes in a way enabling platform neutral access to different user categories. This includes different computing terminals with different systems softwares, such as phones, laptops and other digital assistants.

Other notable features include Platform Resources Sharing for generating resources to be shared through a platform that provisions multiple simultaneous usage scenarios. Physical and virtual resources are dynamically provisioned based on user need. The fundamental aspect in this concept of resource sharing is the idea of abstraction that conceals the site of resources like processing, memory, storage, virtual machines, network bandwidth and connectivity; Quick Elasticity for quickly provisioning resources with high flexibility provisioned.

The system can scale up resources (extra powerful computer) or scale out systems (many similar computers), with capability for automatic or manual scaling. From the user's perspective, the cloud computing platform should appear infinite and can be procured at whatever time and in whichever amounts.

Finally having a metered platform to enable measuring usage of cloud resources, appraising, and conveying to the client in accordance with a metered scheme. A user can be billed based on parameters that are known such as quantity of processing power in use, transactions quantity, storage quantity used and network bandwidth. A user is billed based on the number of services provided.

2.3. Strengths of Cloud Service Computing

According to the US National Standards Institute (2011), cloud service computing offers several advantages, namely, less costs since cloud platforms function at high efficiencies and with great utilization, there are huge cost reductions experienced; ease of deployment based on the kind of service being provisioned, one may not need hardware and software authorization to use their service; Quality of Service which is realized by use of Service Level Agreements signed by the cloud solution provider and the user of the cloud solution.

Cloud solutions also provide: Reliability based on the cloud platforms magnitudes and capability to implement task balancing and failover which increases their reliability, higher than what can be achieved in a solitary organization; Subcontracted Information Technology management since cloud deployments enables someone else to manage computing infrastructure on behalf of another as the owner focuses on managing the business which leads to substantial reductions in IT staff costs; Easy maintenance and upgrade by the fact that the system is located at a central site, it is easy to apply patches and upgrades, therefore have access to updated software versions; and finally fewer obstacles to entry given that initial capital is significantly reduced, making it easy for anyone to significantly expand their businesses at any time.

2.4. Limitations of Cloud Computing

The US National Standards Institute (2011), states disadvantages in cloud service computing as being susceptible to the innate latency that is inherent in their internet links, hence not appropriate for usage scenarios that require huge amount of data transfer.

Software offered through the cloud is not easily customizable, as the client might want; Cloud service computing as a platform is stateless, just like the Internet is. For data to be sent on a distributed infrastructure, it has to flow in one direction. This lack of state makes data to traverse various routes thus arriving out of sequence, in addition various other features permit the interaction to be successful even on a faulty platform. For transactional coherency on the platform, service brokers, transaction managers, and middleware are needed to the system, which introduce additional overheads.

The cloud is also limited on privacy and security concerns since data transits across and is stored on systems that are no longer under the control of the client, interception risk is also increased and malfeasance of others.

2.5. Contemporary Research Trends in Cloud Computing

Explorations in cloud Computing tackles the problems of satisfying the constraints of future generations of private, public and hybrid platforms for cloud, as well as the challenges of letting software and development infrastructure to benefit from merits of cloud services (Nazir, 2012).

Some of the identified challenging research issues include Access Control, Server Consolidation, Reliability and Availability of Service, Service Level Agreement, Management of Energy, Data Management and Security, Data Encryption, virtual machines migration, interoperability, Multi-tenancy, mutually agreed Cloud Standards and finally Management of the cloud platforms.

2.5.1. Service Level Agreements (SLAs)

As the number of cloud users entrusting their operations to cloud platforms increase, Service Level Agreements (SLA) amongst cloud service clients and providers cloud service arise as an important concern. Since the cloud platform is dynamic in nature, constant monitoring on Quality of Service (QoS) parameters are essential to ensure SLAs (Patel, Ranabahu & Sheth, 2009).

The cloud paradigm is governed by service level agreements which permit numerous occurrences of a single application to be duplicated on several servers if there is need, depending on a priority pattern in use, a low level application may be minimized or shut down. A big concern for the user of cloud services is assessing the cloud vendors' SLAs (Nazir, 2012). This was the focus of this research.

2.5.2. Management of Data in the Cloud

Data from cloud systems is normally very huge (especially text based and scientific data), amorphous or semi amorphous, and usually affixed only with erratic updates and thus management of these data is a key research topic in cloud computing (Nazir, 2012).

Further, according to Nazir (2012), given the fact that providers of certain services usually have no access to the physical security infrastructure around data centres, they fully depend on the platform provider to attain full data security.

Factually, the uptake of cloud models makes users give up control of security of the physical systems. Further more, where cloud storage is in public clouds, users share the storage resources (Yahya, Chang, Walters & Wills, 2014).

This applies to virtual private clouds also, where the cloud provider can only indicate the security settings remotely, and is not in a position to confirm whether it is fully implemented. Platform providers, in this setting must attain the objectives of privacy and auditability.

Confidentiality is key for safe access to data and transmission, and auditability, for confirming if security settings for softwares have been interfered with. Confidentiality is attained by use of cryptographic protocols, while auditability could be attained via remote confirmation methods. Nonetheless, in virtual platforms as the clouds, VMs can dynamically move from a certain location to another location; thus the direct use of remote attestation is not adequate. In this scenario, it is vital to develop trust means at each architectural layer in the cloud.

2.5.3. Access Controls

This deals with issues like ensuring password strength and how often the passwords are changed, who configures the rate of password change, the recovery procedure for account names and passwords, how passwords are conveyed to users after being changed, the logs and the capability to review access (Nazir, 2012).

Most of the times, security is the principal concern with regards to records, platform and virtualization because business data is more than just a competitive asset, it most of the times has information on clients, users and staff which if it is accessed by unauthorised persons, may lead to civil liability and potentially criminal charges (Murtaza & Al Masud, 2012). In view of this, many conversations on cloud services target secrecy, confidentially and the separation of data from software logic (Murtaza et al, 2012).

2.5.4. Energy Resource Management

Huge energy saving in cloud infrastructure centres without compromising the services offered are an economic enticement for infrastructure providers and can make a huge influence to environmental sustainability (Nazir, 2012). Design of data centres that are efficient in energy use has attracted considerable attention with the main challenge being how to ensure a better balance between energy saving and platform performance.

2.5.5. Reliability and Availability of Service

In cases where an operator delivers software as a service on a need basis, the service requires a reliability quality factor to enable users review it in any conditions of the network, inclusive of network connections that are slow (Nazir, 2012).

2.5.6. Common Cloud Standards

Standardization in Cloud Computing covers three major areas: technology, personnel and operations. Nazir (2012) points out that at the moment, one major problem is the presence of many fragmented activities ongoing around cloud accreditation, yet a common body to coordinate those activities is not in place. The creation of an accreditation entity to attest cloud platforms and services is a huge task.

2.5.7. Interoperability

Interoperability in cloud computing is brought about by unavailability of common interfaces and open APIs, unavailable open standards for VM formats and service roll out interfaces. These setbacks cause integration challenges between services procured from dissimilar cloud platforms as well as from resources of the cloud and users internal legacy platforms (Ghanam, Ferreira & Maurer 2012).

2.6. Quality of Service Monitoring

The phrase Quality of Service is widely used across industries that deal in service provision. One of the fields where the term is commonly used is in the Information and Communication Technology (ICT) sector, namely, in computer networks and telecommunication.

According to the International Telecommunications Union (ITU 2008), the term QoS in the telecommunication field refers to the entirety of features of a telecommunications service that affect its capability to achieve specified and implied requirements of the telecommunication service user.

According to Cisco Systems Inc. (2003), the phrase QoS in the field of computer networks is the capability of a system to offer enhanced services to certain network traffic across several technologies. The technologies include Ethernet and associated 802.1 networks, Asynchronous Transfer Mode (ATM), the SONET, Frame Relay and all IP-routed networks that use either or all of the stated core technologies. The main objective of QoS, according to Cisco (2003) in this context is to offer priority as well as controlled jitter and latency (to be used by traffic in real-time and interactive basis), dedicated bandwidth, and improved loss features.

With regards to cloud computing, QoS means the extent of reliability, performance and availability presented by a service application and by a service platform or infrastructure on which it is hosted (Ardagna et al, 2014). Generically, it is the extent to which a suite of innate characteristics satisfy requirements (Ramad & Kashyap, 2017).

2.6.1. Quality of Service in Telecommunications

The telecommunications sub sector has two services whose QoS could be monitored, namely, the mobile voice service and the Internet (data) services. The voice QoS metrics are as shown in Table 2.

Table 2: Voice Traffic QoS Metrics

No	Metric	Description
1.	Rate of setting up calls	The ratio of calls effectively set up to a genuine number, well dialed and at time which the busy tone, ring tone, or answer signal is identified at the termination point
2.	Rate of dropped calls	The ratio of calls, that are not deliberately terminated during conversation minus the users involvement
3.	Rate of successful calls	The ratio of calls that are well set up and disconnected by the user
4.	Rate of blocking calls	The ratio of calls that are not set up due to lack of required resources
5.	Time required to set up calls	The duration from when a phone send button is pushed to when the user busy tone, ringing tone or answer signal is established at the user.
6.	Rate of successful handovers	The ratio of effective hand overs, out of the total hand overs requested
7.	Quality of speech	The clearness of the communication conveyed (without noise, echo and interference)

Source: Communications Authority of Kenya (2016).

In the computer networks subsector, the data metrics are as shown in Table 3.

Table 3: Data Traffic QoS Metrics

No	Metric	Description
1.	Rate of successful internet log in	The ratio of effective internet logins that launch an internet period within 40 seconds
2.	Rate of Internet session retention	The ratio of internet periods that are effectively started and continue until terminated by user
3.	Rate of successful internet data transmission	The ratio of successful internet data transmission periods where data is fully transmitted without errors between network points
4.	Transmission time for internet data	The span from when internet data is sent to the network and when it is received
5.	Transmission capacity for internet data	The ratio of the internet data transmission rate advertised by the provider that is actually achieved during a continuous transmission
6.	Latency	The time taken to send data from its source to intended recipient
7.	Packet loss	The vanishing of message units on transit in the network.

Source: Communications Authority of Kenya (2016)

The voice and Internet sub sectors are regulated by various country ICT regulators. In Kenya, they are controlled by Communications Authority of Kenya, through the Kenya Information and Communications Act, No.2 of 1998.

By virtue of them being regulated, the regulator ensures users get value for money by ensuring the service providers deliver on what they have committed to deliver. For example in Kenya, the Communications Authority of Kenya carries out dry tests to ascertain the QoS for voice telephony and data. Service providers found to be offering services below the set QoS threshold are fined a penalty of upto 0.2% of their gross revenue.

For the measurements in table 2, the Authority conducts dry runs, which involve using a server, which contains slots for inserting GSM SIM cards, which also has the database for recorded measurements; the portable slave unit which has slots for inserting mobile phones; Display unit used for configuration of the master and slave. During operation, the SIM cards in the slave are configured to call the master, from different location by moving around the country in a vehicle.

The SIM cards are loaded with airtime the normal way and all the calls made are measured for the various QoS parameters, and recorded in the master for later download. The recorded metrics are later produced in form of a report, showing the regions where the metrics were above the set levels and where they were below the set levels. A high level diagram showing how the measurements are done is as in Figure 5.

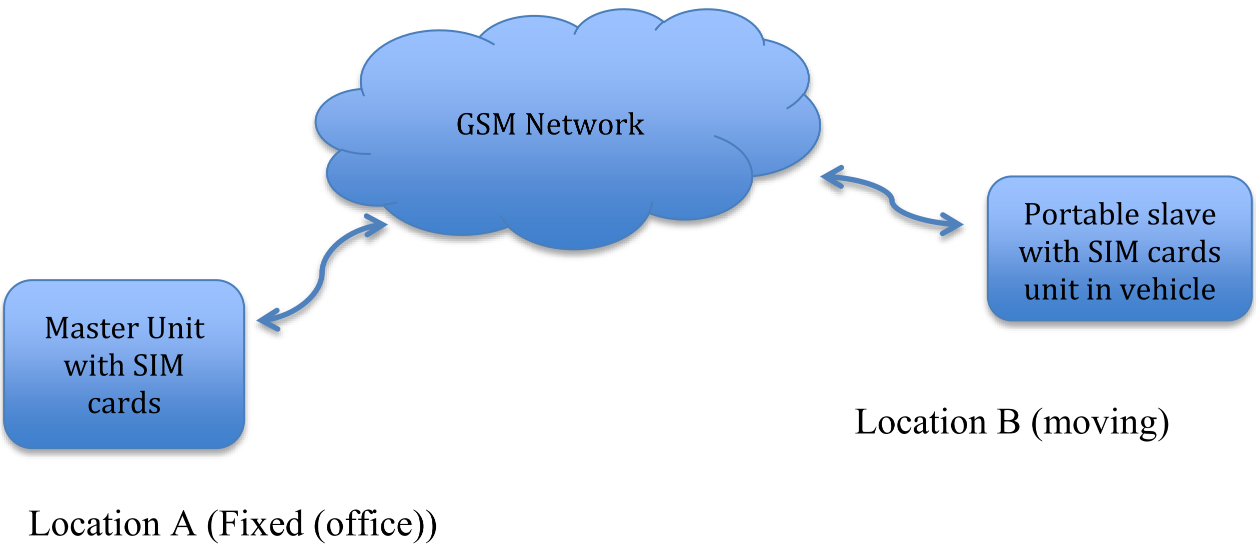


Figure 5: High level diagram for Voice QoS Measurements

Source: Communications Authority of Kenya (2016)

One of the equipment used by Communications Authority of Kenya, is QVoice equipment from Ascom. Sample equipment photos are as depicted in Figure 6 a, b and c.

a) Display Unit

b) Slave (Portable Unit)

c) Server

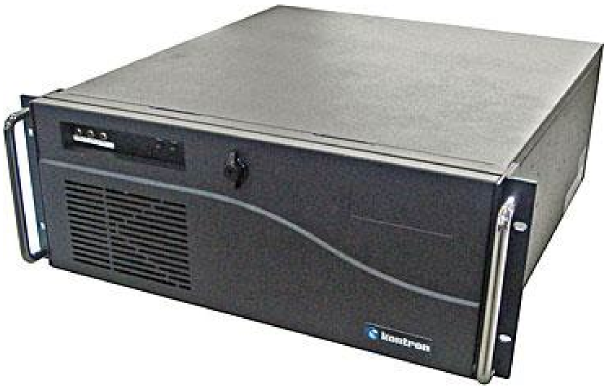


Figure 6: Photos of Sample Equipment

Source: Communications Authority of Kenya (2016)

From Figure 6, the display unit is used for configuration purposes as it provides a graphical user interface that is used for configuring of parameters to measure. During measurements it also displays the network performance as monitored in form of colour codes configured e.g. green for parameters that have met the threshold and red for those parameters that have not met the set threshold.

The Portable (Slave) unit contains SIM cards that have been configured to call or receive calls from the SIM cards inserted in the server unit. The configuration for receiving or calling is done using the display unit. The portable unit is installed in a moving vehicle.

The Server is a stationery unit that contains SIM cards that call or receive call from the slave unit. It also contains the software for reporting, from which one can log in and download the measurement data from the field.

The data QoS monitoring provides for three types of service levels, namely, best effort, differentiated service and guaranteed service. The service levels are the network's capability to provide the service required by particular traffic on the network from end node to end node or edge to edge of the network (Cisco, 2002).

The best effort service level offers no guarantee on the QoS to be offered; the distinguished service, sometimes referred to as soft QoS, offers preferential treatment for some traffic types by applying statistical techniques that ensure quicker handling, increased average bandwidth, and reduced average loss rate; while the guaranteed service, referred to also as hard QoS, uses complete reservation of platform resources for specific traffic (Cisco, 2002).

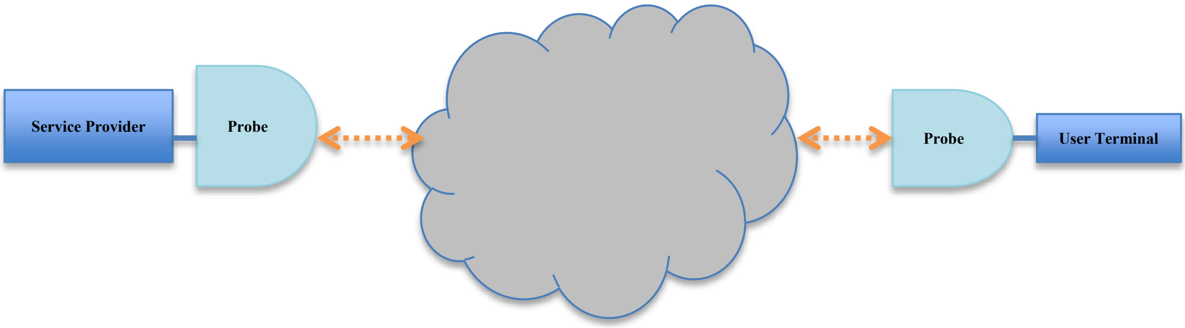
In data networks, the measurements can be active or passive. In active measurement probe packets are generated and send to the network and measurement for important factors like latency, jitter, throughput, packet loss taken (Peuhkuri, 2002). This measurement mode may introduce excess traffic on the network. For passive testing, real traffic is monitored and used to measure QoS parameters.

Both active and passive measurements can be modeled as End-to-End measurement, Hop-by-Hop measurement and Link-by-Link measurement (Peuhkuri, 2002). For End-to-End QoS measurement, the measurement probes are placed at the start and at the end of the path taken by the traffic to be measured, i.e. immediately after the service provider equipment and just before the user terminal.

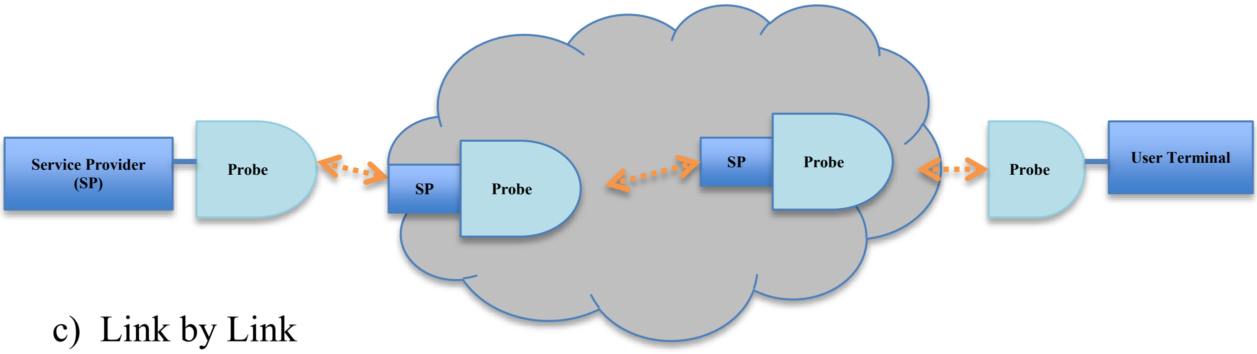
For Hop-to-Hop QoS measurements, the measurement probes are placed immediately after each service provider equipment along the path that the traffic travels, so measurements are done after they leave each equipment.

For link to Link measurements, the measurement probes are placed after the service provider equipment and before the next service provider equipment, so traffic is measurement without the processing delays introduced by the equipment. Figure 7 a, b and c illustrates a high level architecture of these modeling.

a) End to End



b) Hop by Hop



c) Link by Link

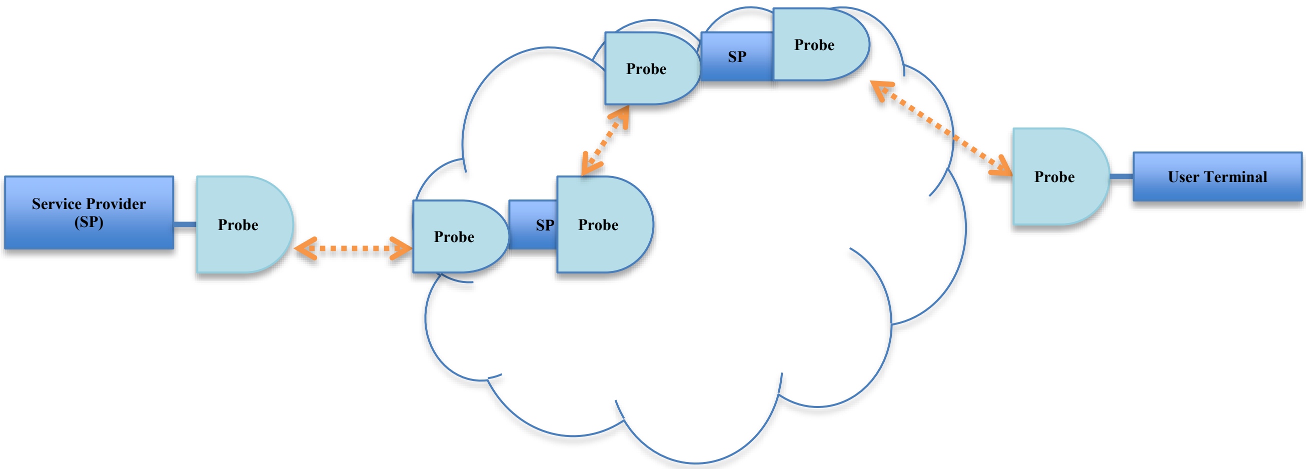


Figure 7 : End to End, Hop by Hop and Link by Link Measurements

Source: Peuhkuri (2002)

2.6.2. Quality of Service in Cloud Computing

Unlike in the telecommunications sub sector where QoS is regulated and the consumer has an entity mandated to ensure that desired levels of QoS are achieved, in cloud computing there is no entity to ensure cloud QoS is realized since cloud computing is currently unregulated.

Amongst the main challenges presented by cloud services is how to manage Quality of Service, referring to the problem of provisioning resources to the user's application to ensure user satisfaction along dimensions such as reliability, performance and availability (Ardagna et al, 2014).

With the swift uptake of cloud computing, various cloud operators have emerged who provide same services at dissimilar costs and levels of performance. Moreover, the dynamic nature of cloud platforms which occur due to the flexibility and demand based provision of cloud resources, there are substantial fluctuations in the Quality of Service levels at each service (Subha et al, 2014). Considering the user's view point, it is not easy for them to select which service is better for them, and which one to use, and what selection parameters to check. Further, ascertaining the best service for a particular task is difficult and mostly defines the achievement of the core business of the consumers (Saravanan et al, 2013).

According to Bardishri and Heshemi (2014), Quality of Service metrics play a critical role in choosing Cloud providers. The same authors argue that to select the best among several Cloud providers, users ought to have a means to monitor and also evaluate vital performance standards, which are necessary to their applications.

QoS parameters can be grouped into two, namely, functional parameters and non-functional ones. Some of the QoS metrics cannot be monitored with ease due to the setup of the clouds (Garg, Versteeg & Buyya, 2011). Further, it is not easy to match which services best fit with all functional and nonfunctional requirements.

The various metrics used to measure QoS can be Qualifiable or Quantifiable. Qualitative metrics are deduced based on experiences from the user. Quantitative measured by use of software and hardware observation tools. This research shall focus on Quantitative metrics.

There are several metrics that are used to rate the Quality of Service delivered by cloud platform providers. The select QoS metrics and as defined by this research are as indicated in Table 4:

Table 4: Cloud Computing QoS Metrics as Defined by this Research

Accuracy	Refers to the level of correctness of the QoS results from the cloud provider's tool as compared with results from the vendor neutral tool.
Adaptability	Refers to the capability and ease with which the cloud provider can amend or enhance the cloud platform features and services based on user's requests.
Availability	Refers to ratio of the number of instances that a user requests for a cloud service and gets the service to the number of times the user requests for the cloud service and does not get the requested service
Reliability	Refers to the availability of the service throughout the duration that the user has initiated a service therefore enabling atomic completion of a given task.
Service Response Time	Refers to the speed with which the requested cloud service loads (Also called service initiation time)
Stability	Refers to the degree of variability in the service response times of the cloud service
Trust	Refers to the level of confidence the user has in the services provided by a given cloud service provider

2.7. Quality of Service Monitoring Models in Cloud Computing

According to Aceto et al (2013), there are seven layers at which QoS measurements of cloud platforms could be performed, namely, hardware, middleware, network, OS, facility, application, and the user. The layers could be viewed as the location of the probes used for examining the system. Consequently, the tier where the probes are positioned directly determines the features that can be scrutinized.

In view of the observation by Aceto et al (2013), monitoring models are modeled around which layer the monitoring probe will be put. The various models are:

2.7.1. Agent Based Model

In this model, software agents are positioned in the virtual machines of the cloud platform. An agent is an independent entity, that has the capability of executing defined duties autonomously, based on explicitly stated instructions or through environment gained knowledge and adapting to variations in the environment through latest knowledge update (Meera & Swamynathan, 2013). They are also defined as self executing codes that work on behalf of the humans (Agrawal & Choubey, 2015).

This model is commonly used in Monitoring as Service Solutions (MaaS). MaaS enables monitoring for purposes of security for example detecting vulnerability, monitoring to aid in trouble shooting, external threats, monitoring to aid in SLA compliance check and QoS (Meera et al, 2013).

Ganglia monitoring system is one of the tools that was developed based on this model, initially used for high performance computing platforms like clusters and Grids, and has now been extended to cloud platforms, using sFlow agents found in the Virtual Machines (Dhingra, Lakshmi & Nandy, 2012). According to Dhingra et al (2012), currently, sFlow agents exists for XCP (Xen Cloud Platform), KVM/libvirt virtualization, Citrix XenServer platforms.

The other monitoring tool that is based on this model is Monitis which implements agents that have been installed on the resources to monitor, to enable users get service performance information and send alerts based on resource scarcity (Aceto et al, 2013). Other tools using this models are: Up.time, Cloudyn, CloudCruiser, Cloudfloor, Boundary, New Relic and DARGOS.

Through literature review, Makokha, Opiyo and Okello-odongo (2017), derived a high level architecture depicting the agent-based model as depicted in Figure 8.

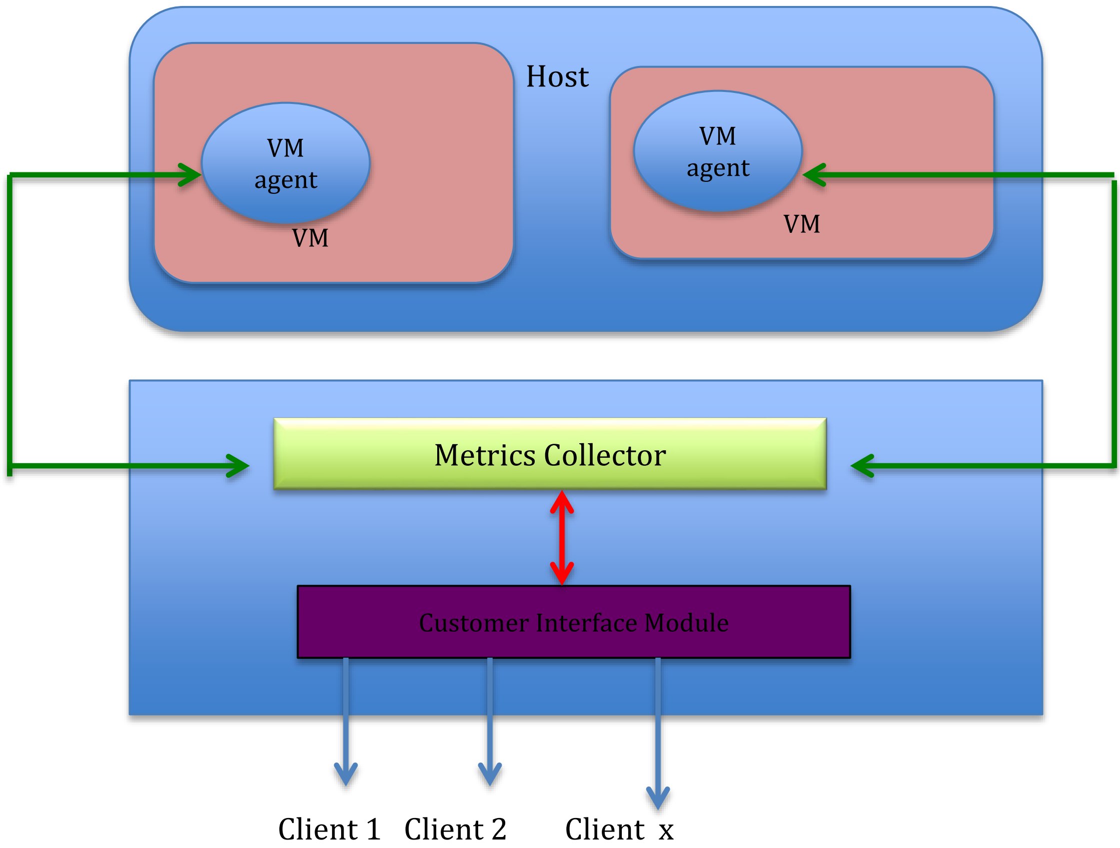


Figure 8: High level Architecture for Agent based Model

Source : Makokha et al (2017)

A detailed implementation architecture for an agent based resource monitoring architecture is depicted in Figure 9.

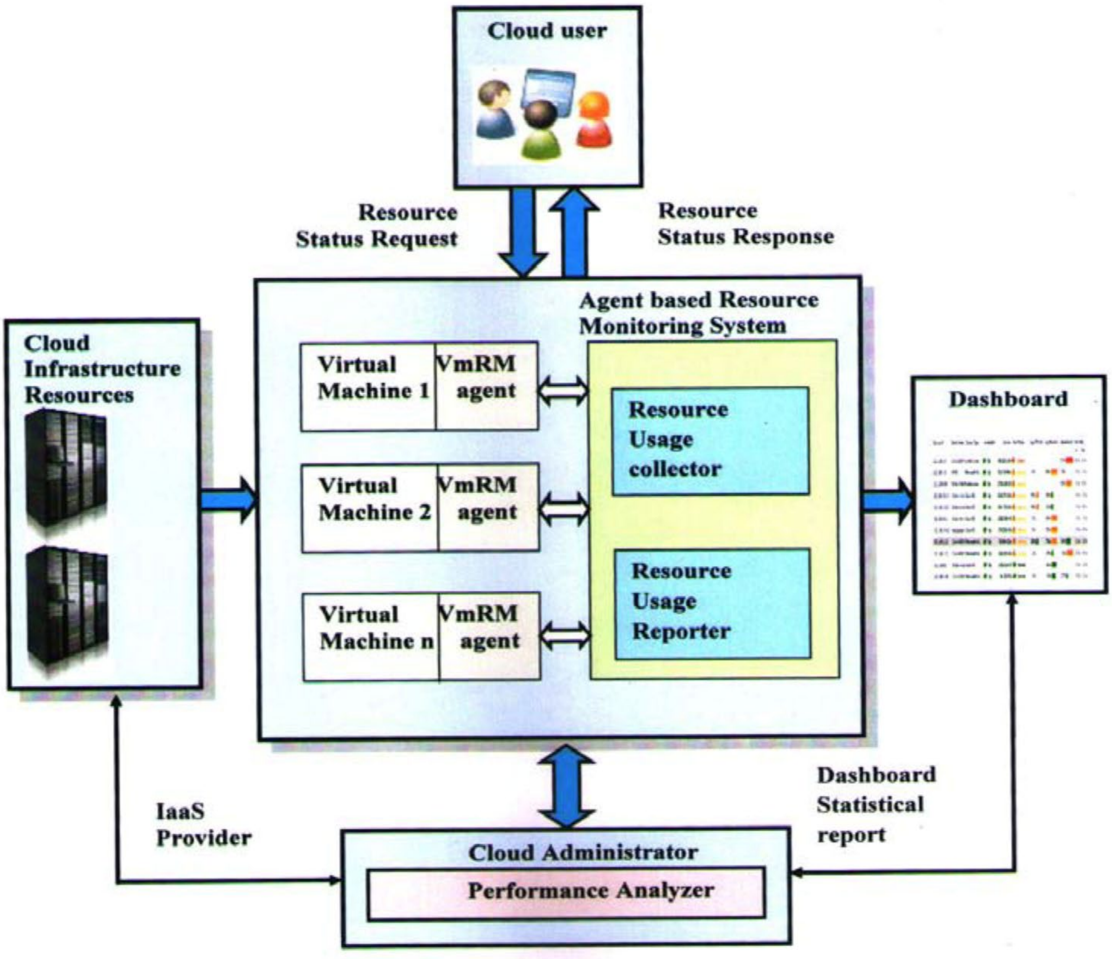


Figure 9 : Implementation of an Agent Based Model in IaaS

Source: Meera and Swamynatha (2013)

From Figure 9, the IaaS cloud is designed to have virtual machines in its platform. A Virtual machine Resource Monitoring agent (VmRM agent) is installed in each Virtual Machine to monitor a specific aspect of the cloud.

The VmRM agent collects the CPU and memory utilization of each virtual machine hosted with different types of applications. It sends the resource usage statistics to the agent based resource monitoring system. Agent based resource monitoring system has two components. The resource usage collector component collates the health information of each VM and sends that to the resource usage reporter.

The resource usage reporter reports the virtual machines status information to the cloud administrator and also displayed in the dashboard. The cloud administrator has a performance analyzer module that analyzes the statistical report in order to measure whether the performance is as per the SLA.

2.7.2. The QoS MONitoring as a Service Model (QoSMONaaS)

It is a portable architecture which implements a trustworthy (neutral, dependable, and timely) infrastructure for checking the QoS as experienced at the business tier on a common cloud infrastructure (Adinolf, Cristaldi, Coppolino & Romano, 2012).

The portability of the architecture is based on the fact that it is possible to migrate it from one infrastructure to a different one after little changes. The infrastructure is presented to all functions running on all as a Service platforms.

Its architecture is made up of the basic interface, the extended interface and two main services, which are authentication and anonymization (Zavol, Jung & Badica, 2013). The basic interface is used by QoSMONaaS to interface with other applications, i.e. the channel that all applications use to request the platform.

The Extended interface is used to collect the information used for QoS monitoring. QoSMONaaS uses the authentication and anonymization services from the underlying platform and which require modification efforts for the QoSMONaaS to be ported on different cloud platforms.

The QoSMONaaS is delivered with a prescribed depiction of the particular business process (KPIs, entities and associations) and a prescribed explanation of the SLA to be guaranteed, that is a suite of controls that must be respected, to enable it monitor the real QoS conveyed by the cloud provider (Cicotti, Coppolino, Cristaldi, Salvatore, & Romano, 2011).

A high level architectural diagram on the implementation of this model is as shown in Figure 10.

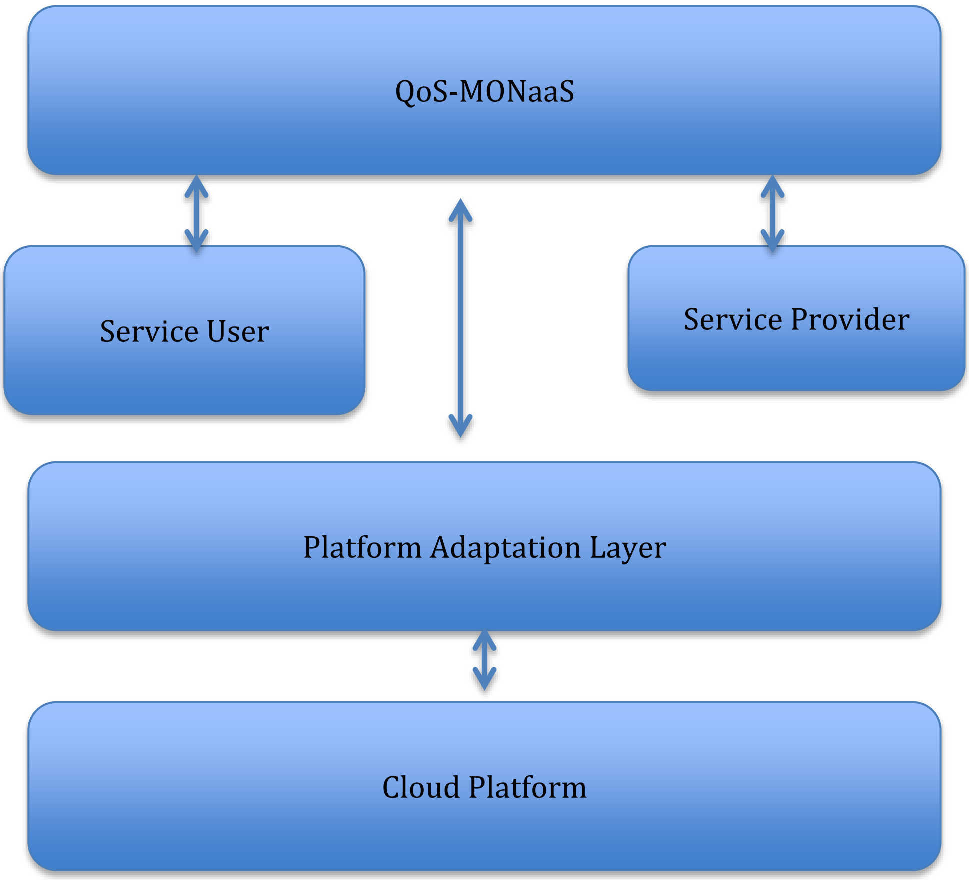


Figure 10 : High Level Architectural Diagram for QoS-MONaaS
Source: Adinolf et al 2012.

A zoomed in view of the QoSMONSaaS as implemented on the Subscription Racing Technology (SRT) platform is as shown in Figure 11.

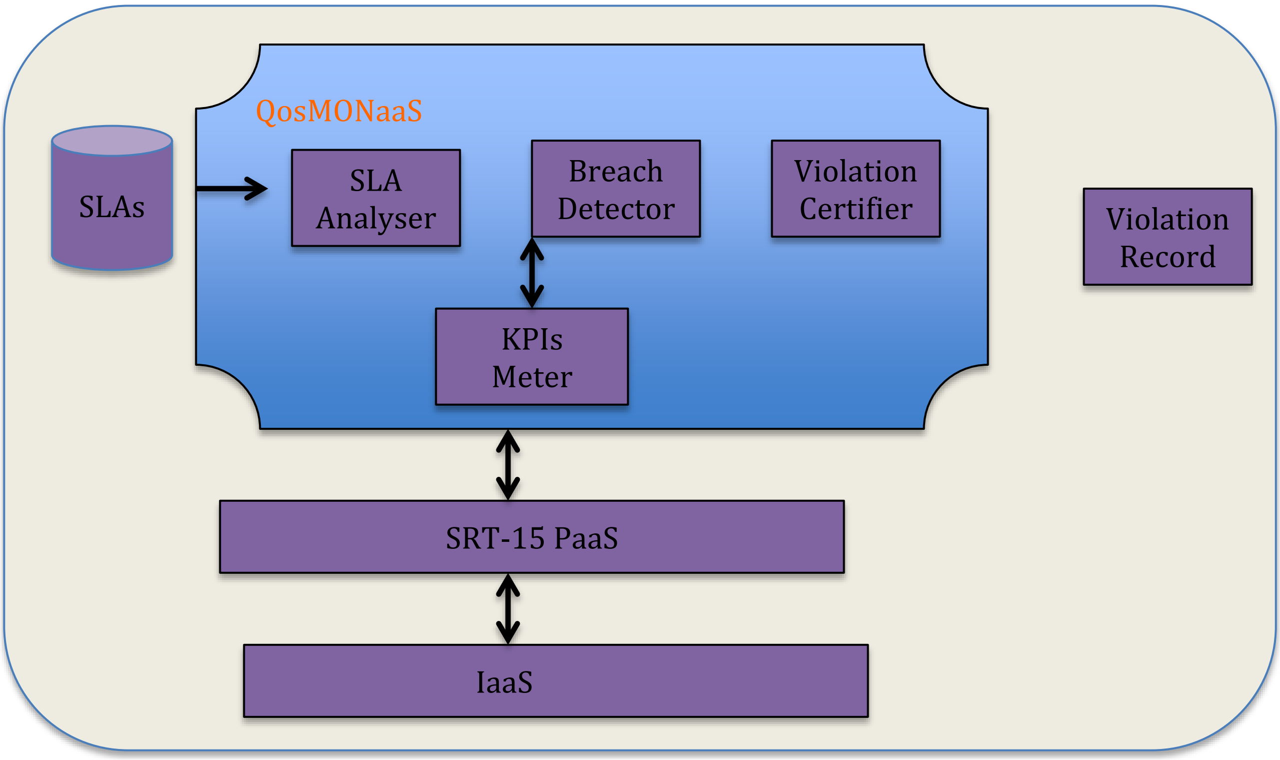


Figure 11 : QoSMONaaS Implementation on SRT project

Source: Cicotti et al 2011.

The SLA Analyzer reads and processes (parsing) SLAs provided as input and also gathers data delivered to the KPI Meter to examine it, while the KPI Meter constantly observes the real value of the KPIs using queries for submission to the SRT-15.

The Breach Detector amalgamates the KPI monitor outputs and the SLA Analyzer conditions to identify contract negations. It reports deviations to the SLA Analyzer and advances all similar data to Violation Certifier.

The Breach Detector outputs are augmented by the Violation Certifier using a timestamp and a digital signature, to enable production of evidence that is not easy to forge and thus usable for forensic purposes.

The SRT-15 being a cloud Platform as a Service solution, enables construction of every software as a service solution. The objective of the Subscription Racing Technology (SRT) for 2015 was to develop a scalable platform for linking enterprise applications and services. The platform aids in enabling the discovery and amalgamation of dynamic business services on the Internet (Cicotti et al, 2011).

2.7.3. CloudQual

Is a model that describes five quality metrics based on six quality dimensions from a service user's perspective (Zheng, Martin, Brohman & Xu, 2014). The dimensions used by this model are service reliability, service usability, service availability, service responsiveness, service security and service elasticity.

The CloudQual model proposed by Jegadeesan and Karuppaiah (2016), has a Usage Monitor, an Aggregate Manager, as well as a Prediction Manager that uses a Generalized Pareto Distribution model (GPD) to envisage performance degradation.

The system modules for the Jegadeesan et al (2016) model are comprised of the Cloud Manager which is in charge of interacting with users to comprehend their service requirements. It is responsible for collecting all requirements as well as performing detection and rating of better services.

Other components are the Monitoring module which does the discovery of services that are capable of meeting user's essential QoS needs. It also supervises the performance of cloud solutions, like for IaaS it oversees scaling latency, memory, the speed of VMs, network latency, storage performance, as well bandwidth. Further, it maintains a record of how SLA needs of clients are being fulfilled by the service provider. There is also the Prediction module for evaluating and modeling short term CPU usage extreme values.

From the reviewed literature, a high level architectural diagram depicting how a QoS monitoring tool developed using the CloudQual model can interface with a cloud provider's infrastructure, is illustrated in Figure 12.

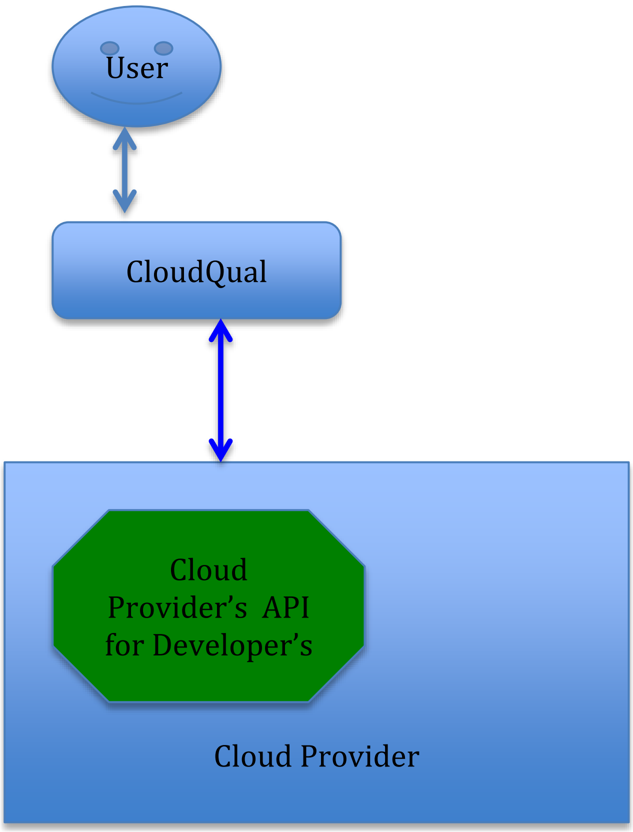


Figure 12: CloudQual High Level Architectural Diagram

Source: Makokha et all (2017)

Mathematically, CloudQual was modeled by Priyanka and Kumar (2016) as :

Let 'B' be the | Cloud Quality system at the final set

B= {I, O, F, $}

Identify the Functions/Modules as,

F= {U, A, Rel, Resp,S, E}

U=Usability

A=Availability

Rel=Reliability

Resp=Responsiveness

S=Security

E=Elasticity.

Identify the Inputs as,I= {c,co,d }

Where,

c=Correlation

co=Consistency

d=Discriminative power

Identify the outputs as,

O= {uv,av,rv,resv,sv,ev }

Where,uv=Usabilityval

av=Availabilityval

rv=Reliabilityval

resv =Responsivenessval

sv=Securityval

ev=Elasticityval

Identify the Constraints as, $= 1

If cloud is secured with firewall then it is difficult to retrieve these parameters to determine its quality. Each parameter was implementented through a separate module as:

1st Module: Usability Module

U= {g,f }

g=gui,

f=features

2nd Module: Availability Module

A= {t,ts,av}

t=Uptime of operational period,

ts=Total time of operational period.

av=Availabiityval.

Formula,av=t/ts

2nd Module: Availability Module

A= {t,ts,av}

t=Uptime of operational period,

ts=Total time of operational period.

av=Availabiityval.

Formula,av=t/ts

3rd Module: Reliability Module

Rel={n,ns,rv}

Where,

n=No. of failed operations,

ns=Total operations occurred in a time interval.

rv=Realiabilityval

Formula,

rv=1-n/ns

4^{th} Module: Responsiveness Module

Resp={fi,ti,tmax,resv}

Where,

fi=Measure central tendency offset of data,

ti=Time between submission and completion,

tmax=Max acceptable time to complete request.

resv=responsivenessval

formula,resv=1-fin=1(ti)/tmax

5^{th} Module: Security Module

S={FT(t),sv}

Where,FT(t)=Cumulative distribution function of random variable T,

t=Time until first security breach occurs.

sv=securityval

Formula,

sv=1-FT(t)

6^{th} Module: Elasticity Module

E={ri1,ri2,n,ev}

Where,

ri1=Amount of resources allocated,

ri2=Amount of resources requested,

n=No. of required resources in operation period.

ev=elasticityval

formula,ev= Σ ni1=1ri1/ Σ ni2=1ri2

The functions 'F' are:

F={Usability (), Availability (), Reliability (), Responsiveness (),
Security (), Elasticity ()}

Usability (h) =P' :: takes the gui.
P' = { h | h takes the gui }

These modules are linked to the cloud provider's API to monitor the various information from the provided cloud services.

2.7.4. Adaptive QoS-driven Monitoring Model

This model has flexibility and offers QoS monitoring services that can be reconfigurable dynamically which are able to adapt to different cloud features (Serhani, Atif & Benharref, 2014). Its architecture has a cloud platform and a setup of hardware functionalities (virtual machines, application servers, storage servers), as well as entities for monitoring, inclusive of Applications Programming Interfaces for smooth communications among numerous architecture's modules as well as with external units to simplify monitoring duties (Serhani et al, 2014).

Various modules are used in this model, they include monitors, SLA verifier, certifier and the driver. According to Serhani et al (2014), the native and universal monitors are responsible for realizing modules (or APIs), each with different functionalities.

The Monitor subsystem monitors performances based on given dimensions, detects violations once they happen; the SLA verifier subsystem, examines the agreement requirements (thresholds) for confirmation if the parameters can be assured before commence of the service monitoring; the Certifier subsystem attests whether a SaaS meets the SLA verification trials, then provides a certificate for the confirmed service provider and the Driver initiates the monitoring process after scoring well in all the required tests.

Amalgamated cloud solutions combined from uniting various single cloud services are watched over by a Multi-monitor-based monitoring platform. The single cloud services could be from one cloud provider or to belong to different cloud operators. A high level diagram depicting the architectural design for this model is as shown in Figure 13.

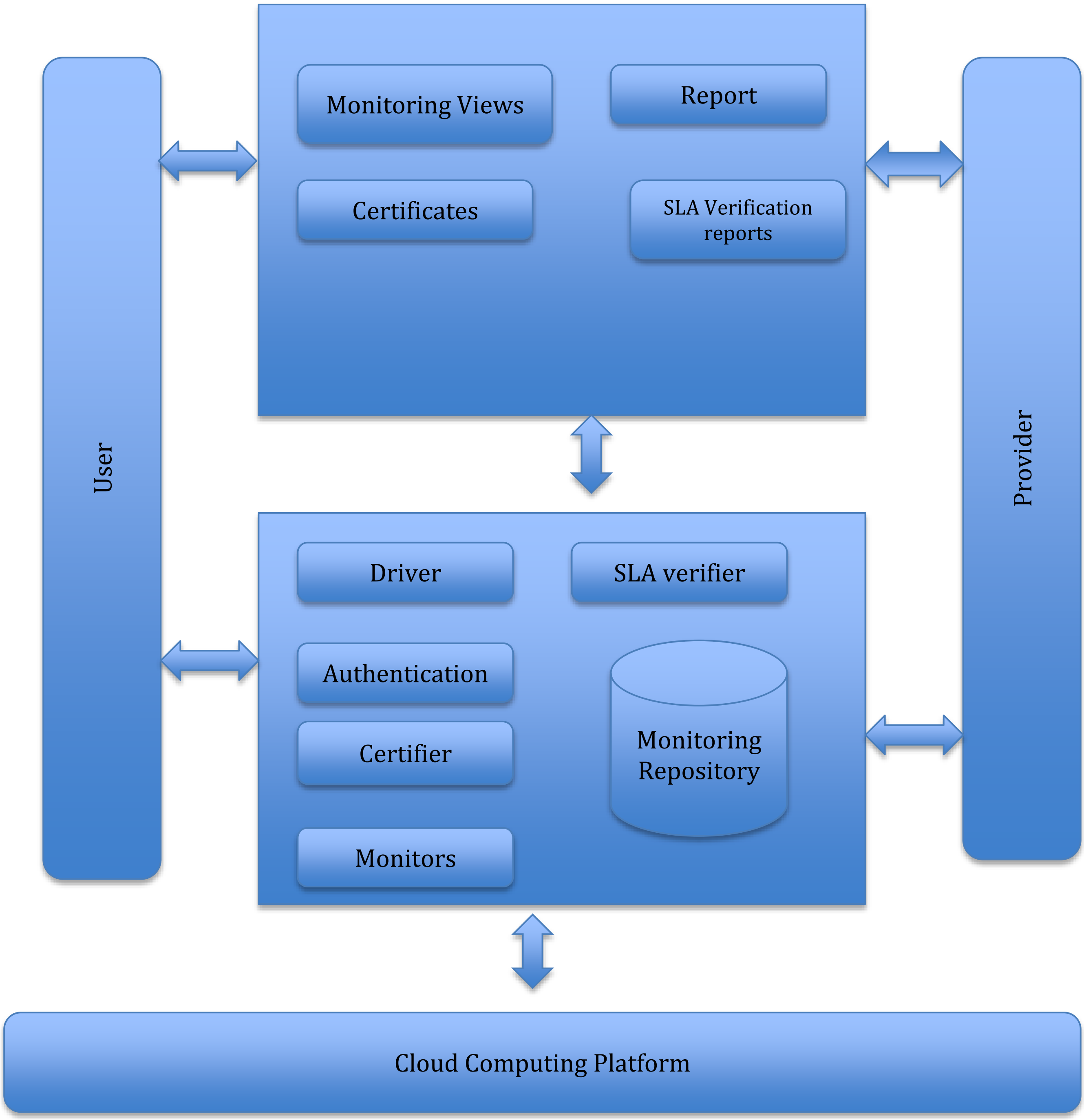


Figure 13 : High Level Architectural Diagram for Adaptive QoS SaaS Model

Source: Serhan et al (2014)

2.8. Vendor Neutrality of the Cloud Quality of Service Monitoring Models

An independent tool that is not tied to any particular vendor platform for checking performance of heterogeneous platforms is a key capability most required by the cloud paradigm (Cicotti et al, 2011)

The presence of trustworthy (timely and reliable) QoS examining tools would enable entities to know whether a failure or performance problem they encounter is caused by the cloud operator, network platform, or design of the software. This can play a key role in the actual take up of cloud technology, since enabling users to get the complete value of cloud facilities would augment the trust level placed in the cloud technology (Cicotti et al, 2011).

A quality model intended for services in the cloud should be computable, unbiased and confirmable, to enable cloud operators measure the QoS delivered, and cloud clients can confirm the QoS experienced (Zheng et al, 2014).

According to Cedillo, Gonzalez-Huerta, Abrahao, & Insfran (2015), in cloud solutions, amongst the shortcomings of QoS measuring tools is in their portability capability. This supports the fact that most cloud service QoS tools are vendor centric and commercial in nature, which makes the tools to be less flexible and portable and this implies that their results are neither extensible nor comparable with other platforms.

A closer look at the highlighted cloud QoS models depicts that they are closely designed based on the internal architecture of the physical platform of the solution provider and thus a performance measuring tool developed based on these models can not be used across different cloud service providers.

In instances where the tool is used across several vendors, it is as a result of the tool being customized for the various cloud service providers through their open APIs. This limits comparability of the measured QoS incase one is at cross roads on which cloud vendor to use. Further incase one has procured various cloud providers for redundancy, direct comparison of performance is not possible with tools developed using this models.

Using the identified cloud QoS monitoring models and their associated monitoring tools, it is possible to derive a QoS Monitoring Framework from which the models are anchored on, thereby converting it from an implicit Conceptual QoS Monitoring Framework to an Explicit Conceptual Monitoring Framework.

According to Vliet (2007), an implicit conceptual model is made of the background knowledge shared by people in the Universe of Discourse. The fact that the knowledge is widely shared leads to ‘of course’ assertions by those within the Universe of Discourse, because this knowledge is taken for granted.

Part of the implicit conceptual model is not articulated and has tacit knowledge, which is skillfully applied and functions in the background. According to Vliet (2007), an implicit conceptual model contains habits, customs, prejudices and even inconsistencies. The explicity in the Conceptual Monitoring Framework is by the fact that it must be able to be communicated to the various stakeholders.

2.9. Existing Cloud QoS Monitoring Framework Formulation

In the context of Information Systems, a framework can be viewed as a structuring of ideas whose value arises from the arrangement it imposes on the ideas in a given Information Systems field (Gorry & Morton, 1971).

An Information System Framework is therefore by definition, a still image, a portrait, and which is not intended to explain how information systems are developed in the various areas. For this purpose one would have to use a process model of information system implementation (Gorry et al, 1971).

According to ISO 13236 on Information Technology -Quality of Service Framework, the standard defines its QoS Framework as a well thought out pool of concepts and how they are related that explains QoS (Quality of Service) thus enabling the partitioning of, and relations between, the themes pertinent to QoS in Information Technology (IT) to be communicated by a common means of explanation (ISO, 1998).

The ISO 13236 Information Technology -Quality of Service Framework states vocabulary and thoughts for QoS in IT, defines how QoS needs can be stated, and finds several QoS mechanisms like the three-party negotiation, usable as components of managing QoS tasks to meet various kinds of QoS requirements, and offers a basis for the description of enhancements and extensions to planned or existing standards (ISO, 1998).

The ISO 13236 Information Technology -Quality of Service Framework does not give a basis of specifying objectives on performance or network signaling of QoS in public communications networks and excludes the detailed specification of QoS mechanisms (ISO, 1998).

ISO Guide 73:2009 on Risk Management -Vocabulary, describes a framework for risk management as a suite of components containing the foundations and arrangements for the organization used in planning, executing, monitoring, assessing and continually refining risk management in the organization (ISO, 2009).

The ISO guide 73:2009 further expounds that the foundations are composed of the policy, related goals, the firm's mandate and its pledge to manage risk; while organizational arrangements comprise of the plans, the relationships, the accountabilities, the resources, the processes and related activities.

A framework can be considered as an integrating metamodel, providing a structure to help in connecting a suite of concepts, models, and methodologies at a higher level of abstraction for their linkages or differences to be displayed to assist in understanding or decision-making (Jayaratna, 1994). The author further defines a methodology as one's thinking and actions that have been structured explicitly.

Frameworks therefore help in an important purpose of organizing ideas and approaches to solving problems in the emerging information systems field (Lucas, Clowes & Kaplan, 1973).

According to Lucas et al (1973), a framework helps structure ideas about systems and facilitates communication among professionals. In Academia frameworks play a critical role for teaching information systems concepts. Frameworks as well provide new directions and trends for research.

Lucas et al (1973) further postulate that since Information Systems exist to support decision making, and therefore a framework should be capable of accommodating dissimilar types of decisions. It should make it possible to get any type of information required for each different decision category in the framework. The same authors also state that a framework should have a theoretical basis, which is the aim of the framework.

Development of an Information System Frameworks needs to adhere to certain guiding principles, namely global consistency to ensure one coherent framework to ensure every concept is linked to every other one in a specific, well-established way; generality to ensure that it is specialisable and extensible in certain situations, to cater for the various specialized subfields; simple and straightforward as possible for easy understanding.

An Information System Framework also needs to be anchored on information system concepts in related fields to avoid creation of an isolated framework incompatible with other related fields, and therefore provide a conceptual foundation**,** to enable it serve as a foundation from which one can build other extensions. (Falkenberg, Hesse, Lindgreen, Nilsson Han Oei, Rolland, Stamper, Van Assche, Verrijn-Stuart & Voss, 1998).

From one Information System Framework, the same solution can be described, for different usages, in dissimilar ways, leading to different types of descriptions (Zachman, 1987).

Thanh and Helfert (2007) in their work on a review of quality frameworks in information systems, proposed an Information System Framework that is anchored on Information System Architecture that considered the perspectives of the User and Developer of the Information System.

Vidgen, Wood and Wood-Harper (1994) suggested a framework to describe software quality anchored on the multiview method of development (Wood, 1992; Wood-Harper & Avison, 1992). The authors postulated that various viewpoints of software quality are needed for one to evaluate product quality effectively. The framework is anchored on user satisfaction, linking the product with its usage as well as the services offered to support it.

Wong and Jeffery (2001) developed a framework for evaluation of software quality based on the motivation behind the evaluation. It was grounded on the belief that evaluators of software are swayed by their roles on the job. According to Wong and Jeffery (2001), participants with dissimilar job roles were found to pay attention on different characteristics sets of the software when assessing software quality.

The theoretical foundation for developing such a framework was anchored on the theory retrievable from cognitive psychology, which was also embraced by Gutman's Means-End Chain Model, that postulates that connections between product features, consequences created by use, and personal ethics of the users determine the process of making decision or, in this instance, the process of software quality evaluation (Wong & Jeffery 2001).

Based on the foregoing literature review on frameworks, this research defines a framework as an encapsulation of ideas, rules, concepts and fundamental principles of a particular domain or system in a static and structured way, and how they are interrelated, to aid in better understanding of the system or domain and in decision making processes.

Further, it can be deduced from the literature reviewed that a framework factors in various aspects of the domain, namely the different stakeholders which influences the view perspective of the framework, the underlying theoretical basis (if any), the various concepts of the system/domain, the underlying principles or set of rules of the domain /system, the aim of the framework, methodologies and the problem to be addressed by the framework.

Using the identified aspects of a framework, the formulation of the existing cloud QoS monitoring framework involved listing all the identified aspects required for a framework, reviewing the various models of QoS monitoring in the cloud and the existing cloud QoS monitoring tools. After identification of the various aspects from the models, an interrelationship between the identified aspects was deduced. The Key aspects considered for the existing framework were stakeholders, view/perspective, aim and concepts.

From the existing cloud QoS monitoring models reviewed, the stakeholders were identified as Cloud solution Providers and the Cloud solution Users. The perspective captured by the existing cloud QoS monitoring models is that of the Cloud Service Providers. The QoS is monitored from the provider's physical platform up to the cloud virtual platform. The QoS from the cloud virtual platform to the end user is not factored.

The aim of the models is to help in enforcement of the Service Level Agreements signed by the Cloud Provider and those using their Cloud Services. The monitoring also helps the Cloud Service providers to know the utilization level (load level) of their physical resources and determine whether to increase or maintain the quantity. The basic concepts addressed by the reviewed models are monitoring layers of the cloud solutions, tests and metrics to be monitored, namely, computation based and network based.

Based on the reviewed Cloud QoS monitoring models an architectural diagram for the existing framework under which monitoring is done, as derived by this research, is as depicted in Figure 14.

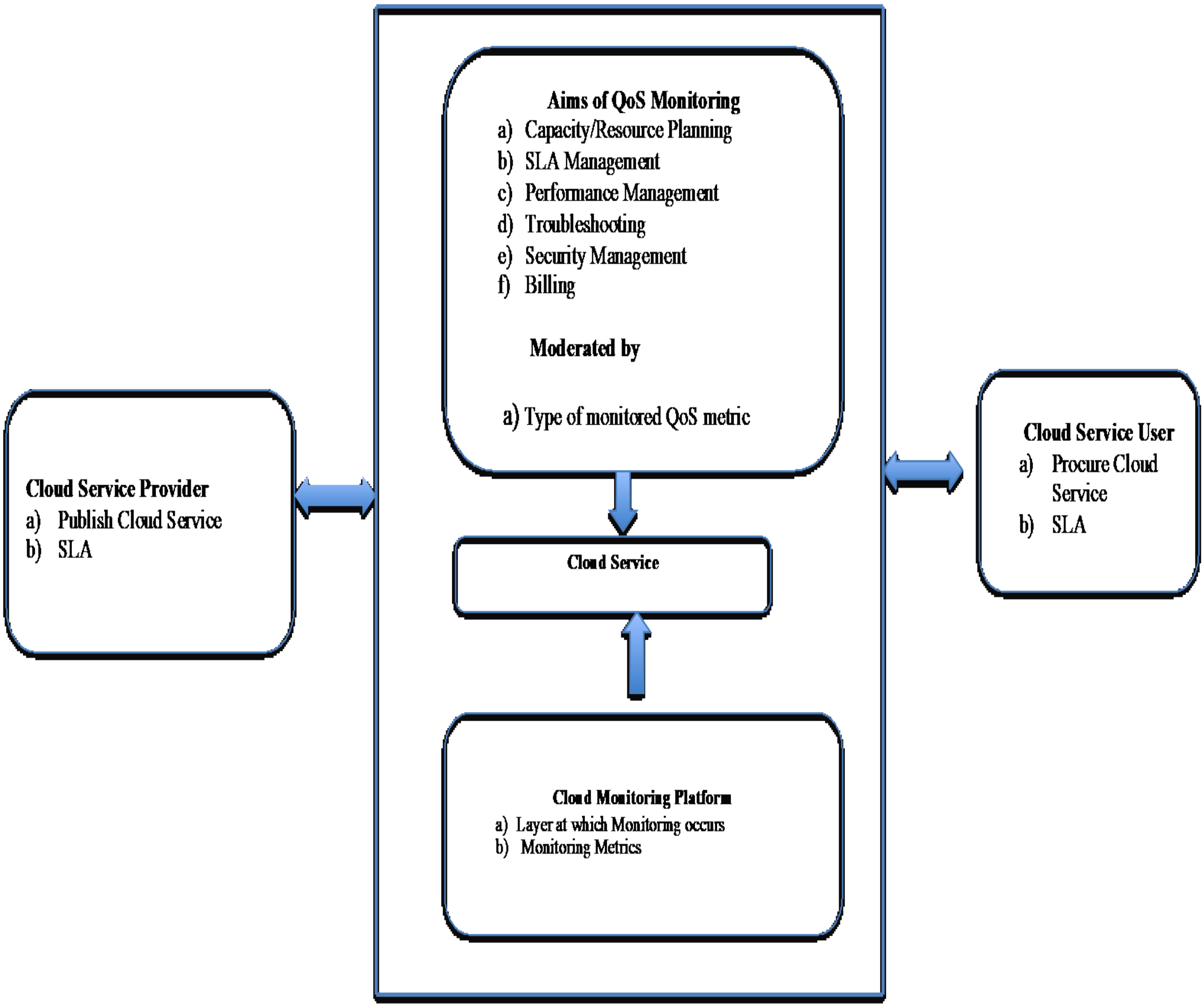


Figure 14: Existing Cloud QoS Monitoring Framework

2.10. Comparable Studies on Developing Multi Cloud QoS Monitoring Frameworks

There have been efforts by other researchers in devising QoS monitoring frameworks and models that can be used for monitoring the QoS of different cloud providers as well as comparing their performance and subsequently ranking them in some instances.

Zeginis, Kritikos, Garefalakis, Konsolaki, Magoutis and Plexousakis (2013) took cognizant of the fact that examining the performance and functionality of services actualized on various cloud providers platforms and modifying them to events produced by various layers of the cloud (PaaS, IaaS and SaaS) in a managed way are research problems for the research community.

To address the challenge, Zeginis et al (2013) proposed an Event Based Multi Cloud Service Applications Framework, which is an events pattern concept for cross-layer cloud services monitoring, which exploits dependencies among layers. The concept distributes mechanism for monitoring across cloud providers by integrating monitoring means in each cloud platform layer and across multiple cloud providers.

The events pattern concept is made of a Monitoring Engine for gathering cross-layer events during service execution, as well as an Adaptation Engine for enabling cross-layer variation actions, that in charge of communicating events via publish/subscribe means.

The model comprises of a manager module, which retrieves results, and then keeps them in a time-series database, after which it reports the noticed violations via the publish/subscribe means to Adaptation Engine instance. The architecture of the Multi Cloud Service Based Application Framework is shown in Figure 15.

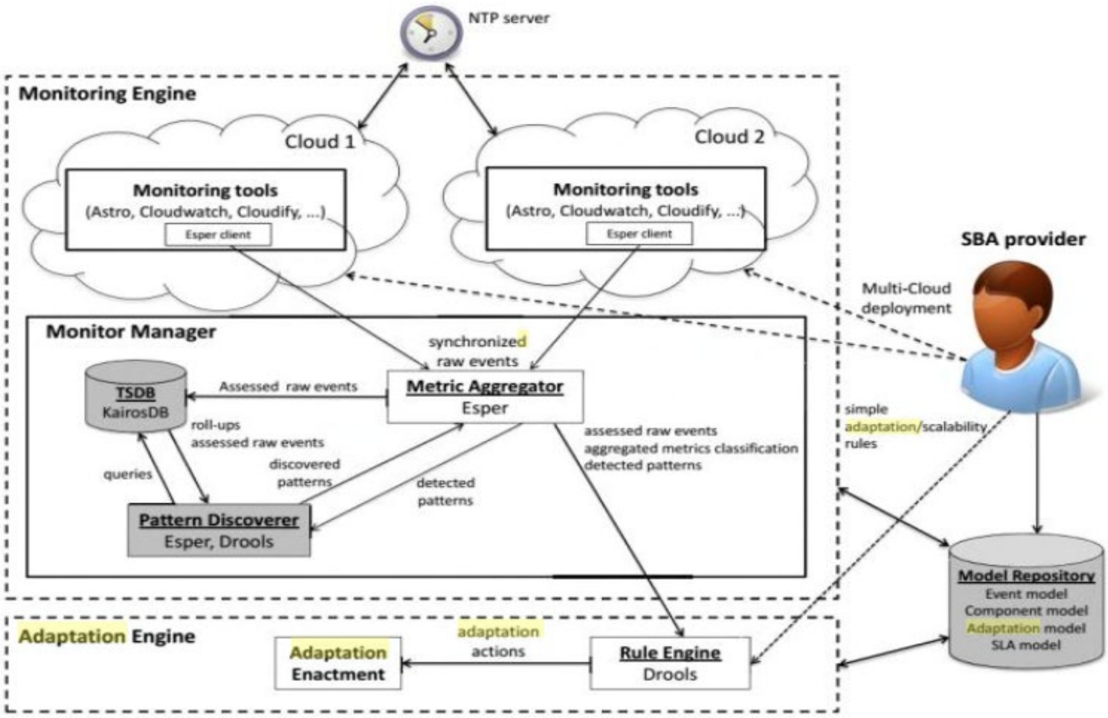


Figure 15: Event Based Multi Cloud Service Based Applications Framework

Source: Zeginis et al (2013)

The Multi Cloud Service Based Framework relies on existing tools, for example cloudify and Amazon cloud watch to perform the actual monitoring. From the description Zeginis et al (2013), it turns out the events pattern concept is a framework that collates data monitored by various tools for analysis. Further, the open source esper client used in monitoring events as captured by the different monitoring tools has to be modified to interface with the various tools.

Since the tools in use are not vendor neutral, they end up monitoring only the clouds for which they have been designed to monitor. Further, the results from these tools cannot be used to compare various cloud providers' performance for choice decision making.

Introducing multiple-cloud platforms like VMware, HyperV and OpenStack, and measuring important features from a centralized location, according to Vicic and Brodnik (2014), is a daunting task. Vicic and Brodnik (2014) argue that cross-cloud monitoring leads to the challenge of upholding compatibility amongst dissimilar properties in different clouds which is compounded by the fact that APIs of different clouds are quite different. From Vicic and Brodnik (2014) it is concluded that every cloud implementation model has unique requirements and needs unique approaches.

To solve the challenges of multi cloud monitoring, Vicic and Brodnik (2014) developed a Multiple-Cloud Monitoring platform for IaaS cloud services that relied on having access to the information concerning the hosts and virtual machines via standardized interfaces namely installed probes and API links to the platforms.

The architecture consists of a control system that is in charge of collecting data and making them available to the SLA control system and to the control dashboard. The control system is capable of communicating directly with the available interface for a virtual platform via additional software installed on the control system. Alternatively, A gateway also known as a translation interface, is implemented for each virtual platform between the virtual platform and the interfaces used by the control system. The architectures of the two possible designs are shown in Figure 16 and 17 respectively.

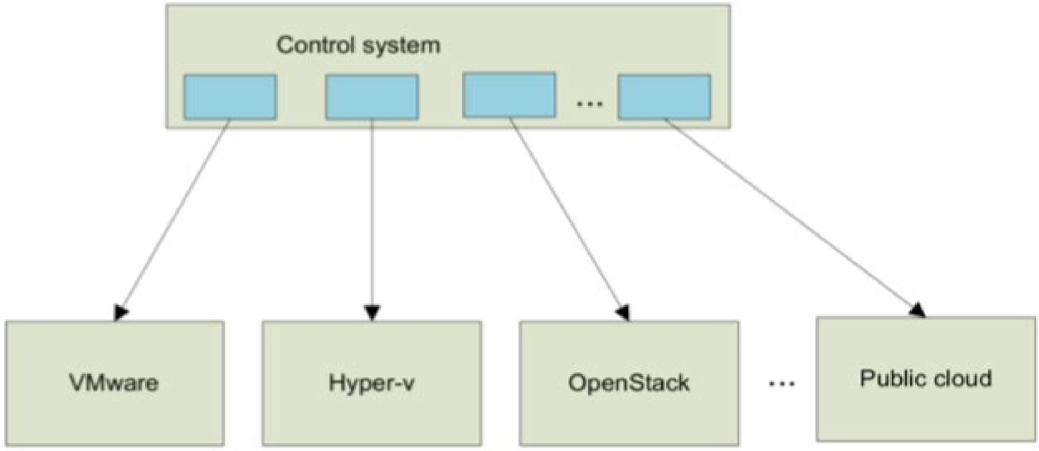


Figure 16: Architecture with Direct Communication

Source: Vicic and Brodnik (2014)

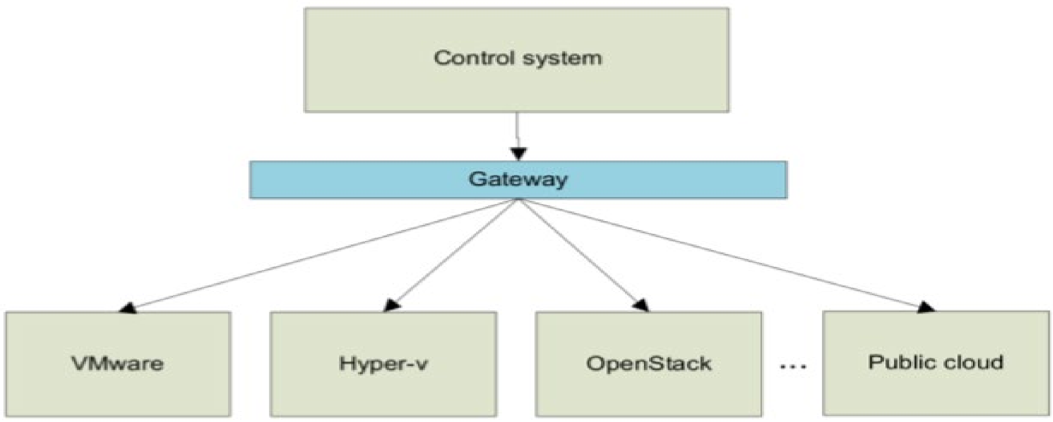


Figure 17: Architecture with Communication through Gateway

Source: Vicic and Brodnik (2014)

The shortcoming of this approach is that it relies on existing vendor specific tools like the ganglia and nagios and collates the monitoring results from the different tools and thus the results can not be used across various vendors for performance comparison. It is thus designed around the architecture of the cloud provider. This approach also introduces additional hardware (probes) in the networks increasing costs and possible points of failure as well as point of attack.

An evaluation and ranking framework, was proposed by Upadhyay (2017), namely, the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), for cloud services to aid in selection of which provider best satisfies the cloud requirement of a customer. The framework is composed of a cloud administrator which is in charge of communicating with the cloud data discovery component to acquire the needed data for service parameter, cloud data discovery that is made up of cloud services, cloud monitor component, history manager component and cloud service discovery.

The cloud administrator component is in charge of evaluating the QoS of the cloud service by ranking cloud services in the form of indices. The measuring component of the cloud service receives the customer's evaluation request for the cloud service.

Tracking of the customer's SLA with the cloud provider is also done by the cloud manager component as well as the fulfillment history of those SLAs. One of more QoS parameters is used by the cloud service measurement component to produce service index on which providers of cloud services best fit to the user service request requirements.

The parameters monitored by this framework are speed of VM, memory, scaling latency, storage performance, network latency and available bandwidth. With each customer specifying their own SLA with regards to the listed parameters, the framework keeps a history of what the customer requested and how the platform performed with regards to those parameters.

The parameters monitored by this framework are speed of VM, memory, scaling latency, storage performance, network latency and available bandwidth. With each customer specifying their own SLA with regards to the listed parameters, the framework keeps a history of what the customer requested and how the platform performed with regards to those parameters.

The modeling type used was mathematical modeling where equations were developed from existing literature and on the weights and importance placed on certain SLA parameters by clients, which the numerical technique of mathematical modeling used in arriving at solutions to the equations.

The deficiency of this framework is in the fact that it relies on the advertised services of the cloud service providers and the history of how the SLAs of previous customers were met by the service provider. Figure 18 illustrates the architecture of the TOPSIS framework.

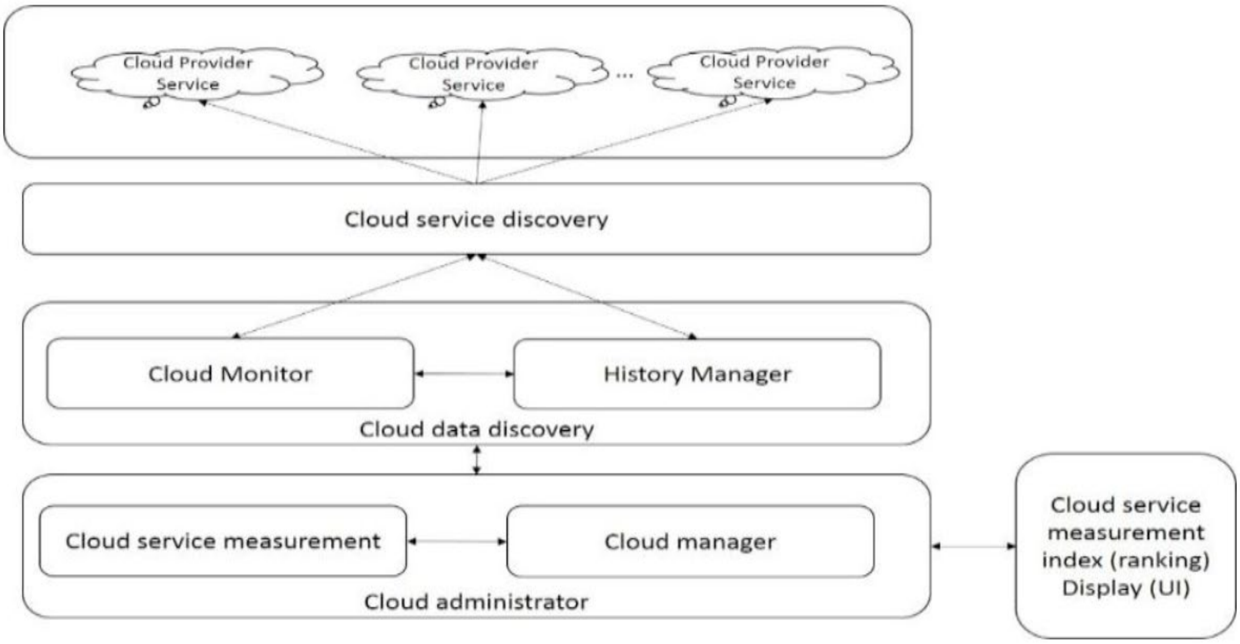


Figure 18 : the TOPSIS Cloud Discovery and Ranking Framework

Source: Upadhyay (2017)

It is given that applications that are dependant on the combined usage of various independent clouds front a challenge of controlling their security due to lack of knowledge on the security measures put in place by the cloud providers, in addition to the need to monitor simultaneously the behavior of various individual components implemented in dissimilar clouds (Rios, Mallouli, Rak, Casola & Ortiz, 2016).

It is on the premise of the security challenges that Rios et al (2016) developed an SLA-led monitoring of multi-cloud application security compliance framework, namely, the Multicloud Security Applications (MUSA) framework.

At design time, during the SLA creation process, security levels of the application, as well as controls and metrics are specified and after the application components are implemented over the multi-cloud they are continuously monitored at run time.

The MUSA framework distinguishes between multi-cloud and federated cloud set-ups in that multi-cloud means the usage of various, sovereign clouds by a user or a service while federated clouds refers to a scenario where a group of cloud service providers willingly interlink their cloud platforms to enable sharing of resources amongst themselves (Grozev & Buyya, 2012).

The monitoring of security service level agreements in MUSA is dependent on usage of various solutions, which are either developed on an ad-hoc basis or are already in existence as open-source or commercial products to get the metrics required and the various indicators needed check their validity (Rios et al, 2016).

To give a holistic approach, the security monitoring is hinged on the Montimage Monitoring Tool (MMT), which uses a combination of Deep Packet Inspection (DPI) and data mining techniques at both network and application component levels to collect and analyse measurements.

The Montimage Monitoring Tool is comprised of monitoring agents positioned in different cloud components for continuously capturing as well as analyzing network communication in addition system status and monitoring libraries for combining data captured from different agents, and computing security-related metrics to check the conformity of service level agreement as well as triggering security alerts or violations based on the event rules (Rios, Iturbe, Mallouli & Rak, 2017). The MUSA architecture is shown in Figure 19.

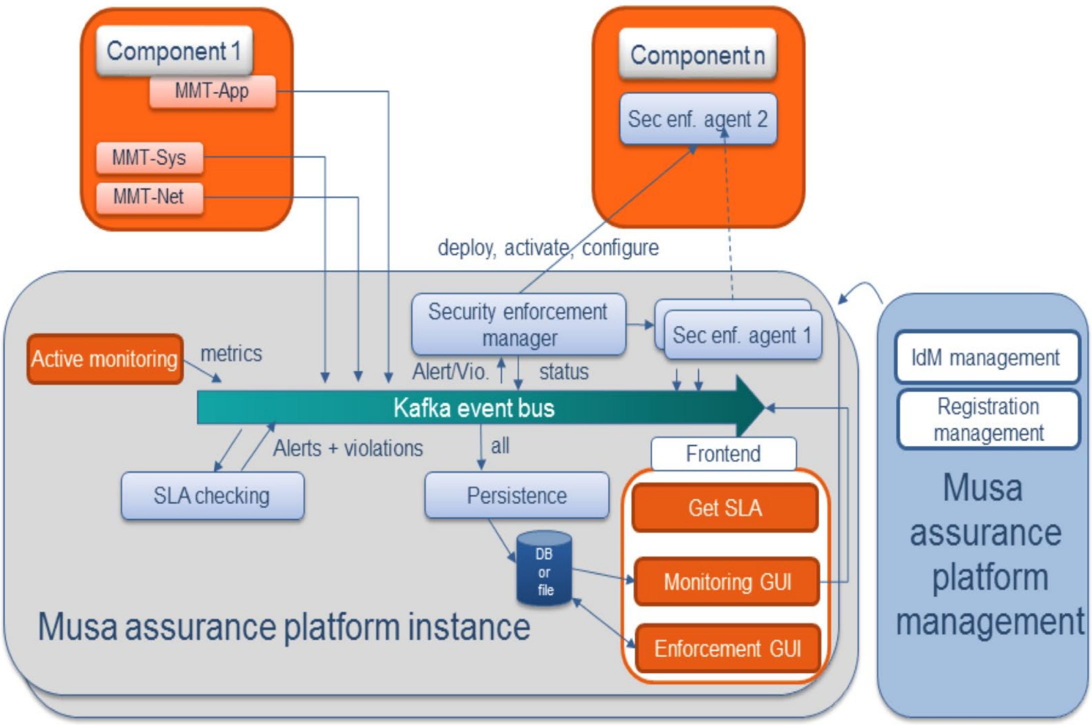


Figure 19: The MUSA Security Assurance Architecture

Source: Rios et al (2017)

The workflow for the MUSA security assurance framework is shown in Figure 20.

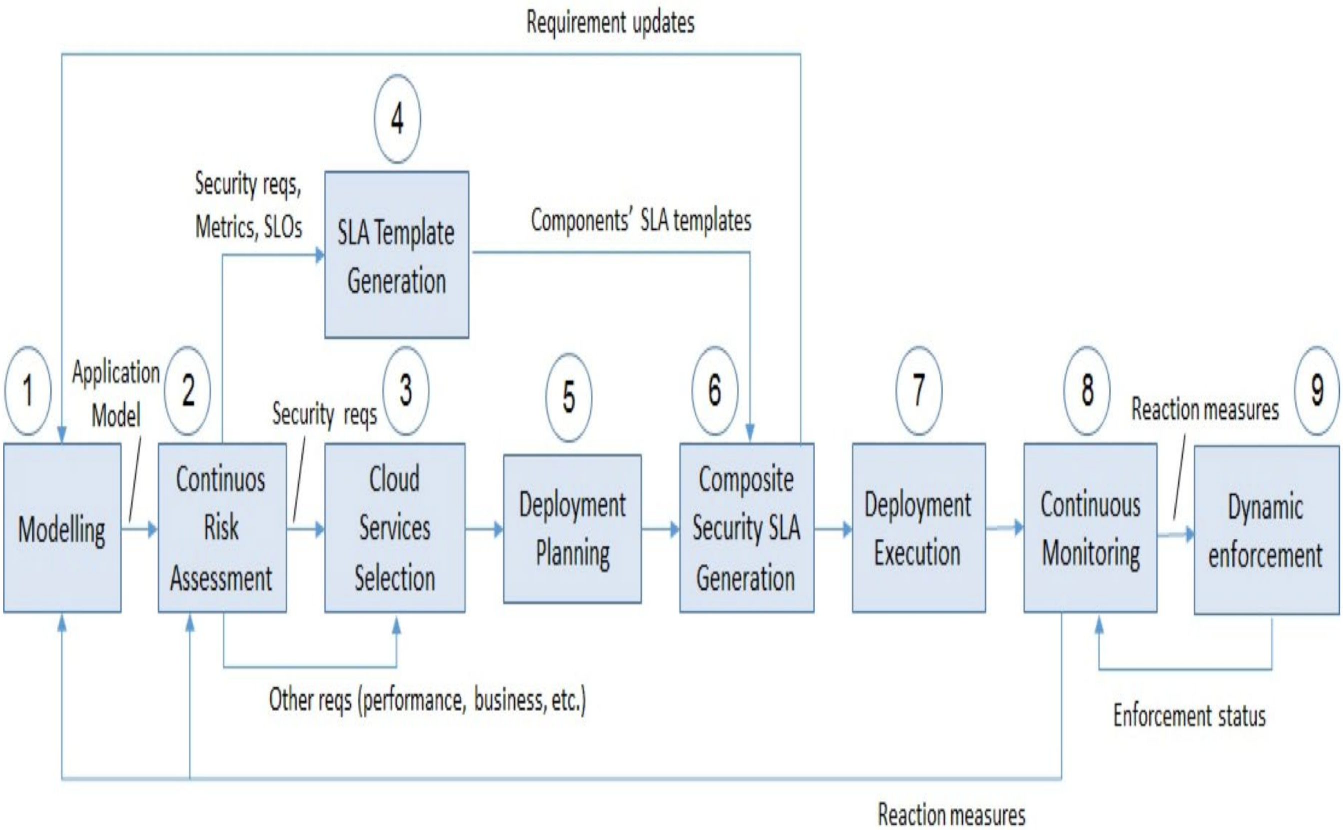


Figure 20 : The MUSA Framework Workflow

Source: Rios et al (2017)

The shortcoming of the MUSA framework is that it is limited to security only as the feature of monitoring of the cloud services, it uses existing commercial tools which are not vendor neutral, and due to the different tools used it is not possible to use it for cross vendor performance appraisal.

Rizvi, Roddy, Gualdoni and Myzyri (2017) postulate that after a company makes the decision to make use of cloud services, the major task ahead is not only choosing the right cloud service provider, but also constantly monitoring the level of services as supplied by the cloud service provider. Rizvi et al (2017) argue that this is due to the fact that the signed cloud service pacts by the cloud user and the cloud service providers can be inflexible and unmaintained.

To abate the stated challenge, Rizvi et al (2017) proposed a third party auditor model whereby a third-party auditing body like cloud brokers, cloud carriers, and cloud auditors can help a cloud user in seeing to it that they receive the assured services from their selected cloud provider. Similar efforts were also done by Mutulu and Kahonge (2018) in their work on Mutlitenancy cloud model using QoS.

The model by Rizvi et al (2017) has a three step approach, consisting of an initial appraisal of any treasured information useful to cloud service agreement, an evaluation of explicit cloud metrics, and quarterly re-evaluations of the cloud service agreements. The model's ultimate goal is building trust amongst the cloud service user and the cloud service provider. A high level diagram depicting the third party auditor is shown in Figure 21.

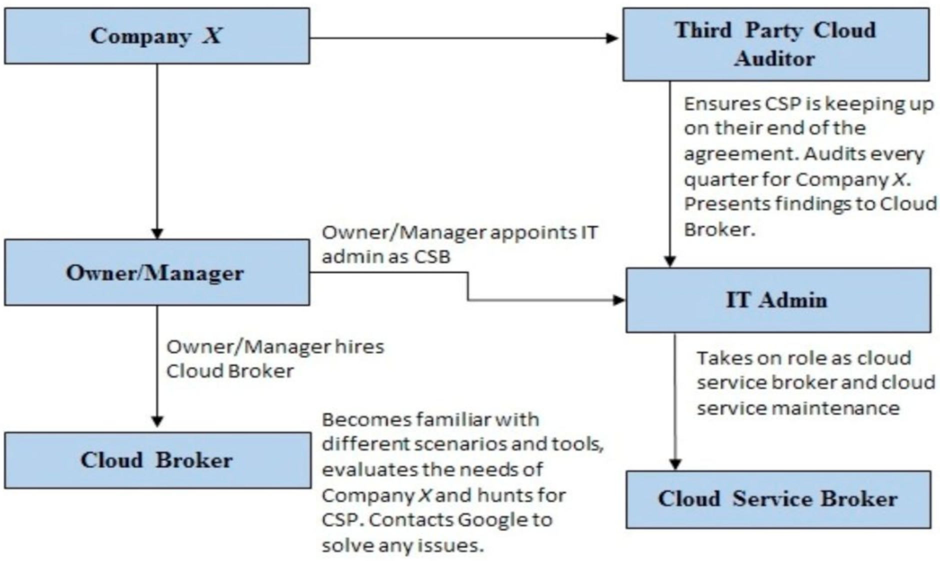


Figure 21: Third Party Audit Model

Source : Rizvi et al (2017)

The challenges posed by the model as proposed by Rizvi et al (2017) is that it introduced overheads and costs in the cloud computing process, which are, the need for cloud service brokers (CSB) to source for best cloud service providers (CSP) for the clients and cloud auditors to perform quarterly review of the cloud services as provided by the cloud providers.

Further, the Rizvi et al (2017) model relies on the tools of some of the cloud providers to perform the actual cloud QoS monitoring, like, Intel's Benchmark Install and Test Tool, IBM's CloudBench and Google's PerfKit. This tools are not only limited in the number of clouds they can monitor, but could pose trust issues in cases where the cloud provider being monitored is the same one being monitored or a competitor's cloud is being monitored.

Because of service selection overload posed by a plethora of cloud applications, Azubuike, Olawande and Adigun (2018), proposed a QoS-based rating and choosing of SaaS applications by use of Heterogeneous Similarity Metrics (HSM).

The Heterogeneous Similarity Metrics (HSM) makes use of combined quantitative and qualitative dimensions for QoS-based rating of cloud-based services by making use of synthetically acquired dataset from cloud services.

The Metrics in the Heterogeneous Similarity Metrics are Heterogeneous Euclidean-Lin Metric (HELM), Heterogeneous Value Difference Metric (HVDM), Heterogeneous Euclidean-Overlap Metric (HEOM), Heterogeneous Euclidean-Eskin Metric (HEEM), and Heterogeneous Euclidean-Goodall Metric (HEGM).

However, the Heterogeneous Similarity Metrics (HSM) shortcoming is based on the fact that it uses artificially generated datasets on the HSM mathematical equations to rank the various cloud services.

According to Ibrahim, Wasim, Varrette and Bouvry (2018), the quality of the offered services offered is not guaranteed by the service level agreement by the fact that it is just a contract.

It is on this premise that Ibrahim et al (2018) proposed an automatic framework named PeRformance Evaluation of SErvices on the Cloud (PRESENCE), to appraise the QoS and service level agreement fulfillment by Web Services obtained from several cloud service providers.

PRESENCE is based on the multi agent system, each agent is responsible for a particular performance metric monitoring a certain aspect of the QoS. Other components of PRESENCE are monitoring and modeling module which is responsible for collecting the data from the agent, stealth module which is responsible for dynamically modifying and balancing the pattern of the workload of the amalgamated metric agents to make the resultant traffic similar to the routine traffic from ordinary users from the cloud service provider viewpoint (Ibrahim et al, 2018).

PRESENCE has also a QoS aggregator virtual in nature and service level agreement checker component, which is in charge of assessing the QoS and service level agreement compliance of the service accessible from the considered cloud service providers and PRESENCE client also known as Auditor that is in charge of relating with the selected cloud service providers and assessing the QoS and service level agreement observance of web services.

An architectural diagram of PRESENCE depicting the various subcomponents is shown in Figure 22.

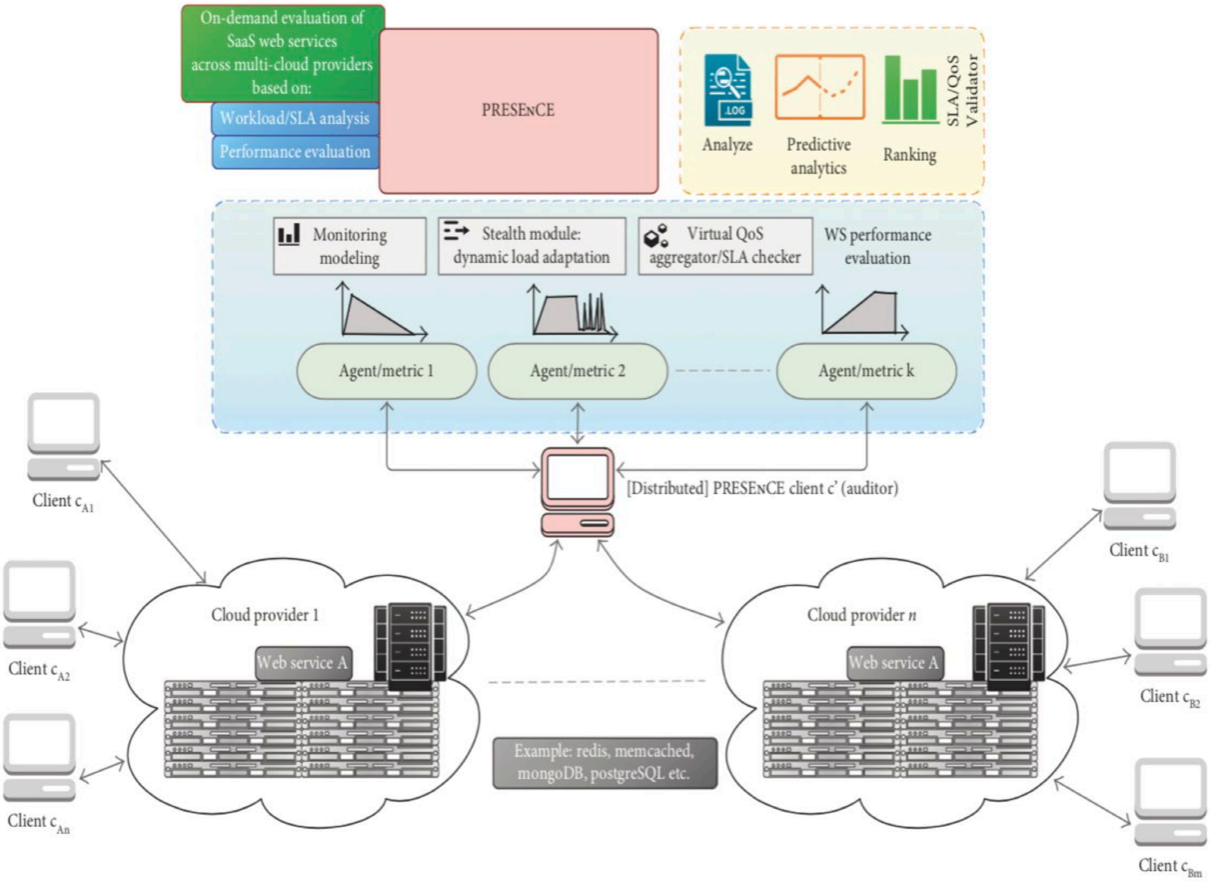


Figure 22: PRESENCE Architectural Diagram

Source: Ibrahim et al (2018)

Whereas the framework can be of use in matching the performance of several cloud service providers, it relies on artificially generated data, which travels alongside the usual natural traffic of users. Further, the fact that the agents have to be customized for each cloud provider, the credibility of comparisons is in doubt since the agents taking the measurements do no have the same configurations and internal set ups.

As acknowledged by Alhamazani, Ranjan, Jayaraman, Mitra, Liu, Rabhi, Georgakopoulos and Wang (2019), contemporary cloud measuring frameworks are by large incompatible across various cloud service providers.
To abate the shortcoming, they proposed Cross-Layer Multi-Cloud Application Monitoring as well as Benchmarking as a Service (CLAMBS). CLAMBS has a capability of service monitoring as well as benchmarking of specific individual components of the application like databases and web servers, that are spread within cloud layers (*-aaS), and spread among various cloud service providers.

According to Alhamazani et al (2019) CLAMBS makes use of an agent based technique for cross-layer, multi-cloud resource or application monitoring plus benchmarking. It is made up of three key components, which are the Monitoring Agent, Manager and Benchmarking Agent.

The manager gathers QoS data from Monitoring Agents while benchmarking information is obtained from benchmarking agents, which run on various virtual machines (VMs) across multi-cloud providers as well as environments.

The monitoring agents reside in the virtual machines executing the application, collecting and sending QoS data as required by the manager. The benchmarking agent has standard functions for measuring network performance between the data center(s) hosting the application service and the user of the application (Alhamazani et al, 2019).

The benchmarking as well incorporates a load-generating component which generates traffic to benchmark the application based on a given workload model. Figure 23 depicts the various components of the CLAMBS model.

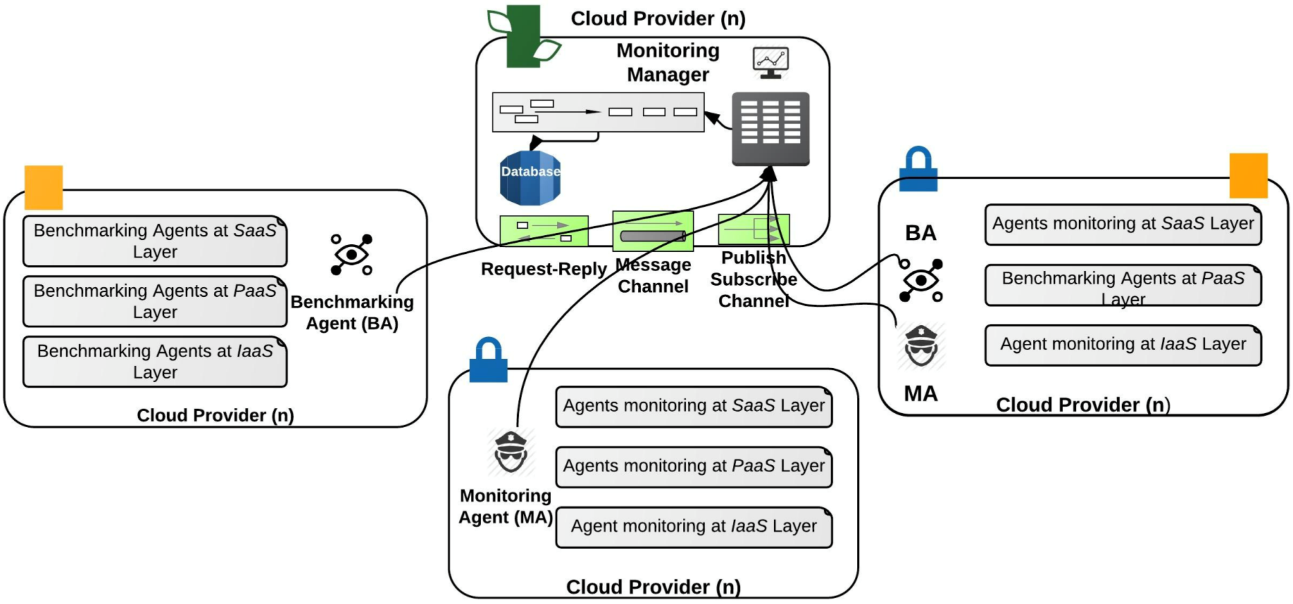


Figure 23: CLAMBS Architecture

Source: Alhamazani et al (2019)

Whereas CLAMBS is multi platform in nature in that it can monitor various cloud platforms, it is not a purely vendor neutral model in the sense that the agents as deployed is the various cloud platforms have to be modified to be able to integrate various different cloud platforms. The user therefore will be limited to the cloud vendors for which the CLAMBS model has already been customized for.

2.11. Evaluating Trust in Information Systems

Works on social virtues and prosperity by Fukuyama (1995) states that trust is the anticipation that ensues within a group of regular, sincere, and behavior that is cooperative in nature, founded on generally shared norms, on a segment of the members of the group.

In developing an integrative model for trust in organizations, Mayer, Davis, and Schoorman (1995) describe trust in terms of the readiness of a party to be susceptible to the actions of another party based on the anticipation that the other will implement a particular action vital to the trustor, regardless of the ability to control or monitor that other party.

Three features of another party in which opinions of trust can be founded, namely, integrity, benevolence and ability were further identified by Mayer et al (1995). The work on responses to crisis in organizations, by Mishra (1996), in particular on the centrality of trust, lists four dimensions of trust, namely, competence, reliability, concern and openness.

Based on the reviewed works, in the context of QoS measurements in Information Systems, this research defines trust as the level of confidence a service user has over the QoS measurements results presented by the service provider.

Trust is considered a non functional property of a service, according to Zainab, Perry and Capretz (2011), which can be used in service selection, in cases where there are similar services on offer.

To evaluate the trustworthiness of a service or service provider, trust metrics are required. Zainab et el (2011) define trust metrics as the information of an entity that is required and used to evaluate the trustworthiness of the entity, with an entity being a service or service provider.

A summary of the trust metrics developed by Zainab et el (2011) is depicted is Figure 24.

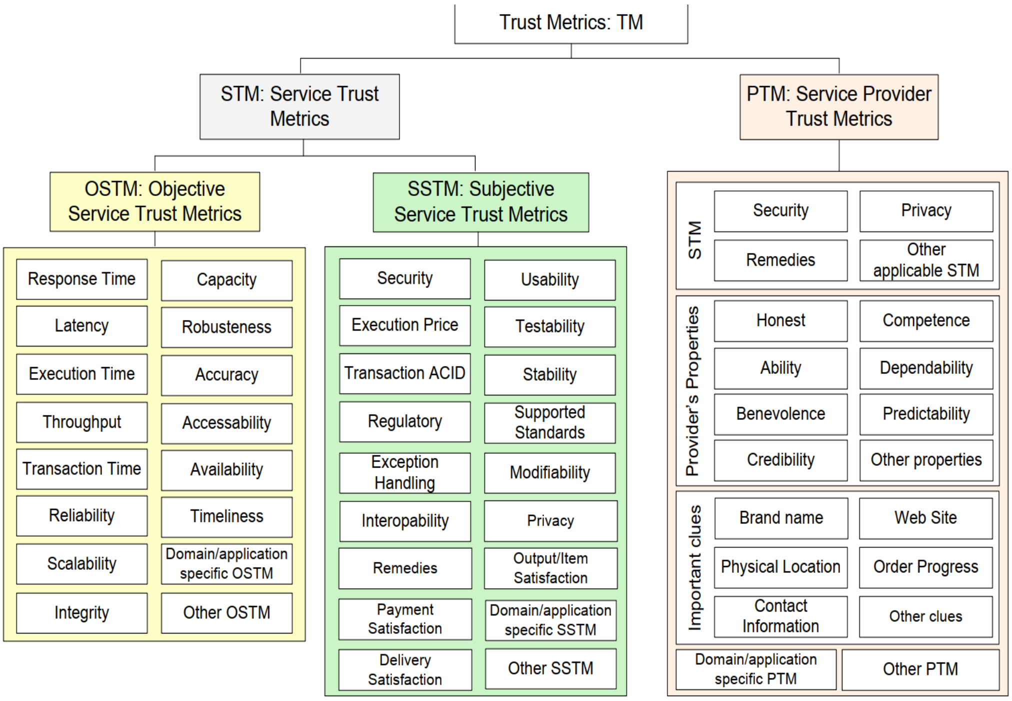


Figure 24 : Trust Metrics

Source: Zainab et el (2011)

Trust based approaches for online service choice, as proposed by Drogani (2009), are Direct Experience, Third Party Trust, a Hybrid approach and Trust Negotiation. This research focuses on the direct experience approach. The approaches are summarized in Figure 25.

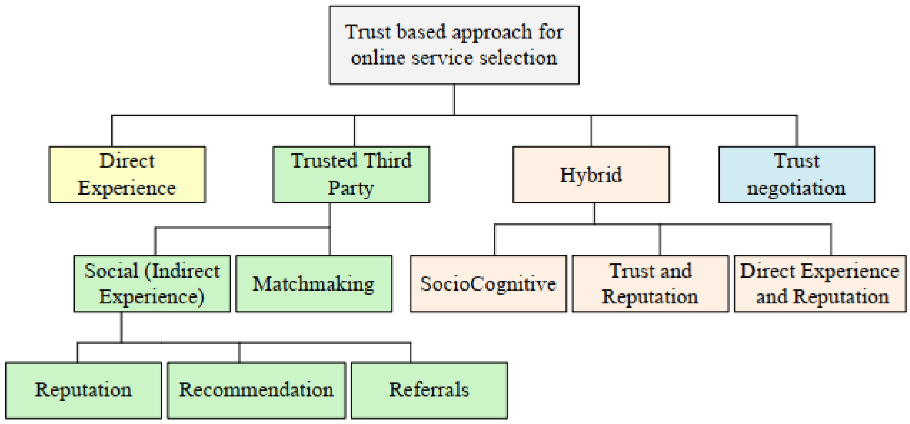


Figure 25: Trust Based Approach for Service Selection

Source: Drogani (2009)

According to Chu, Harley and Xu (2016), key mechanisms for measuring the trustworthiness of a computing platform includes usage of the system of concern and related features, alongside states and behavior; threats comprising of errors, faults and flops caused by intentional actions like attacks or unintentional actions; use of key metrics of trustworthiness; and means to develop trustworthy systems and relationships between assessment like vulnerability assessment, penetration testing, red teaming, and submetrics or attributes of a metric for trustworthiness.

The concept of initial trust, which is trust granted in an unfamiliar entity, system or person, in a setting where the actors do not yet have credible, meaningful information about, or affective bonds with each other was introduced by McKnight, Cummings and Chervany (1998). Credible information is acquired after parties interact with one another for some time.

A trust model on how to Develop and Validate Trust Measures in e-Commerce settings, using an Integrative Typology, containing trusting beliefs, as well as disposition to trust, trusting intentions and institutional based trust, was proposed by McKnight, Choudhury, and Kacmar (2000). The model is as shown in Figure 26.

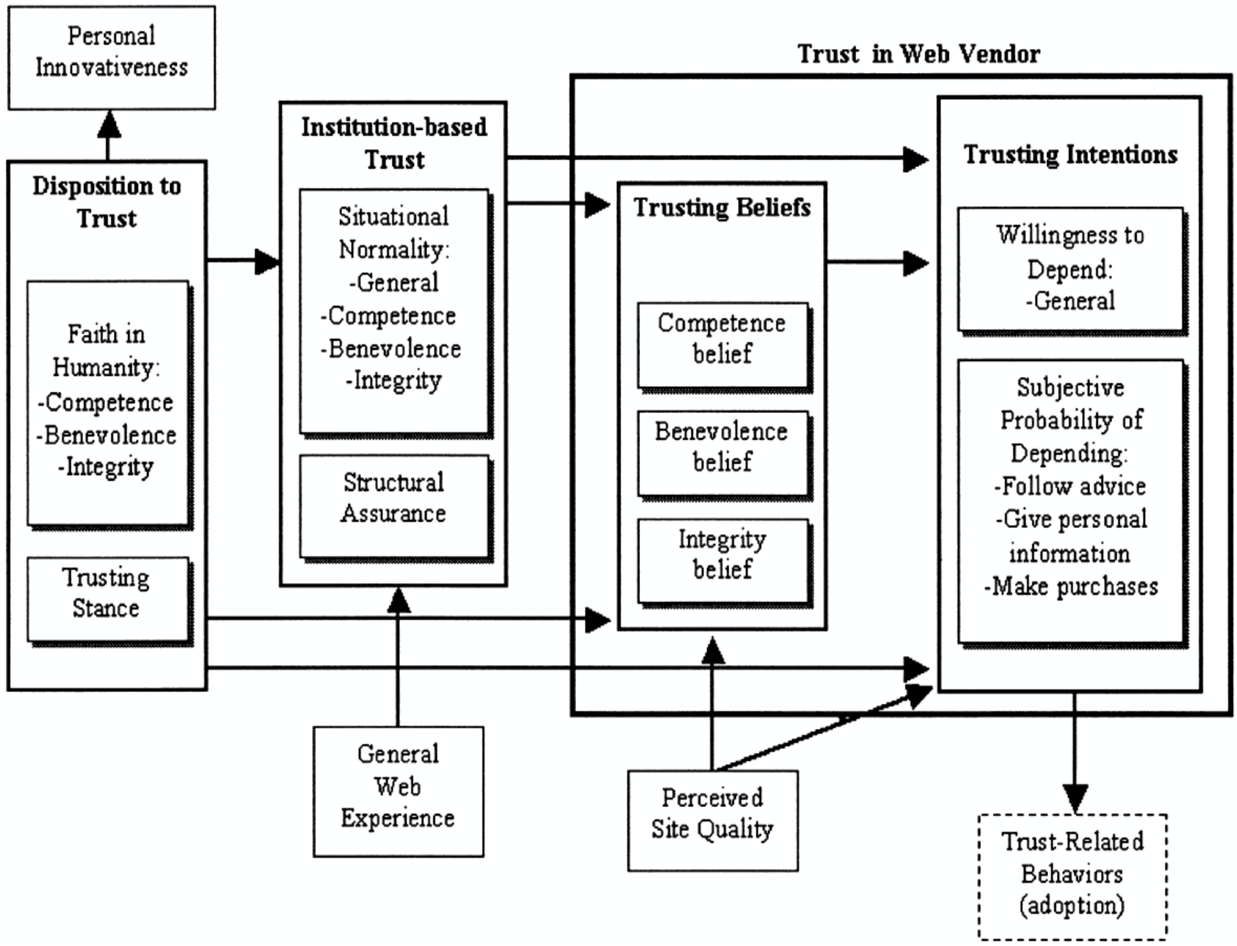


Figure 26 : Web Trust Model

Source: McKnight et al (2000)

A framework for measuring trust in organizations was developed by McEvilya and Tortoriellob (2011), in their work on measuring trust in organizational research. The framework is shown in Figure 27.

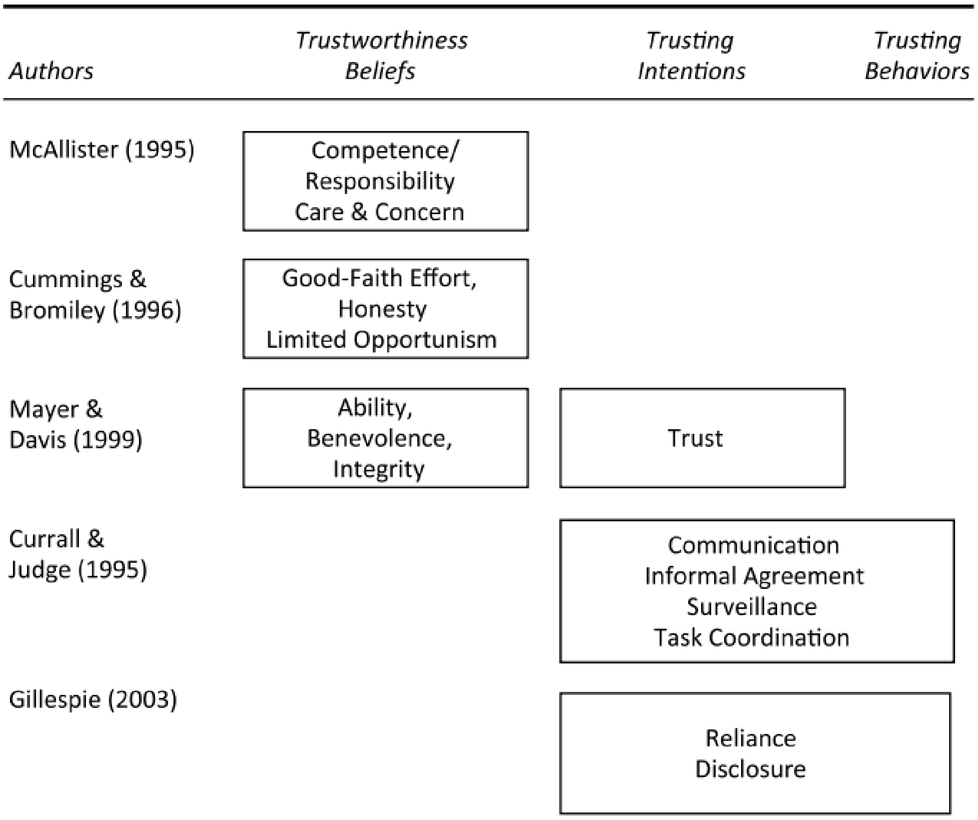


Figure 27 : Framework for Measuring Trust.

Source: McEvilya and Tortoriellob (2011)

The turnaround trust model for measuring trust was postulated by Gholami and Arani (2015). In the turnaround trust model, trust is computed based on equation 1 to 5 as derived by Manuel (2013).

$$Trust\ of\ R_k = W_1 * AV + W_2 * RE + W_3 * D1 + W_4 * TE \qquad \textit{Equation 1}$$

$where$ $W_1 + W_2 + W_3 + W_4 = 1$ and $R_k = resource\ k$

The weight values (W) are assigned depending on their priority as well as trust assessment criteria, with AV representing availability, while RE represents reliability, followed by DI which is data integrity and finally TE is response time performance

The availability of a given resource (R_k) is computed as a ratio of the accepted jobs against the total number of jobs submitted per given time period.

$$\text{Availability of (AV) of } R_k = \frac{A_k(\text{total accepted jobs})}{N_k(\text{total submitted jobs})} \qquad \textit{Equation 2}$$

Reliability of a given resource (R_k) is computed using a ratio of the total completed jobs against the total accepted number of jobs.

$$\text{Reliability (RE) of } R_k = \frac{C_k(\text{total completed jobs})}{A_k(\text{total accepted jobs})} \qquad \textit{Equation 3}$$

Data Integrity of a resource is a computation of the ratio of jobs completed with integrity preserved by a given resource (R_k) against number of total jobs completed.

$$\text{Data Integrity (DI) of } R_k = \frac{D_k(\text{No of Integrity preserved})}{C_k(\text{total completed jobs})} \qquad \textit{Equation 4}$$

Turnaround Efficiency for a job by a given resource (R_k), which is time taken to complete a task computed as:

$$\text{Turnaround Efficiency (TE) of } R_k = \frac{\text{Promised Turnaround}}{\text{Actual Turnaround time}} \qquad \textit{Equation 5}$$

The pictoral resprentation of the turnaround trust model is depicted as shown in Figure 28.

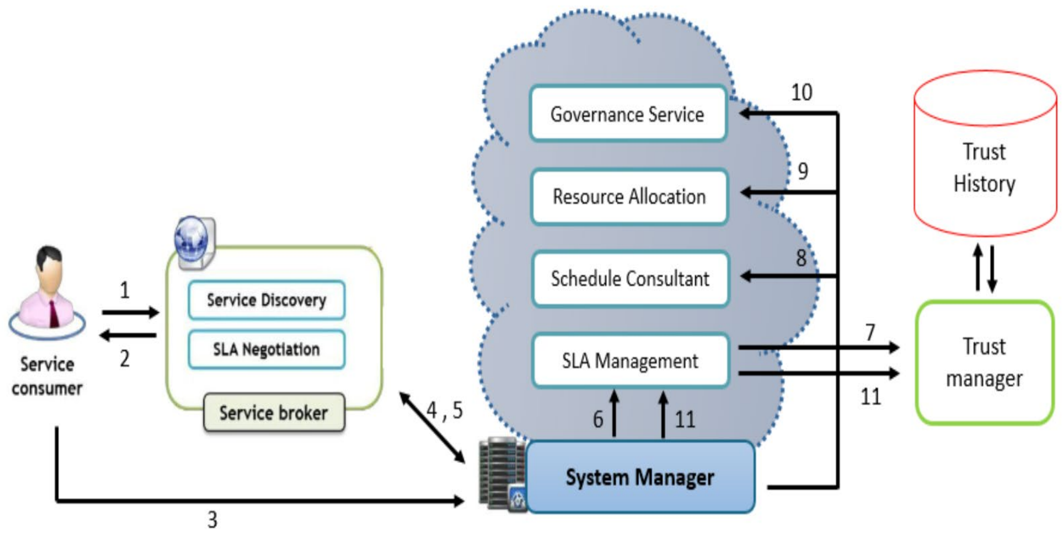


Figure 28 : Turn Around Trust Model.

Source: Gholami et al (2015).

Trust has been noted to be dynamic concept dividable into three growth phases: building of trust, which involves forming trust; stabilizing of trust, in a scenario where trust already exists; and termination of trust, in which case trust ends (Grabner-Kräuter & Kaluscha, 2008).

A Computationally Grounded Quantitative Trust with Time, by Nagat, Jamal and Hongyang (2020), introduces a model for computing the degree of trust. The model, known as the model of Trust Computation Tree Logic (TCTL^G) is defined as a tuple: $MG = (SG, IG, RG, \{\sim i \rightarrow j \mid (i, j) \in Agt2\}, VG)$ where: SG is a non-empty set of attainable global states of the system; $IG \subseteq SG$ is a set of initial global states; $RG \subseteq SG \times SG$ is the conversation relation;

$\sim i \rightarrow j \subseteq SG \times SG$ represents direct trust accessibility relation for each truster-trustee pair of agents $(i, j) \in Agt2$ defined by $s \sim i \rightarrow j\ s'$ iff: $li\ (s)(vi\ (j)) = li\ (s')(vi\ (j))$, and s' is attainable from s using transitions from the transition relation R;

VG: $SG \rightarrow 2AP$ is a labeling function, with AP being a set of non-divisible propositions. The model starts by defining local and corresponding global states of the agents in trustworthy states. Trust of i towards j, ($\sim i \rightarrow j$), exists only if the element values of local and global states of the two agents are same.

This model has a shortcoming with regards to the need to define all possible states in the system states that are considered to be trustworthy from the vision of agent i with regard to agent j. In a multi agent system with many agents, the combinations that will result from this arrangement will be enormous.

The model is also limited to a multi agent system, which is under a single administrative domain. In disparate systems under different domains, it is not possible to define the trust worthy states to be used by agents from the disparate systems.

A quantitative framework for accessing cloud security, using a dependency model that validates both the offered services and customer's requirements validated by checking service conflicts and different Service Level Obligation compatibility issues, is proposed by Taha (2018).

The proposed dependency model is composed of five stages, namely, Security requirements definition, Requirements Quantification, Dependency management approach, Structuring security SLA services using Dependency Structure Matrix and Cloud Service Provider Evaluation.

The proposed framework and model suffers from the limitation of the fact that customers are only able to trust the result of the proposed assessment as long as the information taken as input is reliable (Taha, 2018).

This calls for the use of an independent auditor to perform a third-party attestation of the cloud provider's security SLA through a scheme such as the Cloud Security Alliance Open Certification Framework, as well as the fact that the model is limited to security issues of the cloud based services only.

A composite trust metric, consisting of impression and confidence was introduced by Yefeng, Ping, Lina and Arjan (2017). The authors advance the fact that trust can be composed using algorithms by observing past events, such as good or bad evidence or responses on social platforms.

The proposed framework by Yefeng et al (2017) is based on measurement theory, Dempster –Shafer belief theory, and error propagation theory. The framework has three phases, namely, trust modeling, where trust related information is mapped on trust metrics. For example, reviews and proposition from users of epinions.com, likes and dislikes from users of Facebook.

The second phase is trust inference, which focuses on spreading and combining the collected metrics of trust over the entire network or the portion of interest, while decision making using the measured trust is the final phase.

The widely used metrics for trust depiction are binary metrics, scaled metrics, probability based metrics and similarity based metrics are used (Yefeng et al, 2017). The proposed framework uses a model expressed as: ***T (m, c***), where **m** measures how trustworthy from truster's point of view the trustee is, while, ***c*** which is confidence measures to what extent the truster is in terms of believing in the evaluation of impression/trustworthiness ***m.***

The modeling for the trust values for the epinios.com platform is computed as: To obtain a relation with regards to trust from user ***A*** to user ***Z***, the impression **m** is the mean of assessments that A rates Z's review articles. Which thereafter is converted into a value in [0, 1] as:

$$m_Z^A = \frac{\sum_{i=1}^{i=N} Rating_i}{5*N}$$

For twitter, interactive tweets are used to build trust using sentiment analysis. Using sentistrength, an analysis is constructed for each tweet, which gives a discrete score from −4 to +4 for every tweet. This is then converted into discreet values into the interval [0, 1], using the equation:

((Sentiment+4)/8).

Whereas this model develops measured values for trust, it is a highly subjective process. The reviews, likes, dislikes are all assigned by users based on their perceptions, moods, social cultural inclinations and subjective interpretations. These user perceptions are likely to change with time, or as new information emerges and are thus not objective hence not suitable for use in scientific modeling.

To address the highlighted shortcomings in existing trust models, Makokha, Chepken and Opiyo (2021) proposed an End User Centric Quantitative Trust Model in Cloud Computing. The quantification of trust is meant to evaluate trust and generate a binary value of one (1), if true exists, and Zero (0), if there is no trust

The proposed model is pictorially represented in Figure 29.

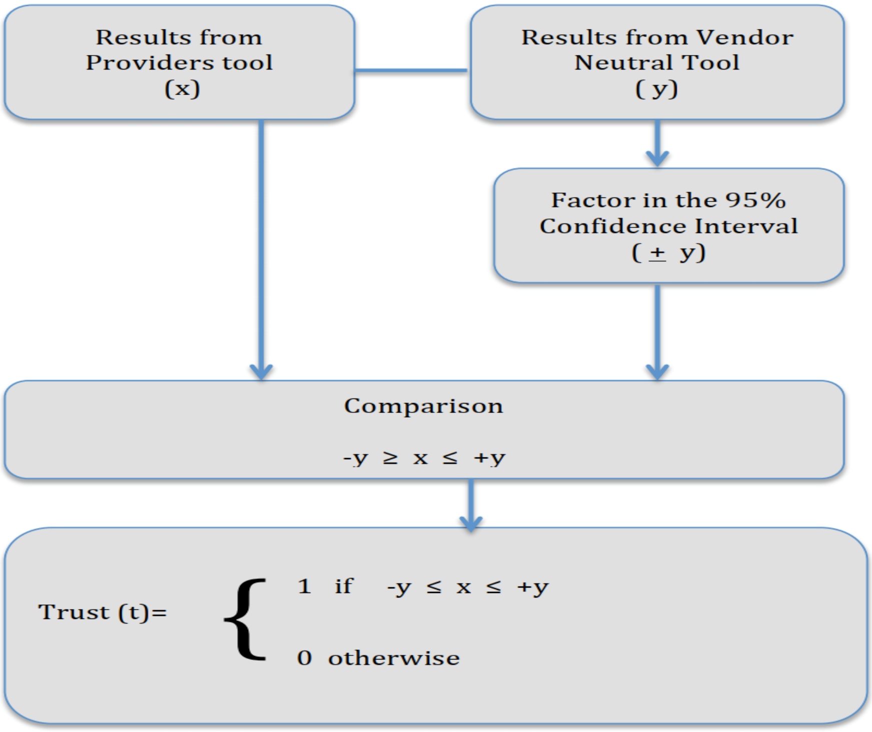


Figure 29 : Proposed Trust Quantification Model

Source: Makokha et al (2021)

The End User Centric Quantitative Trust Model is premised on the fact that cloud computing solutions have embedded capabilities to monitor and measure QoS. The capability measures QoS as provisioned by the provider, the results are then available for users to query from the providers' systems.

A comparison can thus be made with the results from the same cloud platforms obtained using a vendor neutral QoS monitoring model developed by Makokha et al (2019), which measures QoS across all cloud providers. This comparison can then be modeled quantitatively, resolving to one (1) if the results from the service provider are within the 95% confidence interval of the results from the vendor neutral tool, thus signifying trustable results and zero (0) otherwise.

2.12. Research Gulf

From the synthesized literature with regards to QoS monitoring in cloud computing solutions, there exists a gulf in so far development of non intrusive cloud QoS monitoring tools is concerned. All existing models and the associated tools from the models are anchored on the architecture of the cloud platform on which they measure the QoS. This is realized by use of either APIs, customized agents or adaptation layers between the cloud architecture and the monitoring tools

There also exists a gulf in so far as the perspective from which QoS is monitored in cloud computing solutions, as depicted in the existing QoS monitoring framework. The existing cloud QoS monitoring framework monitors QoS from a cloud service provider perspective, making it a vendor centric framework. The existing framework further helps cloud providers to know the utilization levels of the cloud platforms to make decisions on whether to increase physical resources. Unfortunately, the cloud user is left with no option but to rely on information as received from cloud providers. The information received from cloud providers, about the performance of their platforms during SLA evaluation, is not sufficient for the client to build confidence in cloud platforms.

This research set out to address the research gulf of vendor intrusiveness of existing cloud QoS models and tools as well as vendor centricity of the cloud QoS monitoring framework and model, as evidenced by lack of classical and contemporary literature addressing the two issues.

2.13. Chapter Summary

This chapter reviewed existing knowledge in the realm of cloud computing, noting the various cloud computing service models and the reasons as to why an organization may or may not embrace cloud computing.

The chapter also reviewed the concept of quality of service monitoring in the broader Information and Communication Technology sector, before narrowing down to the quality of service monitoring in cloud computing platforms. This resulted in review of various quality of service monitoring models in cloud computing platforms.

The concept of vendor neutrality in so far as quality of service monitoring is concerned was reviewed, with the current framework within which quality of service is monitored being derived. This chapter noted efforts by other researchers to solve the problem of vendor neutrality in cloud quality of service having identified the research gulf that exists.

CHAPTER THREE: METHODOLOGY

A researcher's adopted research methodology is anchored on a certain chosen research philosophy. Whereas there is convergence on the definitions of research methodology, it is difficult to define research philosophy with precision, and the attempt to do so forms an interesting and important part of philosophy itself (Stewart, Blocker & Petrick, 2013). A methodology as embraced by a researcher during a research process, refers to the researcher's own thinking and actions structured explicitly (Jayaratna, 1994). This understanding is echoed by Kothari (2004), who reasons that research methodology refers to the steps adopted by a researcher in solving the research problem and the logic behind the steps taken. This involves selection of certain steps over others, stating the criteria used in selection of those steps and the reason for use of that particular criteria. The thinking, the logic and the actions behind the research methodology, are guided by a research philosophy.

3.1 Research Philosophy

Philosophy has been defined as the use of a rational and reflective method in attempting to get at the most basic underlying principles of a phenomenon and to discover normative criteria (Stewart et al, 2013), while a research philosophy is what a researcher perceives or believes to be truth, reality and knowledge about a phenomenon under study (Gemma, 2018).

Research philosophy has four philosophical dimensions, namely, Ontology which deals with the nature of reality, Epistemology which handles the nature of knowledge and the relationship between the knower and that which would be known, Methodology which deals with the appropriate approach to systematic inquiry and Axiology dealing with the nature of ethics (Krishna, 2020).

The research philosophical dimensions determine the various research paradigms, with the paradigms being defined as set of basic beliefs (or metaphysics) that deal with ultimates or first principles (Lincoln, 1994). They are basic in the sense that they must be accepted simply on faith, however well argued, and there is no way to establish their ultimate truthfulness. They therefore represent the researcher's standpoint and worldview on how the phenomenon under study should be interpreted and understood.

The various paradigms under the Epistemology dimension are Positivism which deals with observable facts, Realism which embraces the fact that what senses see is the reality, Idealism believes that only the mind and its context exist, Interpretivitism which postulates that we interpret phenomenon based on meanings we give to them and Critical Theory which deals with research that challenges those conventional knowledge bases, assumptions, beliefs held by a social group (Saunders, Lewis & Thornhill, 2009).

The Ontology dimension has Objectivism which believes that Social entities exist in reality external to social actors concerned with their existence, Subjectivism which advances that Social phenomenon is created from the perceptions and consequent actions of those social actors concerned with heir existence and Pragmatism which advances the believe that one approach may be better than another in a given research and its possible to work with more than one approach.

The methodology dimension has Case study, Quantitative and Qualitative as the various paradigms, with the Axiology dimension taking into consideration ethics, which is the theory of morality, and aesthetics, the theory of taste and of beauty, as the paradigms.

This research embraces a blend of research paradigms, namely, positivism, which deals with observable facts as its research paradigm and is anchored on the Epistemology dimension of research philosophy, as well as the case study and quantitative research paradigms which are anchored on the methodology research dimension philosophy.

At the generic level of deciding on the methodology, one has to determine whether the research is quantitative or qualitative (Dawson, 2002). Qualitative involves getting an in-depth opinion from research participants through methods search as interviews, questionnaires and focus groups while quantitative research aims to generate statistics.

This research aimed to achieve four main objectives, namely: to develop a high level client trustable QoS monitoring framework for cloud computing systems, to design a vendor neutral model that implements the designed framework for SaaS cloud computing solutions, prototype and evaluate the new vendor neutral cloud performance monitoring tool and finally to develop algorithms for implementing the new vendor neutral cloud performance monitoring model. From the objectives, this research adopted a quantitative research approach.

Kumar (2005) states that qualitative research approaches are often based on deductive logic while quantitative research approaches are based on inductive logic. This research being quantitative in nature will have an inductive logic approach. This chapter highlights, the steps that were used to achieve the four research objectives, how the steps were arrived at and why the chosen steps.

3.2. Development of a Client Trustable Cloud QoS Monitoring Framework

Development of the framework was done through literature review of the existing cloud QoS monitoring framework. The strengths and limitations of the identified framework were analyzed. To aid in better understanding of the existing framework, existing cloud SaaS monitoring models developed from this framework were analyzed and their limitations documented.

Further, sample tools developed from existing cloud QoS monitoring models were highlighted, their applicability, strengths and weaknesses noted. From the literature review a conceptual framework that addresses the highlighted challenges and limitations was designed. The process involved in the development of the client trustable QoS monitoring framework for the cloud, as conceived by this research, is illustrated in Figure 30.

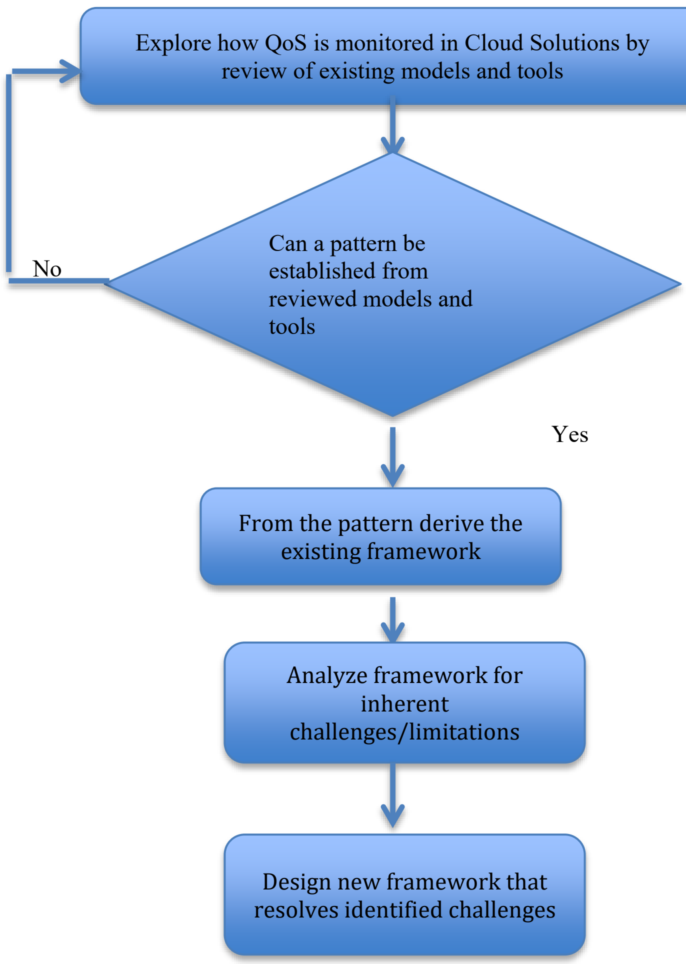


Figure 30: New QoS Monitoring Framework Development Process

3.3. Design of a Vendor Neutral Cloud QoS Monitoring Model

This phase involved both high level designing and low level design of the proposed vendor neutral Quality of Service monitoring model. The proposed model was developed from the proposed cloud QoS monitoring framework bearing in mind the challenges identified in the existing framework of cloud QoS monitoring solutions.

The new framework proposed a change in the location of the QoS monitoring tool from the provider's infrastructure (the cloud) to the user's computing device. This informed the design of the new model by analyzing the access methods in accessing SaaS cloud computing solutions.

Given the focus was on SaaS cloud computing solutions, it turned out the common access method was through the browser, which is situated in the user's computing device. To design the proposed QoS monitoring model based on the browser as an access method, an in depth analysis of the browser architecture and its sub components was done.

From the browser architecture and its subcomponents, it was discovered that a browser's functionality could be augmented through browser extensions. This discovery necessitated a thorough study of the architecture of browser extensions.

This led to a breakthrough on how to integrate a monitoring capability on the users' terminal, with a functionality of monitoring SaaS cloud computing solutions. The process involved in the design of the new SaaS QoS monitoring model, as conceived by this research, is depicted in Figure 31.

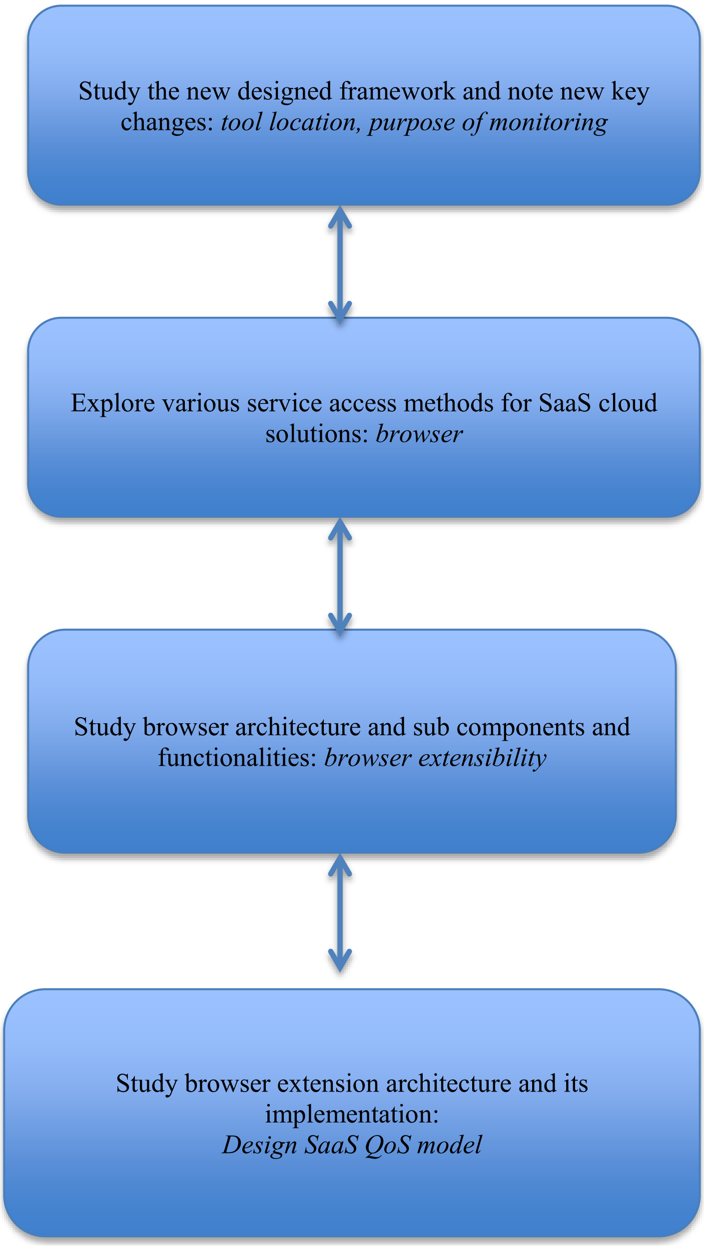


Figure 31: New SaaS Cloud QoS Monitoring Model Design Process

3.4. Implementation the Designed Vendor Neutral Model

This stage involved developing a SaaS cloud QoS monitoring tool that is based on the designed cloud QoS vendor neutral model. The preliminary steps were identification of the cloud QoS parameters to be monitored and exploring if they can be implemented on the new model.

The methodology used in development of the model was prototyping. The main motivation for use of prototyping is based on the fact that prototypes unlock cognitive association mechanisms related to visualization, prior experience, and interpersonal communication in ways that favour iterative learning between peers in the product development community (Berglund & Leifer, 2013).

According to Despa (2014), prototyping is an approach that progressed due to the necessity to outline requirements in a better way, it involves constructing a demonstration portion of the product that possesses the main functions. Early requirements are stated to provide only enough information to construct the prototype.

Further, the prototype helps to improve requirements since it acts as baseline for interaction between project team and project owner. The prototype is therefore not for developing into the final software system.

According to Sommerville (2011), a prototype is a first version of an application used to express ideas and to enable try out of design choices, and discover more on the problem and potential answers.

This methodology was chosen because the main goal of this study is to gather specifications as the model is implemented, to gather only sufficient functionality to enable development of critical model functionality and given that the model to be developed in not a final product but a prototype.

The development process was done through evolutionary prototyping. This was via of browser extension developed using JavaScript and Database browser for SQLite database. SQLite is a library that gets embedded inside the application that makes use of it. Database browser for SQLite was chosen because it is a light weighted database hence it can be easily used as an embedded software with devices like mobile phones as it only loads the required data as opposed to loading entire file, it is fast in terms of read write operations, and does not require installation on the computer on which it is being used.

JavaScript was chosen as the development language because by virtue of it being client-side it executes faster making it run instantly inside the client-side browser, it is a free technology and does not require one to go through any installation or configuration procedure and the fact that it is compatible with all modern browsers.

For purposes of this research the browser chosen was Google chrome. This is because Google chrome is noted as the most extensively used browser having the largest number of extensions that have been made for it (Sanchez-Rola, Satos & Balzarotti, 2017).

According to evaluations by Tamary and Feitelson (2015), using common benchmarks for evaluating browser technical performance, Chrome's rise to supremacy is coherent with technical supremacy over its rivals and with shrewd management of feature selection.

The general prototype development processes adopted by this research are as shown in the Figure 32.

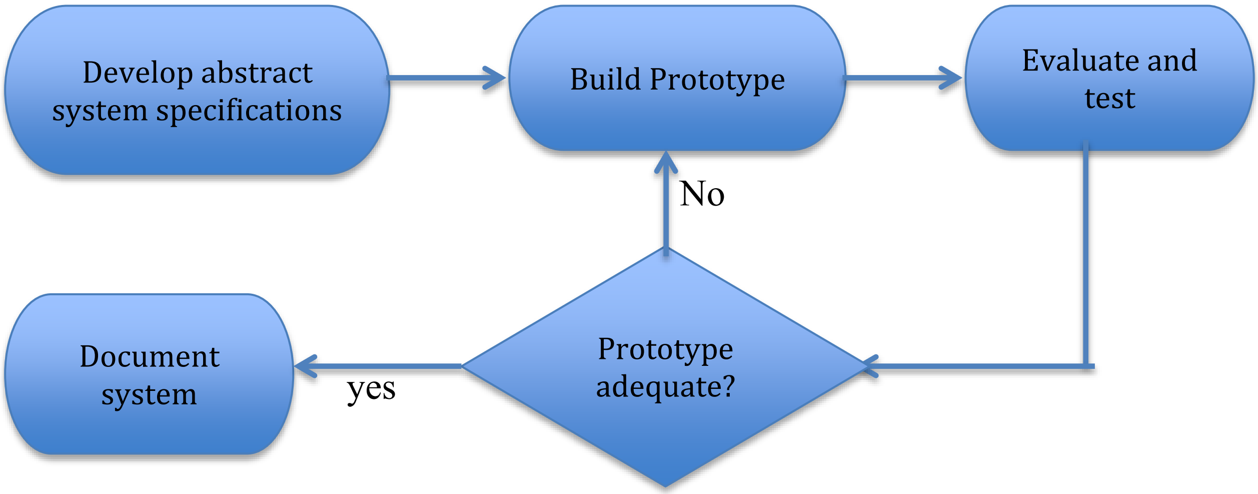


Figure 32: Evolutionary Prototyping

The overall research process adopted by this study is depicted in Figure 33.

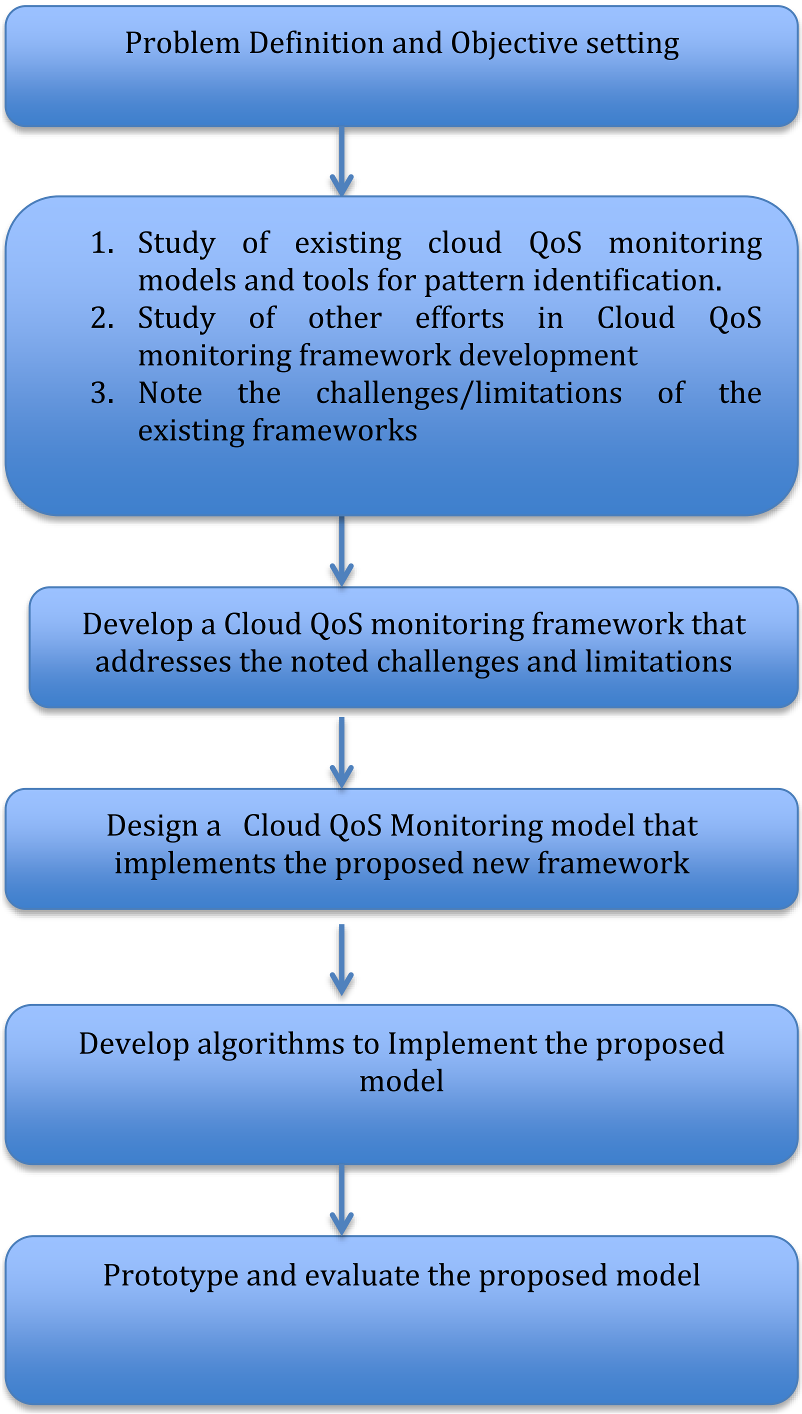


Figure 33: Research Process

3.5. Research Design

The arrangement of condition for collection and analysis of data in a research process has been termed as the research design (Kothari, 2004). It explains how the research will find answers to the research questions, and includes aspects of the research like the study design per se and the logistical arrangements that you propose to undertake, the measurement procedures, the sampling strategy, the frame of analysis and the timeframe (Kumar, 2011). The research design adopted in this study was the descriptive research design where variables are measured without influencing them. In this case, the Internet speeds, and the identified cloud QoS parameters were measured without being influenced by the researcher, since the aim was to find out how each provider is performing along the selected parameters.

3.5.1. Sampling Strategy

According to the TechValidate Survey Report on SaaS Application Trends and Challenges by Akamai (2016), there is a blend of horizontal and vertical applications implemented as SaaS. Of the horizontal applications 47% were service and support, 41% were business intelligence and Analytics, 31% collaboration, 29% for marketing and 24% were for sales.

As for the vertical applications developed 15% were for e-learning applications, 12% for Finance applications, and 10% for Human Resource applications. These statistics are depicted graphically in Figure 34.

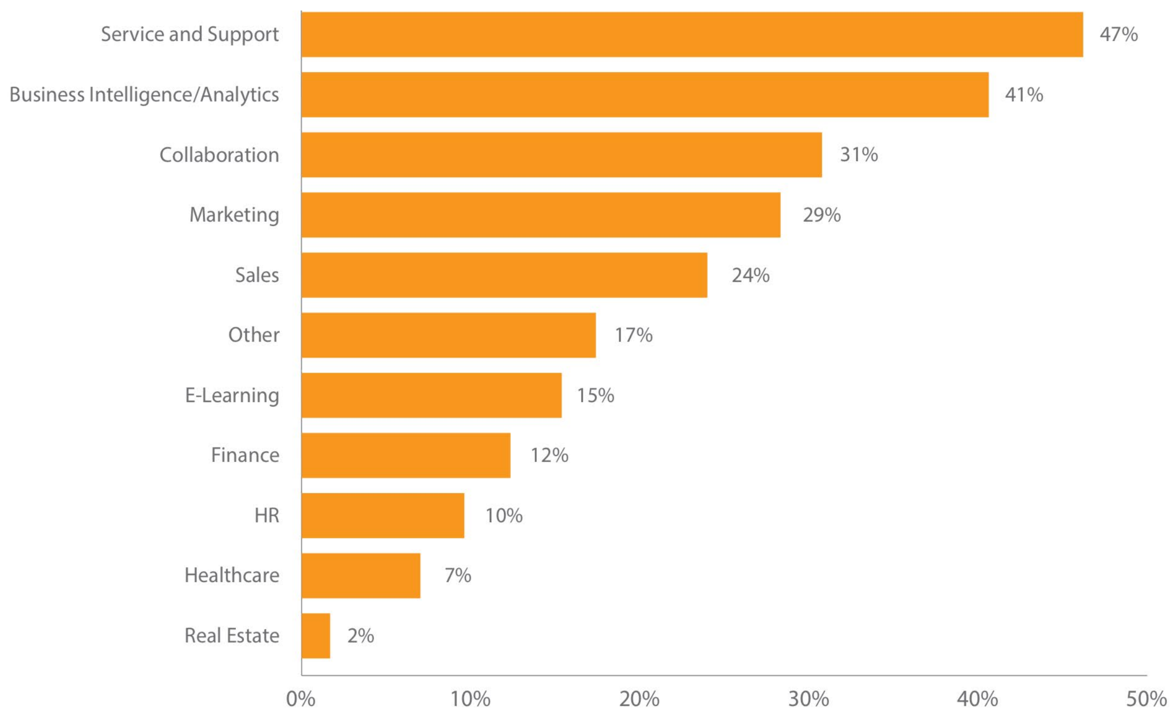


Figure 34: Categories of Services Implemented on SaaS by Percentage.

Source: Akamai (2016).

According to Kazmi (2018), Horizontal SaaS purpose to offer a broad service that can cover a broad scale of the market, in various different industries. It is a more mature model of SaaS having been around for long. Examples comprise of QuickBooks used in accounting, another is Salesforce as CRM service as well as HubSpot used for marketing services.

Kazmi (2018) further postulates that vertical SaaS fabricate software that is meant for use in a very specific industry, purpose-built for clear industry niches and being a recent trend it is not as mature as horizontal SaaS.

Examples include BioIQ an application for testing ones health, Health Assurance Plan an application that enables creation of membership plans for allows dental practices and Guidewire an application used by the insurance sector (Kazmi, 2018).

The other horizontal SaaS example is Office 365 cloud-based productivity tools offered on a subscription basis (Kaplunou 2020). According to Churakova and Mikhramova (2010), the various key providers per SaaS market segment are as shown in Table 5.

Table 5: Key SaaS Providers per Market Segment

SAAS MARKET SEGMENT	KEY PLAYERS
Content, Communication and Collaboration (CCC)	Cisco WebEx, SumTotal, IBM Lotus
Customer Relationship Management (CRM)	Salesforce, Oracle, RightNow
Enterprise Resource Management (ERM)	SAP, NetSuite, Workday
Supply Chain Management (SCM)	Descartes, Ariba, Ketera
Office Suits	Google, Zoho
Digital Content Creation (DCC)	Youtube, Adobe

Source: Churakova and Mikhramova (2010)

From Table 5, and based on the most common application of SaaS from Figure 34, the market segments that were considered for this research are Customer Relationship Management and Office Suites, where the key providers chosen for testing were Salesforce and Google respectively.

Oracle and RightNow were not chosen because they do not provide a platform for trials but only provide an opportunity for demos to be carried out for potential buyers. This would have been difficult for this research to monitor the QoS of the platforms for a prolonged period of time.

Further, from the analysis of Horizontal and Vertical applications, this research focused on horizontal SaaS applications, because they cut across different industries and thus testing results based on them can be used for generalization of SaaS performance.

The most commonly implemented SaaS, namely service and support, Business Intelligence and Analytics, Collaboration, and Sales & Marketing were considered. From examples provided by Kazmi (2018), this research focused on Salesforce, Hubspot Office 365, Google Office suites and Shopify which offers a platform for setting up an online shop with full marketing and CRM features.

The choice of the cloud providers was also made using judgmental sampling where the length of the trial period, presence of free software and the tasks that can be done during the trial period were considered. The trial was performed using solutions from four global SaaS service providers who are Salesforce, Hubspot, Google docs from Google and Shopify.

The leading applications from the two main market segments for SaaS, namely, Customer Relationship Management (CRM) and Office suites were selected, while for performance comparison in the same market segment e.g. office suites, the top two leading SaaS providers, Microsoft and Google were selected. The ranking information was from Datanyze (2021).

The logic for testing with the leading player per market segment is that this would be established firms, with their platforms having matured and therefore the QoS is also expected to be to the satisfaction of the users. Hubspot was chosen because it offers a free Customer Relationship Management System for small enterprises, albeit with limited features. The tasks executed on Hubspot were configuring customers on the Hubspot CRM, setting commodities for selling and configuring prices.

Shopify was chosen because it offers a free platform for setting up an online shop for small enterprises. The tasks executed on Shopify were setting up an online store, designing products and setting their prices and executing sales after generating invoices.

Salesforce.com was chosen because it has a 30 days trial version. The tasks executed were product configuration, price quoting, billing and basic customer relationship management tasks.

Google docs was chosen because the applications provided are part of open, web enabled office software set provided by Google. The tasks executed were opening, using, closing and re opening Ms Excel , Ms word, and Ms Power Point applications. Microsoft office 365 was used for comparison purposes with Google docs.

3.5.2. Study Design

As pointed out by Kumar (2011), a research design should not be confused with a study design, emphasized as study design per se, and involves the set up of the data collection conditions, when to collect the data, how long or how often to collect the data, what data to collect, and whether the researcher should vary the conditions of the set up, whereas the research design also includes other parts which constitute the research process.

This research sought to compare performance of various cloud service providers, using quantitative data. In view of the quantitative nature of required data, the research used cross sectional study design. Cross sectional study design are useful in obtaining an overall picture as it stands at the time of the study (Kumar, 2011). For comparing the performance of two providers offering similar services, like in the case where Microsoft's Ms office was compared with Google docs, the study design used was comparative case study design.

Dimensions of QoS utilized during the testing were response time from the application, availability of the application and application stability. This is because the identified dimensions are the basic QoS metrics in any SLA between a cloud provider and their clients.

The time taken to respond by the service was measured as the time that lapses from the time a user clicks the open button to the time the application is fully open and ready for use. From this value, average response time was also computed.

The Service availability was computed by counting the number of instances when a user clicks an open button and the application returns an error instead of opening the application. Stability was computed from variations in service response time.

Timings were done and recorded during the execution of the tasks to ascertain how long the application takes to initialize and counts were done to note how many times during the execution period is the application not available.

The testing process involved execution of tasks throughout the day, to emulate normal user activities on the cloud, for a period provided by the trial period offer of the cloud service provider. The results were then stored in form of reports per each instance the test was conducted and an average for the trial period computed. The testing period was from 14th September to 27th October 2020, with thirty (30) runs on each of the platforms, amounting to using the platform each working day of the testing duration.

Throughout the testing process, factors that affect the upload and download speeds, namely Internet speed, Internet service provider and the specifications of the computer were kept constant. The only factor left out was the location of servers, which is based on the Content Distribution Network provider used by the cloud service provider.

According to Arie (2021), a CDN, which is a Content Delivery Network, refers to a worldwide network of distributed web servers also called Points of Presence (PoP), aimed at providing faster delivery of content. The content is duplicated and made available in the entire CDN so that users have access to the data stored at a locale closer to them. Benefits of using CDN include reduction in costs of bandwidth, page load times improvement, as well as increasing the global availability of the content.

In this research therefore, the Internet speed and the specifications of the testing process platform were constant in all test and are therefore considered as the independent variables. The QoS values obtained for the various cloud providers' platforms would therefore vary with the Internet speed and are therefore considered dependent variables.

The location of the Content Distribution Network servers used by the different cloud providers is an aspect that affects the QoS values measured. This is based on the providers' choice of the cloud provider on whom to contract for the Content Distribution Network services. This aspect was not part of the test but it affects the values measured and is therefore considered as an extraneous variable.

3.5.3. Testing Procedure

The actual tasks performed on the chosen SaaS cloud providers during the testing process for the various cloud computing platforms involved usage of the platform in a way an ordinary user would use the platform.

For Hubspot, the tasks performed were opening the platform and monitoring how long it took for the platform to be fully initialized, configuring customers and measuring the time used to update the details and configuring products while measuring the time used for updating and loading of the respective tasks.

On the Shopify platform, the tasked involved opening the platform and monitoring the duration taken to fully initialize the platform, creating an online shop and monitoring the duration it took to update the details, configuring products and prices and monitoring the time taken to update details and generating invoices while updating sales, monitoring the time it took for the updates to be effected.

The Salesforce tasks involved opening the website monitoring how long it took before the sales application was fully initialized, configuring product details, monitoring how long it took for the various product details to be captured by the system e.g. time taken for image upload, time taken to save captured details and monitoring the time it took to generate invoice against a given number of orders.

The Google docs testing tasks involved opening the apps website while monitoring the time it took for the apps to be fully initialized, opening the specific online app namely, word, Excel or power point and monitoring the time it took for the app to be fully initialized and ready for use, after using the app, the contents were saved and time it took for the contents to be saved and the app ready for use was monitored as well.

The Office 365 platform testing tasks involved opening the apps website while monitoring the time it took for the apps to be fully initialized, opening the specific online app, namely word, Excel or power point and monitoring the time it took for the app to be fully initialized and ready for use, and after using the app, the contents were saved and the time it took to for the contents to be saved and the app ready for use monitored.

For all the identified tasks, the QoS monitoring tool was capturing the Internet speeds on the user terminal, from the time the user submits data, to the time control is returned to the user for action, and the time taken for the user requests to be responded to and control handed back to the user.

3.6. Verification and Validation Methodology

Within the context of modeling scientific knowledge, verification refers to internal consistency, whereas validation refers to justification of knowledge claims (Barlas & Carpenter, 1990).

The proposed framework was verified by checking for compliance with the general principles of an Information System framework, namely, global consistency to ensure one coherent framework so that every concept is linked to every other one in a specific, well-established way; generality to ensure that it is specialisable and extensible in certain situations, to cater for the various specialized subfields; and finally simple and as straightforward as possible for easy understanding.

Validation of scientific knowledge can take two approaches, the logical empiricist validation, which is a strictly formal, algorithmic, reductionist, and confrontational process, where new knowledge is either true or false; and the relativist validation, which is a semiformal and communicative process, where validation is seen as a gradual process of building confidence in the usefulness of the new knowledge with respect to a purpose (Kjartan, Jan, Reid, Janet & Farookh, 2000).

The proposed framework was validated using the relativist approach, which according to Kjartan et all (2000), is appropriate for open problems, where new knowledge is associated with heuristics and non-precise representations.

The framework was therefore validated on whether it builds confidence in its usefulness with regards to the purpose of cloud user centric QoS monitoring, and whether it provides design solutions correctly (effectiveness) and whether the designed solutions can be realized with less cost and time (efficiency).

The framework was found to build confidence due to its user centricity nature, and the models from it could be realizable effectively through browser extensibility and in an efficient manner due to open source web technology development tools.

According to Kung and Zhu (2008), Software verification and validation are quality guarantee actions in the software development process whose aim to guarantee that the application is made in accordance with a development process that satisfies the user's desires.

The major attributes of software quality are usability, reliability, testability, efficiency, transportability, and maintainability (Adrion, Branstad & Cherniavsky, 1982). The verification and validation process for the model and the monitoring tool derived from the model took place throughout the development processes.

The main phases in the development were requirements phase, design phase and implementation phase. The verification and validation at the requirements stage was geared towards errors discovery in the specification of the requirements and the models used for analysis. The techniques used were reviews of requirements, code inspection, structured walkthrough, and prototyping.

The verification and validation at design phase involved assessing the level of correctness, consistency level, and adequacy of the design with regards to the models of requirements and analysis. This involved review, code inspection, structured walkthrough, formal verification, and use of prototyping techniques.

The activities performed here included, checking for right use of design language, adequacy of the design, non-redundancy, logical consistency and definition-use consistency.

During the implementation phase, verification and validation was done to confirm that the source code implements the right functionality, real time and security constraints, properly handles exceptional instances, satisfies performance. The static verification methods used were code review, inspection, walkthrough and desk checking while testing was used as a dynamic validation method.

The quality of the tool developed was evaluated using the McCall's model, developed by McCall, Richards and Walters (1977). According to the Software Quality Metrics Methodology Standard, by IEEE (2009), software quality is the extent to which an application has a desirable combination of quality traits.

Software quality may also be stated as meeting openly specified functional and performance constraints, openly acknowledged standards for development and implied features that are accepted from all expertly created software (Suman & Wadhwa, 2014).

The McCall model was adopted because based on a comparison study done by Al-Badareen, Selamat, Jabar, Din, and Turaev (2011), the McCall model scored higher than the other models, namely the Boehm, ISO, Dromey and FURPS.

According to the McCall's model, the factors to be considered when evaluating software quality are: Correctness, Flexibility, Integrity, Reliability, Usability Efficiency, Maintainability, Portability, Interoperability, Testability and Reusability. The McCall's model defined the identified metrics as shown in Table 6.

Table 6: Definitions of Software Quality Factors

No	Quality Factor	Definition
1.	Correctness	Is the degree to which the application fulfills its requirements and meets the clients' objectives.
2.	Reliability	The level to which the application performs its designated functionality with desired accuracy.
3.	Efficiency	The quantity of resources as well as instructions needed by an application to execute a task.
4.	Integrity	To what extent can access to the application or data be controlled.
5.	Portability	Effort needed to transfer an application from execution platform to another.
6.	Reusability	The level to which a program can be re-used in other applications.
7.	Interoperability	Effort needed to combine one application with another.
8.	Usability	Effort needed to use an application.
9.	Maintainability	Effort needed to fix errors
10.	Testability	Effort needed to test an application satisfactorily
11.	Flexibility	Effort needed to modify an application as desired.

For purposes of this research, the quality factors considered were: level of correctness, reliability level, efficiency, integrity, ease of use, ease of maintaining, ease of testing and flexibility level.

Correctness was evaluated by comparing the expected output from the developed specifications and the actual output from the system; reliability was evaluated by comparing the variations in the output from the system under similar conditions, namely internet speed at the user end; efficiency was evaluated by verifying the amount of code required to perform a certain function in the system and exploring if there are ways to reduce the code and achieve the same functionality (McCall et al, 1977).

The integrity of the system was evaluated based on whether the data from the system can be accessed and modified externally; usability was evaluated by establishing the average time required for one to learn how to use the system; maintainability was evaluated based on the inline explanations provided in the code on what the code does to enable one locate errors and fix them easily while flexibility was evaluated by the extent of in line documentation that can enable one to understand what the system does and thus modify it in case there is need to (McCall et al, 1977).

3.6.1. Case Study Validation

The overall validation of the developed tool was performed through a case study approach. A case study is a pragmatic probe that explores a contemporary occurrence in its actual-life setting (Yin, 1984). According to Vissak (2010), case studies do not essentially have to depend on prior literature or prior experimental evidence.

It is on the basis of the strength of case studies as sighted by Vissak (2010), coupled by the fact that this is a new research area with no previous empirical data for comparison that a case study approach was used for validation.

The results from the designed and implemented model were compared with those from the vendor's tool with regards to the same parameters measured by the new tool. To validate the results obtained from the vendor agnostic tool and compare its comparison from other cloud QoS tools, a case study was conducted using existing tools on Gsuite, Salesforce, Hubspot, Shopify and Microsoft. The method for conducting the case studies was through testing using cloud service owner's platform.

The methodology used for testing involved creating a new account in the cloud service owner's platform and thereafter using the services in a manner that a conventional user would use the cloud services. In instances where difficulties were encountered or clarifications required during usage of system, video calls, online chats, and emails were handy in getting aid from the cloud service providers. Sample conversations with the sales and technical teams of the cloud providers are shown in appendix 1.

On Gsuite, the procedure consisted of opening forms, sheets, Google docs, and slides. The running apps were put into use in a manner that an ordinary user would initiate and make use of the apps, close and re-open them.

Salesforce was used by making an account on the platform, configuring commodities for sales, setting prices, giving clients quotations and giving feedback to questions from buyers.

Hubspot usage consisted of creating a new account on the cloud service owner's platform and inputting customer details in the Hubspot Customer Relationship Management system, inputting commodities for marketing and fixing their prices.

Shopify usage consisted of involved configuring a new account on the platform, creating an online store, inputting products and setting their prices and as well as executing sales and finally generating invoices.

3.7. Chapter Summary

This chapter elaborated the research approach and actions taken during the research for attainment of the solutions to the research objectives. The chapter mapped the researcher's actions and thinking to the existing research philosophies and therefore anchoring the approaches on positivism, case study and quantitative research paradigms, which are, based on Epistemology and Methodology research philosophies, respectively.

From the research philosophy, the methodology used for developing the client trustable quality of service monitoring in the cloud was highlighted, as well as the research design for the development of the vendor neutral quality of service monitoring model.

The chapter also highlighted how the model was implemented, how the quality of service parameters were chosen, how the testing platforms were selected, how the testing was designed and how the vendor neutral model was validated and all aspects of the study design per se.

CHAPTER FOUR: REALIZATION OF THE VENDOR NEUTRAL MODEL

Design in itself being a problem solving activity, is a matter of trial and error, and therefore, there should be no confusion between the product of the design process and the process itself (Vlient, 2007). Whereas during the demonstration of a mathematical proof, successive steps dovetail perfectly into each other and everything fits into place at the end, the real discovery of the proof was possibly quite different (Vlient, 2007).

The outcome of the design progression is therefore a logical reconstruction of the design process, with the design process being an imaginative one, and the quality and expertise of the designers as well being a key determinant for its success.

This research used two research design techniques, namely, descriptive research design and case study. The descriptive research design was used in conceptualization of a user centric cloud QoS monitoring framework, based on the shortcoming of the existing provider centric framework, and comparison of the vendor neutral QoS monitoring results with those from the cloud provider's integrated QoS monitoring tool. It was also used in conceptualization of a vendor neutral SaaS cloud QoS monitoring model based on the proposed cloud QoS monitoring architecture, testing of the vendor neutral tool on select global SaaS cloud providers.

The case study research design was used for comparison of results from select global SaaS cloud providers monitoring tools with the results from the vendor neutral tool developed by this research.

4.1 Formulation of A Client Trustable Cloud QoS Monitoring Framework

Based on the existing framework, depicted in Figure 13, it is noted that the existing cloud-monitoring framework relies solely on the cloud provider's perspective of QoS monitoring. The monitoring is done by the cloud provider for the purpose of Service Level Agreement management, cloud resources provisioning and billing.

The existing framework empowers the cloud provider, while the cloud user is left to rely on the information from the cloud provider. To empower the cloud user as well, the new framework proposes cloud monitoring with the user perspective considered, and the location of the cloud QoS monitoring tool to enable end to end QoS monitoring. Further, the aim of QoS monitoring shifts from cloud providers centric goals to user centric goals, like cloud provider pre-selection comparison and cloud provider QoS report validation.

The three aspects included in the new framework, namely, the tool location at the cloud customer's end for end node -to-end node QoS monitoring, the ability to compare different clouds performance prior to selection of the cloud provider that meets desired cloud user goals, and the ability to authenticate the QoS report from the cloud solution provider's monitoring tool, are aspects meant to empower the cloud service user.

The proposed Client Trustable Framework, by this research, is illustrated in Figure 35.

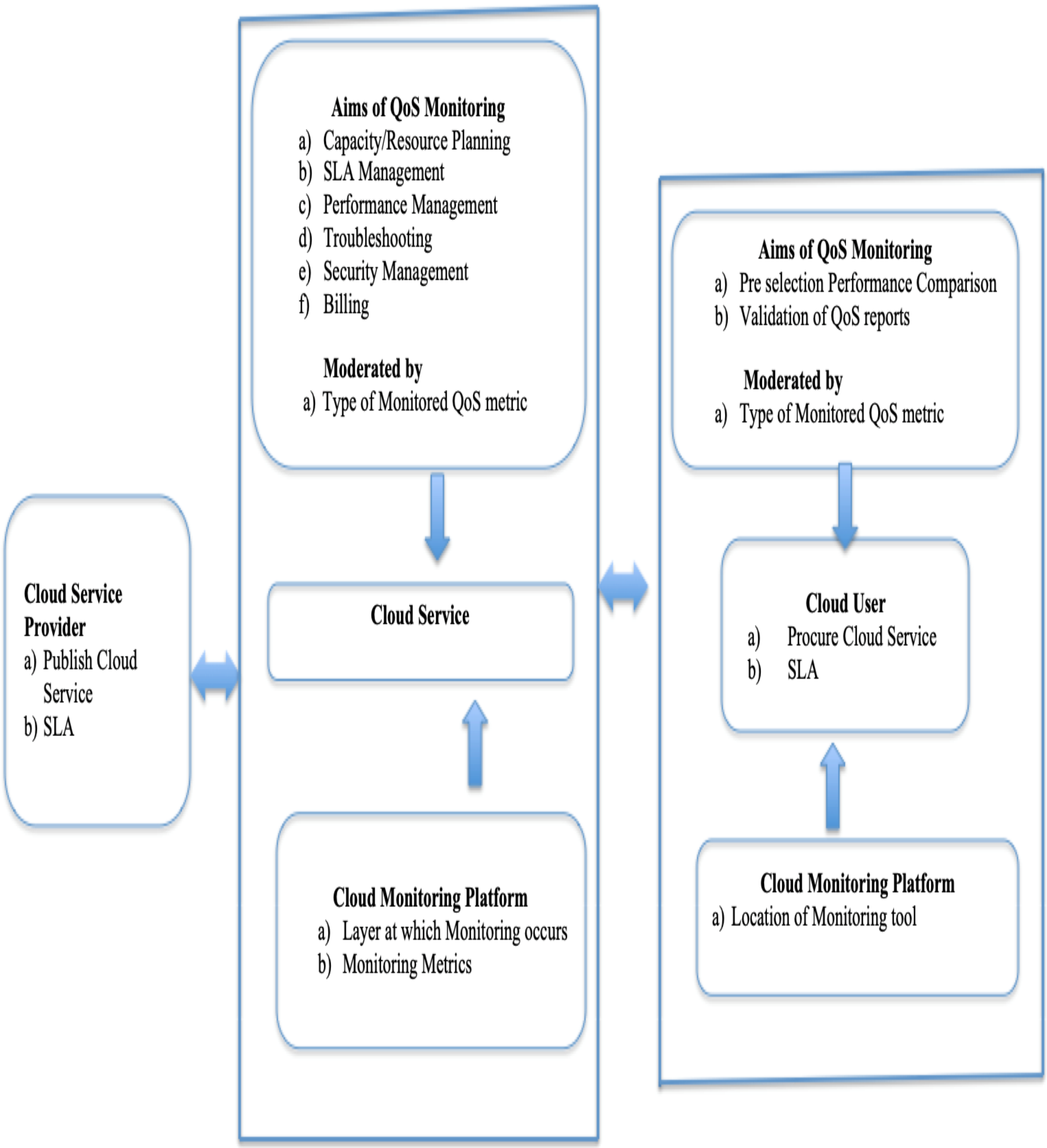


Figure 35: Proposed Client Trustable Cloud QoS Monitoring Framework.

4.2. Appraisal of Trust in the New Framework

From the reviewed literature, this research focused on directly experienced trust approach, since the user and the provider have no prior encounters that would form the basis for any trust.

To enable quantitative comparison of trust among the global cloud service providers, this research used an End User Centric Quantitative Trust Model in Cloud Computing (Makokha, Chepken & Opiyo, 2021).

Using the results from the vendor agnostic QoS Monitoring solution for the cloud, and applying the most widely used confidence interval of 95% proposed by Hazra (2017), on the results from the vendor neutral tool, and comparing them with the results from Google and Microsoft QoS tools, a quantitative value was realized based on how close or far the results are from each other. The comparison was also enhanced by the user experience during usage of the services.

4.3. Formulation of the Proposed Vendor Agnostic SaaS Cloud QoS Monitoring Model

According to Makokha, Opiyo and Okello-odongo (2017), the contemporary models for QoS monitoring in the cloud currently in use are the Quality of Service MONitoring as a Service Model (QoSMONaaS), CloudQual, Adaptive QoS-driven Monitoring Model and the Agent Based Model. All the existing models are linked to the physical platform of the cloud solution provider, and therefore a QoS measuring solution derived from all the listed models cannot be used across multiple cloud vendors. This implies that the major draw back of cloud monitoring tools is portability.

Cloud solution models can be grouped into three main categories: PaaS (Platform as a Service), IaaS (Infrastructure as a Service), plus SaaS (Software as a Service) (Gorelek, 2013). According to Kumar and Goudar (2012), these models can be presented using an architectural diagram in the Figure 36.

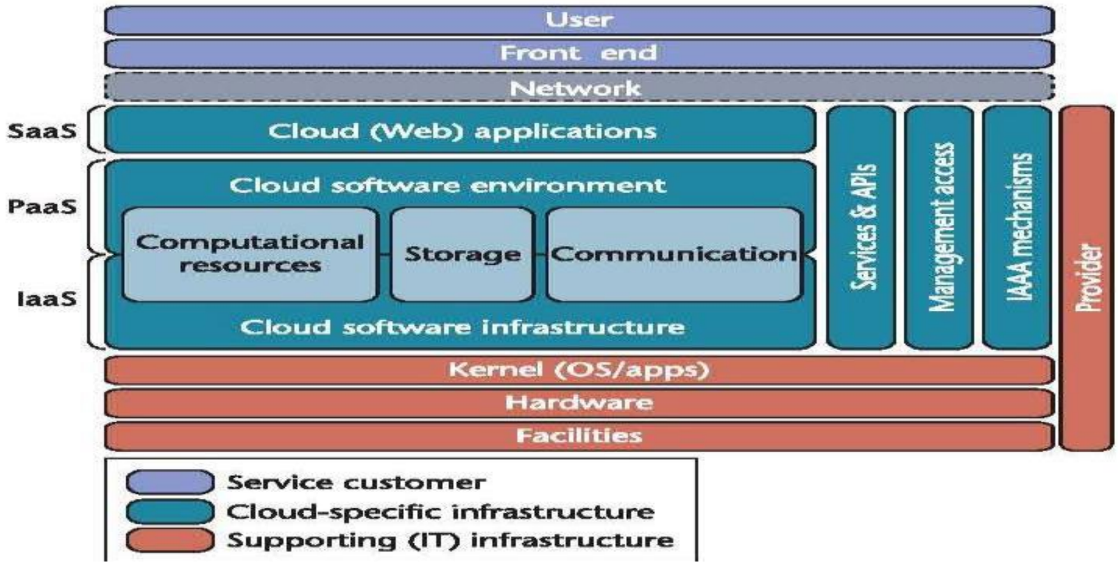


Figure 36: Cloud Reference Architecture

Source: Kumar and Goudar (2012).

From the analysis done by Makokha and Opiyo (2018), the existing architecture depicted from a bird's eye view design, of contemporary cloud QoS monitoring models is as illustrated in Figure 37.

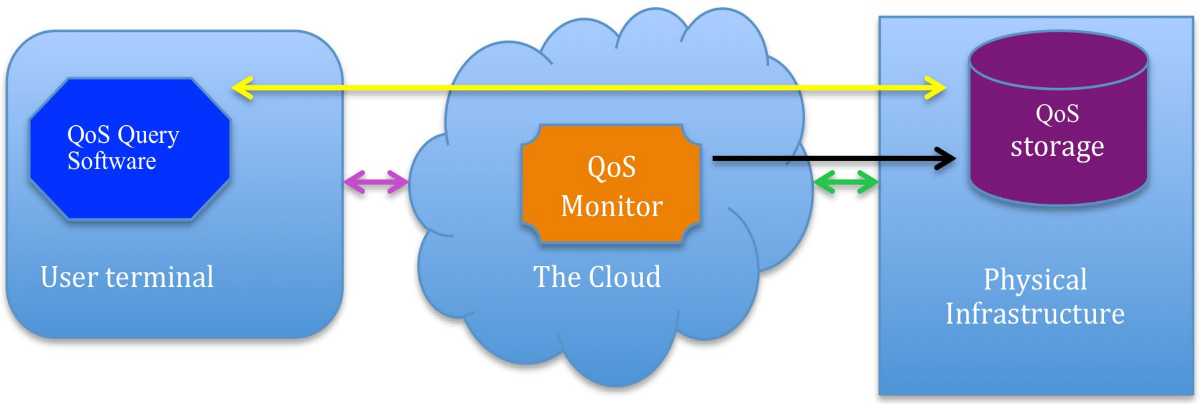


Figure 37 : A Bird's Eye View of QoS Monitoring using Existing Models

Source: Makokha and Opiyo (2018)

From Figure 37, the developed tools for cloud QoS monitoring are situated within the cloud, where they monitor the QoS as perceived by the client and keep the QoS values in the provider's platform for subsequent retrieval by the cloud client. This indeed portends the likelihood of vendor bias in view of the fact that the cloud service owner as well as the owner of the tool for measuring the cloud QoS is the same firm, and further, the cloud service provider stores the QoS values in their platform, prior the client querying the values. In an environment in which the Service Level Agreements (SLA) is rigorous, issues of trust around the monitored QoS will arise.

In addition, based on the illustration 37, it is evident that the solution is tightly coupled with the physical platform of the cloud facility from where it is executing. This indicates that the QoS tool is not portable to any other dissimilar cloud provider's infrastructure and therefore in a scenario where the user of the service would like to equate the QoS measures of various cloud providers of similar services, it would not be feasible to use the same tool.

To abolish potential instances of cloud owner bias, there is need for designing a model which is not linked to the infrastructure of any particular cloud service provider. Moreover, the QoS measurements as monitored has to be transmitted in real time to the cloud service user with no requirement for prior storage on the cloud provider's infrastructure.

Using the new proposed framework, shown in Figure 35, the possible architecture designed at high level for the solution to the challenge of non portability of cloud QoS Monitoring tools due to vendor tied models, as visualized by Makokha and Opiyo (2018) is depicted in Figure 38.

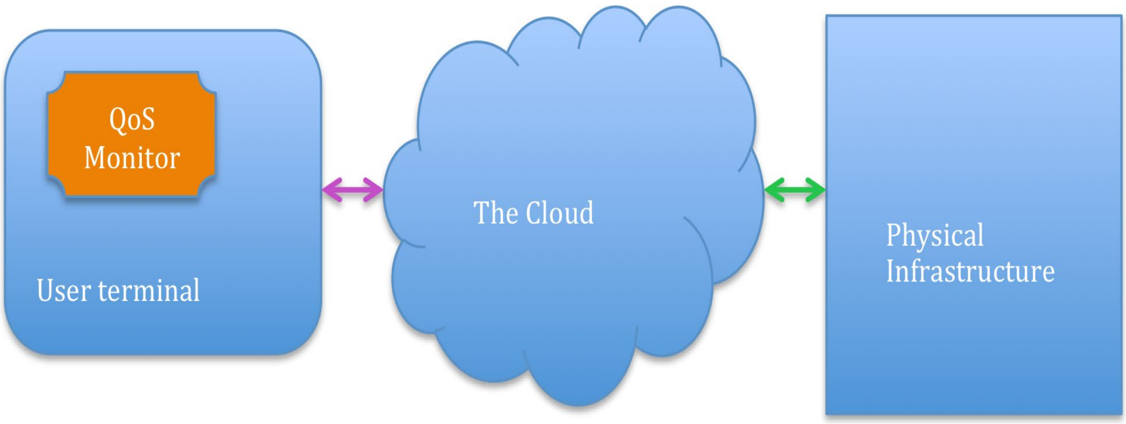


Figure 38: High Level Architecture of Cloud QoS Monitoring using a Vendor Agnostic Model

Source: Makokha and Opiyo (2018)

From Figure 38, the QoS Monitor is located on the terminal of the user and monitors the cloud service as the user interacts with the cloud. The results are stored on the terminal which is being used by the user and thus no querying is needed. The tool also measures end of service node to end of service node QoS. Given the tool is located on the user's terminal, it is not tightly coupled to the architecture of any cloud provider. This makes the tool vendor neutral and thus usable across all cloud providers.

The solution to the puzzle on how to realize this architecture rests on the methods used to access the cloud services. It is noted that the three identified cloud service models, namely, IaaS, PaaS, and SaaS, are accessible by the cloud users through two methods, which are by use of a cloud owner specific software running on the user's service access device and by use of a browser for accessing the web (Ashraf, 2014).

The named access techniques to the stated cloud configuration models are illustrated in Figure 39.

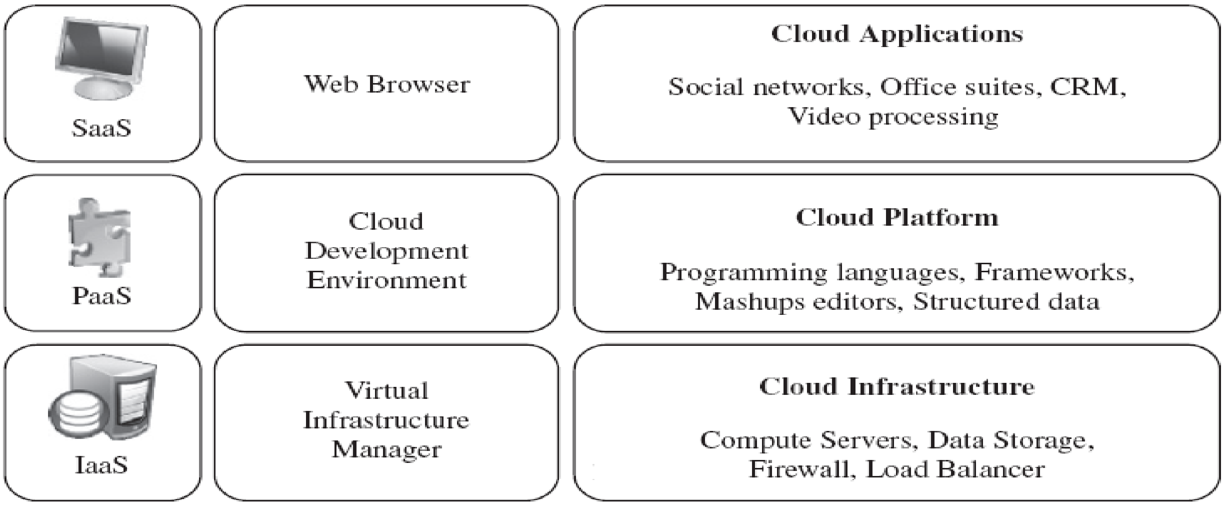


Figure 39: Cloud Services Access Models

Source: Ashraf (2014)

A thorough review of this usage techniques depicts that usage of the cloud service owner specific application approach is also reliant on the owner of cloud service and is therefore not vendor agnostic. This leaves only one access method deemed to be vendor agnostic which is access by browser method. According to Buyya, Broberg and Goscinski (2011), cloud services that are offered by SaaS providers are accessible to users via portals on the web.

The web browser access method opens prospects for designing a vendor agnostic model that can be used for measuring cloud solutions' QoS. Actualizing this requires a deeper analysis of the blueprint of different web browsers for a thorough comprehension of the different architectural components that encompass the browser, which will guide on third party tool integration in the browser for purposes of extending browser functionality to contain cloud QoS measurement.

4.3.1. The Web Browsers Architecture

According to Junghoon, Seungbong and Sangjin (2011), a browser for accessing the web is an indispensible application required to be used for Internet access. A web browser is an application that reads as well as fetches documents from local sites and sites around the world through the Internet (Vetter, Spell & Ward, 1994).

Grosskurth and Godfrey (2005) define a web browser as an application that gets data from the World Wide Web stored in distant storage servers then presents it in the browser window on the user's screen or passes the data to an external specialized application for opening the particular document.

Taking cognizant of these definitions, this research defines a web browser as user application with a graphical user interface from where the user interacts with Internet content by indicating the location of the content using the content address.

A generic high level design depicting a browser for the web is as illustrated as shown in Figure 40.

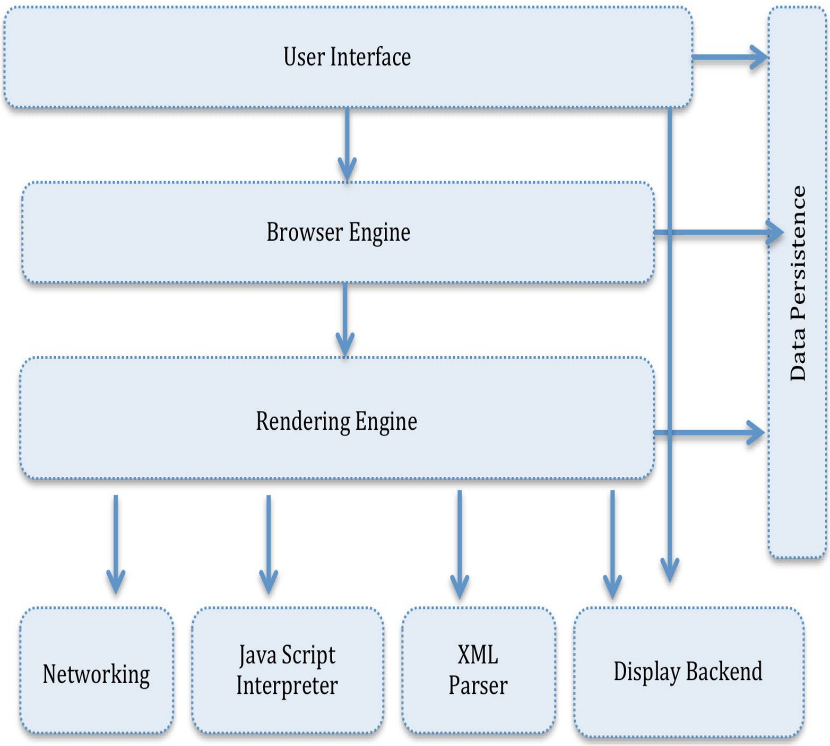


Figure 40: High Level Architecture of the Web Browser

Source: Grosskurth et al (2005)

4.3.2. Web Browser Sub Components

According to Grosskurth et al (2005), each of the sub components listed in Figure 40 plays a critical role in the browser.

The User Interface subsystem resides amid the Browser Engine and user. It has features like visually showing page-loading progress, printing, toolbars, smart handling of downloads, and preferences. It is sometimes amalgamated in the desktop environment for communication with other desktop seawares and browser session management (Grosskurth et al, 2005).

The Browser Engine subsystem is a module that is embeddable enabling it to provide the Rendering Engine with a high level interface. It loads the user provided URI and enables basic browsing functionalities like back, forward and reload features. It has hooks used for observing different aspects in the browsing session like the status load progress of the current page and alerts from JavaScript. It as well enables querying and processing settings of the Rendering Engine (Grosskurth et al, 2005).

The Rendering Engine subsystem brings forth a visual representation of the provided URI. It has capability of showing XML and HTML documents that have been by option designed using Cascading Style Sheets (CSS), in addition to insert content like imageries. Further, it is also in charge of page layout and it may also contain reflow algorithms that are responsible for incrementally adjusting the location of elements on the page. The HTML parser is also contained in this subsystem (Grosskurth et al, 2005).

The Networking subsystem executes protocols for transferring files like HTTP and FTP. It converts from one character set to another, as well as resolving media types like MIME for files. Also included is a cache of recently retrieved resources (Grosskurth et al, 2005).

The JavaScript Interpreter is responsible for evaluating JavaScript, known as ECMAScript code that is sometimes embedded in web pages. This scripting language was developed by Netscape. Certain functionality of JavaScript like popup windows opening, can be disabled for security purposes by the Engine of the Browser or the Rendering Engine (Grosskurth et al, 2005).

The other subsystem, XML Parser, analyses files that are in the XML format to a tree like structure called the Document Object Model (DOM). It is among the subsystems that are most reusable in the blueprint. Realizations of most browsers make use of already in existence XML Parser, instead of rewriting theirs from scratch (Grosskurth et al, 2005).

The Display Backend subsystem offers windowing and drawing primitives, a suite of interface widgets for the user, including a set of fonts. It is sometimes tightly coupled to the user device Operating System (Grosskurth et al, 2005).

The last subsystem, Data Persistence, collects for storage different data sets related to the browsing session on disk including data that is high level in nature such as bookmarks or toolbar locations and security certificates, cookies and cache data which is lower level in nature (Grosskurth et al, 2005).

4.3.3. Browser Extensibility

According to Lerner (2011), an extensible platform is one that allows future amendments to the formerly devised base system, which could be in form of new additions, new improvements upon, or substitutions of current functions. Contemporary browsers possess three techniques of enhancing these functionalities, namely via plugins, or extensions, as well widgets.

In computer science, a plug-in , also called add-in or addin or plugin or extension or add-on / addon) is an application segment which enhances a particular capability of a currently in use software (Jain, 2015). A detailed differentiation of the terms extension, plug-in, add-on and widget and patch, based on the literature reviewed by this research is depicted in Table 7.

Table 7: Noted Distinctions Between Plug-in, Widget, Add-on and Extension

No	Term	Description	Examples	Key feature
1.	Plug in	Application designed to process and display content that a web browser is not by default designed to process and display (non HTML content)	a. Adobe acrobat b. QuickTime Player c. Real player d. Winamp e. Java	Works in background Not visible to user
2.	Widget	Drag and drop Content blocks that enhance site layout and functionality mostly used to display dynamic content, such as feeds of recent blog posts, comments, search boxes and blog posts archives, as well as the frontend display of plugins that have been activated. It is implemented as a plug in	a. A “purchase now” icon on smart phones, Weather, maps, clock b. A calendar c. Search bar d. Social media sharing button	Visible to user, drag and drop
3.	Extension	An application meant to increase the functions a browser can perform	a. DownThemAll for Firefox b. Firebug for Firefox c. Google Voice extension for Chrome d. Let there be Comic Sans for Safari	Browser specific
4.	Add on	Generic term for Extension, Plug in and Widget		
5.	Patch	A segment of an application designed to update another application or its supporting data and operating system, for purposes of fixing or improving it		

Source: Jain (2015)

An architectural diagram, containing a provision for an add-on, as visualized by Vrbanec, Kirić and Varga (2013) is as shown in Figure 41.

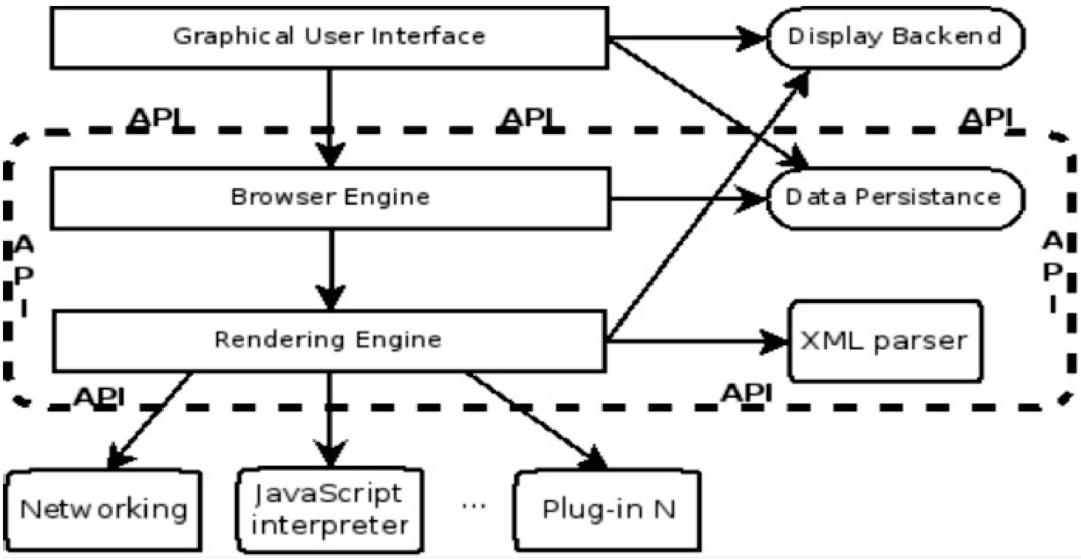


Figure 41: Browser Architecture with Add-on Sub Structure.

Source: Vrbanec et al (2013)

The Generic arrangement on the interfacing between an add-on and a program already in use is a shown in Figure 42.

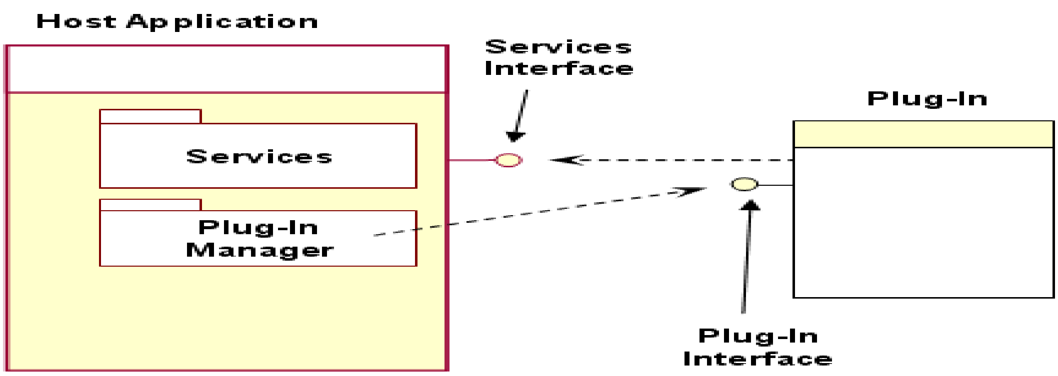


Figure 42: Generic Interfacing of an add on to a host application

Source: Jain (2015)

Based on the explanations provided in Table 7, a vendor agnostic model to be used for monitoring cloud QoS is at best conceptualized and actualized as a software extension. This is because the monitoring functionality will be incorporated in the browser and it will monitor any cloud service accessed by that browser.

4.3.4. The Architecture of a Browser Extension

The basic blueprint of an extension of web browser, as conceived by Barth, Porter Felt, Saxena and Boodman (2010) is as illustrated in Figure 43.

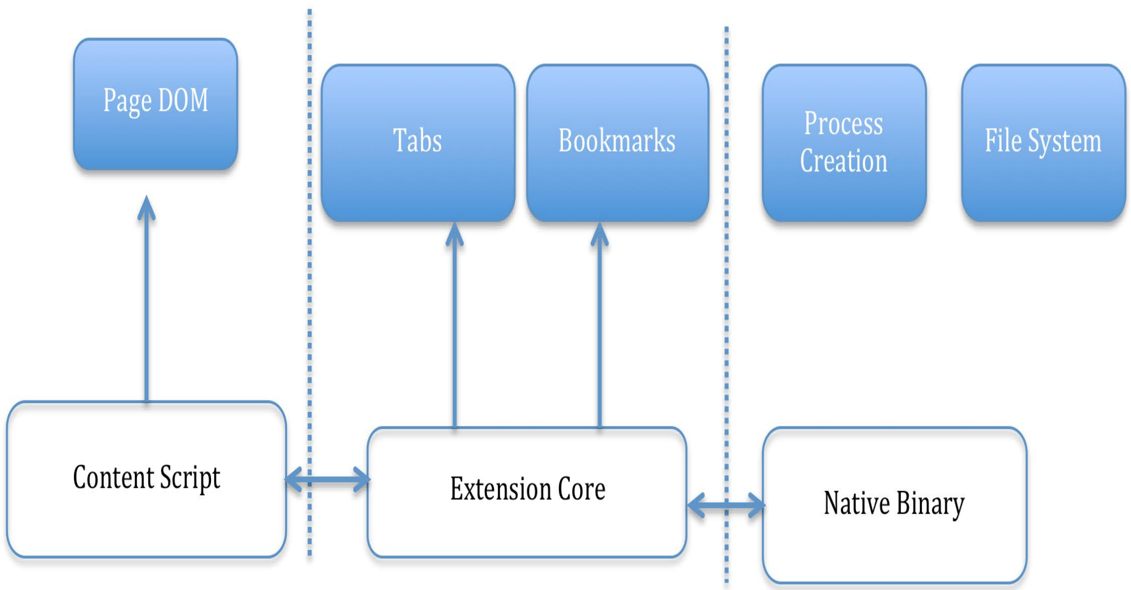


Figure 43: Sub Structures of a Browser Extension

Source: Barth et al (2010)

Content scripts, as depicted in Figure 43, are constrained to relating with content from the untrusted Web only and thus run without privileges; while the Extension core is in charge of implementing features specific to the extension like modification of browser User Interface (UI), relating with resources at system level via Chrome's Application Programming Interface (API) extension and in this regard runs using full privileges of the extension; with the native binary code interacting with the computer of the host.

According to Liu, Zhang, Yan and Chen (2012), a browser example, Chrome, segregates privileges amongst various components of an extension. Case in point, web contents can directly interact with content script of an extension. However, naturally it lacks the authorizations to enable it access browser modules, save for the fact that it can interconnet via postMessage to the core of the extension.

Despite the core of extension having the most allocated privileges, it is protected from pages on the web. It therefore has to rely on content scripts as well as use XMLHttpRequest for communicating with the content on the web. This native binary of an extension, while running as an NPAPI plugin, contains the most privileges to enable it execute any arbitrary code as well as to access any files (Liu et al, 2012).

The privilege segregation phenomenon with a multi component blueprint was presented in contemporary browsers to mitigate the challenges of security in old age browsers that were monolithic, and therefore whose extension code as well as the code that was linking to Web page content were executing as a unified heap of JavaScript (Liu et al, 2012).

4.4. Proposed Vendor Neutral Cloud QoS Monitoring Model

To attain development of a vendor agnostic model which can be used for monitoring cloud QoS, the desired model has to be realized as a software extension, that would be anchored to a precise browser. Figure 44 depicts a high level blueprint, as visualized by this research, of this proposed model.

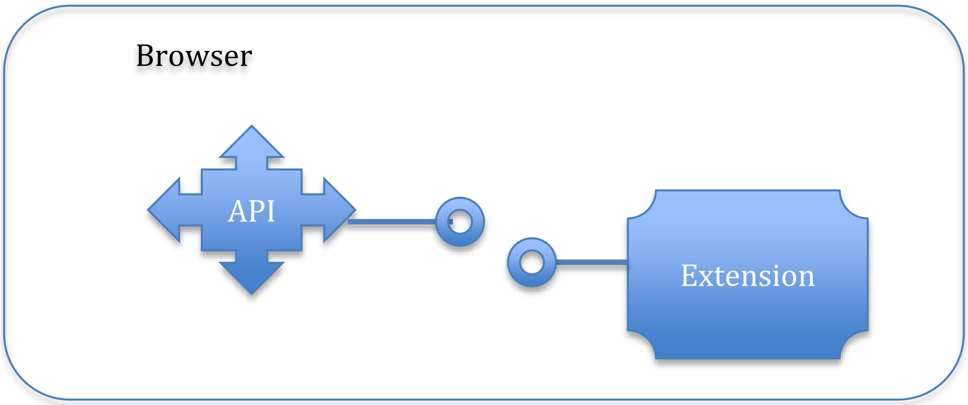


Figure 44: A High Level Blueprint for the Proposed Vendor Agnostic Model

From Figure 44, the API is provided by the developer of the browser. They come built-in with the browsers and allow developers to perform complex operations without dealing with the sophisticated lower-level code. In this case it adds the QoS monitoring functionality to the browser.

The Extension component contains the modules that comprise the functions to monitor various metrics of the cloud QoS monitoring. The functions are linked to the browser via the browser's inbuilt APIs.

An expanded viewpoint of the sub component of the Extension, as designed by this research, is as illustrated in Figure 45.

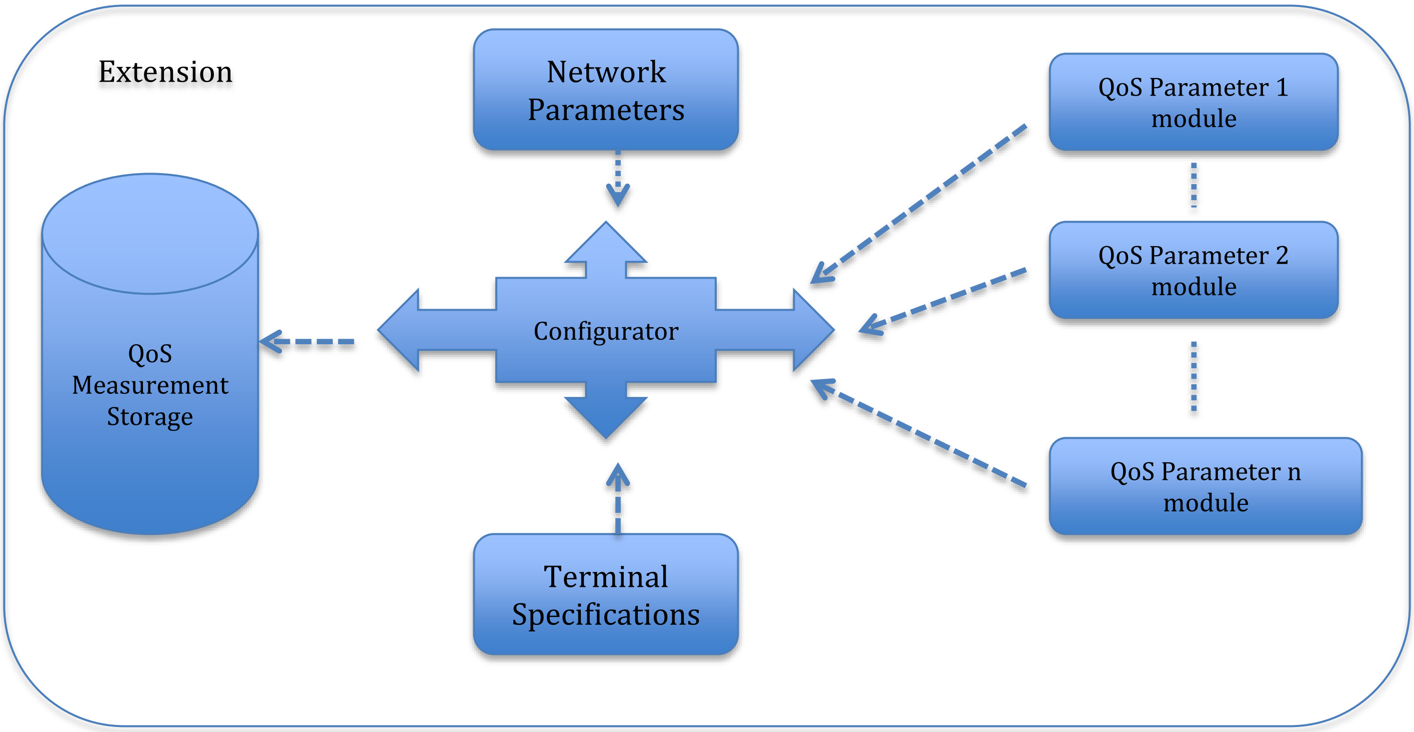


Figure 45 : A Zoomed-in Diagram of the Proposed Model for Cloud QoS Monitoring

From Figure 45, all configurations of the various parameters to be monitored are set in the configurator. The terminal specifications sub component is used to obtain the parameters of the system (computer) from where the extension is running like RAM memory capacity and the speed of CPU.

The Network parameters module measures the user's Internet speeds and any other network parameters at the monitoring time. The significance of this is that in scenarios where the QoS derived from the cloud is impacted by the end user device that monitored the QoS values. The QoS parameter component measures the particular parameter it is programmed to measure and keeps the values in the module for reporting.

The QoS parameter Module contains functions that would measure the particular QoS metric that has been coded, for example the service response time, availability and stability in the case of this research. The monitored QoS metrics values are then stored in the Storage module containing a database linked to the browser.

4.5. Actualization of the Proposed Vendor Neutral Model for Cloud QoS

The suggested vendor agnostic model was realized as a browser extension on the Google chrome browsers. Designing and realization tools made use of comprised of regular technologies use in web development like CSS, JavaScript, HTML, SQLite database and Node JS.

4.5.1. Algorithms Development

The algorithms for the cloud QoS extension were derived from the developed prototype after an iterative process that ensured the developed prototype achieved its intended purpose. The QoS model has three main algorithms, namely, the algorithm for recording terminal specifications, the algorithm for monitoring Internet connections, and the algorithm for monitoring the time taken to accomplish various tasks as configured in the configurator.

The terminal specification-monitoring algorithm collects the details of the terminal on which the QoS extension has been installed. Algorithm 1 details the steps involved.

Algorithm 1: Terminal Specification Collecting Algorithm

```
START
    On Extension Installation
        Create and Assign Client_ID
            Get Client_ID details as
                cpu_numberOfCore
                cpu_archName
                cpu_modelName
                ram_size
                date_joined
        Create SQLite Database Table
            Log Client_ID
            Log Client_ID    details
STOP
```

A sample JavaScript code snippet for the implementation is shown in appendix 2.

The Internet monitoring algorithm monitors the network connection parameters and log in the database. Algorithm 2 indicates the algorithm details.

Algorithm 2: Internet Connection Parameters Collection Algorithm

---

```
START
    While the monitoring status is turned on
            Create operation_ID

            Check supplied url

            Check internet connection status
                    If  Internet Connection is up
                        Get and log connection parameters as:
                            Round trip time
                            Downlink
                            EffectiveType
                        Loop until monitoring status is turned off
                        Compute average of the connection parameters collected
                    Else
                        Report no Internet Connection
                    End if
     End While
STOP
```

A sample JavaScript code snippet for the implementation is shown in appendix 3.

The cloud QoS monitoring algorithm monitors the specific QoS parameter and logs the metrics in the database. Algorithm 3 indicates the details.

Algorithm 3: QoS Monitoring Algorithm

```
START
        While url is valid and internet connection is ON
              While url is loading
                    Log the start of loading time and end of   loading time
              End while
              On complete of url loading
                    Listen to user mouse and button events
                          On user event executed:
                          Log the start of user event and time of completion of user event
                    At end of user events
                            Compute:
                                      Average service response time as service response time.
                                      Compute variations in service response time using
                                      standard deviation for service stability determination.
                                      Compute Service Availability using recorded system
                                      outage instances due to inordinate response times.
            End While
STOP
```

A sample JavaScript code snippet for the implementation is shown in appendix 4.

4.5.2. Tool Integration into the Browser

The installation of the vendor agnostic QoS monitoring tool for the cloud into the chrome browser is triggered by typing " *chrome://extensions* " on the Google chrome web browser:

Upon hitting the enter button, an option for either packaging an extension or loading extension that is not packaged was provided. Noting that this is a trial session extension one was required to choose loading an extension that is not packed and then precede to the location of the package one wished to be load.

A screenshot from the interface shown at the time of integrating the developed tool for QoS into chrome browser is illustrated by Figure 46.

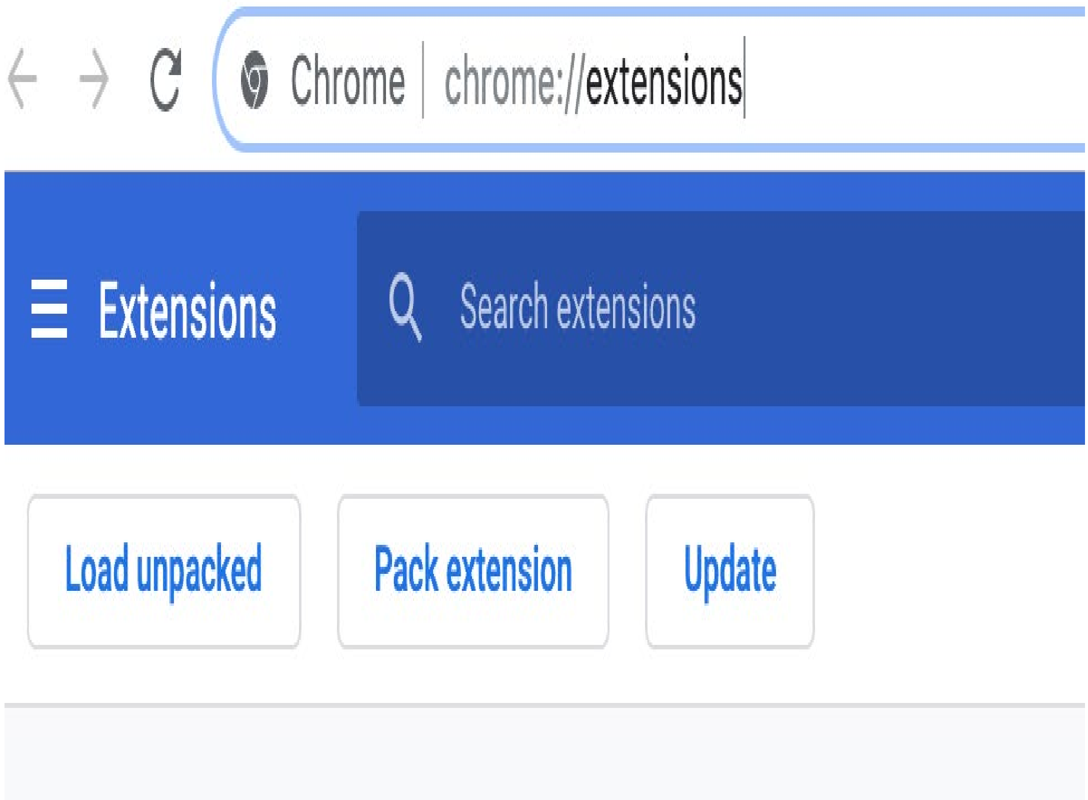


Figure 46: Integration Interface for the QoS tool into Chrome Browser

Upon integrating the tool, it appeared alongside other extensions previously installed on Google chrome as illustrated in Figure 47.

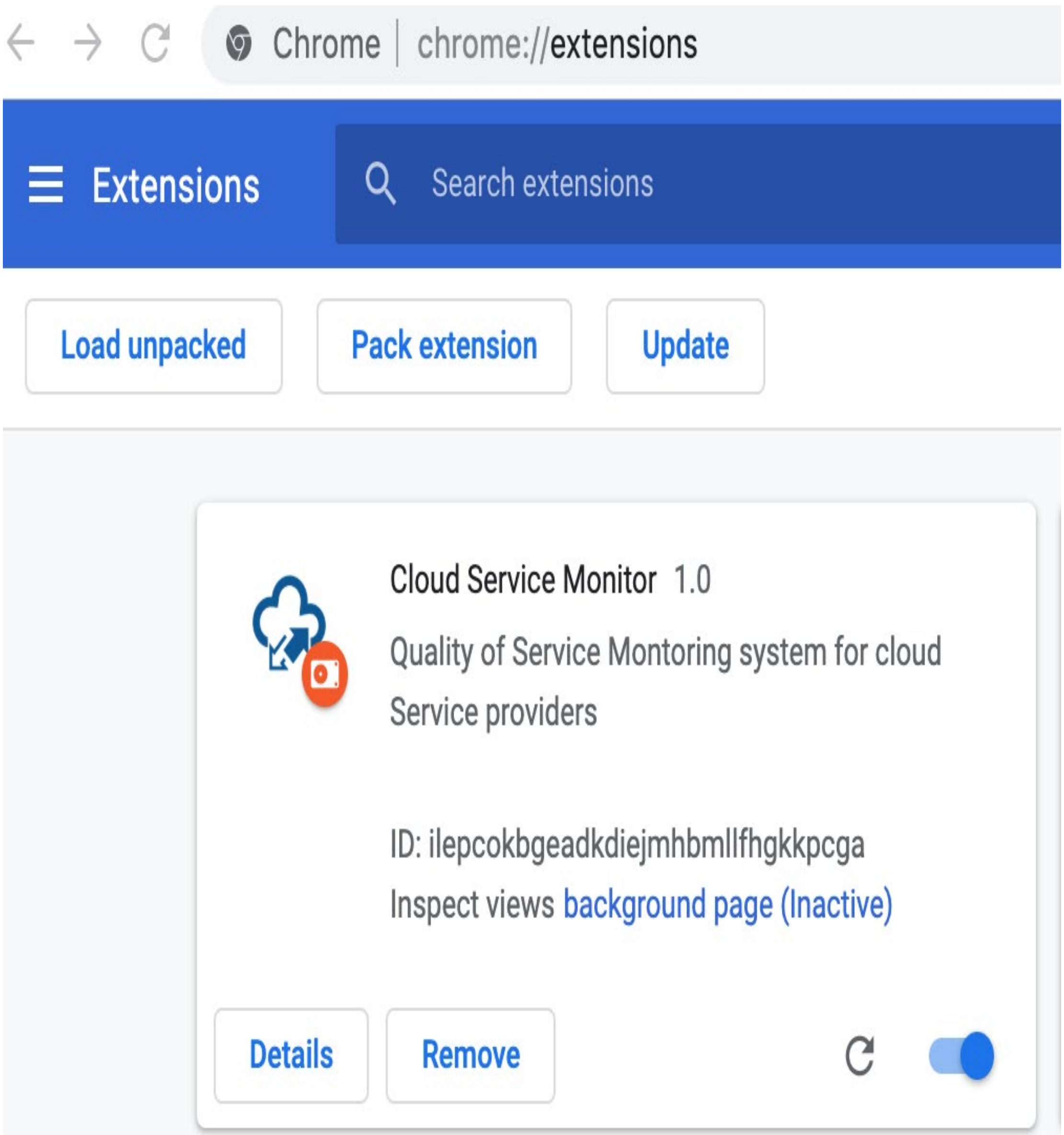


Figure 47: Cloud QoS Service Monitor as Integrated in Google Chrome

After having successfully installed, the service monitor was executed using the computer terminal by using the command written as : *npm run dev* as illustrated by Figure 48.

```
Abus-MacBook-Pro:qos-webapplication abuhamza$ npm run dev

> qosmonitor@1.0.0 dev /Users/abuhamza/Desktop/QoSaPP/qos-webapplication
> nodemon bin/www

[nodemon] 1.19.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node bin/www`
QoS Application running on port 8484
```

Figure 48: Starting the QoS Monitoring Tool

4.5.3. Testing with the Vendor Agnostic Cloud QoS Tool

Based on Figure 48, it was noted that the monitoring tool was executing on port 8484 as local host accessible from the browser. On opening the application, it appears as illustrated in Figure 49.

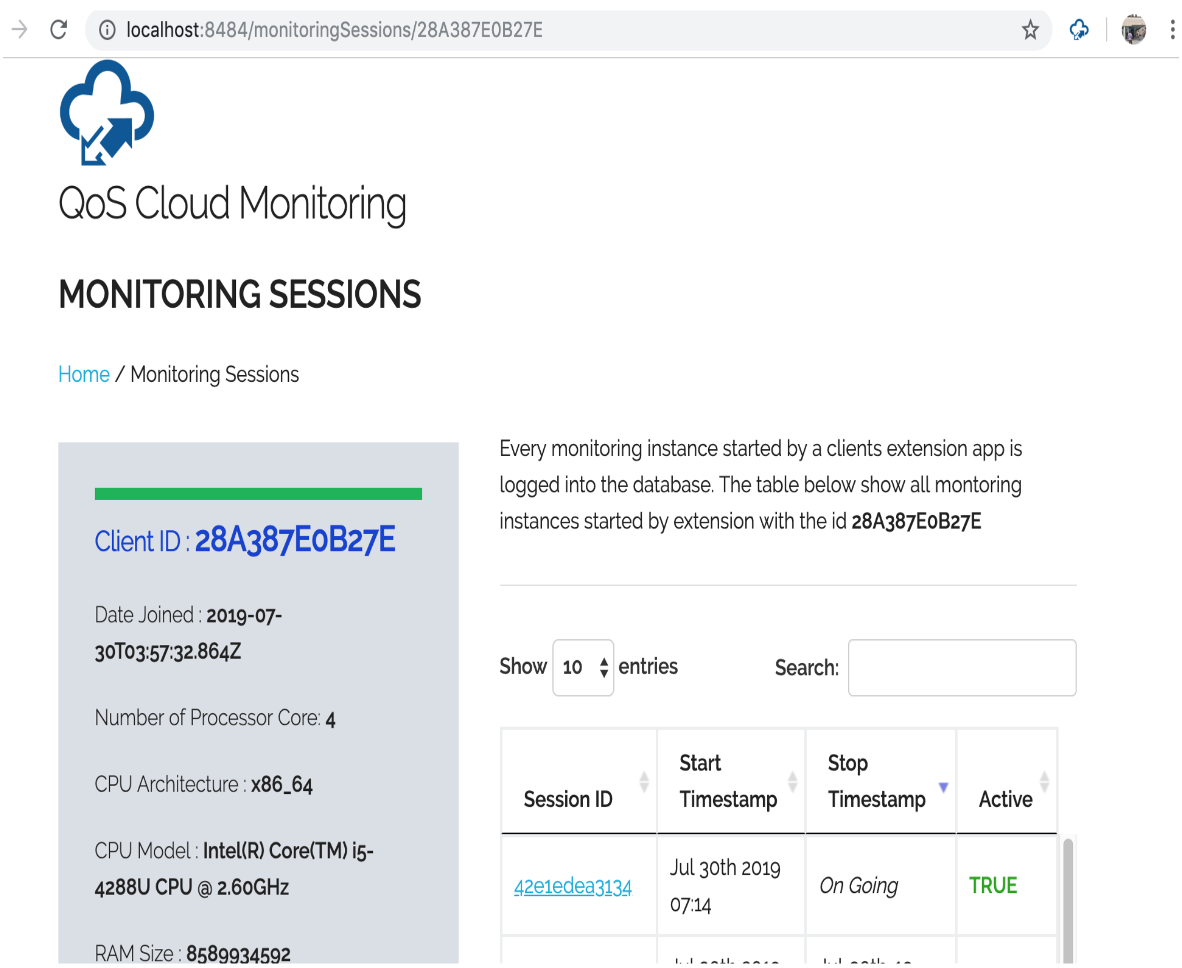


Figure 49: Active QoS Monitoring Platform on the Browser

Once the platform is executed to run, configurations were made for any cloud service sites that required to be monitored. The tool measured and stored the results for set parameters for QoS in the database created automatically at the time of tool installation.

4.6. Chapter Summary

This chapter detailed the research journey used in conceptualizing, visualizing and realizing the client trustable cloud quality of service monitoring framework, the vendor neutral cloud quality of service monitoring model, the associated quality of service monitoring tool derived from the vendor neutral model and the algorithms for realizing the quality of service monitoring tool. A thorough review of the architecture of web browsers was presented and an explanation on how an extension can be embedded into a web browser. The chapter, as well, highlighted the technique used in quantitatively evaluating trust in the proposed cloud quality of service monitoring framework.

CHAPTER FIVE: RESULTS AND DISCUSSION

The testing set up consisted of a test platform and testing conditions that were maintained constant for all tests and therefore were considered as independent variables. The QoS metrics obtained from the tests were then considered to be dependent variables, which are also affected by an extraneous variable, which is the location of servers of the Content Distribution Network (CDN) providers procured by the cloud service providers. The testing was conducted using the same end user terminal, which was a laptop made by Apple with specifications of Intel(R) Core(TM) i5-4288U CPU, of 2.60GHz speed categorized as a MacBook pro and the same Internet conditions, namely, an average of Internet effective type 3G from the same Internet service provider.

One of the key principles of cloud computing, as pointed out by Buyya, Brobger and Goscinski (2011) is trust, and according to the said authors, the most critical issue to address before cloud computing can become the preferred computing paradigm is that of establishing trust, and therefore mechanisms to build and maintain trust between cloud computing consumers and cloud computing providers, as well as among cloud computing providers themselves, are essential for the success of any cloud computing offering. The proposed QoS monitoring model by this research therefore provides a platform to realize this principle.

With regards to QoS monitoring, the main principle of QoS monitoring as postulated by the European Telecommunications Standards Institute (ETSI) is use of parameters. The selected parameters may be used for various purposes like specifying the level of quality of service in customer telecommunication service contracts or in the description of terms and conditions of the service.

Further the selected parameters may as well be used in comparing the quality of service of different service providers, comparing the quality of service aspects of different service offers and preparing long term studies on the quality of service aspects of a specific service. This study has used parameters for specifying the levels of QoS and for comparison of QoS of different cloud providers.

ETSI also mentions Data Collection Period as a principle of QoS measurement, with recommendations being that for measurements to be used for long term comparisons, it is recommended that QoS data should be collected and calculated on a quarterly basis starting on 1 January, 1 April, 1 July and 1 October, while for shorter periods being advisable for QoS aspects where frequent and fast changes in quality are likely to occur.

Noting the nature of SaaS cloud services, this study used shorter periods in view of the fact that frequent changes are likely to occur in the course of usage of SaaS services.

5.1 Cloud QoS Monitoring with the Vendor Neutral Model Tool

With regards to the tests performed on Google docs, Salesforce, Shopify and Hubspot cloud solution, under same platform and Internet conditions as independent variables, the average results are as illustrated by Table 8.The number of tests done was thirty (30) runs for all the cloud platforms, with the testing having been done between 14th September 2020 to 27th October 2020, amounting to usage of each platform once per working day of the duration of the test. Sample raw results from which the averages were computed are shown in appendix 5.

Table 8: Cloud Providers QoS Monitoring Results

Number	Provider of the Cloud	Response Time	Availability	Stability
1.	Salesforce	2.93 sec	100%	0.252 sec (stable)
2.	Google (docs)	4.83 sec	100%	1.654 sec (stable)
3.	Hub Spot	2.45 sec	100%	1.574 sec (stable)
4.	Shopify	2.59 sec	100%	1.3 sec (stable)

Based on Table 8, response time of the service refers to the mean time taken from the time the user requested for a service until the time the service was initialized and ready to be used. Service availability was taken as the sum total of instances when the user demanded for a service and got the service compared to the sum total of instances the demanded service was not available. Stability refers to variations in the service response time, computed using standard deviation from the average response time.

The test results for Salesforce show that on average, during the entire test duration, for all the tasks executed on its platform, the time taken for any of the requested service to be initialized and ready for use by the user was 2.93 seconds.

As for Google, the results show that on average, during the entire test duration, for all tasks executed on its platform, the time taken for any requested service to be initialized and ready to for use by the user was 4.83 seconds.

For Hubspot, the results show that on average, during the entire test duration, for all tasks executed on its platform, the time taken for any requested service to be initialized and ready to for use by the user was 2.45 seconds.

With regards to Shopify, the results show that on average, during the entire test duration, for all tasks executed on its platform, the time taken for any requested service to be initialized and ready to for use by the user was 2.59 seconds.

For service availability testing purposes, and to prevent a situation where the service may take too long to load, the maximum load time was set to ten (10) seconds. Any service response beyond ten (10) seconds the service was tagged as unavailable.

According to Munyaradzi, Maxmillan and Mutembedza (2016), the average website load time must be eight (8) seconds in order to increase stakeholder satisfaction and thus be perceived to be within desired Quality of service.

Further, according to tests done by Sukhpuneet, Kulwant and Parminder (2016), using the Site Speed Checker, on the performance of identified websites showed a maximum load time of 10.82 seconds.

Nielson (2007) advocates for a maximum waiting time of 10 seconds since this is the limit time to keep the attention of the user to keep focusing on the dialogue. Longer wait delays make users want to execute other tasks while awaiting for the computer to conclude, so that requires that they be given feedback indications on when the computer expects to be finish.

Having considered the three stated time frames, and considering that cloud computing is not just about website content but specialized services, this research settled on 10 seconds to factor in the specialized nature of the website content to be loaded.

Results from Table 8 show that during the entire testing period, all the cloud platforms were available for use, indicated as an availability of 100%. This implies that at any time during the test duration that user wished to use the service, it was available to the user.

The stability of the Service was calculated using the standard deviation metric. Getting a standard deviation value larger compared to the mean implies the platform is not stable as compared to a standard deviation value found to be lower than the mean which means the platform is stable.

From Table 8, Salesforce stability evaluated to 0.252 seconds, which is below the average service response time of 2.93 seconds, and therefore the platform was considered to be stable in so far as service response times are concerned.

The Google platform stability evaluated to 1.654 seconds, which is below the average service response time of 4.83 seconds, and therefore the platform was considered to be stable in so far as service response times are concerned

The Hubsport stability evaluated to 1.574 seconds, which is below the average service response time of 2.45 seconds, and therefore the platform was considered to be stable in so far as service response times are concerned.

Likewise, for Shopify, the stability evaluated to 1.3 seconds, which is below the average service response time of 2.59 seconds, and therefore the platform was considered to be stable in so far as service response times are concerned.

5.2. Results from Existing Cloud Computing Platform Integrated Tools

The QoS monitoring results from cloud provider's integrated QoS monitoring tools for select cloud service providers obtained during the same time as when the vendor agnostic cloud QoS monitoring tools, are detailed herein.

5.2.1. Gsuite

Gsuite is Software as a Service (SaaS) solution that amalgamates all the cloud-based productivity and collaboration solutions established by Google used by enterprises, institutions, and nonprofits firms. Alongside each subscription one gets access to customized Gmail addresses, Sheets, Docs, Calendar, Slides and Drive, Sites.

G-suite provides its users with a dashboard that contains the present performance status of the solution they are using, which is accessible using the link: https://www.google.com/appsstatus#hl=en&v=status

The performance metrics for Gsuite are amalgamated as No Issues, Service Disruption and Service outage. Gsuite users therefore look out on the dashboard for any of these performance metrics whenever they are reported, and are therefore part of the SLA with Google.

No issues means the solution is on and executing normally, Service Disruption means the solution has been switched off briefly for the sake of maintenance while Service outage means the solution is not operational due to a technical issue. A sample snapshot for the dashboard is as illustrated in Figure 50.

Current status	10/18/19	10/19/19	10/20/19	10/21/19	10/22/19	10/23/19	10/24/19
Google Analytics							
App Maker							
Google Maps							
Blogger							
Google Sync for Mobile							
Classroom							
Google Realtime API							
						« Older	Newer »

All times are shown in your local timezone unless otherwise noted. RSS Feed

No Issues Service disruption Service outage

Figure 50: QoS Monitoring Platform for Gsuite

The Google Service Level Agreement for Gsuite states that if Google fails to realise the GSuite SLA, while the client realizes their responsibilities under this GSuite SLA, the client will be eligible to get Service Credits (Google, 2019).

The SLA defines two key terms, namely, Downtime for a domain which refers to when client error rate is greater than five percent. It is measured basing on server side rate of error; and Monthly Uptime, measured in Percentage, which refers to sum total of minutes in a month subtract the number of Downtime Minutes encountered in a month, divided by the sum total number of minutes in a month. These are elaborated as in Table 9.

Table 9: Google Service Level Agreement based on uptime.

Monthly Percentage Uptime	Service days added to the Service end term (or monetary credit equal to the value of days of service for monthly postpay billing clients), at no charge to the client
< 99.9% - >= 99.0%	3
< 99.0% - >= 95.0%	7
< 95.0%	15

Whereas these percentages have been defined, the dashboard does not provide the users with direct view of the uptime percentages. This means the user has to request the information from Google or the reseller once they notice the service is down. Upon receiving the percentages the user has no means of validating the percentages as provided by Google.

5.2.2. Salesforce

Salesforce provides its customers a platform to confirm on the status of the services to which they have subscribed. Four notable metrics are found on the platform, namely Available**,** Performance degradation, Service disruption and Maintenance.

The dashboard is accessible using the link: https://status.salesforce.com/products/all, while the terms of service can be accessed via the link: https://c1.sfdcstatic.com/content/dam/web/en_ie/www/documents/services-training/SSC-EU-%20Success%20Cloud%20Compare%20Plans%20-%20687-final.pdf.

Available refers to the fact that the service is on and in execution, Performance degradation means the service is running but at below expected quality of service, service disruption means the service is unavailable due to system failure, while maintenance means the service is unavailable for maintenance purposes.

Whereas the platform informs the user of performance degradation, it does not provide the exact level or extent of performance degradation. The user is therefore not able to gauge or quantify the level performance degradation. Sample screen shots from the platform are as shown in Figure 51.

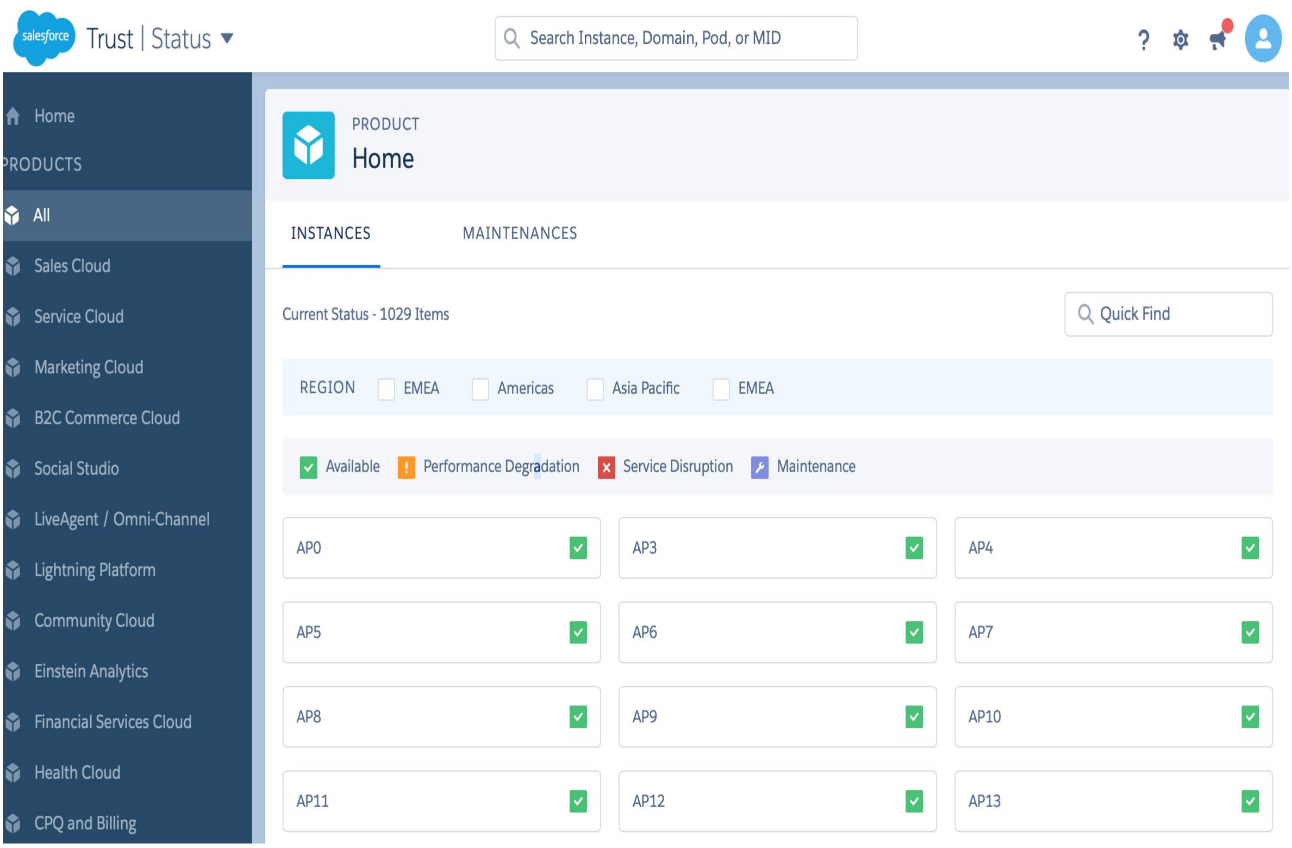


Figure 51: Sample Screenshot for Salesforce QoS Monitoring Platform

Further, Salesforce has plans that categorize its clients' level of service level agreement based on the pricing, namely standard, premier, premier plus and priority. The standard client gets a response to a reported issue within 2 days and has 12 hours 5 days a week online support (12/5).

The premier client gets support within one hour of reported critical incident and has 24 hours, 7days a week (24/7) of phone and online support. The premier plus client has similar support as premier plus an additional access to admin services. Priority has a minute 15 critical response and has 24 hours, 7 days a week (24/7) of phone and online support. The clients pay different prices for the various service level agreements plans (Salesforce, 2019). In situations where the services are down the Salesforce approach does not provide a cloud service user with the actual gauge of the performance degradation.

5.2.3. Hubspot

Hubspot offers a cloud solution where its clients can see the execution status of the cloud solutions it is offering. The execution status are grouped in order as either Operational but Degraded Performance, followed by Partial Outage, then Major Outage and finally Maintenance. The platform is accessible using the link: https://status.hubspot.com. A snapshot of the monitoring platform for Hubspot QoS is illustrated by Figure 52.

The operational but degraded performance means the user is able to get all the services but with low performance standards than the usual standards like longer load times.

Partial outage means some services are not available and so the client should expect to use only a fraction of the services they have subscribed to, while major outage means all the services are not available to the users.

Maintenance is used when the platform is deliberately made unavailable for a predetermined amount of time, which is communicated to users in advance, with the aim of either fixing earlier identified issues or upgrading the platform.

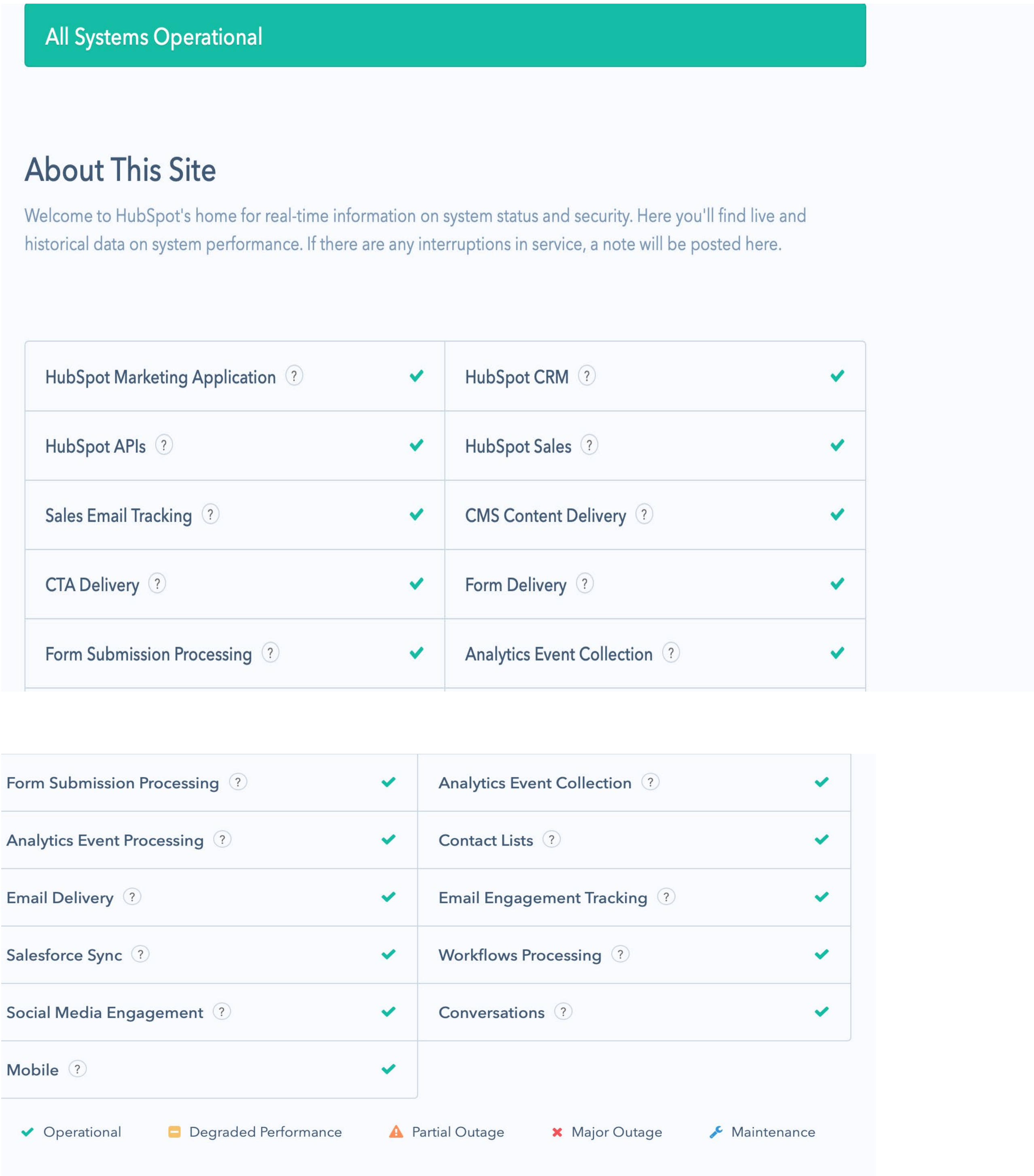


Figure 52: Hubspot Platform for QoS Monitoring

The platform further summarizes the quantitative metrics as illustrated by Figure 53.

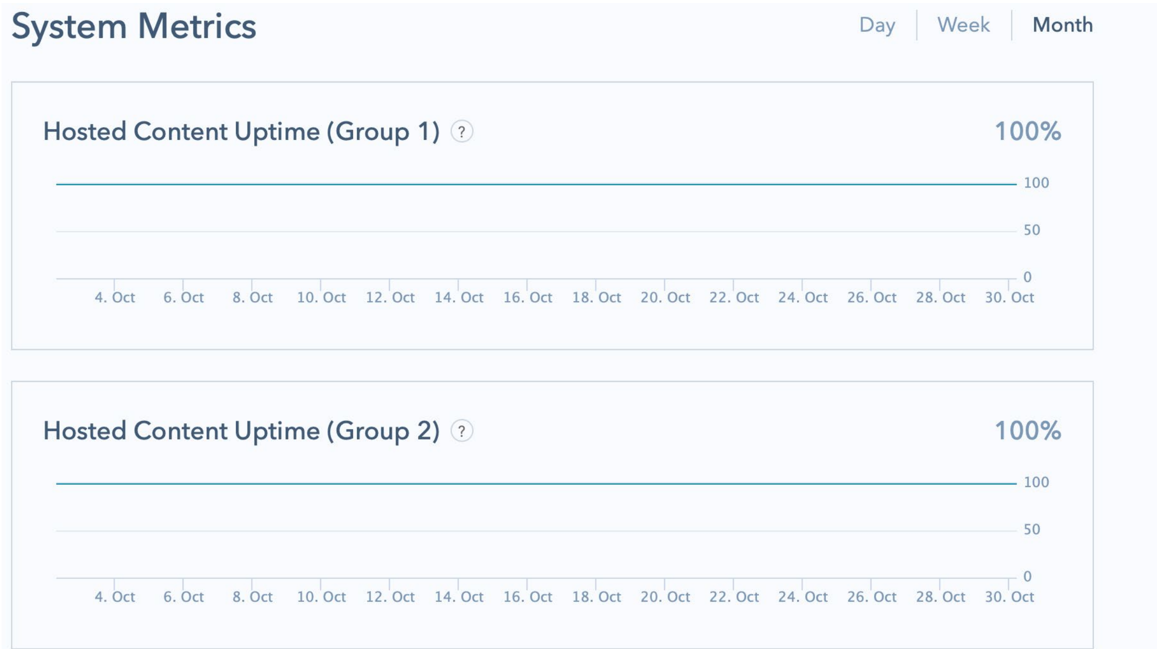


Figure 53: Summary of QoS Metrics Monitored by Hubspot Platform

The Platform also provides a summary of historical incidences as experienced by users and actions taken by the provider to address the incidences. This is as shown in Figure 54.

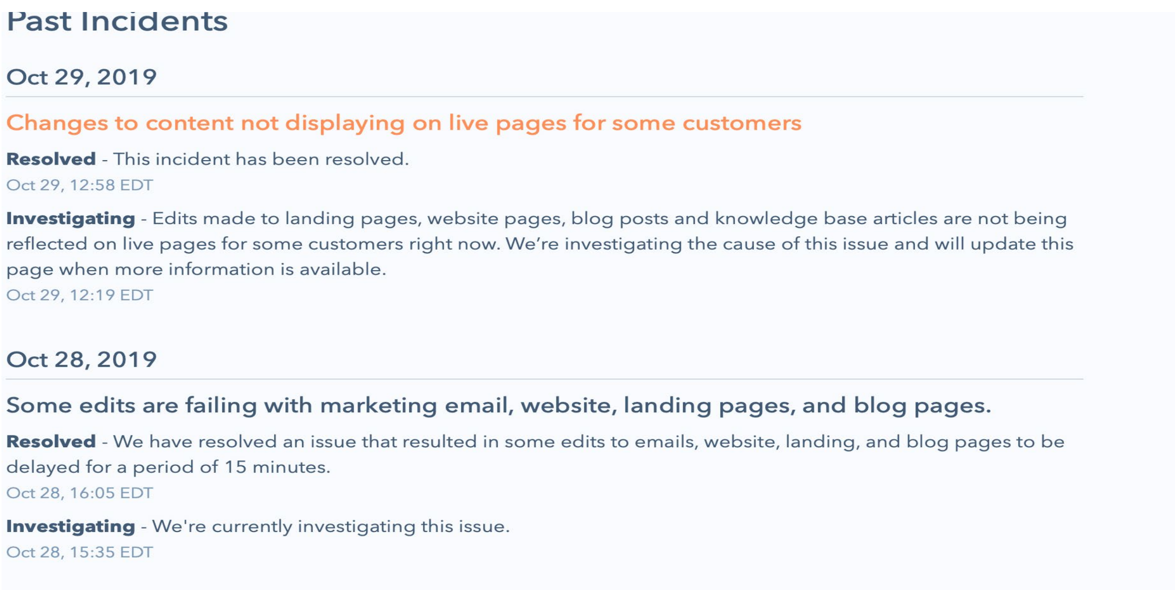


Figure 54: Summary of Hubspot Past Incidences

5.2.4. Shopify

Shopify provides a platform from which customers can view any issues affecting the platform at any particular moment. The platform also provides a quantitative metric for the average time the platform took to reply to the requests from the user. Viewing the functioning status of the platform is accessible using the link https://status.shopify.com.

A snapshot of the platform is depicted by Figure 55.

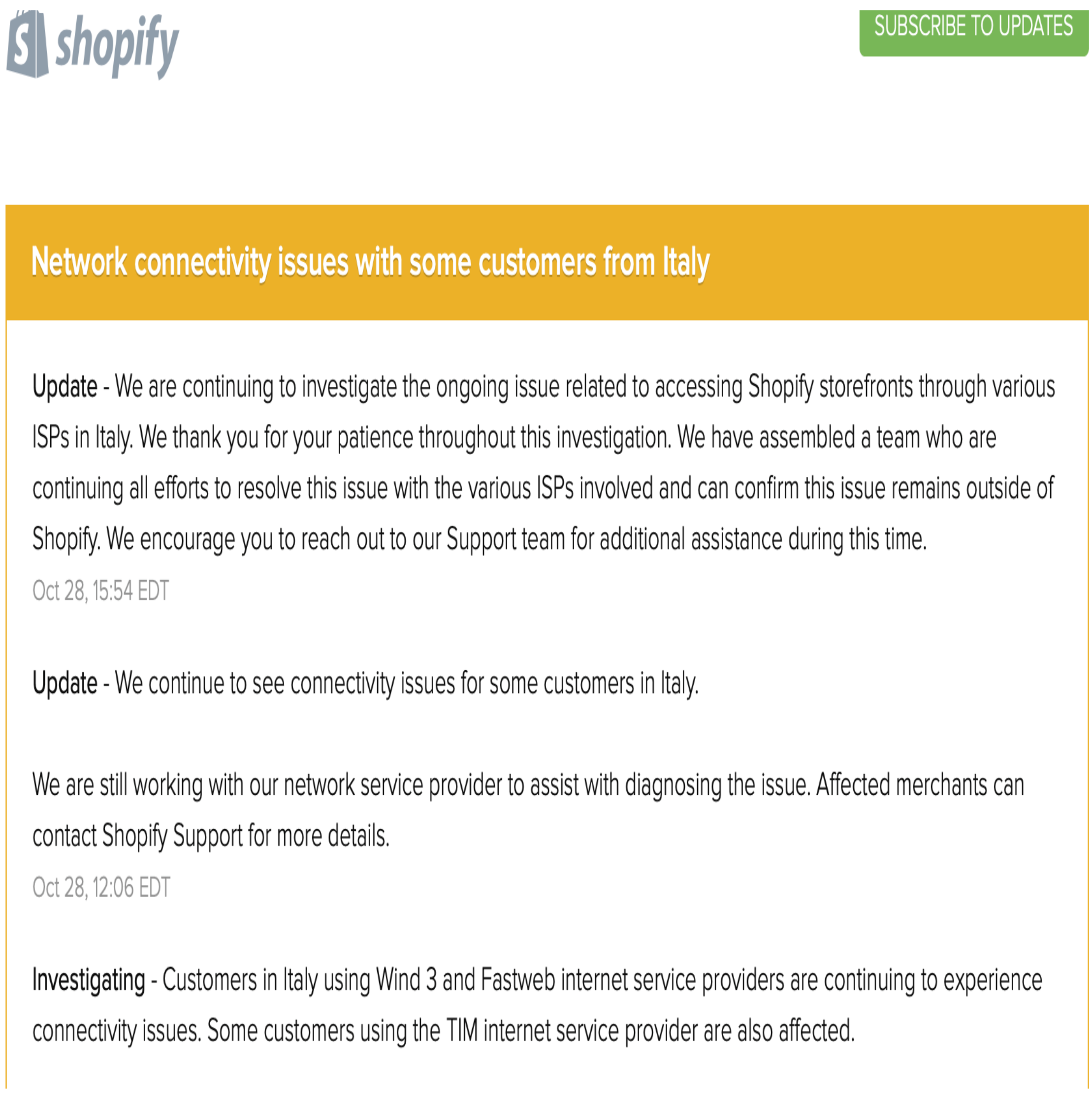


Figure 55: Shopify Platform for QoS Monitoring

A screenshot for the Quantifiable metrics, namely service response time, is as depicted in Figure 56.

Figure 56: Shopify Quantifiable QoS Metrics.

The platform also provides users with a historical view of past incidences and the actions taken to remedy the incidences. A screenshot is as shown in Figure 57.

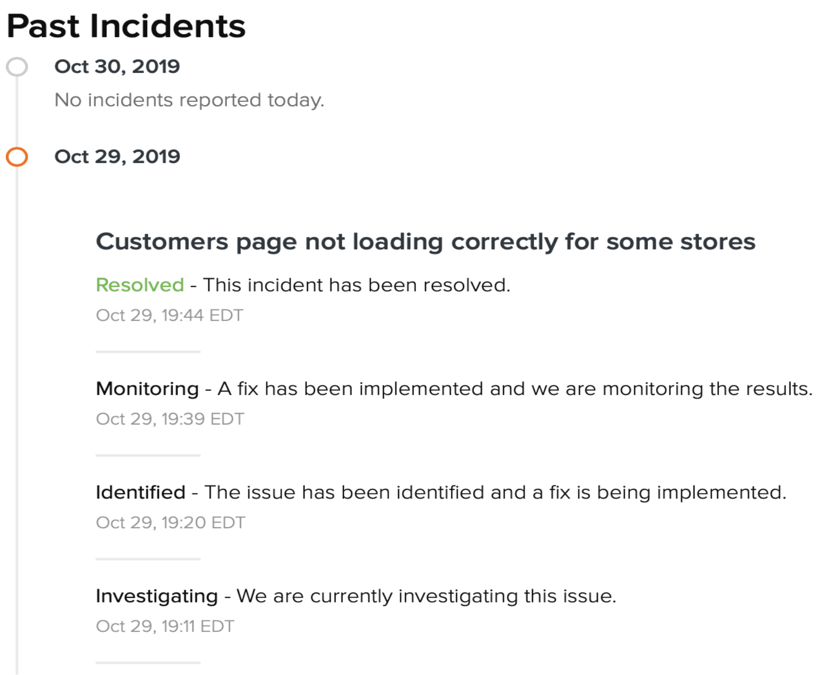


Figure 57: Shopify Historical Incidences.

A summary of the QoS aspects depicted by the Shopify platform are: No known issues, Maintenance, Degraded, Partial Outage and Outage as shown in Figure 58.

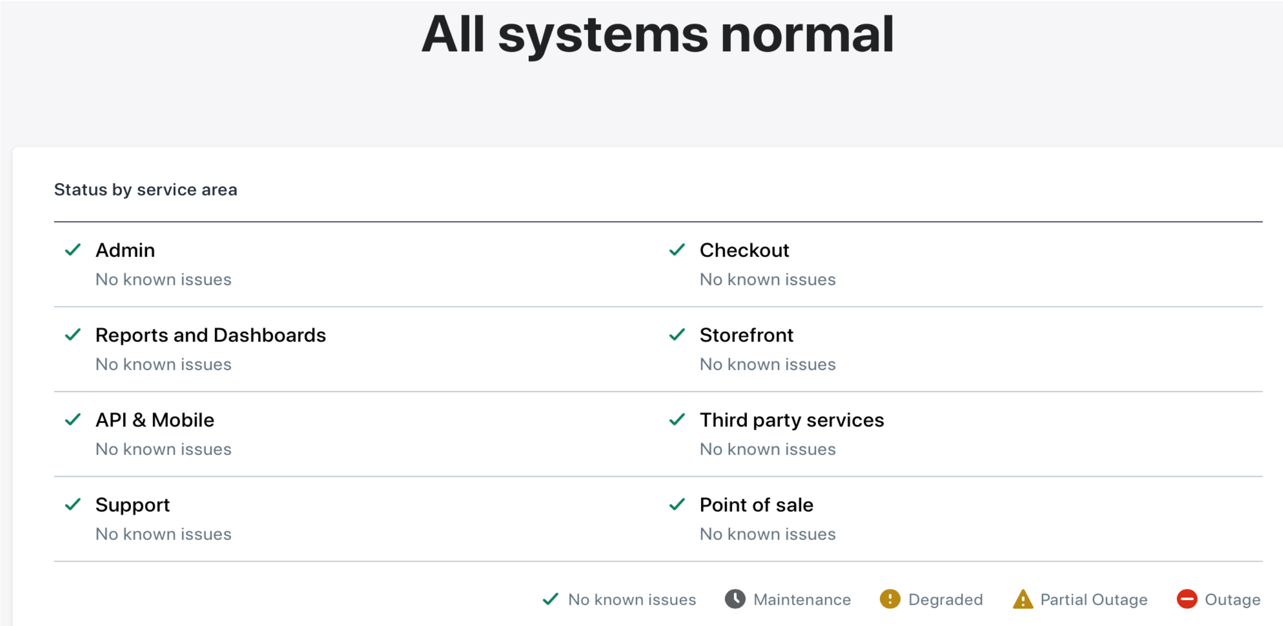


Figure 58 : Shopify QoS Metrics

5.2.5. Microsoft

Microsoft provides an opportunity for users to check the health status of its services, namely Microsoft services, consisting of Yammer, Microsoft Dynamics CRM, Office on the web, and mobile device management cloud services, on the Service health page accessible through the Microsoft 365 admin center.

It is used whenever one is experiencing a problem with a cloud service, to check the service health to ascertain on whether this is a known issue whose resolution is in progress before calling the support team or spending time in troubleshooting it.

It is accessed via Microsoft 365 admin center at https://admin.microsoft.com, and the health state of each cloud service is illustrated in a Table format as depicted in Figure 59.

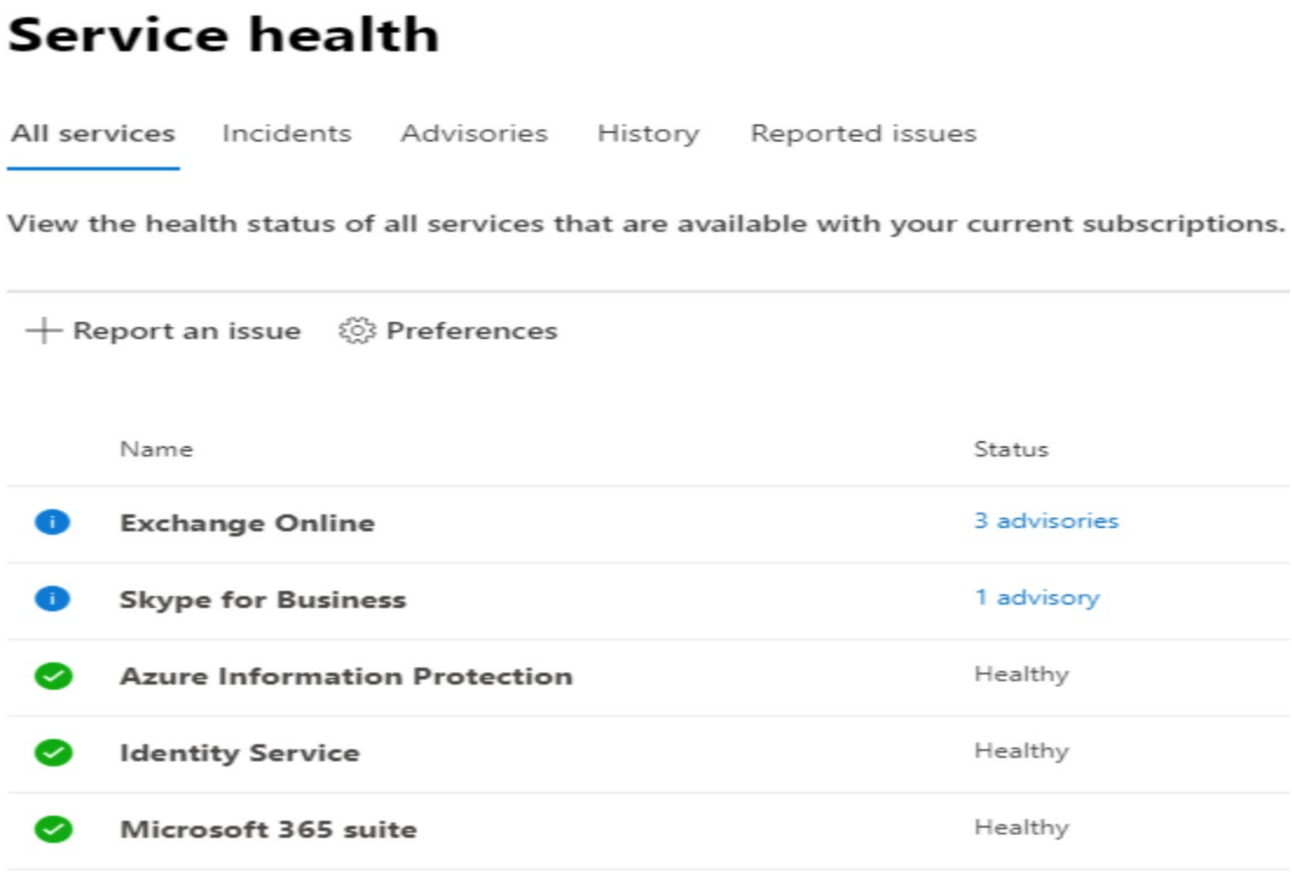


Figure 59 : The Health Status of Microsoft Cloud Services.

From Figure 59, services currently up and running as expected as shown as healthy, while the incidences tab will show services that have a reported problem and are thus not functioning as expected.

Microsoft defines a service incident as an event that impacts on the delivery of a service. These Service incidents may be occasioned by hardware or software failure in the Microsoft data center or a faulty network connection between the client and Microsoft, or even a major data center issue like regional catastrophe, flood, or fire.

Once an incident has been reported, Microsoft performs an impact assessment to determine which specific features affected by the incident and how they are affected. This is then posted as an advisory for the services that are available albeit with degraded performance. A sample snapshot for the advisory page is as illustrated in Figure 60.

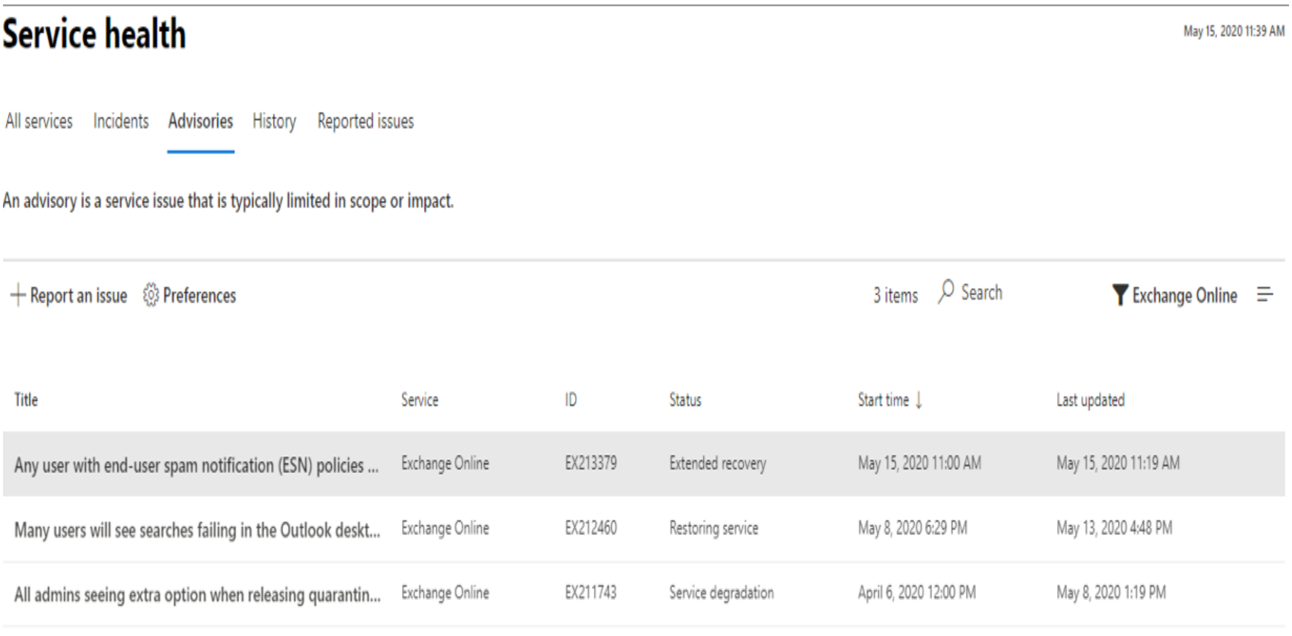


Figure 60 : Advisory Page for Microsoft Services

5.3. Analysis of the Testing Results

From the case studies of the select global cloud service providers, summarized results from the capability of different QoS measuring tools in use by the selected four global cloud solution providers is depicted in Table 10.

Table 10: Comparative Summary of Cloud QoS Monitoring Tools Capabilities

No	QoS Metric	Cloud Provider's Tool Capability					All Cloud Providers
		Gsuite	SalesForce	Hubspot	Shopify	Microsoft	Vendor Neutral Model
1.	Service Response Time	✖	✖	✖	✔	✖	✔
2.	Service Availability	✔	✔	✔	✔	✔	✔
3.	Service Stability	✖	✖	✖	✖	✖	✔

From Table 10, Gsuite, Salesforce, Hubspot and Microsoft QoS Monitoring tools have a one QoS metric measuring capability, which is Service Availability. While Shopify QoS Monitoring tool has capabilities of two metrics, which are service response time and service availability.

Based on Table 10, a client on Gsuite, Salesforce and Hubspot who wishes to know the service response time of the services they are receiving will not be able to know. Further a client who wishes to compare the performance of the various providers will not be able since the tools are provider specific and thus inter cloud comparison is not possible. Fortunately, the vendor neutral tool, measures all three metrics, and can be used for cross vendor comparison.

5.4. Application of the Vendor Neutral Model Tool in Cloud Provider Choice

To equate the functioning of cloud service providers providing the same services, this research focused on Microsoft Office 365 and Google docs for comparison purposes. The selection was done based on the fact that the said providers offer similar office applications, namely Word, Excel and PowerPoint and are the leading providers in that market segment.

The comparative test was done using the same terminal, at the same times where the applications are opened on different tabs of the same browser and under the same Internet conditions.

The testing was done to resemble an ordinary user who would want to use the said applications at random times of the day, between 6th October 2020 to 27th October 2020, with a sample size of sixty (60) runs having been used, amounting to platform usage of the platform three times (morning, afternoon and evening) per working day of the testing duration. This was aimed at emulating the way SLAs are evaluated after a certain period of time, like quarterly or monthly before payments are done. The average results for the comparison are as shown in Table 11. Sample raw results from which the averages were computed are shown in appendix 6.

Table 11: Comparison Results Between Microsoft office and Google Docs

Platform	Average Service Response Time (Seconds)	Average Availability	Stability
Google	4.47	100%	Stable (2.003 sec)
Microsoft	6.04	100%	Stable (5.966 sec)

From the analysis in Table 11, the average service response time for Google is 4.47 while for Microsoft is 6.04 seconds. Both platforms had an availability of 100% since at no time during testing did any of the platform report a platform failure leading to outage of services.

Whereas the availability is 100%, the fluctuations in the service response time are higher for Microsoft at 5.966 seconds than for Google at 2.003 seconds, meaning the Google platform was more stable than the Microsoft platform.

In summary it is found that Google performed better than Microsoft. Where a decision is to be made on whose services to procure, the user can factor in their decision making process this performance measures.

The snapshot reports from the monitoring tool are shown in Figure 61 for Google and 62 for Microsoft respectively.

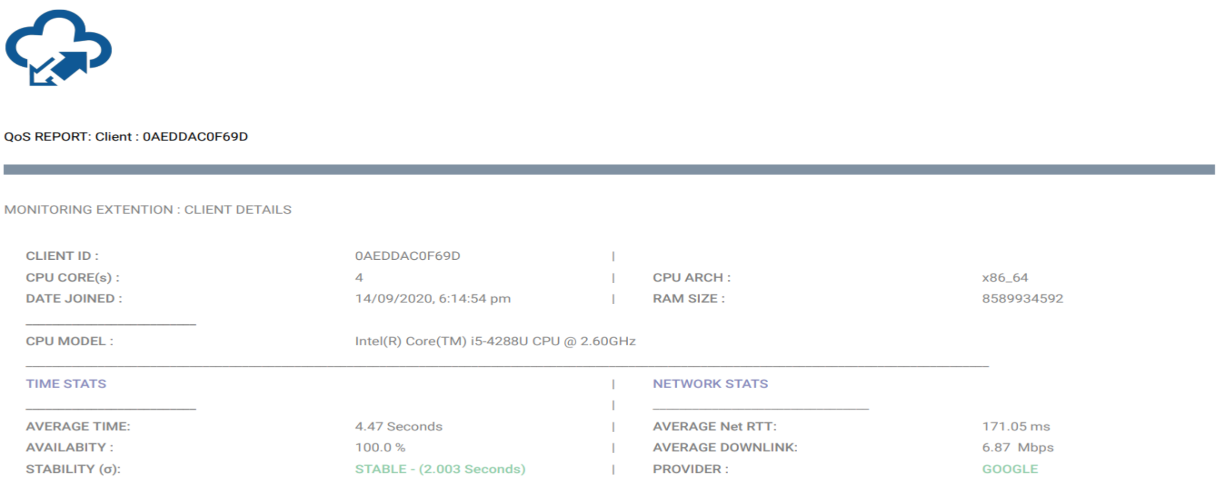
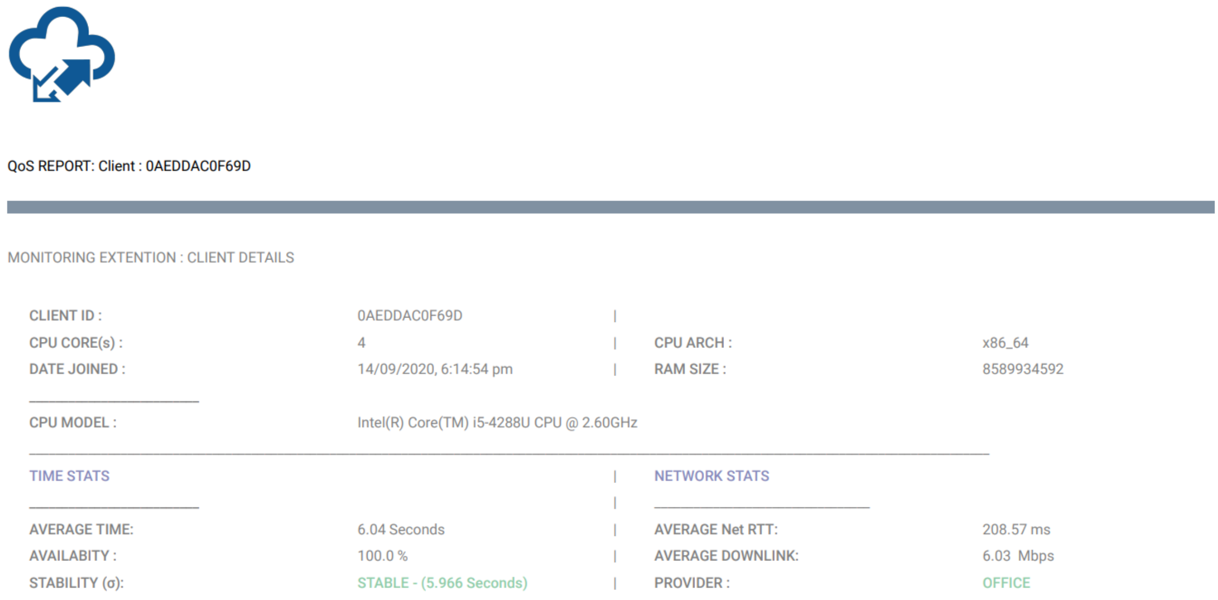


Figure 61: Google Performance Report



QoS REPORT: Client : 0AEDDAC0F69D

MONITORING EXTENTION : CLIENT DETAILS

CLIENT ID :	0AEDDAC0F69D	|		
CPU CORE(s) :	4	|	CPU ARCH :	x86_64
DATE JOINED :	14/09/2020, 6:14:54 pm	|	RAM SIZE :	8589934592
CPU MODEL :	Intel(R) Core(TM) i5-4288U CPU @ 2.60GHz			
TIME STATS		|	NETWORK STATS	
AVERAGE TIME:	6.04 Seconds	|	AVERAGE Net RTT:	208.57 ms
AVAILABITY :	100.0 %	|	AVERAGE DOWNLINK:	6.03 Mbps
STABILITY (σ):	STABLE - (5.966 Seconds)	|	PROVIDER :	OFFICE

Figure 62: Microsoft Performance Report

The QoS monitoring tools of the vendors also reported continuous system availability of the services and thus the user could build trust to the results from the provider's tool due to similarity in the results from the vendor agnostic tool and those of the cloud provider.

5.5. Application of the Vendor Neutral Model Tool in Cloud Providers Trust Computations

To compute trust based on the reported results by the chosen cloud providers, Google and Microsoft, tests were done as from 6th October 2020 to 27th October 2020. The results from the tests were subjected to the trust quantification model depicted in Figure 29. The QoS results from the vendor agnostic tool, which were used for trust quantification, are depicted in Table 12.

Table 12: Measured QoS Results by the Vendor Agnostic Tool

Platform	Average Response Time	Average Availability	Stability
Google	4.39	100%	Stable (1.986 sec)
Microsoft	5.99	100%	Stable (5.845 sec)

From the analysis in Table 12, the average service response time, time required to process and complete a service request, for Google is 4.39 seconds while for Microsoft is 5.99 seconds.

Both platforms had an availability of 100%, which means at no time during testing period did any of the platform report a service failure leading to outage of services and therefore making the user unable to access the services they wished to use.

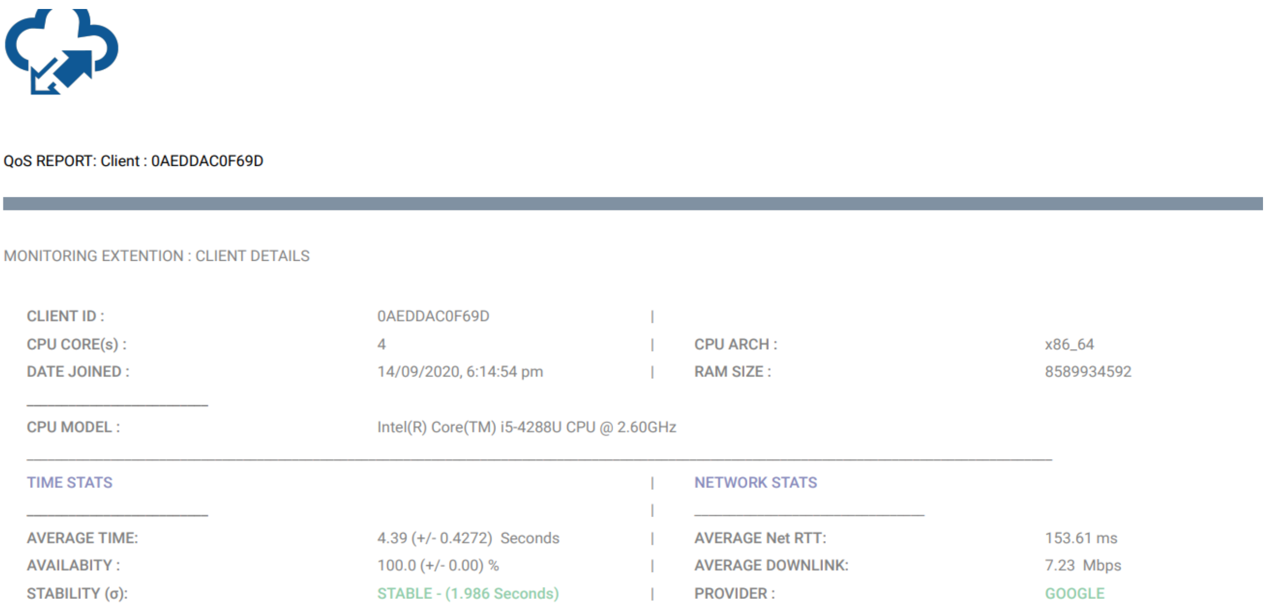
Whereas the availability is 100%, the stability, fluctuations in the service response time, computed using standard deviation, are higher for Microsoft at 5.845 seconds than for Google at 1.986 seconds, meaning the Google platform was more stable than the Microsoft platform.

From the studies done by Makokha et al (2019), a common metric between the vendor agnostic cloud QoS measuring solution and the cloud provider integrated QoS measuring solutions is the service availability.

During the testing period, Google, using its QoS platform at: https://www.google.com/appsstatus, reported no issues during the entire time, translating to 100% availability.

Similarly, Microsoft, through its QoS monitoring platform, https://admin.microsoft.com, showed the status of office suites to be healthy during the entire time, translating to 100% availability.

The QoS value screenshots from the vendor neutral model for Microsoft and Google platforms are as shown in Figure 63 and 64 respectively.



QoS REPORT: Client : 0AEDDAC0F69D

MONITORING EXTENTION : CLIENT DETAILS

CLIENT ID :	0AEDDAC0F69D		
CPU CORE(s) :	4	CPU ARCH :	x86_64
DATE JOINED :	14/09/2020, 6:14:54 pm	RAM SIZE :	8589934592
CPU MODEL :	Intel(R) Core(TM) i5-4288U CPU @ 2.60GHz		
TIME STATS		NETWORK STATS	
AVERAGE TIME:	4.39 (+/- 0.4272) Seconds	AVERAGE Net RTT:	153.61 ms
AVAILABITY :	100.0 (+/- 0.00) %	AVERAGE DOWNLINK:	7.23 Mbps
STABILITY (σ):	STABLE - (1.986 Seconds)	PROVIDER :	GOOGLE

Figure 63: QoS Screenshot Results for Microsoft

QoS REPORT: Client : 0AEDDAC0F69D

MONITORING EXTENTION : CLIENT DETAILS

CLIENT ID :	0AEDDAC0F69D		
CPU CORE(s) :	4	CPU ARCH :	x86_64
DATE JOINED :	14/09/2020, 6:14:54 pm	RAM SIZE :	8589934592
CPU MODEL :	Intel(R) Core(TM) i5-4288U CPU @ 2.60GHz		

TIME STATS		NETWORK STATS	
AVERAGE TIME:	5.99 (+/- 1.3055) Seconds	AVERAGE Net RTT:	196.75 ms
AVAILABITY :	100.0 (+/- 0.00) %	AVERAGE DOWNLINK:	6.37 Mbps
STABILITY (σ):	STABLE - (5.845 Seconds)	PROVIDER :	OFFICE

Figure 64: QoS Screenshot Results for Google

Using the Quantitative Trust Model by Makokha et al (2019), and the service availability QoS metric, which is the common QoS Metric between the vendor agnostic solution and cloud providers' integrated solutions, trust quantification values are as in Table 13.

Table 13: Quantitative Trust Values

Platform	**Vendor Neutral results**	**Cloud Provider Results**	**Trust Value**
Google	100%(± 0)	100%	1
Microsoft	100%(± 0)	100%	1

From Table 13, a cloud user can trust the results from the cloud providers due to the fact that they are within the confidence interval of the vendor neutral tool. This is critical for the trust building phase as highlighted by Grabner-Kräuter and Kaluscha, (2008) and also augments the direct experience trust concept advanced by Dragoni (2009) since the user will have experienced the services from the providers during the usage phase.

5.6. Summary of Results and Discussion

From the results highlighted the vendor neutral tool is capable of monitoring the three vital QoS metrics of SaaS cloud providers, namely, service response time, service availability and service stability. The results for the Google docs, Salesforce, Shopify and Hubspot are indicated as 4.83 seconds, 2.93 seconds, 2.45 seconds and 2.59 seconds, respectively in so far as service response time is concerned. The tool as well computed the availability and stability of the platforms and all were found to be 100% available and stable throughout the entire testing time.

For platforms offering similar services, case in point Google docs and Microsoft office suite a performance comparison reveals that the Google platform is better in terms of service response time and the stability of the platform, than Microsoft platform.

Quantification of trust using the vendor neutral model evaluates to a trust value of one (1) implying the user can trust the QoS values as reported by the cloud provider by the fact that they are within the 95% confidence level of those measured by the vendor neutral model.

5.7. Chapter Summary

This chapter highlighted the conditions and platform under which the testing was done and the principles of cloud quality of service monitoring that were put in consideration during the testing. The chapter also presented the testing results for the four selected cloud computing platforms and the capability of the vendor neutral tool as compared to cloud provider platform integrated tools. Application of the tool in trust evaluation and cloud provider selection was also presented in this chapter.

CHAPTER SIX: CONCLUSION AND FURTHER RESEARCH

This research aimed at addressing the problem of cross vendor cloud QoS monitoring, and used four research objectives to solve the problem, namely, develop a high level client trustable QoS Monitoring Framework for cloud computing systems; design a Vendor Neutral Model that implements the designed Framework for SaaS Cloud Computing solutions; Prototype and Evaluate the QoS Monitoring tool developed from the new Vendor Neutral Cloud Performance Monitoring Model; and Develop Algorithms for a SaaS monitoring tool that implements the new Vendor Agnostic Cloud Performance Monitoring Model. In the course of carrying out this research, five (5) publications in international journals were made, and are indicated in the linked publications section of this report.

6.1 Conclusion

The first objective on development of a client trustable QoS Monitoring Framework was met by first deriving the existing cloud QoS monitoring framework from existing explicitly documented cloud QoS monitoring models. Upon deriving the existing framework, and relying on identified shortcomings, a proposed framework with trust factored, was introduced by developing a framework that is user centric and that factors in the reason of QoS monitoring from the user perspective. The output of the first objective was therefore a proposed client trustable cloud QoS monitoring framework.

The second objective being development of a vendor neutral QoS monitoring model for SaaS cloud solutions, was developed after review of existing SaaS QoS monitoring models and noting their limitations. This aimed at transferring the location of the monitoring tools from the cloud provider's infrastructure and locating it in the users' devices used to access the cloud services. Enabling end-to-end QoS monitoring. The output was a model anchored on the browser, and implemented as a browser extension.

Objective three involved prototyping, implementing and evaluating a vendor neutral QoS monitoring tool from the proposed vendor neutral model. The output was a prototype, and was developed using CSS, JavaScript, HTML, SQLite database and Node JS, as the development tools. The prototype was implemented on chrome browser and tested on Shopify, Google, Microsoft Hubspot and Salesforce SaaS platforms. The selected platforms hold the larger market share of their respective market segments and therefore have mature solutions. The tool was then validated through a comparative case study by comparing its results with the results from the cloud providers' integrated tools.

After validation of the tool, objective four, which was development of algorithms that implement the model, was realized by deriving the algorithms from the prototype code. The output of this objective was therefore a set of algorithms that implement the vendor neutral SaaS cloud QoS model.

From the evaluation done, the tool developed from the Client based Vendor Neutral Model has an advantage in that it has the capability of providing monitoring results for all the three key QoS parameters, comprising of response time of the service, availability of the service and stability of the service.

The tools from the Vendor Neutral Model, being vendor agnostic also, can be used for cross-cloud QoS performance comparison since they are not tightly coupled to the underlying facility architecture of the cloud platform.

The tools could also be used to validate the reported QoS performance from the provider's tool. Further, the fact that the Client based Vendor Neutral tool is located at user's end, the results are trustable to the user. This is crucial in enhancing trust between the cloud providers and clients. This is reinforced by the quantitative trust model that if used and the results evaluate to one (1), then clients will develop trust in the cloud providers platforms.

6.2. Knowledge Contribution to Computer Science

This research advances new knowledge in QoS monitoring in the Cloud Computing field of study, by introducing a novel perspective to cloud QoS monitoring, namely, the user centric perspective of QoS monitoring that fosters trust in the cloud services and the QoS values reported by cloud service providers. The new perspective is anchored on the proposed client trustable QoS monitoring framework and the associated vendor neutral SaaS QoS Monitoring Model.

This is a new realm that could be researched further in the field of Computer Science, with the developed client centric QoS monitoring framework, acting as an anchor framework for development of other cloud QoS monitoring models and their associated tools.

6.3. Implications on Theory, Practice and Policy

This research introduces a new framework, dubbed the Client Trustable QoS Monitoring Framework, from which future principles of cloud QoS monitoring can be anchored on, especially with regards to monitoring from the client's perspective, cloud vendor pre-selection monitoring and validation of cloud provider reported QoS results.

The proposed model, the Vendor Neutral SaaS QoS monitoring model, influences future practice in that the practice of double QoS measurement in the cloud computing set up, where there exists the provider integrated cloud QoS monitoring tool and the vendor neutral tool, monitoring the cloud QoS simultaneously, will be vital during SLA evaluation.

For cloud provider pre-selection purposes, the procurement policy on cloud services can benefit from the ability to perform testing on the performance of various cloud QoS providers before making a choice on whose services to procure. The results from the vendor neutral tool can therefore for part of the evaluation criteria with a certain apportioned weight to the overall score. The SLA can also incorporate the concept of validation using both the results from cloud provider QoS tools and the vendor neutral QoS tools.

6.4. Future Studies

The developed model was limited to the Software as a Service (SaaS) cloud applications. Additional studies could be done to extend or develop new models based on the same Client Trustable Framework for Platform as a Service applications (PaaS) and Infrastructure as a Service applications (IaaS).

The developed model was tied to the browsers on the user's terminal, to extend this research, explorations could also be done to identify common applications on user terminals like the operating systems or user terminal utilities on which other new vendor neutral QoS monitoring models could be pegged on, using the same client trustable framework.

References

Aceto, G., Botta, A., Donato, W. & Pescapè, A. (2013) Cloud monitoring: A survey, *Computer Networks*: *The International Journal of Computer and Telecommunications Networking*. Volume 57, Issue 9.

Adinolf, O., Cristaldi, R., Coppolino, L. & Romano, L. (2012). QoS-MONaaS: A Portable Architecture for QoS Monitoring in the Cloud. *Signal Image Technology and Internet Based Systems (SITIS), 2012 Eighth International Conference on, Napes, Italy*. IEEE.

Adrian, W. R., Branstad, A. M. & Cherniavsky, C. J. (1982) Validation, Verification, and Testing of Computer Software. *Computing Surveys*, Vol. 14, No. 2, June 1982.

Agrawal & Choubey (2015) Enhanced Agent Based Scheduling & Monitoring System in Cloud Computing. *International Journal of Computer Science And Technology*. Vol. 6, Issue 1.

Akamai (2016). *TechValidate Survey Report: SaaS Application Trends and Challenges*. https://www.akamai.com/de/de/multimedia/documents/report/techvalidate-saas-survey-results.pdf [accessed on 21st September 2020].

Ardagna, D., Casale, G., Ciavotta, M., Pérez, J. F., & Wang, W. (2014). Quality-of-service in cloud computing: modeling techniques and their applications. *Journal of Internet Services and Applications 2014*. Volume 5, Issue 11.

Al-Roomi, M., S Al-Ebrahim, S., Buqrais, S. & Ahmad, I. (2013). Cloud Computing Pricing Models: A Survey. *International Journal of Grid and Distributed Computing*.Vol.6, No.5.

AL-Badareen A.B., Selamat M.H., A. Jabar M., Din J. & Turaev S. (2011) Software Quality Models: A Comparative Study. In: Mohamad Zain J., Wan Mohd W.M., El-Qawasmeh E. (eds) Software Engineering and Computer Systems. ICSECS 2011. *Communications in Computer and Information Science*, Vol 179. Springer: Berlin, Heidelberg.

Alhamazani, K., Ranjan, R., Mitra, K., Rabhi, F., Jayaraman, P., Khan, S., Guabtni, A. & Bhatnagar, V. (2014). An Overview of the Commercial Cloud Monitoring Tools: Research Dimensions, Design Issues, and State-of-the-Art. *Computing*. 97. 10.1007/s00607-014-0398-5.

Alhamazani, K., Ranjan, R., Jayaraman, P., Mitra, K., Liu, C., Rabhi, F., Georgakopoulos, D. & Wang, L. (2019) Cross-Layer Multi-Cloud Real-Time Application QoS Monitoring and Benchmarking As-a-Service Framework, in *IEEE Transactions on Cloud Computing*, vol. 7, no. 1, pp. 48-61, 1 Jan.-March 2019.

Arie, D. (2021) *Content Delivery Network Explained.* https://www.globaldots.com/resources/blog/content-delivery-network-explained/ [accessed on 21st May 2021].

Armbrust, M., Fox, A., Griffithth, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I. & Zaharia, M. (2009): *Above the clouds: A Berkeley View of Cloud Computing*. Technical Report No. UCB/EECS-2009-28. EECS Department, University of California, Berkeley.

Ashraf, I. (2014). An Overview of Service Models of Cloud Computing. *International Journal of Multidisciplinary and Current Research*. Vol.2 (July/Aug 2014) issue.

Barlas, Y. & S. Carpenter (1990). Philosophical Roots of Model Validation: Two Paradigms. *System Dynamics Review* Vol. 6, No. 2, pp. 148-166.

Berglund, A. & Leifer, L. (2013) Why we Prototype! An International Comparison of the Linkage between Embedded Knowledge and Objective Learning. *Engineering Education*, 8:1, 2-15, DOI: 10.11120/ened.2013.00004.

Barth, A., Porter Felt, A., Saxena, P. & Boodman, A. (2010), Protecting Browsers from Extension Vulnerabilities: *Proceedings of the 17th Network and Distributed Systems Security Symposium*, San Diego, CA (Feb 28th – March 3rd, 2010).

Bardsiri, K. A. & Hashemi, M. S. (2014). QoS Metrics for Cloud Computing Services Evaluation. *International Journal of Intelligent Systems and Applications*, Vol. 6, No. 12.

Buyya, R. (Ed.), *High Performance Cluster Computing: Architectures and Systems*. Volume 1, Prentice Hall, Upper Saddle River, USA, 1999.

Buyya, R., Broberg, J. and Goscinski, A. (2011) Eds. *Cloud Computing: Principles and Paradigms*. Hoboken, New Jersey: John Wiley & Sons, Inc.

Chekfoung T., Kecheng L. & Lily S. (2013) A design of evaluation method for SaaS in cloud computing, *Journal of Industrial Engineering and Management*, JIEM, 2013 – 6(1): 50-72.

Cicotti, G., Coppolino, L., Cristaldi, R., D'Antonio, S. & Romano, L. (2011). QoS Monitoring in a Cloud Services Environment: the SRT-15 Approach in *11th Proceedings of the 2011 international conference on Parallel Processing, Bordeaux, France*. Springer-Verlag Berlin, Heidelberg. Pages 15-24.

Cisco Systems (2002). *Internetworking Technologies Handbook*. Cisco Press, Indianapolis, Indiana.

Cedillo, P., Gonzalez-Huerta, J., Abrahao, S., & Insfran, E. (2015). A Monitoring Infrastructure for the Quality Assessment of Cloud Services. In D. Vogel, X. Guo, C. Barry, M. Lang, H. Linger, & C. Schneider (Eds.), *Information Systems Development: Transforming Healthcare through Information Systems (ISD2015 Proceedings).* Hong Kong, SAR: Department of Information Systems.

Churakova, I. & Mikhramova, R. (2010) *Software as a Service: Study and Analysis of SaaS Business Model and Innovation Ecosystems*, M.Sc., Ghent University, Belgium.

Cho, J., Hurley, P. M. & Xu, S. (2016). Metrics and Measurement of Trustworthy Systems *in* MILCOM 2016 - 2016 IEEE Military Communications Conference, Baltimore, MD, 2016, pp. 1237-1242, doi: 10.1109/MILCOM.2016.7795500.

Communications Authority of Kenya (2016) https://ca.go.ke/consumers/ca-you/quality-of-service/ [accessed on 12th July 2016].

Dash, S. B., Saini, H., Panda, T. C., & Mishra, A. (2014). Service Level Agreement Assurance in Cloud Computing: A Trust Issue. *(IJCSIT) International Journal of Computer Science and Information Technologies*, Vol. 5 (3).

Datanyze (2021) Top competitors of Zoho CRM [online] https://www.datanyze.com/market-share/customer-relationship-management--33/zoho-crm-market-share [accessed on August 2021].

Dawson, C. (2002) *Practical Research Methods*, UBS Publishers' Distributors: New Delhi.

Desai, A. Oza, R. Sharma, P. & Patel, B. (2013) Hypervisor: A Survey on Concepts and Taxonomy. *International Journal of Innovative Technology and Exploring Engineering* (IJITEE) ISSN: 2278-3075, Volume-2, Issue-3, February 2013.

Despa, L., M. (2014). Comparative Study on Software Development Methodologies. *Database Systems Journal,* Vol. V, No. 3.

Dhingra, M., Lakshmi, J., & Nandy, S. K. (2012). Resource Usage Monitoring in Clouds in *12th Proceedings of the 2012 ACM/IEEE 13th International Conference on Grid Computing*. IEEE Computer Society: Washington. Pages 184-191.

Dragoni, N. (2009). Toward trustworthy web services - approaches, weaknesses and trust-by-contract framework. *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Volume 3, pp. 599–606, 2009.

Dumbill, E. (2012) *Big Data Now.* Sebastopol: O'Reilly Media, Inc.

Endrei, M., Ang, J., Arsanjani, A., Chua, S., Comte, P., Krogdahl, P., Luo, M. and Newling, T. (2004). *Patterns: Service-Oriented Architecture and Web Services*. North Castle Drive Armonk, NY: IBM Corporation.

Esteves. R. (2011) A taxonomic analysis of cloud computing. In *1st Doctoral Workshop in Complexity Sciences ISCTE- IUL/FCUL*, Lisbon.

European Telecommunications Standards Institute (ETSI) *Speech Processing, Transmission and Quality Aspects (STQ); User related QoS parameter definitions and measurements,* ETSI EG 202 057-1 V1.2.1 (2005-10), ETSI Guide.

Ezenwoke, A., Olawande, D. & Adigun, M. (2018) QoS-based Ranking and Selection of SaaS Applications using Heterogeneous Similarity Metrics. *Journal of Cloud Computing:Advances, Systems and Applications,* 7:15.

Falkenberg, E. D., Hesse, W., Lindgreen, P., Nilsson, B. E., Han Oei, J. L. Rolland, C., Stamper, R. K., Van Assche, F. J. M., Verrijn-Stuart , A. A. & Voss, K. (1998) *A Framework of Information System Concepts, The FRISCO Report* (Web Edition). https://www.researchgate.net/profile/Frans_Van_Assche/publication/2937450_FRISCO_A_framework_of_information_system_concepts/links/00b4951e4051b019d0000000/FRISCO-A-framework-of-information-system concepts.pdf [accessed on 1st October 2020].

Firdhous, M., Hassan, M., & Ghazali, O. (2013). Monitoring, Tracking and Quantification of Quality of Service in Cloud Computing. *International Journal of Scientific & Engineering Research, Volume 4, Issue 5.*

Foster, I., Kesselman, C. & Tuecke, S. (2001). The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International Journal of High Performance Computing Applications archive.* Volume 15, Issue 3.

Fukuyama, F. (1995). *Trust: The social virtues and the creation of prosperity*. The Free Press :New York.

Ghanam, Y., Ferreira, J., & Maurer, F. (2012). Emerging Issues & Challenges in Cloud Computing— A Hybrid Approach. *Journal of Software Engineering and Applications.* Volume 5, No. 11A.

Garg, K. S., Versteeg, S. & Buyya, R. (2011). SMICloud: A Framework for Comparing and Ranking Cloud Services. *2011 Fourth IEEE International Conference on Utility and Cloud Computing*.

Gholami, A. & Arani, M. G. (2015) A Trust Model Based on Quality of Service in Cloud Computing Environment. *International Journal of Database Theory and Application,* Vol. 8 No.5.

Google (online) Gsuite Service Level Agreement. https://gsuite.google.com/terms/sla.html [accessed on 12th November 2019].

Gouda, K. C., Dwivedi, D., Patro, A. & Bhat, N. (2014) Migration Management in Cloud Computing. *International Journal of Engineering Trends and Technology* (IJETT) – Volume 12 Issue 9.

Gorelek, E.(2013) Cloud Computing Models Comparison of Cloud Computing Service and Deployment Models. M. Eng., Massachusetts Institute of Technology http://web.mit.edu/smadnick/www/wp/2013-01.pdf.

Gorry, G. A. & Morton, S. S. M. (1971) .A Framework for Management Information Systems. Working Paper 510-71, Afred P. Sloan school of Management, Massachusetts Institute of Technology

Grabner-Kräuter, S (2000) Consumer Trust in Electronic Commerce: Conceptualization and Classification of Trust Building Measures, *in* Kautonen, T. and Heikki Karjaluoto, H (2008) *Trust And New Technologies*, Cheltenham, United Kingdom.

Grabner-Kräuter, S. & Kaluscha, A. E.. (2008). Consumer trust in electronic commerce: Conceptualization and classification of trust building measures, *in* Teemu K. and Heikki K. (Eds.) *Trust and New Technologies*. Edward Elgar Publishing 2008, Cheltenham, United Kingdom.

Grozev, N. & Buyya, R. (2012) Inter-Cloud architectures and application brokering: taxonomy and survey, *Software - Practice and Experience*, Vol. 44, no. 3, pp. 369—-390, 2012.

Gurudatt, K., Maheshchandra, J., Sadanand, B. & Hemant, B. (2013) Communication As Service Cloud. *International Journal of Computer Networking, Wireless and Mobile Communications* (IJCNWMC). Vol. 3. Issue 1.

Grosskurth, A. & Godfrey, M. (2005). A Reference Architecture for Web Browsers. Software Maintenance, 2005. ICSM'05. *Proceedings of the 21st IEEE International Conference Software Maintenance*, Budapest, Hungary, IEEE Computer Society, Los Vaqueros Circle, CA.

Habib, S. M., Hauke, S., Ries, S. & Mu˙hlha¨user, M. (2012) *Journal of Cloud Computing: Advances, Systems and Applications.* 1:19.

Hazra, A (2017) Using the Confidence Interval Confidently. *Journal of Thoracic Disease*, Volume 9 Issue 10.

Hurwitz, J., Nugent, A., Halper, F. & Kaufman, M. (2013). *Big Data For Dummies.* Hoboken, New Jersey: John Wiley & Sons, Inc.

Ibrahim, A. A. Z. A., Wasim, M. U., Varrette, S. & Bouvry, P. (2018) PRESENCE: Monitoring and Modelling the Performance Metrics of Mobile Cloud SaaS Web Services. *Mobile Information Systems*, Volume 2018, Article ID 1351386.

Institute of Electrical and Electronics Engineers (IEEE) (2009). *Standard for a Software Quality Metrics Methodology, IEEE Std 1061 TM-1998 (R2009)*, New Jersey, USA.

International Telecommunication Union (2008) *Series E: overall Network Operation, Telephone Service, Service Operation and Human Factors*. ITU-T E.800, 09/2008.

International Telecommunication Union (2015) *Series Y: Global Information Infrastructure, Internet Protocol Aspects and Next-Generation Networks: Big data – Cloud computing based requirements and capabilities*. ITU-T Y.3600, 11/2015.

International Organization for Standardization (ISO) *Information technology — Quality of service: Framework,* ISO/IEC 13236:1998(E): ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland.

International Organization for Standardization (ISO) *Risk Management-Vocabulary,* ISO/Guide 73:2009(E): ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland.

Jain, J. (2015) *Security Plug-ins Handbook: A Student's Guide*, Chicago: InfoSec Institute [Available at: http://resources.infosecinstitute.com/wpcontent/uploads/Security-Plugins-Handbook-1.pdf [viewed on 3rd November 2017].

Jayaratna, N. (1994). Understanding and evaluating methodologies. NIMSAD: A systemic approach. Berkshire: McGraw-Hill.

Jegadeesan, J. & Karuppaiah, V. (2016) Quality of Service Monitoring and Prediction in Cloud Computing Environments. *International Journal of Emerging Technology in Computer Science & Electronics* (IJETCSE), Volume 22, Issue 3.

Jones, D. (2010) Definitive Guide to Monitoring the Data Center, Virtual Environments, and the Cloud. *Nimsoft from the data center to the Cloud*. Realtime Publishers: San Francisco, CA.

Junghoon, O., Seungbong, L., & Sangjin L. (2011). Advanced evidence collection and analysis of web browser activity: *Proceedings of The Digital Forensic Research Conference*, DFRWS 2011 USA, New Orleans, LA (Aug 1st - 3rd).

Kaplunou , P (2020) The Difference Between Vertical SaaS and Horizontal SaaS (online). https://smart-it.io/blog/vertical-saas-vs-horizontal-saas/ [Accessed on 22nd September 2020].

Kjaratan, P., Jan, E., Reid, B. , Janet , K. & Farookh, M. (2000). Validating Design Methods & Research: The Validation Square. *Proceedings of DETC '00 2000 ASME Design Engineering Technical Conferences*, September 10-14, 2000, Baltimore, Maryland

Kaur, R. R. (2015). A Review of Computing Technologies: Distributed, Utility, Cluster, Grid and Cloud Computing. *International Journal of Advanced Research in Computer Science and Software Engineering.* Volume 5, Issue 2.

Karim, R. (2015) *Techniques And Tools For Secure Web Browser Extension Development*. PhD, The State University of New Jersey.

Kazmi, R. (2018) What Are Vertical and Horizontal SaaS? (Online) https://www.koombea.com/blog/vertical-horizontal-saas/ [accessed on 22nd September 2020].

Kothari, C. R. (2004) *Research Methodology: Methods and Techniques*. New Age International (P) Ltd., Publishers: New Delhi.

Kumar, R. (2005) *Research Methodology-A Step-by-Step Guide for Beginners*, (2nd ed.), Pearson Education: Singapore.

Kumar, R. (2011) *Research Methodology-A Step-by-Step Guide for Beginners*, (3rd ed.), SAGE Publications Ltd: London.

Kumar, S. & Goudar, R. H. (2012) Cloud Computing – Research Issues, Challenges, Architecture, Platforms and Applications: A Survey. *International Journal of Future Computer and Communication,* Vol. 1, No. 4, December 2012.

Kung, D. & Zhu, H. (2008) Software Verification and Validation. In Wah, B. (Ed.) *Encyclopedia of Computer Science and Engineering*, John Wiley & Sons, Inc.: New Jersey.

Laskey, K. B. & Laskey, K. (2009), *Service Oriented Architecture*. WIREs Comp Stat, 1: 101–105. doi:10.1002/wics.8.

Lerner, B. S. (2011) *Designing for Extensibility and Planning for Conflict: Experiments in Web-Browser Design*. PhD., University of Washington.

Liu, L., Zhang, Vuclip, I., Yan, G & Chen, S. (2012) Chrome Extensions: Threat Analysis and Countermeasures, NDSS '12: *Proceedings of the 19th Network and Distributed System Security Symposium*. San Diego, California February 5-8, 2012.

Lucas, H. C., Clowes, K. W., & Kaplan, R. B. (1973) Frameworks for Information Systems. Working Paper No. 142, Stanford Graduate Business School. https://www.gsb.stanford.edu/faculty-research/working-papers/frameworks-information-systems [accessed on 1st October 2020].

Makokha, F., Opiyo, E. & Okello-odongo (2017). Challenges of Quality of Service Monitoring in Cloud Computing Solutions. *International Journal of Computer and Information Technology* Vol 06 Issue 06.

Makokha, F. & Opiyo, E. (2018). A Vendor Neutral QoS Monitoring Model for SaaS Cloud Computing Solutions. *International Journal of Computer and Information Technology* Vol 07 Issue 01.

Makokha, F., Opiyo, E. & Chepken, C. (2019) Browser Integrated Vendor Neutral Cloud QoS Monitoring System. *International Journal of Computer and Information Technology*, Vol 8, No 6.

Makokha, F., Chepken, C. K. & Opiyo, E. T. (2021) End User Centric Quantitative Trust Model in Cloud Computing. *American Journal of Computer Science and Engineering*. Vol. 7, No. 1, 2021, pp. 1-7.

Manuel, P (2013) A trust model of cloud computing based on Quality of Service. *Annals of Operations Research*. 233. 1-12. 10.1007/s10479-013-1380-x.

Mayer, R. C., Davis, J. H and Schoorman, F. D. (1995). An Integrative Model of Organizational Trust, *The Academy of Management Review,* Vol. 20, No. 3.

Mcknight, D.H., Cummings, L.L. and Chervany, N. L (1998). Initial Trust Formation in New Organizational Relationships. *Academic Management Review* 23(3) 473–490.

McEvilya, B. & Tortoriellob, M. (2011) Measuring Trust in Organisational Research: Review and Recommendations, *Journal of Trust Research* 1(1).

Meera, A. & Swamynathanb, S. (2013) Agent based Resource Monitoring system in IaaS Cloud Environment. *International Conference on Computational Intelligence: Modeling Techniques and Applications,* (CIMTA). Elsevier Technology: Amsterdam, The Netherlands.

Mell, P. & Grance, T. (2011). The NIST Definition of Cloud Computing. *NIST Special Publication 800-145.*

Mishra, A. K. (1996). Organizational Responses to Crisis: The Centrality of Trust. R. M. K. and T. R. Tyler (eds), Sage, Thousand Oaks, CA, 261-287.

Mondal, K. R. & Sarddar, D. (2015). Utility Computing. *International Journal of Grid Distribution Computing,* Vol. 8, No.4.

Mamoun, H. M. & Eslam M. I. (2014). A Proposed Framework for Ranking and Reservation of Cloud Services. *International Journal of Engineering and Technology, Volume* 4, No. *9.*

McCall, J. A., Richards, P. K. & Walters, G. F. (1977) Factors In Software Quality: Concept and Definitions of Software Quality. *RADC-TR-77-369*, Vol. I (of three) Final Technical Report. Massachusetts: Hanscom Air Force Base.

Meng, S. & Liu, L. (2013) Enhanced Monitoring-as-a-Service for Effective Cloud Management. *IEEE Transactions on Computers.* Volume 62 No. 9.

Misra, A., Sharma, A., Gulia, P., & Bana, A. (2014). Big Data: Challenges and Opportunities. *International Journal of Innovative Technology and Exploring Engineering* (IJITEE) Volume-4 Issue-2.

Mutulu, M. P & Kahonge, M. A. (2018). A Multi-tenancy Cloud Trust Model using Quality of Service Monitoring – a Case of Infrastructure as a Service (IaaS). M.Sc. University of Nairobi.

Munyaradzi, Z., Maxmillan, G. & Mutembedza N. A. (2013). Effects of Web Page Contents on Load Time over the Internet. *International Journal of Science and Research (IJSR), India Online ISSN: 2319-7064,* Volume 2 Issue 9, September 2013.

Murtaza, S. & Al Masud, R. (2012) An Extended and Granular Classification of Cloud's Taxonomy and Services. *International Journal of Soft Computing and Engineering (IJSCE).* Volume 2, Issue 2.

Naeem, M. M., Memon, F. M. H., Siddique, M. & Rauf, A. (2016) An Overview of Virtualization and Cloud Computing. *Science International (Lahore)*, Volume 28 Issue 4.

Nagat, D., Jamal B. & Hongyang Q (2020) Computationally Grounded Quantitative Trust with Time. *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems* (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand© 2020.

National Institute of Standards and Technology (2011), The NIST Definition of Cloud Computing (NIST Special Publication 800-145), Computer Security Division, Gaithersburg, USA.

Navarro, A. C., Kahler, H. N., & Mateu, L. (2014). A Survey on Parallel Computing and its Applications in Data-Parallel Problems Using GPU Architectures. *Communications in Computational Physics.* Volume. 15, No. 2.

Nazir, M. (2012). Cloud Computing: Overview & Current Research Challenges. *IOSR Journal of Computer Engineering (IOSR-JCE)* Volume 8, Issue 1.

Nielsen, J. (2000) *Designing Web Usability: the practice of simplicity*. Indianapolis: New Riders.

Odun-Ayo, I., Ajayi, O. & Falade, A. (2018) Cloud Computing and Quality of Service: Issues and Developments. *Proceedings of the International Multi Conference of Engineers and Computer Scientists,* 2018. Vol I IMECS 2018, March 14-16, 2018, Hong Kong.

Owansa, T. & Walubengo, J (2017) Regional Academic Network on IT Policy (RANTIP) –Cloud Computing Research Case of Kenya (Cloud User Perspectives) Nov 2017.

Omwansa, T., Waema, T. & Omwenga, B. (2014). Cloud Computing in Kenya A 2013 Baseline Survey. University of Nairobi, School of Computing and Informatics (SCI) & Computing for Development Lab (C4DLab). http://erepository.uonbi.ac.ke/bitstream/handle/11295/77612/Omwansa%20[at.al]_Cloud%20Computing%20in%20Kenya.pdf?sequence=1

Patel, P., Ranabahu, A. & Sheth, A. (2009). Service level agreement in cloud computing, in*: Cloud Workshops* at OOPSLA09, Orlando, Florida, USA, October 25–29.

Peuhkuri, M (2002). *Internet Traffic Measurements – Aims, Methodology, and Discoveries*. M.Sc., Helsinki University of Technology.

Portnoy, M. (2012). *Virtualization Essentials.* Indianapolis, Indiana: John Wiley & Sons, Inc.

Qi, H. & Gani, A. (2012). Research on Mobile Cloud Computing: Review, Trend and Perspectives: in *Proceedings of the Second International Conference on Digital Information and Communication Technology and its Applications* (DICTAP), IEEE, Pages 195-202.

Ramad, H. H. & Kashyap, D. (2017) Quality of Service (QoS) in Cloud Computing. (IJCSIT) *International Journal of Computer Science and Information Technologies*, Vol. 8 (3).

Rios, E., Mallouli, W., Rak, M., Casola, V. & Ortiz, M. A (2016) SLA-Driven Monitoring of Multi-cloud Application Components Using the MUSA Framework, in *2016 IEEE 36th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, Nara, 2016, pp. 55-60.

Rios, E., Iturbe, E., Mallouli, W. & Rak, M.(2017) Dynamic Security Assurance in Multi-cloud DevOps, *2017 IEEE Conference on Communications and Network Security (CNS)*, Las Vegas, NV, 2017, pp. 467-475.

Rizvi, S., Roddy, H., Gualdoni, J. & Myzyri, L (2017) Three-Step Approach to QoS Maintenance in Cloud Computing Using a Third-Party Auditor, in *Procedia Computer Science* 114 (2017) 83–92, Complex Adaptive Systems Conference with Theme: Engineering Cyber Physical Systems, CAS October 30 – November 1, 2017, Chicago, Illinois, USA.

Salesforce (2019) Trust Status. https://status.salesforce.com [accessed on 13th October 2019].

Saravanan, K. & Kantham, L. (2013). An enhanced QoS Architecture based Framework for Ranking of Cloud Services. *International Journal of Engineering Trends and Technology (IJETT),* Volume 4, Issue 4.

Sanchez-Rola, I., Santos, I., & Balzarotti, D. (2017). Extension Breakdown: Security Analysis of Browsers Extension Resources Control Policies: *Proceedings of the 26th USENIX Security symposium* (2017), Vancouver, BC, Canada.

Serhani, M.A., Atif, Y. & Benharref, A. (2014). Towards an adaptive QoS-driven monitoring of cloud SaaS. *International Journal on Grid and Utility Computing*, Vol. 5, No. 4.

Sosinsky, B (2011). *Cloud Computing Bible*. Indianapolis, Indiana: Wiley Publishing, Inc.

Sommerville, I. (2011). *Software Engineering*, 9th Ed. Boston: Pearson Education, Inc.

Subha, M. & Banhu, U. M. (2014). A Survey on QoS Ranking in Cloud Computing. *International Journal of Emerging Technology and Advanced Engineering.* Volume 4, Issue 2.

Suman, Wadhwa, M. (2014) A Comparative Study of Software Quality Models. *International Journal of Computer Science and Information Technologies*, Vol. 5 (4).

Sukhpuneet, K., Kulwant, K. & Parminder, K.(2016) An Empirical Performance Evaluation of Universities Website. *International Journal of Computer Applications* (0975 – 8887) Volume 146 – No.15, July 2016.

Taha, A. (2018) Quantitative Trust Assessment in the Cloud. MSc., Technische Universität Darmstadt, Germany.

Tamary, J. and Feitelson, D. G. (2015) The rise of Chrome. PeerJ Computer Science 1:e28 https://doi.org/10.7717/peerj-cs.28.

Vetter, R. J., Spell, C., & Ward, C. (1994). Mosaic and the World Wide Web. *IEEE Computer*, Volume: 27, Issue: 10.

Thanh, T. P. T & Helfert, M (2007). A Review of Quality Frameworks in Information Systems *in proceedings of Information Systems Technology and its Applications, 6th International Conference* ISTA'2007, May 23-25, 2007, Kharkiv, Ukraine.

Upadhyay, N. (2017) Managing Cloud Service Evaluation and Selection. *Procedia Computer Science* 122 (2017) 1061–1068.

Vicic, J. & Brodnik, A. (2014) Multiple-cloud platform monitoring. *Elektrotehnis Ki Vestnik* 81(3): 94–100, 2014.

Vidgen, R., Wood, J., & Wood-Harper, A. (1994). Customer satisfaction: The need for multiple perspectives of information system quality. Chance of Receiving Desired Consequences. *Management II,* Vol 1. Computational Mechanics Publications.

Vissk, T. (2010) Recommendations for Using the Case Study Method in International Business Research. *The Qualitative Report*, Volume 15 Number 2 March 2010 370-388.

Vliet, H. V (2007) *Software Engineering: Principles and Practice*. John Wiley & Sons: New Jersy.

Vouk, A. M. (2008). Cloud Computing – Issues, Research and Implementations. *Proceedings of the ITI 2008 30th International Conference on Information Technology Interfaces,* June 23-26, Cavtat, Croatia.

Vidhya, V.(2013) Recent Trends in Cloud Computing: A Survey. *International Journal of Advances in Computer Science and Technology.* Volume 2, No.5.

Vrbanec, T., Kirić, N. & Varga, V. (2013) The evolution of web browser architecture, in M Mokrys, S Badura, A Lieskovsky (Eds.), *SCIENCOF 2013 :Proceedings of The 1st International Virtual Scientific Conference*. Publishing Society: Slovakia.

Vyawahare, D. G., Bende, R. B., Bhajipale, D. N., Bharsakle, R. D. & Salve, A. G. (2016) A Survey on Security Challenges and Solutions in Cloud Computing. *International Journal of Innovative Research in Computer and Communication Engineering.* Vol. 4, Issue 3.

Wong B. & Jeffery R. (2002) A Framework for Software Quality Evaluation. In: Oivo M., Komi-Sirviö S. (eds) Product Focused Software Process Improvement. PROFES 2002. *Lecture Notes in Computer Science*, Vol 2559. Springer, Berlin, Heidelberg.

Yahya, F., Chang, V., Walters, R. J. & Wills, G. B. (2014) Security Challenges in Cloud Storage. In 2014 *IEEE 6th International Conference on Cloud Computing Technology and Science.*

Yefeng R., Ping Z., Lina A. & Arjan D. (2017) Measurement Theory-Based Trust Management Framework for Online Social Communities. *ACM Transactions on Internet Technology*, Vol. 17, No. 2, Article 16, Publication date: March 2017.

Yin, R. (1984). Case study research: *Design and methods* (1st ed.). Beverly Hills, CA: Sage Publishing.

Zachman, J. A (1987) A framework for information systems architecture. *IBM Systems Journal*, Volume 26. No 3, 1987.

Zainab, A., Perry, M. & Capretz, M. A (2011). Trust Metrics for Services and Service Providers *in The Sixth International Conference on Internet and Web Applications and Services* ICIW 2011 March 20-25, 2011 - St. Maarten, The Netherlands Antilles.

Zavol, F., Jung, J. J. & Badica, C. (2013). Intelligent Distributed Computing VII, in *Proceedings of the 7th International Symposium on Intelligent Distributed Computing (IDC), Prague , Czech Republic*. Springer: London.

Zeginis, C., Kritikos, K., Garefalakis, P., Konsolaki, K., Magoutis, K. & Plexousakis, D. (2013) Towards Cross-Layer Monitoring of Multi-Cloud Service-Based Applications, In *Proceedings of the 2nd European Conference on Service-Oriented and Cloud Computing (ESOCC 2013*), Malaga, Spain, 11-13 September 2013.

Zheng, X., Martin, P., Brohman, K. & Xu, L. D. (2014). CLOUDQUAL: A Quality Model for Cloud Services. *IEEE Transactions on Industrial Informatics*, Vol. 10, No. 2.

Zia, A. Naeem, M. & Khan, A. (2012). Identifying Key Challenges in Performance Issues in Cloud Computing. *International Journal of Modern Education and Computer Science*, Volume 10.

Appendix 1: Email Conversations and Chats with Cloud Providers

Extreme Computing Solutions & Salesforce

more details »

Hello Frankline

Thanks a lot for your time today.

This is the invitation for your call with the territory manager **Valerio Forliano on Thursday 12th of September at 10 AM Kenyan Time / 8 AM Irish Time.**

He will call you on this number: **+254 724 528176**

Do not hesitate to revert back to me if you have any question

When	Thu Sep 12, 2019 10am – 10:30am East Africa Time - Nairobi
Joining info	meet.google.com/phn-cgcv-gjx
	Or dial: +353 1 571 2439 PIN: 653470# More phone numbers
Calendar	goldmedalist321@gmail.com
Who	• vbannino@salesforce.com - organizer • goldmedalist321@gmail.com • Valerio Forliano

Going (goldmedalist321@gmail.com)? **Yes** - **Maybe** - **No** more options »

Chat transcript on https://offers.hubspot.com/contact-sales

1 message

Rhys <inboundsaleschat@hubspot.hs-inbox.com> Wed, Oct 30, 2019 at 7:38 AM
To: goldmedalist321@gmail.com

Thanks for chatting!

Here's your chat transcript from https://offers.hubspot.com/contact-sales

Ready to chat software? That's what I'm here for, so don't be shy.

12:03 AM

hi

Ready to chat software? That's what I'm here for, so don't be shy.

12:03 AM

hi

You 12:03 AM

how are you Rhys

You 12:04 AM

Hi Frankline. I'm well thank, and youself?

Rhys 12:05 AM

am good

You 12:06 AM

Good to hear. How can I help you today?

Rhys 12:06 AM

I would like to know suppose HubSpot platform is down /out of service for say a long time , on an issue being resolved , do I get compensated as a client ?

HubSpot has not had such a period of downtime. As per our Customer Terms of Service, we try to make the Subscription Service available 24 hours a day, 7 days a week.

Rhys 12:11 AM

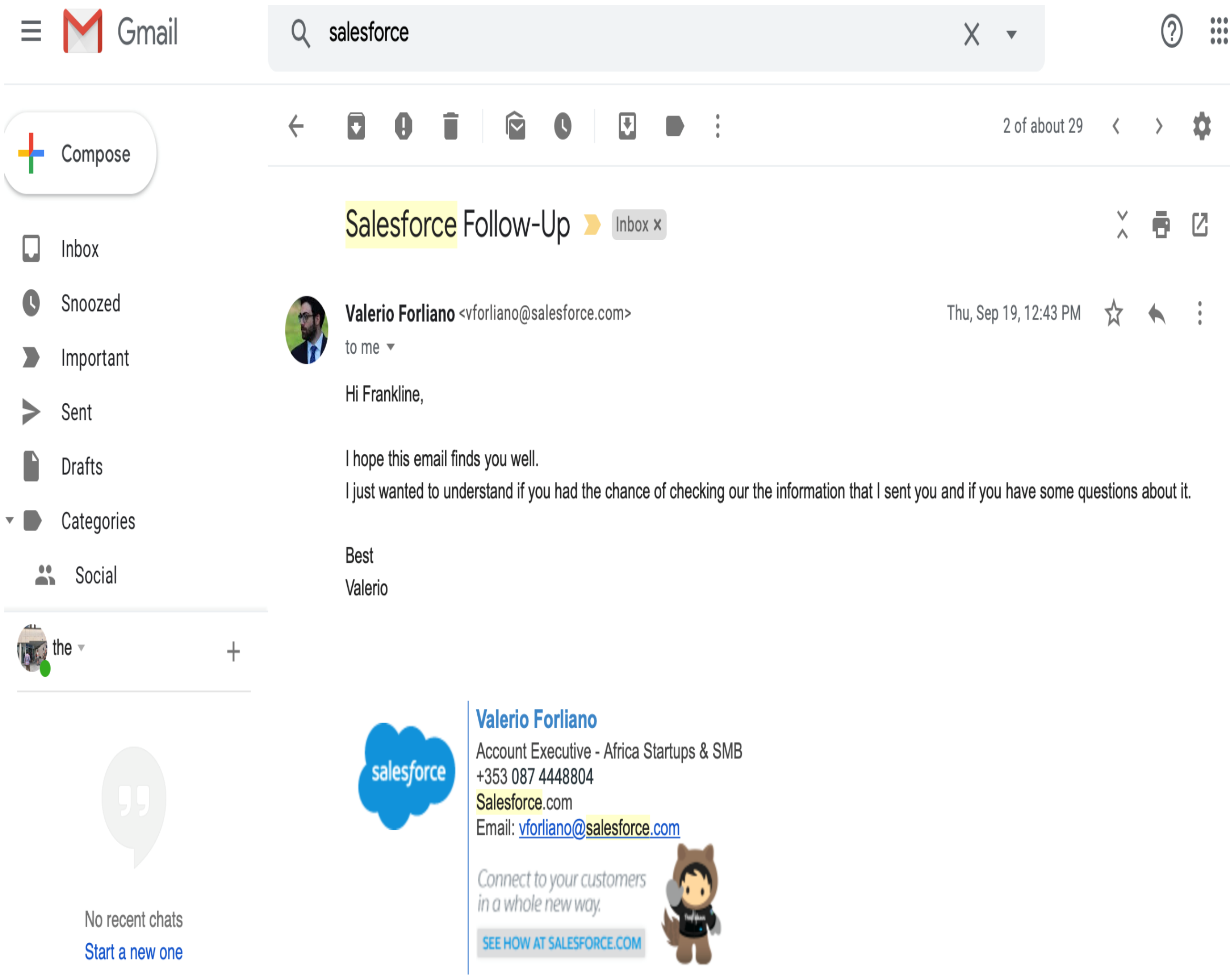
okay thanks , but in the unlikely even that it happens ?will HubSpot compensate ?

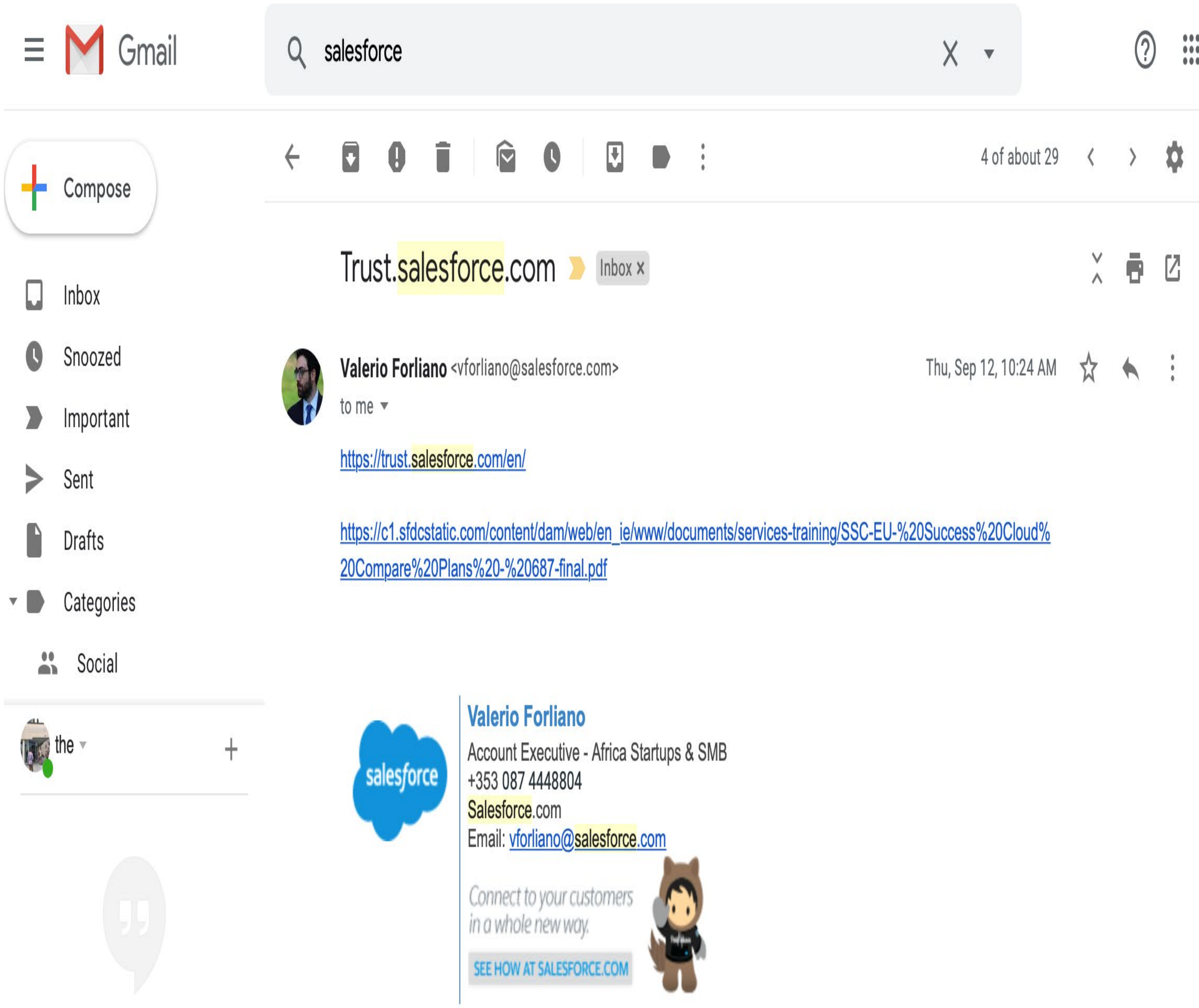
You 12:17 AM

still there Rhys ?

You 12:25 AM

I've been looking into this for you. It appears that there will not be compensation.


Gmail
salesforce
Compose
Inbox
Snoozed
Important
Sent
Drafts
Categories
Social
the
No recent chats
Start a new one
2 of about 29
Salesforce Follow-Up
Inbox x
Valerio Forliano <vforliano@salesforce.com>
to me
Thu, Sep 19, 12:43 PM
Hi Frankline,
I hope this email finds you well.
I just wanted to understand if you had the chance of checking our the information that I sent you and if you have some questions about it.
Best
Valerio
salesforce
Valerio Forliano
Account Executive - Africa Startups & SMB
+353 087 4448804
Salesforce.com
Email: vforliano@salesforce.com
Connect to your customers in a whole new way.
SEE HOW AT SALESFORCE.COM



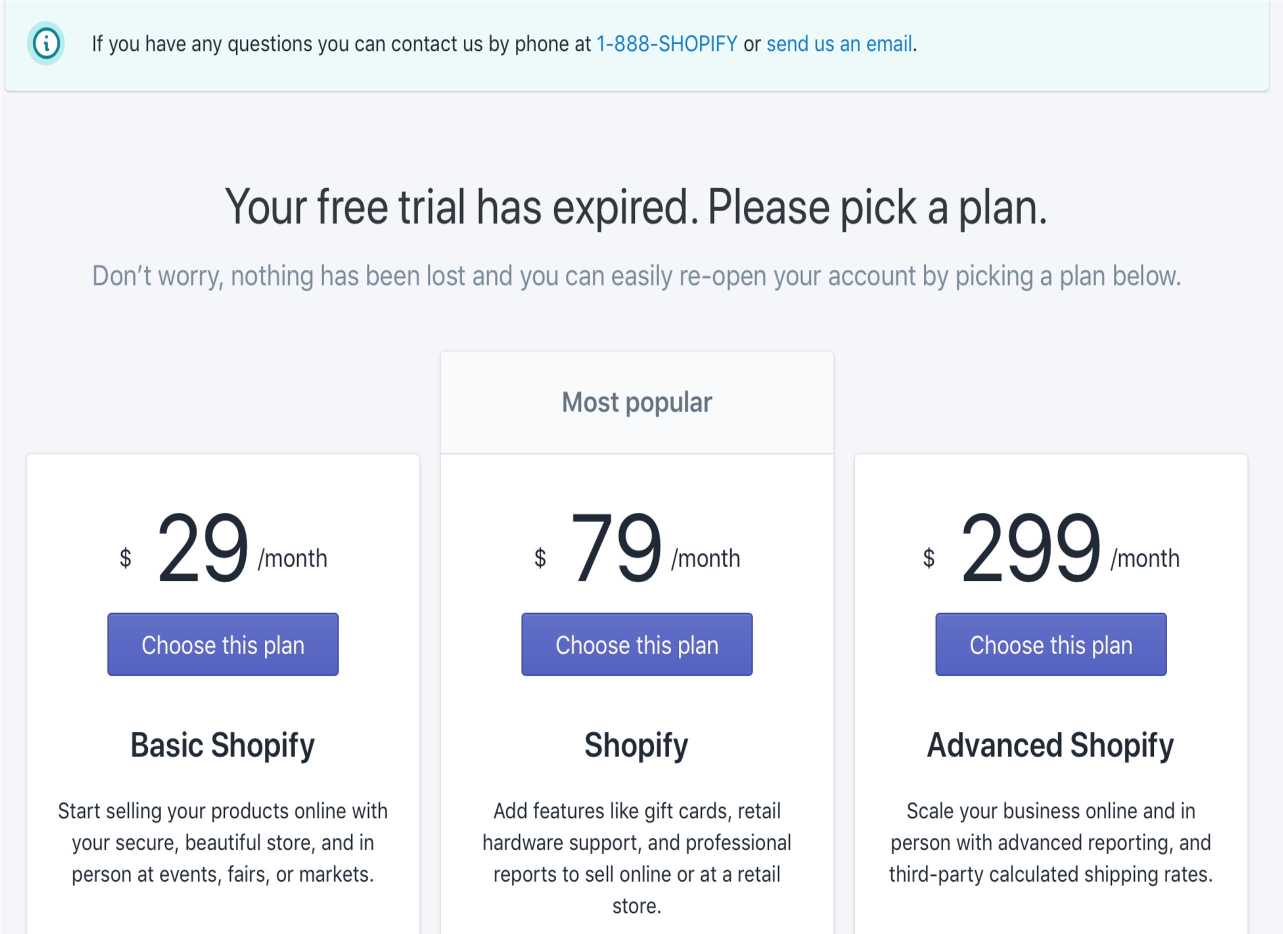
Trust.salesforce.com

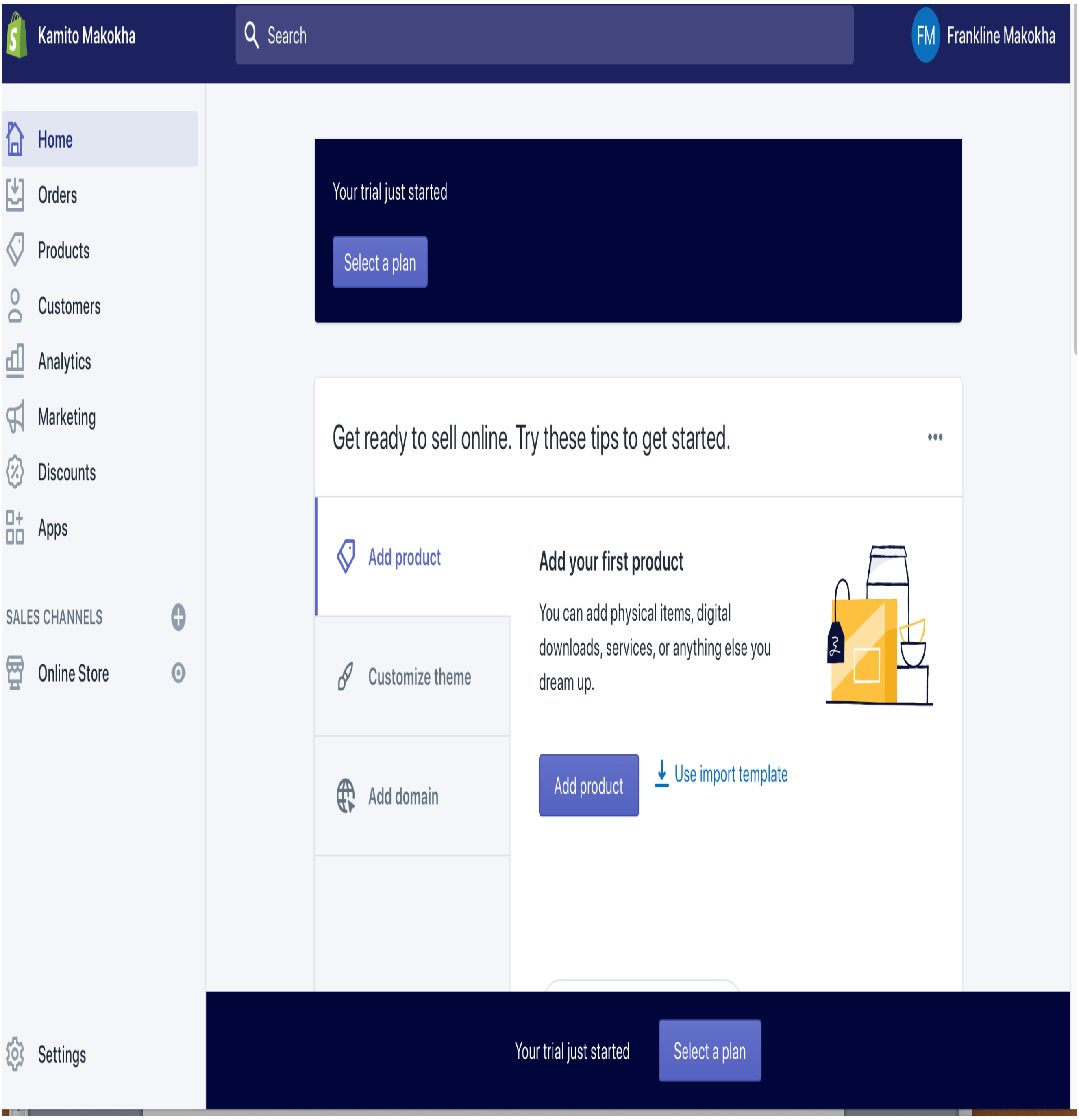
Valerio Forliano <vforliano@salesforce.com>
Thu, Sep 12, 10:24 AM
to me

https://trust.salesforce.com/en/

https://c1.sfdcstatic.com/content/dam/web/en_ie/www/documents/services-training/SSC-EU-%20Success%20Cloud%20Compare%20Plans%20-%20687-final.pdf

Valerio Forliano
Account Executive - Africa Startups & SMB
+353 087 4448804
Salesforce.com
Email: vforliano@salesforce.com


If you have any questions you can contact us by phone at 1-888-SHOPIFY or send us an email.
Your free trial has expired. Please pick a plan.
Don't worry, nothing has been lost and you can easily re-open your account by picking a plan below.
Most popular
$ 29 /month
Choose this plan
Basic Shopify
Start selling your products online with your secure, beautiful store, and in person at events, fairs, or markets.
$ 79 /month
Choose this plan
Shopify
Add features like gift cards, retail hardware support, and professional reports to sell online or at a retail store.
$ 299 /month
Choose this plan
Advanced Shopify
Scale your business online and in person with advanced reporting, and third-party calculated shipping rates.


Kamito Makokha
Search
FM Frankline Makokha
Home
Orders
Products
Customers
Analytics
Marketing
Discounts
Apps
SALES CHANNELS
Online Store
Settings
Your trial just started
Select a plan
Get ready to sell online. Try these tips to get started.
Add product
Customize theme
Add domain
Add your first product
You can add physical items, digital downloads, services, or anything else you dream up.
Add product
Use import template
Your trial just started
Select a plan

Appendix 2: JavaScript Code Snippet for Getting Terminal Specifications

```
chrome.system.cpu.getInfo(function (info) {
  chrome.system.memory.getInfo(function (memory) {
    if (details.reason == "install") {
      var now = new Date();
      $.ajax({
        url: `${portalURL}/api/addClientInstance`,
        data: {
          runtime_id: chrome.runtime.id,
          cpu_numberOfCore: info.numOfProcessors,
          cpu_archName: info.archName,
          cpu_modelName: info.modelName,
          ram_size: memory.capacity,
          date_joined: new Date(now).toISOString()
        },
```

Appendix 3: JavaScript Snippet for getting Internet Connection Parameters

```
$.ajax({
    url: `${portalURL}/api/addSiteMonitored`,
    data: {
      operation_id: _operationId,
      name: mainUrl.split('.')[1],
      mainUrl: mainUrl,
      partialUrl: sender.url.split(mainUrl)[1],
      loadtime: totalTime,
      net_rtt: window.navigator.connection.rtt,
      net_downlink: window.navigator.connection.downlink,
      net_effectiveType: window.navigator.connection.effectiveType
    },
```

Appendix 4: Sample JavaScript Snippet for Monitoring Cloud Platform QoS

```
chrome.runtime.onMessage.addListener(
  function(request, sender, sendResponse) {
    var mainUrl = sender.url.split('//')[1].split('/')[0];

    console.log("Data Sent",request);
    if (urlList.includes(mainUrl)) {
    var totalTime = (request.timing.duration/1000).toFixed(3);
    chrome.storage.local.get("switch", function (data) {
          var _switch = data.switch;
         if(_switch == "_ON_"){
           if(navigator.onLine){

    // console.log("Data Sent",request);
             chrome.storage.local.get("operationId", function (data) {
              var _operationId = data.operationId;

        $.ajax({
          url: `${portalURL}/api/addSiteMonitored`,
          data: {
            operation_id: _operationId,
            name: mainUrl.split('.')[1],
            mainUrl: mainUrl,
            partialUrl: sender.url.split(mainUrl)[1],
            loadtime: totalTime,
            net_rtt: window.navigator.connection.rtt,
            net_downlink: window.navigator.connection.downlink,
            net_effectiveType: window.navigator.connection.effectiveType
          },
```

```
        },
        dataType: "json",
        type: "post",
        success: function (data) {
          var notifOptions = {
            type: "basic",
            iconUrl: "static/icon.png",
            title: "Cloud QoS Monitor: "+ mainUrl +"[Complete]",
            message: totalTime + " seconds"
          };
        chrome.notifications.create((Math.floor(Math.random() * 90000) + 10000).toString(), notifOptions,

        });
        },
        error: function (e) {
          chrome.storage.local.set({ switch: "_OFF_" });
          alert("Problem Connecting to QoS Server");
        }
      });
    });
  } else{
        var notifOptions3 = {
          type: "basic",
          iconUrl: "static/icon.png",

          title: "Network Error",
          message: "Your Computter is offline"
        };
        chrome.notifications.create((Math.floor(Math.random() * 90000) + 10000).toString(), notifOptions3
          chrome.storage.local.set({ switch: "_OFF_" });

      });
    }
  }
});
```

```
// START monitoring proccess
startMonitoringSession = function () {
  // Get clint id from local storage
  chrome.storage.local.get("clientId", function (data) {
    var _clientId = data.clientId;

    // Create new operation id
    var _operationId = guid();
    var now = new Date();

    // Save monitoring session in the web api using AJAX call
    $.ajax({
      url: `${portalURL}/api/addMonitingLog`,
      data: {
        operation_id: _operationId,
        client_id: _clientId,
        start_timestamp: new Date(now).toISOString()
      },
      dataType: "json",
      type: "post",
      success: function (data) {
        //On request successful check server responce: SUCCESS/ERROR
        if (data.message == "SUCCESS") {
// console.log(data);
          // Store created operation id to chrome local storage
          chrome.storage.local.set({
            operationId: _operationId
          });
```

```
                urlList = [];
                for(var i = 0; i < data.urls.length; i++){
                  urlList.push(data.urls[i].mainURL);
                  uniqiueStingList.push(data.urls[i].partialURL);
                }
              } else if(data.message == "NOURL"){

                chrome.storage.local.set({ switch: "_OFF_" });
                alert("Please Add URL to Be Monitored");

              } else if(data.message == "ERROR"){

                chrome.storage.local.set({ switch: "_OFF_" });
                alert("Problem Connecting to QoS Server");
              }
            },
            error: function (e) {
              chrome.storage.local.set({ switch: "_OFF_" });

              alert("Problem Connecting to QoS Server");
            }
          });
        });
      };
```

```
stopMonitoringSession = function () {
  chrome.storage.local.get("operationId", function (data) {
    var _operationId = data.operationId;
    var now = new Date();
    $.ajax({
      url: `${portalURL}/api/updateMonitorStopStauts/${_operationId}`,
      data: {
        stop_timestamp: new Date(now).toISOString()
      },
      type: "get",
      success: function (data) {
        chrome.storage.local.set({
          operationId: "NULL"
        });
      }
    });
  });
};
```

Appendix 5: Sample Raw QoS Monitoring Results For All Cloud QoS Platforms

QoS Cloud Monitoring

Home / Monitoring Sessions / Sites Monitored

ALL MONITORED SITES

For **ACTIVE** monitoring sessions, evry navigation done within the saved URL are monitored and logged. The information stored is the network status, URL info and the loading time. This data is for client with the ID, **0AEDDAC0F69D**

The table below is a list of all sites monitoired by this client

PRINT PDF

Search:

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
5A30D6F0F69D	888dc8c7ff89	Sep 14th 06:17	www.shopify.com	/	3.193	250	1.7	4g	PDF
6134F3A0F69D	888dc8c7ff89	Sep 14th 06:17	www.shopify.com	/market	Infinity	250	1.7	4g	PDF
B3B4A440F69D	c51f9e842777	Sep 14th 06:19	docs.google.com	/document/d/1cE9DQ2NNqtRcYglvsUQhUBf9uDNzbrAN-049uC1mUaQ/edit	3.366	250	8.5	4g	PDF
D49B3020F69D	079693d1d21b	Sep 14th 06:20	docs.google.com	/document/d/1cE9DQ2NNqtRcYglvsUQhUBf9uDNzbrAN-049uC1mUaQ/edit	4.832	250	8	4g	PDF
32DD21C0F69E	cf6ff77c8bdb	Sep 14th 06:23	www.shopify.com	/	2.054	300	5.8	4g	PDF
52566200F69E	fc5303dfbc2d	Sep 14th 06:24	docs.google.com	/document/d/1cE9DQ2NNqtRcYglvsUQhUBf9uDNzbrAN-049uC1mUaQ/edit	4.362	250	5.3	4g	PDF
63970A10F69E	85605fdcb7b5	Sep 14th 06:24	www.shopify.com	/online	1.746	250	4.9	4g	PDF
655FE330F69E	85605fdcb7b5	Sep 14th 06:24	www.shopify.com	/online	2.559	250	4.9	4g	PDF
678422C0F69E	85605fdcb7b5	Sep 14th 06:24	www.shopify.com	/email-marketing	2.261	250	4.9	4g	PDF
6C486190F69E	85605fdcb7b5	Sep 14th 06:24	www.shopify.com	/sell	1.774	250	4.9	4g	PDF
7F785680F69E	c206a5d3e782	Sep 14th 06:25	www.shopify.com	/examples	2.257	250	5	4g	PDF
8197C4D00799	cf45c07d1dec	Oct 6th 09:02	www.office.com	/?auth=1	1.998	450	1.4	3g	PDF
A33F7DD00799	3646b9c9ce92	Oct 6th 09:03	docs.google.com	/document/d/18VTjqWvvwkwiosam_OhjeE7SwGuR7fTdMuUI8eTCO2s/edit	5.132	250	6.7	4g	PDF
F524EA400799	2ca65a45c041	Oct 6th 09:05	www.office.com	/launch/word?auth=1	1.821	200	10	4g	PDF
102377D0079A	f43ea98a74c8	Oct 6th 09:06	docs.google.com	/document/d/11_WnF0sTPtEnv6NJJx38cuBVH3uQIVvSaj4QQfML78Y/edit	7.357	200	10	4g	PDF

Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Search

Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
3F533DB0079A	88d5f3ccb091	Oct 6th 09:07	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!163&ithint=file%2cdocx&action=editnew&wdNewAndOpenCt=1601964342348&wdPreviousSession=4c87d3db-cb2e-446f-bb48-55da84995f60&wdOrigin=OFFICECOM-WEB.START.NEW	6.898	200	9.9	4g	PD
50BF1470079A	24a91746e108	Oct 6th 09:08	docs.google.com	/document/d/1ywVmZ4el3nkrc4wajJLUFqx-qy8Q8slUDnkZVXq9zcQ/edit	4.624	250	10	4g	PD
56861D40079A	24a91746e108	Oct 6th 09:08	www.office.com	/launch/word?auth=1	3.54	250	10	4g	PD
8A3EC6A0079A	6f2bc5e28de2	Oct 6th 09:09	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!168&ithint=file%2cdocx&action=editnew&wdTpl=TM00002109&wdlcid=2057&wdNewAndOpenCt=1601964509077&wdPreviousSession=8e968b31-ea05-4b5a-9d0b-457e0e23a5d7&wdOrigin=OFFICECOM-WEB.START.TEMPLATES	9.336	250	10	4g	PD
B332EFF0079A	c16a2931adcb	Oct 6th 09:10	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!168&ithint=file%2cdocx&action=editnew&wdTpl=TM00002109&wdlcid=2057&wdNewAndOpenCt=1601964509077&wdPreviousSession=8e968b31-ea05-4b5a-9d0b-457e0e23a5d7&wdOrigin=OFFICECOM-WEB.START.TEMPLATES	11.374	300	1.55	3g	PD
D95BEB50079A	b437d06c138b	Oct 6th 09:12	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!170&ithint=file%2cxlsx&action=editnew&wdNewAndOpenCt=1601964700190&wdPreviousSession=0f966f46-c478-4fd0-ae6c-6fb1e12e91a1&wdOrigin=OFFICECOM-WEB.MAIN.NEW	3.56	200	7.9	4g	PD
DF169810079A	b437d06c138b	Oct 6th 09:12	docs.google.com	/document/u/0/	1.899	200	7.9	4g	PD
1BFB4730079B	5d8181e126e1	Oct 6th 09:13	docs.google.com	/spreadsheets/d/1onHQau1lFLStma2oRuftPt01l-beVJPz1A5QmCJfJwo/edit#gid=0	5.363	150	5.5	4g	PD

Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
2E995C60079B	f04a668a15cf	Oct 6th 09:14	docs.google.com	/spreadsheets/d/1onHQau1lFLStma2oRuftPt01l-beVJPz1A5QmCJfJwo/edit#gid=0	9.458	200	4.95	4g	PD
C02CAA10079B	dfd42b53040e	Oct 6th 09:18	docs.google.com	/presentation/d/1Udgv90jnfw83N7sBgj8S4dznRgiZ1PtFDSQG3UterfE/edit#slide=id.p	8.742	650	1.55	3g	PD
C0BDED90079B	dfd42b53040e	Oct 6th 09:18	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!173&ithint=file%2cpptx&action=editnew&wdNewAndOpenCt=1601965061950&wdPreviousSession=6bd0b728-95d4-4acb-8a3a-90fe91159f72&wdOrigin=OFFICECOM-WEB.START.NEW	42.397	500	1.55	3g	PD
F9442490079B	a0542e958265	Oct 6th 09:20	onedrive.live.com	/edit.aspx?action=edit&resid=6937FA95C77474EF!173&ithint=file%2cpptx&action=editnew&wdNewAndOpenCt=1601965061950&wdPreviousSession=6bd0b728-95d4-4acb-8a3a-90fe91159f72&wdOrigin=OFFICECOM-WEB.START.NEW	13.339	250	4.6	4g	PD
03282FB0079C	3e6418d52140	Oct 6th 09:20	docs.google.com	/presentation/d/1Udgv90jnfw83N7sBgj8S4dznRgiZ1PtFDSQG3UterfE/edit#slide=id.p	7.364	250	1.35	3g	PD
8FED5380079C	51da551264d9	Oct 6th 09:24	onedrive.live.com	/edit.aspx?action=edit&resid=6937FA95C77474EF!173&ithint=file%2cpptx&action=editnew&wdNewAndOpenCt=1601965061950&wdPreviousSession=6bd0b728-95d4-4acb-8a3a-90fe91159f72&wdOrigin=OFFICECOM-WEB.START.NEW	8.919	250	1.4	4g	PD
98BD2F80079C	757d34695d2b	Oct 6th 09:24	docs.google.com	/presentation/d/1Udgv90jnfw83N7sBgj8S4dznRgiZ1PtFDSQG3UterfE/edit#slide=id.p	4.415	250	1.4	4g	PD
DC50D670079C	f3fd91666993	Oct 6th 09:26	docs.google.com	/forms/u/0/d/1cX2HGCkOFtq8r36J-reJLw6vfxnsfDDJnBuhiKr2l5M/edit	2.448	200	5.1	4g	PD
DD3A7320079C	f3fd91666993	Oct 6th 09:26	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!176&cid=6937fa95c77474ef	4.615	200	5.1	4g	PD

Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Search

Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
02CB2620079D	7570908b255e	Oct 6th 09:27	onedrive.live.com	/Edit.aspx?resid=6937FA95C77474EF!176&wd=target(Research%20notes.one%7C5e0ca259-dcdb-4f52-9004-6f417c26d1d5/Sample%20Research%20Notes%7C494f65eb-bd10-48c3-9d88-48e11208c185/)	3.727	200	5.1	4g	PD
AB9451E00A14	c42d3a4ed976	Oct 9th 12:49	www.office.com	/?auth=1	11.067	200	5.55	4g	PD
B06FC2300A14	c42d3a4ed976	Oct 9th 12:49	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!173&cid=364dba8f-5284-45ab-88ba-e63e017e4f06&ithint=file%2cpptx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	8.285	200	5.55	4g	PD
BA8871E00A14	c42d3a4ed976	Oct 9th 12:49	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!170&cid=256528b1-4fdc-422f-9333-96a2e18dd392&ithint=file%2cxlsx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	5.398	200	5.55	4g	PD
BC8468F00A14	c42d3a4ed976	Oct 9th 12:49	docs.google.com	/document/u/0/	6.58	200	5.55	4g	PD
C3B2C9F00A14	c42d3a4ed976	Oct 9th 12:49	docs.google.com	/document/d/1QZv7JCJSJBKK_WwBeKBjH3ogyq4eGw1DTCVFM3oFp78/edit	6.904	200	5.55	4g	PD
E9A1C9900A14	a0525ddc05de	Oct 9th 12:50	docs.google.com	/spreadsheets/d/12SHUOcKsv89wZpZRnygf6YuvQzJ3Pi7LwcuF35BHZAY/edit#gid=1386834576	3.851	200	6.05	4g	PD
ED9A2CE00A14	a0525ddc05de	Oct 9th 12:50	docs.google.com	/spreadsheets/u/0/	3.368	200	6.05	4g	PD
EFB1E9500A14	a0525ddc05de	Oct 9th 12:51	docs.google.com	/forms/u/0/	2.175	200	6.05	4g	PD
F32ABC600A14	a0525ddc05de	Oct 9th 12:51	docs.google.com	/forms/u/0/	1.203	200	6.05	4g	PD
F8669C300A14	a0525ddc05de	Oct 9th 12:51	docs.google.com	/presentation/u/0/	3.159	200	6.05	4g	PD
FF6DC6200A14	a0525ddc05de	Oct 9th 12:51	docs.google.com	/presentation/d/1zUNnO8MuceatabZwBb0opPdXZyZXQJiCODHNZ7ilw3w/edit	8.466	200	5.85	4g	PD
2BC7FDD00A15	240fb9d1c5ff	Oct 9th 12:52	www.office.com	/?auth=1	2.019	200	5.9	4g	PD
32CFEB100A15	240fb9d1c5ff	Oct 9th 12:52	www.office.com	/launch/powerpoint?auth=1	1.553	200	5.9	4g	PD

Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
39A704500A15	240fb9d1c5ff	Oct 9th 12:53	onedrive.live.com	/edit.aspx?action=edit&resid=6937FA95C77474EF!184&ithint=file%2cpptx&action=editnew&wdTpl=TM16401370&wdlcid=2057&wdNewAndOpenCt=1602237175097&wdPreviousSession=d907cc97-9900-4013-b250-89a2b4cfe125&wdOrigin=OFFICECOM-WEB.START.TEMPLATES	7.119	200	5.9	4g	PDF
419050E00A15	240fb9d1c5ff	Oct 9th 12:53	docs.google.com	/presentation/u/0/	3.272	250	3.55	4g	PDF
446D32100A15	240fb9d1c5ff	Oct 9th 12:53	docs.google.com	/document/u/0/	3.439	250	3.55	4g	PDF
49F313300A15	240fb9d1c5ff	Oct 9th 12:53	docs.google.com	/document/d/1v4G5ljoMFWOhABEkzt6MJMJrbgJ9WuQtJc52DmvIQvg/edit	5.501	250	3.55	4g	PDF
2A2D6B300A16	af129825903b	Oct 9th 12:59	www.office.com	/?auth=1	1.064	150	8.05	4g	PDF
2DD194000A16	af129825903b	Oct 9th 12:59	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!170&cid=256528b1-4fdc-422f-9333-96a2e18dd392&ithint=file%2cxlsx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	3.345	150	8.05	4g	PDF
332E69500A16	af129825903b	Oct 9th 01:00	www.office.com	/launch/word?auth=1	1.609	150	8.05	4g	PDF
36B964D00A16	af129825903b	Oct 9th 01:00	docs.google.com	/document/d/1v4G5ljoMFWOhABEkzt6MJMJrbgJ9WuQtJc52DmvIQvg/edit	5.844	150	8.05	4g	PDF
A821EC900AC6	f80f70763b2d	Oct 10th 10:03	www.office.com	/?auth=1	27.321	400	1.25	3g	PDF
BCD8CFA00AC6	9465708b5cad	Oct 10th 10:03	docs.google.com	/document/d/1xpfXYvxqKNW_iURcMG1h4ppOKmTxpvXrIVDhXqziZP0/edit	9.246	450	1.55	3g	PDF
BF8774E00AC6	9465708b5cad	Oct 10th 10:03	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!170&cid=3941edd8-df60-426f-b701-d019ceae991e&ithint=file%2cxlsx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	6.767	450	1.55	3g	PDF
D72D3A800AC6	1fb10dc1bd8f	Oct 10th 10:04	docs.google.com	/document/d/1xpfXYvxqKNW_iURcMG1h4ppOKmTxpvXrIVDhXqziZP0/edit	5.263	400	1.35	3g	PDF
DC99EE500AC6	1fb10dc1bd8f	Oct 10th 10:04	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!170&cid=3941edd8-df60-426f-b701-d019ceae991e&ithint=file%2cxlsx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	5.736	450	1.35	3g	PDF

Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
F7AA56300AC6	0d70a221e65a	Oct 10th 10:05	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!173&cid=23d4a9a6-a8db-4e23-9a3c-1eac379919d8&ithint=file%2cpptx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	9.444	550	1.5	3g	PD
04A761200AC7	0d70a221e65a	Oct 10th 10:05	docs.google.com	/document/d/1xpfXYvxqKNW_iURcMG1h4ppOKmTxpvXrIVDhXqziZP0/edit	4.366	400	1.5	3g	PD
09240FA00AC7	0d70a221e65a	Oct 10th 10:05	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!170&cid=3941edd8-df60-426f-b701-d019ceae991e&ithint=file%2cxlsx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	4.074	400	1.5	3g	PD
0FC9DFB00AC7	0d70a221e65a	Oct 10th 10:06	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!173&cid=23d4a9a6-a8db-4e23-9a3c-1eac379919d8&ithint=file%2cpptx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	9.452	400	1.5	3g	PD
37ABFCC00AC7	9fd323027584	Oct 10th 10:07	docs.google.com	/presentation/d/12JDdFn4aJzG-zcW8T79LlCuEJaKRByx7DHcuXTaHukU/edit#slide=id.p	5.173	250	1.3	3g	PD
3BBC7BF00AC7	9fd323027584	Oct 10th 10:07	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!170&cid=3941edd8-df60-426f-b701-d019ceae991e&ithint=file%2cxlsx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	3.64	250	2.55	4g	PD
41FA89800AC7	9fd323027584	Oct 10th 10:07	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!173&cid=23d4a9a6-a8db-4e23-9a3c-1eac379919d8&ithint=file%2cpptx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	7.164	250	2.55	4g	PD
4793CB900AC7	9fd323027584	Oct 10th 10:07	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!168&cid=2b15d1c6-d630-4783-acd1-15330c379e7e&ithint=file%2cdocx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	3.447	250	2.55	4g	PD
B0E111500E98	630d65ee6536	Oct 15th 06:44	docs.google.com	/document/u/0/	9.099	50	10	4g	PD
B177AC000E98	630d65ee6536	Oct 15th 06:44	www.office.com	/?auth=1	7.18	50	10	4g	PD
CF6591500E98	8993a1f3ff96	Oct 15th 06:45	www.office.com	/?auth=1	1.241	50	10	4g	PD

Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
D372B5200E98	8993a1f3ff96	Oct 15th 06:45	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!173&cid=30285e96-f6c2-4e20-905d-a378102c444f&ithint=file%2cpptx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	9.489	50	10	4g	PDF
E6E7C9600E98	fe1b764005e8	Oct 15th 06:45	docs.google.com	/document/d/1b-59QrMACbVKpJ7b9ZyAbsQZfT3Yg0jJQkAi6Bi8_YI/edit	4.993	50	10	4g	PDF
EA9F7A300E98	fe1b764005e8	Oct 15th 06:45	docs.google.com	/document/u/0/	4.281	50	10	4g	PDF
EF7963E00E98	fe1b764005e8	Oct 15th 06:45	docs.google.com	/spreadsheets/u/0/	2.394	50	10	4g	PDF
F3BE12700E98	fe1b764005e8	Oct 15th 06:46	docs.google.com	/spreadsheets/u/0/d/1leTsFkH7-JeselHJhMTuXk140xPOM5D18vRY678z5V4/edit?ntd=1&usp=sheets_home&ths=true	2.202	50	10	4g	PDF
F80706C00E98	fe1b764005e8	Oct 15th 06:46	docs.google.com	/spreadsheets/u/0/	2.492	50	10	4g	PDF
FAE128D00E98	fe1b764005e8	Oct 15th 06:46	docs.google.com	/spreadsheets/u/0/d/1UC9bfvGysnVwX1q1j1YwhJQGM5mtsJdoa8OmRV7L7C0/edit?ntd=1&usp=sheets_home&ths=true	2.052	50	10	4g	PDF
FE8D67F00E98	fe1b764005e8	Oct 15th 06:46	docs.google.com	/spreadsheets/u/0/	2.299	50	10	4g	PDF
02E502400E99	fe1b764005e8	Oct 15th 06:46	docs.google.com	/spreadsheets/d/1SvOd4lWNlMa42UqOBZdWbZGi-lsrhPEJCtBlkZgSSK0/edit	5.108	50	10	4g	PDF
22D520300E99	bb1fc7af9134	Oct 15th 06:47	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!173&cid=30285e96-f6c2-4e20-905d-a378102c444f&ithint=file%2cpptx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	10.952	50	10	4g	PDF
45E9EE00111B	b8bd2cb5a0e0	Oct 18th 11:23	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!168&cid=60d8288c-07d9-4577-a768-9bf92f08707b&ithint=file%2cdocx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	3.376	50	9.95	4g	PDF
4B6C73C0111B	b8bd2cb5a0e0	Oct 18th 11:24	docs.google.com	/document/u/0/	3.528	50	9.95	4g	PDF
51A43FC0111B	b8bd2cb5a0e0	Oct 18th 11:24	docs.google.com	/document/d/1xyy7-8R5zl5RlVIcJb_XWHV_gQXjxQF9hrHb1LSosMs/edit	8.674	50	9.95	4g	PDF
51C81B70111B	b8bd2cb5a0e0	Oct 18th 11:24	www.office.com	/launch/powerpoint?auth=1	2.006	50	9.95	4g	PDF

Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
598527E0111B	b8bd2cb5a0e0	Oct 18th 11:24	onedrive.live.com	/edit.aspx?action=edit&resid=6937FA95C77474EF!186&ithint=file%2cpptx&action=editnew&wdNewAndOpenCt=1603009463555&wdPreviousSession=4513434b-29c8-460c-8099-e6247f5844dd&wdOrigin=OFFICECOM-WEB.START.NEW	7.046	100	10	4g	PD
5E82C630111B	b8bd2cb5a0e0	Oct 18th 11:24	docs.google.com	/document/u/0/	4.16	100	10	4g	PD
61004950111B	b8bd2cb5a0e0	Oct 18th 11:24	docs.google.com	/presentation/u/0/	3.196	100	10	4g	PD
67E19BC0111B	b8bd2cb5a0e0	Oct 18th 11:24	docs.google.com	/presentation/d/1VamdAAy7nlxhrQ_bZ83cfJE1sEhNid5J0G63i43j7Gk/edit	7.86	100	10	4g	PD
9204A820111B	c38fe58383c7	Oct 18th 11:26	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!188&ithint=file%2cxlsx&action=editnew&wdTpl=TM16400654&wdlcid=2057&wdNewAndOpenCt=1603009518305&wdPreviousSession=744f77d4-06a8-4f58-9e66-e662190a3e77&wdOrigin=OFFICECOM-WEB.START.TEMPLATES	10.693	100	10	4g	PD
7052263011CA	8b66babb8ae4	Oct 19th 08:17	docs.google.com	/document/u/0/	4.136	300	1.3	3g	PD
7213854011CA	8b66babb8ae4	Oct 19th 08:17	www.office.com	/?auth=1	4.426	300	1.3	3g	PD
75FC312011CA	8b66babb8ae4	Oct 19th 08:18	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!188&cid=68048cc8-b9ef-4314-8448-a53e9ea5680a&ithint=file%2cxlsx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	8.717	300	1.3	3g	PD
8FB98C2011CA	87dfdf092f35	Oct 19th 08:18	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!188&cid=68048cc8-b9ef-4314-8448-a53e9ea5680a&ithint=file%2cxlsx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	4.299	300	1.25	3g	PD
953B9CB011CA	87dfdf092f35	Oct 19th 08:18	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!168&cid=cb0215bc-c75a-407c-adf1-c4bc8147cba1&ithint=file%2cdocx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	2.519	300	1.25	3g	PD
97DD97C011CA	87dfdf092f35	Oct 19th 08:18	docs.google.com	/document/d/1A_DgG80LSuMOiW-TNvClgSWxjXOjNf9bJ-b1-UV180E/edit	6.617	300	1.25	3g	PD

Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
9F985A4011CA	87dfdf092f35	Oct 19th 08:19	docs.google.com	/document/d/1A_DgG80LSuMOiW-TNvCl9SWxjXOjNf9bJ-b1-UV180E/edit	4.828	300	1.25	3g	PDF
A686F32011CA	87dfdf092f35	Oct 19th 08:19	docs.google.com	/presentation/u/0/	3.994	300	1.25	3g	PDF
B864024011CA	f749519e0498	Oct 19th 08:19	www.office.com	/launch/forms?auth=1	1.021	300	1.4	3g	PDF
BCC5127011CA	f749519e0498	Oct 19th 08:20	docs.google.com	/presentation/d/14wsZ6VyyBh5HfrRYR6eImXH_Lbade3T6pSR1n59sA-0/edit	4.868	300	1.4	3g	PDF
057E521011CB	47d0332c43ff	Oct 19th 08:22	www.office.com	/?auth=1	1.067	300	1.45	3g	PDF
0C7883B011CB	47d0332c43ff	Oct 19th 08:22	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!188&cid=68048cc8-b9ef-4314-8448-a53e9ea5680a&ithint=file%2cxlsx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	7.686	300	1.45	3g	PDF
0E5CD41011CB	47d0332c43ff	Oct 19th 08:22	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!168&cid=cb0215bc-c75a-407c-adf1-c4bc8147cba1&ithint=file%2cdocx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	7.828	300	1.45	3g	PDF
127FC9D011CB	47d0332c43ff	Oct 19th 08:22	www.office.com	/launch/forms?auth=1	2.674	300	1.45	3g	PDF
1845E84011CB	47d0332c43ff	Oct 19th 08:22	docs.google.com	/presentation/d/14wsZ6VyyBh5HfrRYR6eImXH_Lbade3T6pSR1n59sA-0/edit#slide=id.p	5.233	300	1.45	3g	PDF
1B6A80D011CB	47d0332c43ff	Oct 19th 08:22	docs.google.com	/presentation/u/0/	4.186	300	1.45	3g	PDF
1DD7B04011CB	47d0332c43ff	Oct 19th 08:22	docs.google.com	/forms/u/0/	2.222	300	1.45	3g	PDF
2568558011CB	47d0332c43ff	Oct 19th 08:22	docs.google.com	/forms/u/0/d/1-y0-heNVY1APZTGzSkUD3NoDJDyjPtWRWM-5RI6qMkY/edit?ntd=1&usp=forms_home&ths=true	4.163	300	1.45	3g	PDF
4D95646014E3	03cb3a27a5ef	Oct 23rd 06:53	www.office.com	/?auth=1	4.646	250	10	4g	PDF
6698E54014E3	b3ec2311384b	Oct 23rd 06:54	docs.google.com	/document/d/1GZgKt-GpaYRksXxOXcSaxdY3RcuIfEcd0FMTqN41Xk0/edit	4.543	50	10	4g	PDF
7A0F802014E3	fa1fa4965b69	Oct 23rd 06:54	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!168&cid=5c488221-7f33-465e-84a5-e7c33e365f23&ithint=file%2cdocx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	4.292	50	10	4g	PDF
81D8C19014E3	fa1fa4965b69	Oct 23rd 06:54	www.office.com	/launch/word?auth=1	2.582	50	10	4g	PDF
86CBD89014E3	fa1fa4965b69	Oct 23rd 06:55	docs.google.com	/document/u/0/	3.665	50	10	4g	PDF
8960B44014E3	fa1fa4965b69	Oct 23rd 06:55	docs.google.com	/presentation/u/0/	3.338	50	10	4g	PDF
Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
9216518014000	fa1fa4965b69	Oct 23rd 06:55	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!190&ithint=file%2cxlsx&action=editnew&wdNewAndOpenCt=1603425315598&wdPreviousSession=6fd51b49-588b-4366-97bf-d7db52e24d41&wdOrigin=OFFICECOM-WEB.START.NEW	3.006	50	10	4g	PD
9A9A577014E3	fa1fa4965b69	Oct 23rd 06:55	docs.google.com	/presentation/d/1nJXEdRi0eWUWpPFpo0Vy7dfmumK3p3HFz69EtGlnMWY/edit	6.368	50	10	4g	PD
C0844E0014E3	80a6cee29d7f	Oct 23rd 06:56	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!192&ithint=file%2cxlsx&action=editnew&wdNewAndOpenCt=1603425373873&wdPreviousSession=13321c98-4f94-49ac-b9e3-b1ec9c8e9766&wdOrigin=OFFICECOM-WEB.START.NEW	6.527	50	10	4g	PD
F4C0583014E3	28fc76e03a46	Oct 23rd 06:58	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!192&ithint=file%2cxlsx&action=editnew&wdNewAndOpenCt=1603425373873&wdPreviousSession=13321c98-4f94-49ac-b9e3-b1ec9c8e9766&wdOrigin=OFFICECOM-WEB.START.NEW	5.634	50	10	4g	PD
FD439AD014E3	28fc76e03a46	Oct 23rd 06:58	docs.google.com	/spreadsheets/u/0/d/1VoKsamkXURNVgXATtB6pHol7UNj90Gpdor54bh1EANQ/edit?ntd=1&usp=sheets_home&ths=true	2.192	50	10	4g	PD
00CF329014E4	28fc76e03a46	Oct 23rd 06:58	docs.google.com	/spreadsheets/u/0/	2.682	50	10	4g	PD
033B298014E4	28fc76e03a46	Oct 23rd 06:58	docs.google.com	/spreadsheets/u/0/create?usp=sheets_home&ths=true	1.961	50	10	4g	PD
05E8482014E4	28fc76e03a46	Oct 23rd 06:58	www.office.com	/?auth=1	1.269	50	10	4g	PD
0C925DF014E4	28fc76e03a46	Oct 23rd 06:58	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!168&cid=32c9ae99-4e05-4acb-a12a-cdd6915e0aa9&ithint=file%2cdocx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	6.124	50	10	4g	PD

Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
12D32AA014E4	28fc76e03a46	Oct 23rd 06:58	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!194&ithint=file%2cpptx&action=editnew&wdTpl=TM16401379&wdlcid=2057&wdNewAndOpenCt=1603425530623&wdPreviousSession=c9f0efd4-7211-49dd-bc7d-3bfbe9057729&wdOrigin=OFFICECOM-WEB.START.TEMPLATES	4.495	50	10	4g	PDF
26AD424014E4	28fc76e03a46	Oct 23rd 06:59	docs.google.com	/spreadsheets/u/0/	2.906	50	10	4g	PDF
3F3C085014E4	8afd8e25db83	Oct 23rd 07:00	docs.google.com	/document/d/1Ttj4bVOKseE5R1wCGqr1TnPfpioXqdNsYnEGADwkKPA/edit	5.1	0	10	4g	PDF
41AD085014E4	8afd8e25db83	Oct 23rd 07:00	www.office.com	/?auth=1	1.506	0	10	4g	PDF
502B0DF014E4	8afd8e25db83	Oct 23rd 07:00	www.office.com	/?auth=1	1.094	0	10	4g	PDF
D58AE9A01752	b9e894cbd74e	Oct 26th 09:16	docs.google.com	/document/d/1PZXZwMDKFu1eFhm3yqxJYtH_bNClbLD8hYhlAtWQuKs/edit	6.257	300	0.7	3g	PDF
D8DBE9601752	b9e894cbd74e	Oct 26th 09:16	www.office.com	/?auth=1	5.971	300	0.7	3g	PDF
E0705F301752	b9e894cbd74e	Oct 26th 09:17	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!168&cid=51948194-5ba0-4b59-b123-41da2beaac94&ithint=file%2cdocx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	7.861	300	0.7	3g	PDF
0EFEDC001753	88d927aee763	Oct 26th 09:18	docs.google.com	/presentation/d/1OgkEFH6LhnSGy6e_cXZylZYaGml1oi6AvbEF5BAoOig/edit	7.66	350	1.45	3g	PDF
3B9C37301753	78574580904d	Oct 26th 09:19	www.office.com	/?auth=1	1.42	250	10	4g	PDF
3F1383A01753	78574580904d	Oct 26th 09:19	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!198&ithint=file%2cxlsx&action=editnew&wdTpl=TM16400654&wdlcid=2057&wdNewAndOpenCt=1603693175615&wdPreviousSession=a0098dfc-55a7-4a8e-acbb-64d6122de764&wdOrigin=OFFICECOM-WEB.START.TEMPLATES	8.722	250	10	4g	PDF

Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
47138C801753	785745809O4d	Oct 26th 09:20	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!200&ithint=file%2cdocx&action=editnew&wdTpl=TM00002138&wdlcid=2057&wdNewAndOpenCt=1603693194524&wdPreviousSession=0d8d6139-10b8-44f0-a57a-c71e23b12939&wdOrigin=OFFICECOM-WEB.START.TEMPLATES	4.013	250	10	4g	PDF
4C99E2D01753	785745809O4d	Oct 26th 09:20	docs.google.com	/presentation/u/0/	2.564	250	10	4g	PDF
Infinity	785745809O4d	Oct 26th 09:20	docs.google.com	/spreadsheets/u/0/	2.881	250	10	4g	PDF
8D7966401753	4854f0fc46d5	Oct 26th 09:21	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!198&ithint=file%2cxlsx&action=editnew&wdTpl=TM16400654&wdlcid=2057&wdNewAndOpenCt=1603693175615&wdPreviousSession=a0098dfc-55a7-4a8e-acbb-64d6122de764&wdOrigin=OFFICECOM-WEB.START.TEMPLATES	4.832	250	1.3	3g	PDF
Infinity	4854f0fc46d5	Oct 26th 09:22	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!192&cid=d3589169-0a8a-4097-aeff-5ce04c86dd65&ithint=file%2cxlsx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	5.762	250	1.3	3g	PDF
942B44401753	4854f0fc46d5	Oct 26th 09:22	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!200&ithint=file%2cdocx&action=editnew&wdTpl=TM00002138&wdlcid=2057&wdNewAndOpenCt=1603693194524&wdPreviousSession=0d8d6139-10b8-44f0-a57a-c71e23b12939&wdOrigin=OFFICECOM-WEB.START.TEMPLATES	3.668	250	1.3	3g	PDF
B8DB34301753	11aabd3d9fe8	Oct 26th 09:23	onedrive.live.com	/edit.aspx?action=edit&resid=6937FA95C77474EF!196&ithint=file%2cpptx&action=editnew&wdTpl=TM16401379&wdlcid=2057&wdNewAndOpenCt=1603693052896&wdPreviousSession=eed5b12d-c045-40a3-855e-02ef53c9dd7a&wdOrigin=OFFICECOM-WEB.START.TEMPLATES	6.906	300	10	4g	PDF
D4EA31301820	6da81201c645	Oct 27th 09:51	www.office.com	/?auth=1	1.494	250	1.6	4g	PDF
Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Search

Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
DDEB08401820	6da81201c645	Oct 27th 09:51	docs.google.com	/document/u/0/	5.621	50	10	4g	PD
E34C71701820	6da81201c645	Oct 27th 09:51	onedrive.live.com	/edit.aspx? resid=6937FA95C77474EF!192&cid=3b5a9b0e-3b8b-4020-ad1f-bb3f3c7a1611&ithint=file%2cxlsx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	8.323	50	10	4g	PD
E60A0AD01820	6da81201c645	Oct 27th 09:51	www.office.com	/?auth=1	1.198	50	10	4g	PD
EBA855F01820	6da81201c645	Oct 27th 09:52	onedrive.live.com	/edit.aspx? resid=6937FA95C77474EF!194&cid=c61ef867-315f-448f-b901-1189c1e7ff6b&ithint=file%2cpptx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	5.603	50	10	4g	PD
EC6A1F501820	6da81201c645	Oct 27th 09:52	docs.google.com	/document/d/13eDwCnmsYOhoEvRzDXleZtmIgMxx2M5hqSC3iJfbWLA/edit	2.903	50	10	4g	PD
181A0D401821	edf84b3f5441	Oct 27th 09:53	docs.google.com	/presentation/d/1cIFmiC5O5WGSUhkOSfEdHDZLXfKJyW7oF8S9bPd_QHw/edit#slide=id.gcb9a0b074_1_0	4.865	50	10	4g	PD
1F9A37C01821	edf84b3f5441	Oct 27th 09:53	www.office.com	/launch/word?auth=1	1.764	50	10	4g	PD
270357D01821	edf84b3f5441	Oct 27th 09:53	docs.google.com	/presentation/u/0/	3.132	50	10	4g	PD
2AA1DB501821	edf84b3f5441	Oct 27th 09:53	docs.google.com	/document/u/0/	3.601	50	10	4g	PD
2FAA4F101821	edf84b3f5441	Oct 27th 09:53	docs.google.com	/document/u/0/	3.363	50	10	4g	PD
8CAA4AD01821	f58a77ea539e	Oct 27th 09:56	docs.google.com	/presentation/d/1GW73q1C_fi1uMr4Img_HKS8z8rIzr5l5GHTkPH2l14c/edit#slide=id.gcb9a0b074_1_0	4.394	50	10	4g	PD
90777F701821	f58a77ea539e	Oct 27th 09:56	docs.google.com	/presentation/u/0/	3.474	50	10	4g	PD
92806ED01821	f58a77ea539e	Oct 27th 09:56	docs.google.com	/forms/u/0/	1.819	50	10	4g	PD
961666D01821	f58a77ea539e	Oct 27th 09:56	docs.google.com	/forms/u/0/d/1d0vFJ4Qnr7cVNJijwC37Clev6FK79KPOgV7kkmR-IeU/edit? ntd=1&usp=forms_home&ths=true	3.217	50	10	4g	PD
9B4771301821	f58a77ea539e	Oct 27th 09:56	docs.google.com	/forms/u/0/	1.626	50	10	4g	PD
7EE8C1F01822	992de6a679a6	Oct 27th 10:03	docs.google.com	/document/d/1ZOSpSwpIHiAt55vKERdMdpSe-SiMJSVKs7cqpn-qHb8/edit	5.199	50	10	4g	PD
824B9C001822	992de6a679a6	Oct 27th 10:03	docs.google.com	/forms/u/0/	1.5	50	10	4g	PD
86A3AB801822	992de6a679a6	Oct 27th 10:03	docs.google.com	/forms/u/0/d/1O2gEe8rDY2xcQE7tVK8jj-MM13cIkzdFZfy3FcI8uWM/edit? ntd=1&usp=forms_home&ths=true	3.683	50	10	4g	PD

Monitor ID | Operations ID | Timestamp | Main URL | Partial URL | Load Time (sec) | Net RTT | Net Downlink | Net ET

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
A2CF47601822	fd652f9cee13	Oct 27th 10:04	docs.google.com	/spreadsheets/d/1Ypv7XHMg9QI-QrDUrbu_kbQqwDWx5gNTzE-T0TONVC4/edit#gid=1386834576	5.156	50	10	4g	PDF
A5C555E01822	fd652f9cee13	Oct 27th 10:04	docs.google.com	/spreadsheets/u/0/	3.488	50	10	4g	PDF
A83719301822	fd652f9cee13	Oct 27th 10:04	docs.google.com	/presentation/u/0/	3.375	50	10	4g	PDF
AEB082101822	fd652f9cee13	Oct 27th 10:04	docs.google.com	/presentation/d/132lYgNrBWq_fOMYXIFREX96WBD89ExpxV2mAUkKFiHY/edit	6.123	50	10	4g	PDF
B1FC2AA01822	fd652f9cee13	Oct 27th 10:04	docs.google.com	/presentation/u/0/	2.829	50	10	4g	PDF
B3E7A6F01822	fd652f9cee13	Oct 27th 10:04	docs.google.com	/forms/u/0/	2.392	50	10	4g	PDF
B7245B601822	fd652f9cee13	Oct 27th 10:04	docs.google.com	/forms/d/1mD0aeg3zTefhfoEfivdCZNPhz9el04oLI_HRQIT1bwc/edit	3.106	50	10	4g	PDF
C3106B801822	fd652f9cee13	Oct 27th 10:05	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!192&cid=9aef3ef1-5bfc-4dc6-8d83-70779d8eef45&ithint=file%2cxlsx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	5.594	50	10	4g	PDF
C60457201822	fd652f9cee13	Oct 27th 10:05	www.office.com	/?auth=1	14.863	50	10	4g	PDF
C9E90B601822	fd652f9cee13	Oct 27th 10:05	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!196&cid=3822b402-0c55-4e04-ba19-dbe1b8667894&ithint=file%2cpptx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	6.865	50	10	4g	PDF
CF78B0801822	fd652f9cee13	Oct 27th 10:05	www.office.com	/launch/excel?auth=1	1.871	50	10	4g	PDF
D37E81501822	fd652f9cee13	Oct 27th 10:05	www.office.com	/launch/word?auth=1	1.567	50	10	4g	PDF
D982E1401822	fd652f9cee13	Oct 27th 10:05	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!202&ithint=file%2cdocx&action=editnew&wdTpl=TM00002138&wdlcid=2057&wdNewAndOpenCt=1603782343791&wdPreviousSession=59037d58-08f4-49aa-b636-e7f55dcd4103&wdOrigin=OFFICECOM-WEB.START.TEMPLATES	6.361	50	10	4g	PDF
FD5F06701822	a230baf49c09	Oct 27th 10:06	www.office.com	/launch/powerpoint?ui=en-US&rs=GB&auth=1	1.237	250	8.4	4g	PDF
Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Showing 1 to 173 of 173 entries

QoS Cloud Monitoring

Home / Monitoring Sessions / Sites Monitored

ALL MONITORED SITES

For **ACTIVE** monitoring sessions, evry navigation done within the saved URL are monitored and logged.
The information stored is the network status, URL info and the loading time. This data is for client with the ID, **906EEE40CD69**

The table below is a list of all sites monitoired by this client

PRINT PDF

Search:

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
9C8C6EA0CD69	28d810ab79ed	Sep 2nd 01:08	app.hubspot.com	/login?loginPortalId=&loginRedirectUrl=https%3A%2F%2F	2.796	400	1.5	3g	PI
A3C52FE0CD69	28d810ab79ed	Sep 2nd 01:08	docs.google.com	/document/d/1-b4TO_NTiCUtxbYLM4_68HrtST_TILb6VzUT6-v6k_I/edit	5.551	400	1.5	3g	PI
A6B17A60CD69	28d810ab79ed	Sep 2nd 01:08	docs.google.com	/document/u/0/	2.544	400	1.5	3g	PI
AEB6DA70CD69	28d810ab79ed	Sep 2nd 01:08	docs.google.com	/document/d/1-b4TO_NTiCUtxbYLM4_68HrtST_TILb6VzUT6-v6k_I/edit	6.399	400	1.5	3g	PI
EE105A20CD69	da6e8c17907d	Sep 2nd 01:10	www.shopify.com	/	2.231	250	2.2	4g	PI
F51CDB40CD69	da6e8c17907d	Sep 2nd 01:10	www.shopify.com	/	4.798	250	2.2	4g	PI
FBF6B3A0CD69	da6e8c17907d	Sep 2nd 01:11	www.shopify.com	/	1.773	250	2.2	4g	PI
01675F10CD6A	da6e8c17907d	Sep 2nd 01:11	www.shopify.com	/	1.54	250	2.2	4g	PI
0697CD30CD6A	da6e8c17907d	Sep 2nd 01:11	app.hubspot.com	/login?loginPortalId=&loginRedirectUrl=https%3A%2F%2F	1.635	250	2.2	4g	PI
0BECB340CD6A	da6e8c17907d	Sep 2nd 01:11	docs.google.com	/document/d/1-b4TO_NTiCUtxbYLM4_68HrtST_TILb6VzUT6-v6k_I/edit	4.831	250	2.2	4g	PI
12122FC0CD6A	da6e8c17907d	Sep 2nd 01:11	app.hubspot.com	/login?loginPortalId=&loginRedirectUrl=https%3A%2F%2F	1.671	250	2.2	4g	PI
1BA534B0CD6A	da6e8c17907d	Sep 2nd 01:12	app.hubspot.com	/login/googleLoginRedirect#state=%7B%22client%22:%22HUBSPOT%22,%22rememberLogin%22:false,%22loginRedirectUrl%22:%22https://	2.672	250	2.2	4g	PI
1D83DFC0CD6A	da6e8c17907d	Sep 2nd 01:12	app.hubspot.com	/home-beta	0.946	250	2.2	4g	PI
250C47A0CD6A	da6e8c17907d	Sep 2nd 01:12	app.hubspot.com	/getting-started/5131353	9.79	250	2.2	4g	PI
7E876260CD6A	a33046c168b1	Sep 2nd 01:14	app.hubspot.com	/email/5131353/edit-dd/12599520521	5.408	200	2.7	4g	PI
Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
B6E8D350CD6A	ef681884e898	Sep 2nd 01:16	www.salesforce.com	/services/learn/overview/	3.183	250	2.15	4g	PI
BD2C1100CD6A	ef681884e898	Sep 2nd 01:16	www.salesforce.com	/events/webinars/	2.68	200	2.9	4g	PI
8B48EDE0CE0D	e48a04385a7e	Sep 3rd 08:41	app.hubspot.com	/home-beta	1.643	200	1.5	4g	PI
8CC66620CE0D	e48a04385a7e	Sep 3rd 08:42	app.hubspot.com	/getting-started/5131353	1.315	200	1.5	3g	PI
91C34120CE0D	e48a04385a7e	Sep 3rd 08:42	app.hubspot.com	/email/5131353/manage	1.669	350	1.5	3g	PI
979C7260CE0D	e48a04385a7e	Sep 3rd 08:42	app.hubspot.com	/analytics/5131353/tools	3.016	350	1.5	3g	PI
B400AA70CE0D	23a41befe712	Sep 3rd 08:43	app.hubspot.com	/upgrade/5131353/sequences	0.985	350	1.45	3g	PI
B9CAC080CE0D	23a41befe712	Sep 3rd 08:43	app.hubspot.com	/chatflows/5131353	2.67	350	1.45	3g	PI
BFDA9220CE0D	23a41befe712	Sep 3rd 08:43	app.hubspot.com	/contacts/5131353/contacts	4.939	350	1.45	3g	PI
C4B28000CE0D	23a41befe712	Sep 3rd 08:43	app.hubspot.com	/import/5131353	1.217	350	1.45	3g	PI
805BA75001DA	aeb181803d8e	Nov 8th 06:47	docs.google.com	/document/u/0/	3.465	50	10	4g	PI
8A90B8A001DA	aeb181803d8e	Nov 8th 06:47	docs.google.com	/document/d/1mh0Cbz4ijmToQi4aPjELU_KTsLbOGLxC3QLWflw7VRU/edit	4.571	50	10	4g	PI
90175D1001DA	aeb181803d8e	Nov 8th 06:48	docs.google.com	/document/u/0/	2.888	50	10	4g	PI
9460034001DA	aeb181803d8e	Nov 8th 06:48	docs.google.com	/document/d/1lw3rX1TPbaRA4mFClPtGXjdVuXVSGqV05qPqb1YVY6U/edit	4.168	50	10	4g	PI
99E1ECC001DA	aeb181803d8e	Nov 8th 06:48	docs.google.com	/document/u/0/	2.768	50	10	4g	PI
9D4F271001DA	aeb181803d8e	Nov 8th 06:48	docs.google.com	/document/d/1lw3rX1TPbaRA4mFClPtGXjdVuXVSGqV05qPqb1YVY6U/edit	3.849	50	10	4g	PI
B62E08F001DA	39e0c48098a5	Nov 8th 06:49	app.hubspot.com	/login/?loginRedirectUrl=https%3A%2F%2F	0.105	50	10	4g	PI
B91C012001DA	39e0c48098a5	Nov 8th 06:49	app.hubspot.com	/login/?loginRedirectUrl=https%3A%2F%2F	0.88	50	10	4g	PI
C129025001DA	39e0c48098a5	Nov 8th 06:49	app.hubspot.com	/login/googleLoginRedirect#state=%7B%22client%22:%22HUBSPOT%22,%22rememberLogin%22:false,%22loginPortalId%22:5131353,%22loginRedirectUrl%22:%22https://	2.464	50	10	4g	PI
C207064001DA	39e0c48098a5	Nov 8th 06:49	app.hubspot.com	/email/5131353/manage	0.934	150	10	4g	PI
DA34998001DA	19487f70e2b8	Nov 8th 06:50	app.hubspot.com	/reports-dashboard/5131353/sales	0.919	150	10	4g	PI
E25FAA0001DA	19487f70e2b8	Nov 8th 06:50	app.hubspot.com	/reports-dashboard/5131353/sales	0.694	150	10	4g	PI
E50AF3E001DA	19487f70e2b8	Nov 8th 06:50	app.hubspot.com	/reports-dashboard/5131353	0.76	150	10	4g	PI
E9710D2001DA	19487f70e2b8	Nov 8th 06:50	app.hubspot.com	/contacts/5131353/tickets	3.239	200	10	4g	PI
394A35400601	348d664ca468	Nov 13th 01:34	docs.google.com	/document/d/16Y6Bv6ouLwoJsVVfnv5uyCw-S2WPX5poivtRNsSDfBw/edit	6.163	100	9.75	4g	PI

Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
3E2BE7200601	348d664ca468	Nov 13th 01:34	docs.google.com	/document/d/16Y6Bv6ouLwoJsVVfnv5uyCw-S2WPX5poivtRNsSDfBw/edit	3.993	100	9.75	4g	PI
431958D00601	348d664ca468	Nov 13th 01:35	docs.google.com	/document/u/0/	4.026	150	10	4g	PI
45E6D0B00601	348d664ca468	Nov 13th 01:35	docs.google.com	/document/d/14MAeKKzW8MgrL-3KLH7DWa1qfL8udhultcGNNf0jPso/edit	4.637	150	10	4g	PI
49E6FC300601	348d664ca468	Nov 13th 01:35	app.hubspot.com	/reports-dashboard/5131353/sales	2.83	150	10	4g	PI
4A39FE300601	348d664ca468	Nov 13th 01:35	app.hubspot.com	/login/?loginRedirectUrl=https%3A%2F%2F	0.307	150	10	4g	PI
4D77C4100601	348d664ca468	Nov 13th 01:35	app.hubspot.com	/login/?loginRedirectUrl=https%3A%2F%2F	1.5	150	10	4g	PI
84E3BCB0075A	6ad691000e11	Nov 15th 06:46	docs.google.com	/document/d/11eZQDRE0K_VhJGaxA2eRQYja5lCNnedGRMbOXi4YXXc/edit	2.419	50	10	4g	PI
88EAC600075A	6ad691000e11	Nov 15th 06:46	app.hubspot.com	/reports-dashboard/5131353/sales	1.028	50	10	4g	PI
89D39F60075A	6ad691000e11	Nov 15th 06:46	app.hubspot.com	/login/?loginRedirectUrl=https%3A%2F%2F	1.467	50	10	4g	PI
62D349401B02	f3f0069838bf	Dec 10th 07:06	docs.google.com	/document/u/0/	2.515	50	10	4g	PI
678622F01B02	f3f0069838bf	Dec 10th 07:06	docs.google.com	/document/d/1UcRKL0LYpkbL_p_BbcTq_48ilFJKPxn53BBrCjQu8b8/edit	4.916	50	10	4g	PI
86F4DBE01B02	0af06920a962	Dec 10th 07:07	docs.google.com	/document/d/1Wvm2_7mEiw75Q-7r_uLjmWuuZmAL7akRTgAtCiP5Tlw/edit	3.468	50	10	4g	PI
8AAD28F01B02	0af06920a962	Dec 10th 07:07	docs.google.com	/document/u/0/	2.889	50	10	4g	PI
8E7CF5A01B02	0af06920a962	Dec 10th 07:07	docs.google.com	/document/d/1O25lAWa8DFN89XajJNv1nMDpT8S0CX_odtK3ZkY9Nss/edit	1.481	50	10	4g	PI
943299501B02	0af06920a962	Dec 10th 07:07	docs.google.com	/document/d/1O25lAWa8DFN89XajJNv1nMDpT8S0CX_odtK3ZkY9Nss/edit	3.353	50	10	4g	PI
97058F701B02	0af06920a962	Dec 10th 07:07	docs.google.com	/document/u/0/	2.835	50	10	4g	PI
993F2BC01B02	0af06920a962	Dec 10th 07:07	docs.google.com	/document/u/0/	1.846	50	10	4g	PI
9E291B001B02	0af06920a962	Dec 10th 07:07	docs.google.com	/document/u/0/	2.354	50	10	4g	PI
B62F50201B02	d32698edd030	Dec 10th 07:08	app.hubspot.com	/login/?loginRedirectUrl=https%3A%2F%2F	0.416	50	9.85	4g	PI
B7E034701B02	d32698edd030	Dec 10th 07:08	app.hubspot.com	/login/?loginRedirectUrl=https%3A%2F%2F	0.231	50	9.85	4g	PI
BB856EB01B02	d32698edd030	Dec 10th 07:08	app.hubspot.com	/login/?loginRedirectUrl=https%3A%2F%2F	2.07	50	9.85	4g	PI

Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
D359BE601B02	ea0e473d7733	Dec 10th 07:09	app.hubspot.com	/login/?loginRedirectUrl=https%3A%2F%2Fwww.hubspot.com%2Fpricing%2Fmarketing	2.653	200	6.45	4g	PI
DB8173801B02	ea0e473d7733	Dec 10th 07:09	app.hubspot.com	/login/googleLoginRedirect#state=%7B%22client%22:%22HUBSPOT%22,%22rememberLogin%22:false,%22loginRedirectUrl%22:%22https://www.hubspot.com/pricing/marketing%22%7D&id_token=eyJhbGciOiJSUzI1NiIsImtpZCI6IjViNWRkOWJlNDBiNWUxY2YxMjFlMzU3M2M4ZTQ5ZjEyNTI3MTgzZDMiLCJ0eXAiOiJKV1QifQ.eyJpc3MiOiJhY2NvdW50cy5nb29nbGUuY29tIiwiYXpwIjoiNDY2NjgwNTI2NDc0LmFwcHMuZ29vZ2xldXNlcmNvbnRlbnQuY29tIiwiYXVkIjoiNDY2NjgwNTI2NDc0LmFwcHMuZ29vZ2xldXNlcmNvbnRlbnQuY29tIiwic3ViIjoiMTEzNzQzOTE2MDU5MzI0MTkyMzM1IiwiZW1haWwiOiJnb2xkbWVkYWxpc3QzMjFAZ21haWwuY29tIiwiZW1haWxfdmVyaWZpZWQiOnRydWUsIm5vbmNlIjoiQ05tWF9lX3VMUklaQUQ1d01XOHRCREt2am9mdTdqRXpvMFJCc3FQbnVZMUdkUSIsImlhdCI6MTU3NTk1MDk2NiwiZXhwIjoxNTc1OTU0NTY2LCJqdGkiOiI0NWFiMzE2MDQzOTg5OGRhYmE2OTkyMjYxODU1YzcxNjg4NjY5OGRjIn0.v9gqs84cPJFHbY80MCAIdY8S_iMQtfO8OPtrF_rGd5RJhIDPKMwfg7YtfQsgHpVjxIYlTRkkT42OSyS_Qgpl0BD1shuaozNj3KARK9mGuoc1sDtopAu-nTanFKHqdhp7pTPlR_wxjFq34TUHKQHgRuz_o9WzFaixmYkpc-R_StmMxyNnFtp250Q9p-EWNaAqd5f59NT1_gyxbq85VOsMdS_pA4PPRSFSzCAMGOTSFBpitMG43pRgiCxe5z5j5P3dbV4lL4AwyzV_1VUsq-X9w5767_79s9OM4vvBDW_hsya6J0cGa6kfDLIK7LeTfID7VEh6rd33dJOD3Hx3rln1A&authuser=0&session_state=61e2e5473d932ca68ea322cc088caea88900a934..c41c&prompt=none	2.539	50	6.3	4g	PI
DEAD86201B02	ea0e473d7733	Dec 10th 07:09	app.hubspot.com	/pricing/5131353/marketing	2.745	50	6.3	4g	PI
E2DB51501B02	ea0e473d7733	Dec 10th 07:09	app.hubspot.com	/email/5131353/manage	1.428	50	10	4g	PI
Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
E5B60FA01B02	ea0e473d7733	Dec 10th 07:09	app.hubspot.com	/upgrade/5131353/sequences	1.175	50	10	4g	PI
EA6789C01B02	ea0e473d7733	Dec 10th 07:09	app.hubspot.com	/contacts/5131353/lists	2.947	50	10	4g	PI
ECFD28C01B02	ea0e473d7733	Dec 10th 07:09	app.hubspot.com	/reports-dashboard/5131353	1.054	50	10	4g	PI
075E8BF01B03	22a7c45bf20c	Dec 10th 07:10	app.hubspot.com	/contacts/5131353/deals	2.285	350	1.25	3g	PI
12E9DE701B03	22a7c45bf20c	Dec 10th 07:11	www.shopify.com	/free-trial?ref=geardeal-online2	5.869	50	5.2	4g	PI
31ACC7001B03	b47b8f6b974d	Dec 10th 07:11	www.shopify.com	/free-trial?ref=geardeal-online2	1.928	50	5.8	4g	PI
371EAAF01B03	b47b8f6b974d	Dec 10th 07:12	www.shopify.com	/free-trial?ref=geardeal-online2	1.946	50	5.8	4g	PI
DD4FAE601B03	737f48e2da79	Dec 10th 07:16	boyjumaboy.myshopify.com	/admin/checkouts	3.393	200	4.7	4g	PI
072892601B04	72ff6d67600a	Dec 10th 07:17	boyjumaboy.myshopify.com	/admin	3.085	200	4.4	4g	PI
284103B01B04	1f089013b5d3	Dec 10th 07:18	boyjumaboy.myshopify.com	/admin	3.151	200	4.65	4g	PI
5E87AAF01B04	4955d7e5c2fc	Dec 10th 07:20	boyjumaboy.myshopify.com	/admin	3.498	200	5.1	4g	PI
6159DDC01B04	4955d7e5c2fc	Dec 10th 07:20	boyjumaboy.myshopify.com	/admin	2.797	200	5.1	4g	PI
64CC21201B04	4955d7e5c2fc	Dec 10th 07:20	boyjumaboy.myshopify.com	/admin	3.885	200	5.1	4g	PI
9A3F99E01B04	6eb0415b8065	Dec 10th 07:21	www.salesforce.com	/	3.815	250	1.45	3g	PI
A210F1A01B04	6eb0415b8065	Dec 10th 07:22	www.salesforce.com	/products/what-is-salesforce/	5.001	250	2.6	4g	PI
D9649C101B04	c01765202dbc	Dec 10th 07:23	www.salesforce.com	/products/	4.472	200	1.8	4g	PI
DCD77BB01B04	c01765202dbc	Dec 10th 07:23	www.salesforce.com	/company/about-us/	3.709	200	3.35	4g	PI
F52122701B04	887d108a46a8	Dec 10th 07:24	www.salesforce.com	/events/	3.827	200	3.6	4g	PI
FD27BB001B04	887d108a46a8	Dec 10th 07:24	www.salesforce.com	/services/overview/	3.819	200	3.6	4g	PI
026018601B05	887d108a46a8	Dec 10th 07:24	www.salesforce.com	/events/	3.526	200	4.45	4g	PI
085A18101B05	887d108a46a8	Dec 10th 07:25	www.salesforce.com	/customer-success-stories/	6.531	200	6	4g	PI
Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Showing 1 to 86 of 86 entries

Appendix 6: Sample Raw QoS Results For Google and Microsoft Comparison

QoS Cloud Monitoring

Home / Monitoring Sessions / Sites Monitored

ALL MONITORED SITES

For **ACTIVE** monitoring sessions, evry navigation done within the saved URL are monitored and logged. The information stored is the network status, URL info and the loading time. This data is for client with the ID, **oAEDDACoF6gD**

The table below is a list of all sites monitoired by this client

PRINT PDF

Search: oct

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
8197C4D00799	cf45c07d1dec	Oct 6th 09:02	www.office.com	/?auth=1	1.998	450	1.4	3g	PDF
A33F7DD00799	3646b9c9ce92	Oct 6th 09:03	docs.google.com	/document/d/18VTjqWvvwkwiosam_OhjeE7SwGuR7fTdMuUl8eTCO2s/edit	5.132	250	6.7	4g	PDF
F524EA400799	2ca65a45c041	Oct 6th 09:05	www.office.com	/launch/word?auth=1	1.821	200	10	4g	PDF
102377D0079A	f43ea98a74c8	Oct 6th 09:06	docs.google.com	/document/d/11_WnFosTPtEnv6NJJx38cuBVH3uQIVvSaj4QQfML78Y/edit	7.357	200	10	4g	PDF
3F533DB0079A	88d5f3ccb091	Oct 6th 09:07	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!163&ithint=file%2cdocx&action=editnew&wdNewAndOpenCt=1601964342348&wdPreviousSession=4c87d3db-cb2e-446f-bb48-55da84995f60&wdOrigin=OFFICECOM-WEB.START.NEW	6.898	200	9.9	4g	PDF
50BF1470079A	24a91746e108	Oct 6th 09:08	docs.google.com	/document/d/1ywVmZ4el3nkrc4wajJLUFqx-qy8Q8sIUDnkZVXq9zcQ/edit	4.624	250	10	4g	PDF
58861D40079A	24a91746e108	Oct 6th 09:08	www.office.com	/launch/word?auth=1	3.54	250	10	4g	PDF
8A3EC6A0079A	6f2bc5e28de2	Oct 6th 09:09	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!168&ithint=file%2cdocx&action=editnew&wdTpl=TM00002109&wdlcid=2057&wdNewAndOpenCt=1601964509077&wdPreviousSession=8e968b31-ea05-4b5a-9d0b-457e0e23a5d7&wdOrigin=OFFICECOM-WEB.START.TEMPLATES	9.336	250	10	4g	PDF
Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
B332EFF0079A	c16a2931adcb	Oct 6th 09:10	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!168&ithint=file%2cdocx&action=editnew&wdTpl=TM00002109&wdlcid=2057&wdNewAndOpenCt=1601964509077&wdPreviousSession=8e968b31-ea05-4b5a-9d0b-457e0e23a5d7&wdOrigin=OFFICECOM-WEB.START.TEMPLATES	11.374	300	1.55	3g	PDI
D95BEB50079A	b437d06c138b	Oct 6th 09:12	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!170&ithint=file%2cxlsx&action=editnew&wdNewAndOpenCt=1601964700190&wdPreviousSession=0f966f46-c478-4fd0-ae6c-6fb1e12e91a1&wdOrigin=OFFICECOM-WEB.MAIN.NEW	3.56	200	7.9	4g	PDI
DF169810079A	b437d06c138b	Oct 6th 09:12	docs.google.com	/document/u/0/	1.899	200	7.9	4g	PDI
1BFB4730079B	5d8181e126e1	Oct 6th 09:13	docs.google.com	/spreadsheets/d/1onHQau1IFLStma2oRuftPt01l-beVJPz1A5QmCJfJwo/edit#gid=0	5.363	150	5.5	4g	PDI
2E995C60079B	f04a668a15cf	Oct 6th 09:14	docs.google.com	/spreadsheets/d/1onHQau1IFLStma2oRuftPt01l-beVJPz1A5QmCJfJwo/edit#gid=0	9.458	200	4.95	4g	PDI
C02CAA10079B	dfd42b53040e	Oct 6th 09:18	docs.google.com	/presentation/d/1Udgv90jnfw83N7sBgj8S4dznRgiZ1PtFDSQG3UterfE/edit#slide=id.p	8.742	650	1.55	3g	PDI
C0BDED90079B	dfd42b53040e	Oct 6th 09:18	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!173&ithint=file%2cpptx&action=editnew&wdNewAndOpenCt=1601965061950&wdPreviousSession=6bd0b728-95d4-4acb-8a3a-90fe91159f72&wdOrigin=OFFICECOM-WEB.START.NEW	42.397	500	1.55	3g	PDI
F9442490079B	a0542e958265	Oct 6th 09:20	onedrive.live.com	/edit.aspx?action=edit&resid=6937FA95C77474EF!173&ithint=file%2cpptx&action=editnew&wdNewAndOpenCt=1601965061950&wdPreviousSession=6bd0b728-95d4-4acb-8a3a-90fe91159f72&wdOrigin=OFFICECOM-WEB.START.NEW	13.339	250	4.6	4g	PDI
Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
03282FB0079C	3e6418d52140	Oct 6th 09:20	docs.google.com	/presentation/d/1Udgvg0jnfw83N7sBgj8S4dznRgiZ1PtFDSQG3UterfE/edit#slide=id.p	7.364	250	1.35	3g	PDI
8FED5380079C	51da551264d9	Oct 6th 09:24	onedrive.live.com	/edit.aspx?action=edit&resid=6937FA95C77474EF!173&ithint=file%2cpptx&action=editnew&wdNewAndOpenCt=160196506195O&wdPreviousSession=6bd0b728-95d4-4acb-8a3a-90fe91159f72&wdOrigin=OFFICECOM-WEB.START.NEW	8.919	250	1.4	4g	PDI
98BD2F80079C	757d34695d2b	Oct 6th 09:24	docs.google.com	/presentation/d/1Udgvg0jnfw83N7sBgj8S4dznRgiZ1PtFDSQG3UterfE/edit#slide=id.p	4.415	250	1.4	4g	PDI
DC50D670079C	f3fd91666993	Oct 6th 09:26	docs.google.com	/forms/u/0/d/1cX2HGCkOFtq8r36J-reJLw6vfxnsfDDJnBuhlKr2l5M/edit	2.448	200	5.1	4g	PDI
DD3A7320079C	f3fd91666993	Oct 6th 09:26	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!176&cid=6937fa95c77474ef	4.615	200	5.1	4g	PDI
02CB2620079D	7570908b255e	Oct 6th 09:27	onedrive.live.com	/Edit.aspx?resid=6937FA95C77474EF!176&wd=target(Research%20notes.one%7C5e0ca259-dcdb-4f52-9004-6f417c26d1d5/Sample%20Research%20Notes%7C494f65eb-bd10-48c3-9d88-48e11208c185/)	3.727	200	5.1	4g	PDI
AB9451E00A14	c42d3a4ed976	Oct 9th 12:49	www.office.com	/?auth=1	11.067	200	5.55	4g	PDI
B06FC2300A14	c42d3a4ed976	Oct 9th 12:49	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!173&cid=364dba8f-5284-45ab-88ba-e63e017e4f06&ithint=file%2cpptx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	8.285	200	5.55	4g	PDI
BA8871E00A14	c42d3a4ed976	Oct 9th 12:49	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!170&cid=256528b1-4fdc-422f-9333-98a2e18dd392&ithint=file%2cxlsx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	5.398	200	5.55	4g	PDI
BC8468F00A14	c42d3a4ed976	Oct 9th 12:49	docs.google.com	/document/u/0/	6.58	200	5.55	4g	PDI
C3B2C9F00A14	c42d3a4ed976	Oct 9th 12:49	docs.google.com	/document/d/1QZv7JCJSJBKK_WwBeKBjH30gyq4eGw1DTCVFM30Fp78/edit	6.904	200	5.55	4g	PDI

Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
E9A1C9900A14	a0525ddc05de	Oct 9th 12:50	docs.google.com	/spreadsheets/d/12SHUOcKsv8gwZpZRnygf6YuvQzJ3Pi7LwcuF35BHZAY/edit#gid=1386834576	3.851	200	6.05	4g	PDI
ED9A2CE00A14	a0525ddc05de	Oct 9th 12:50	docs.google.com	/spreadsheets/u/0/	3.368	200	6.05	4g	PDI
EFB1E9500A14	a0525ddc05de	Oct 9th 12:51	docs.google.com	/forms/u/0/	2.175	200	6.05	4g	PDI
F32ABC600A14	a0525ddc05de	Oct 9th 12:51	docs.google.com	/forms/u/0/	1.203	200	6.05	4g	PDI
F8669C300A14	a0525ddc05de	Oct 9th 12:51	docs.google.com	/presentation/u/0/	3.159	200	6.05	4g	PDI
FF6DC6200A14	a0525ddc05de	Oct 9th 12:51	docs.google.com	/presentation/d/12UNnO8MuceatabZwBb0opPdXZyZXQJiCODHNZ7ilw3w/edit	8.466	200	5.85	4g	PDI
2BC7FDD00A15	240fb9d1c5ff	Oct 9th 12:52	www.office.com	/?auth=1	2.019	200	5.9	4g	PDI
32CFEB100A15	240fb9d1c5ff	Oct 9th 12:52	www.office.com	/launch/powerpoint?auth=1	1.553	200	5.9	4g	PDI
39A704500A15	240fb9d1c5ff	Oct 9th 12:53	onedrive.live.com	/edit.aspx?action=edit&resid=6937FA95C77474EF!184&ithint=file%2cpptx&action=editnew&wdTpl=TM16401370&wdlcid=2057&wdNewAndOpenCt=1602237175097&wdPreviousSession=d907cc97-9900-4013-b250-89a2b4cfe125&wdOrigin=OFFICECOM-WEB.START.TEMPLATES	7.119	200	5.9	4g	PDI
419050E00A15	240fb9d1c5ff	Oct 9th 12:53	docs.google.com	/presentation/u/0/	3.272	250	3.55	4g	PDI
446D32100A15	240fb9d1c5ff	Oct 9th 12:53	docs.google.com	/document/u/0/	3.439	250	3.55	4g	PDI
49F313300A15	240fb9d1c5ff	Oct 9th 12:53	docs.google.com	/document/d/1v4G5ljoMFWOhABEkzt6MJMJrbgJ9WuQtJc52DmvlQvg/edit	5.501	250	3.55	4g	PDI
2A2D6B300A16	af129825903b	Oct 9th 12:59	www.office.com	/?auth=1	1.064	150	8.05	4g	PDI
2DD194000A16	af129825903b	Oct 9th 12:59	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!170&cid=256528b1-4fdc-422f-9333-96a2e18dd392&ithint=file%2cxlsx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	3.345	150	8.05	4g	PDI
332E69500A16	af129825903b	Oct 9th 01:00	www.office.com	/launch/word?auth=1	1.609	150	8.05	4g	PDI
36B964D00A16	af129825903b	Oct 9th 01:00	docs.google.com	/document/d/1v4G5ljoMFWOhABEkzt6MJMJrbgJ9WuQtJc52DmvlQvg/edit	5.844	150	8.05	4g	PDI
A821EC900AC6	f80f70763b2d	Oct 10th 10:03	www.office.com	/?auth=1	27.321	400	1.25	3g	PDI
BCD8CFA00AC6	9465708b5cad	Oct 10th 10:03	docs.google.com	/document/d/1xpfXYvxqKNW_iURcMG1h4ppOKmTxpvXrlVDhXqziZP0/edit	9.246	450	1.55	3g	PDI

Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
BF8774E00AC6	9465708b5cad	Oct 10th 10:03	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!170&cid=3941edd8-df60-426f-b701-d019ceaegg1e&ithint=file%2cxlsx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	6.767	450	1.55	3g	PDI
D72D3A800AC6	1fb10dc1bd8f	Oct 10th 10:04	docs.google.com	/document/d/1xpfXYvxqKNW_iURcMG1h4ppOKmTxpvXrlVDhXqziZP0/edit	5.263	400	1.35	3g	PDI
DC99EE500AC6	1fb10dc1bd8f	Oct 10th 10:04	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!170&cid=3941edd8-df60-426f-b701-d019ceaegg1e&ithint=file%2cxlsx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	5.736	450	1.35	3g	PDI
F7AA56300AC6	0d70a221e65a	Oct 10th 10:05	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!173&cid=23d4a9a6-a8db-4e23-9a3c-1eac37991 9d8&ithint=file%2cpptx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	9.444	550	1.5	3g	PDI
04A761200AC7	0d70a221e65a	Oct 10th 10:05	docs.google.com	/document/d/1xpfXYvxqKNW_iURcMG1h4ppOKmTxpvXrlVDhXqziZP0/edit	4.366	400	1.5	3g	PDI
09240FA00AC7	0d70a221e65a	Oct 10th 10:05	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!170&cid=3941edd8-df60-426f-b701-d019ceaegg1e&ithint=file%2cxlsx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	4.074	400	1.5	3g	PDI
0FC9DFB00AC7	0d70a221e65a	Oct 10th 10:06	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!173&cid=23d4a9a6-a8db-4e23-9a3c-1eac379919d8&ithint=file%2cpptx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	9.452	400	1.5	3g	PDI
37ABFCC00AC7	9fd323027584	Oct 10th 10:07	docs.google.com	/presentation/d/12JDdFn4aJzG-zcW8T79LlCuEJaKRByx7DHcuXTaHukU/edit#slide=id.p	5.173	250	1.3	3g	PDI
3BBC7BF00AC7	9fd323027584	Oct 10th 10:07	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!170&cid=3941edd8-df60-426f-b701-d019ceaegg1e&ithint=file%2cxlsx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	3.64	250	2.55	4g	PDI

Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
41FA89800AC7	gfd323027584	Oct 10th 10:07	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!173&cid=23d4a9a6-a8db-4e23-9a3c-1eac379919d8&ithint=file%2cpptx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	7.164	250	2.55	4g	PDF
4793CB900AC7	gfd323027584	Oct 10th 10:07	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!168&cid=2b15d1c6-d630-4783-acd1-15330c379e7e&ithint=file%2cdocx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	3.447	250	2.55	4g	PDF
B0E111500E98	630d65ee6536	Oct 15th 06:44	docs.google.com	/document/u/0/	9.099	50	10	4g	PDF
B177AC000E98	630d65ee6536	Oct 15th 06:44	www.office.com	/?auth=1	7.18	50	10	4g	PDF
CF6591500E98	8993a1f3ff96	Oct 15th 06:45	www.office.com	/?auth=1	1.241	50	10	4g	PDF
D372B5200E98	8993a1f3ff96	Oct 15th 06:45	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!173&cid=30285e96-f6c2-4e20-905d-a378102c444f&ithint=file%2cpptx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	9.489	50	10	4g	PDF
E6E7C9600E98	fe1b764005e8	Oct 15th 06:45	docs.google.com	/document/d/1b-59QrMACbVKpJ7b9ZyAbsQZfT3Yg0jJQkAi8Bi8_Yl/edit	4.993	50	10	4g	PDF
EA9F7A300E98	fe1b764005e8	Oct 15th 06:45	docs.google.com	/document/u/0/	4.281	50	10	4g	PDF
EF7963E00E98	fe1b764005e8	Oct 15th 06:45	docs.google.com	/spreadsheets/u/0/	2.394	50	10	4g	PDF
F3BE12700E98	fe1b764005e8	Oct 15th 06:46	docs.google.com	/spreadsheets/u/0/d/1leTsFkH7-JeselHJhMTuXk140xPOM5D18vRY678z5V4/edit?ntd=1&usp=sheets_home&ths=true	2.202	50	10	4g	PDF
F80706C00E98	fe1b764005e8	Oct 15th 06:46	docs.google.com	/spreadsheets/u/0/	2.492	50	10	4g	PDF
FAE128D00E98	fe1b764005e8	Oct 15th 06:46	docs.google.com	/spreadsheets/u/0/d/1UCgbfvGysnVwX1q1j1YwhJQGM5mtsJdoa8OmRV7L7C0/edit?ntd=1&usp=sheets_home&ths=true	2.052	50	10	4g	PDF
FE8D67F00E98	fe1b764005e8	Oct 15th 06:46	docs.google.com	/spreadsheets/u/0/	2.299	50	10	4g	PDF
02E502400E99	fe1b764005e8	Oct 15th 06:46	docs.google.com	/spreadsheets/d/1SvOd4lWNIMa42UqOBZdWbZGi-lsrhPEJCtBlkZgSSK0/edit	5.108	50	10	4g	PDF

Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
22D520300E99	bb1fc7af9134	Oct 15th 06:47	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!173&cid=30285e96-f6c2-4e20-905d-a378102c444f&ithint=file%2cpptx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	10.952	50	10	4g	PDI
45E9EE00111B	b8bd2cb5a0e0	Oct 18th 11:23	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!168&cid=60d8288c-07d9-4577-a768-9bf92f08707b&ithint=file%2cdocx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	3.376	50	9.95	4g	PDI
4B6C73C0111B	b8bd2cb5a0e0	Oct 18th 11:24	docs.google.com	/document/u/0/	3.528	50	9.95	4g	PDI
51A43FC0111B	b8bd2cb5a0e0	Oct 18th 11:24	docs.google.com	/document/d/1xyy7-8R5zl5RlVlcJb_XWHV_gQXjxQF9hrHb1LSosMs/edit	8.674	50	9.95	4g	PDI
51C81B70111B	b8bd2cb5a0e0	Oct 18th 11:24	www.office.com	/launch/powerpoint?auth=1	2.006	50	9.95	4g	PDI
598527E0111B	b8bd2cb5a0e0	Oct 18th 11:24	onedrive.live.com	/edit.aspx?action=edit&resid=6937FA95C77474EF!186&ithint=file%2cpptx&action=editnew&wdNewAndOpenCt=1603009463555&wdPreviousSession=4513434b-29c8-460c-8099-e6247f5844dd&wdOrigin=OFFICECOM-WEB.START.NEW	7.046	100	10	4g	PDI
5E82C630111B	b8bd2cb5a0e0	Oct 18th 11:24	docs.google.com	/document/u/0/	4.16	100	10	4g	PDI
61004950111B	b8bd2cb5a0e0	Oct 18th 11:24	docs.google.com	/presentation/u/0/	3.196	100	10	4g	PDI
67E19BC0111B	b8bd2cb5a0e0	Oct 18th 11:24	docs.google.com	/presentation/d/1VamdAAy7nlxhrQ_bZ83cfJE1sEhNid5J0G63i43j7Gk/edit	7.86	100	10	4g	PDI
9204A820111B	c38fe58383c7	Oct 18th 11:26	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!188&ithint=file%2cxlsx&action=editnew&wdTpl=TM16400654&wdlcid=2057&wdNewAndOpenCt=1603009518305&wdPreviousSession=744f77d4-06a8-4f58-9e66-e662190a3e77&wdOrigin=OFFICECOM-WEB.START.TEMPLATES	10.693	100	10	4g	PDI
7052263011CA	8b66babb8ae4	Oct 19th 08:17	docs.google.com	/document/u/0/	4.136	300	1.3	3g	PDI
7213854011CA	8b66babb8ae4	Oct 19th 08:17	www.office.com	/?auth=1	4.426	300	1.3	3g	PDI

Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
75FC312011CA	8b66babb8ae4	Oct 19th 08:18	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!188&cid=68048cc8-b9ef-4314-8448-a53e9ea5680a&ithint=file%2cxlsx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	8.717	300	1.3	3g	PDF
8FB98C2011CA	87dfdf092f35	Oct 19th 08:18	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!188&cid=68048cc8-b9ef-4314-8448-a53e9ea5680a&ithint=file%2cxlsx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	4.299	300	1.25	3g	PDF
953B9CB011CA	87dfdf092f35	Oct 19th 08:18	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!168&cid=cb0215bc-c75a-407c-adf1-c4bc8147cba1&ithint=file%2cdocx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	2.519	300	1.25	3g	PDF
97DD97C011CA	87dfdf092f35	Oct 19th 08:18	docs.google.com	/document/d/1A_DgG8oLSuMOiW-TNvClgSWxjXOjNfgbJ-b1-UV18oE/edit	6.617	300	1.25	3g	PDF
9F985A4011CA	87dfdf092f35	Oct 19th 08:19	docs.google.com	/document/d/1A_DgG8oLSuMOiW-TNvClgSWxjXOjNfgbJ-b1-UV18oE/edit	4.828	300	1.25	3g	PDF
A686F32011CA	87dfdf092f35	Oct 19th 08:19	docs.google.com	/presentation/u/0/	3.994	300	1.25	3g	PDF
B864024011CA	f749519e0498	Oct 19th 08:19	www.office.com	/launch/forms?auth=1	1.021	300	1.4	3g	PDF
BCC5127011CA	f749519e0498	Oct 19th 08:20	docs.google.com	/presentation/d/14wsZ6VyyBh5HfrRYR6elmXH_Lbade3T6pSR1n59sA-o/edit	4.868	300	1.4	3g	PDF
057E521011CB	47d0332c43ff	Oct 19th 08:22	www.office.com	/?auth=1	1.067	300	1.45	3g	PDF
0C7883B011CB	47d0332c43ff	Oct 19th 08:22	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!188&cid=68048cc8-b9ef-4314-8448-a53e9ea5680a&ithint=file%2cxlsx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	7.686	300	1.45	3g	PDF
0E5CD41011CB	47d0332c43ff	Oct 19th 08:22	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!168&cid=cb0215bc-c75a-407c-adf1-c4bc8147cba1&ithint=file%2cdocx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	7.828	300	1.45	3g	PDF
127FC9D011CB	47d0332c43ff	Oct 19th 08:22	www.office.com	/launch/forms?auth=1	2.674	300	1.45	3g	PDF
1845E84011CB	47d0332c43ff	Oct 19th 08:22	docs.google.com	/presentation/d/14wsZ6VyyBh5HfrRYR6elmXH_Lbade3T6pSR1n59sA-o/edit#slide=id.p	5.233	300	1.45	3g	PDF
Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
1B6A80D011CB	47d0332c43ff	Oct 19th 08:22	docs.google.com	/presentation/u/0/	4.186	300	1.45	3g	PDI
1DD7B04011CB	47d0332c43ff	Oct 19th 08:22	docs.google.com	/forms/u/0/	2.222	300	1.45	3g	PDI
2568558011CB	47d0332c43ff	Oct 19th 08:22	docs.google.com	/forms/u/0/d/1-y0-heNVY1APZTGzSkUD3NoDJDyjPtWRWM-5Rl6qMkY/edit?ntd=1&usp=forms_home&ths=true	4.163	300	1.45	3g	PDI
4D95646014E3	03cb3a27a5ef	Oct 23rd 06:53	www.office.com	/?auth=1	4.646	250	10	4g	PDI
6698E54014E3	b3ec2311384b	Oct 23rd 06:54	docs.google.com	/document/d/1GZgKt-GpaYRksXxOXcSaxdY3RcuIfEcdoFMTqN41Xk0/edit	4.543	50	10	4g	PDI
7A0F802014E3	fa1fa4965b69	Oct 23rd 06:54	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!168&cid=5c488221-7f33-465e-84a5-e7c33e365f23&ithint=file%2cdocx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	4.292	50	10	4g	PDI
81D8C19014E3	fa1fa4965b69	Oct 23rd 06:54	www.office.com	/launch/word?auth=1	2.582	50	10	4g	PDI
86CBD89014E3	fa1fa4965b69	Oct 23rd 06:55	docs.google.com	/document/u/0/	3.665	50	10	4g	PDI
8960B44014E3	fa1fa4965b69	Oct 23rd 06:55	docs.google.com	/presentation/u/0/	3.338	50	10	4g	PDI
9216518014000	fa1fa4965b69	Oct 23rd 06:55	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!190&ithint=file%2cxlsx&action=editnew&wdNewAndOpenCt=1603425315598&wdPreviousSession=6fd51b49-588b-4366-97bf-d7db52e24d41&wdOrigin=OFFICECOM-WEB.START.NEW	3.006	50	10	4g	PDI
9A9A577014E3	fa1fa4965b69	Oct 23rd 06:55	docs.google.com	/presentation/d/1nJXEdRioeWUWpPFpo0Vy7dfmumK3p3HFz69EtGlnMWY/edit	6.368	50	10	4g	PDI
C0844E0014E3	80a6cee29d7f	Oct 23rd 06:56	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!192&ithint=file%2cxlsx&action=editnew&wdNewAndOpenCt=1603425373873&wdPreviousSession=13321c98-4f94-49ac-b9e3-b1ec9c8e9766&wdOrigin=OFFICECOM-WEB.START.NEW	6.527	50	10	4g	PDI

Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
F4C0583014E3	28fc76e03a46	Oct 23rd 06:58	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!192&ithint=file%2cxlsx&action=editnew&wdNewAndOpenCt=1603425373873&wdPreviousSession=13321c98-4f94-49ac-b9e3-b1ecgc8e9766&wdOrigin=OFFICECOM-WEB.START.NEW	5.634	50	10	4g	PDI
FD439AD014E3	28fc76e03a46	Oct 23rd 06:58	docs.google.com	/spreadsheets/u/0/d/1VoKsamkXURNVgXATtB6pHol7UNj90Gpdor54bh1EANQ/edit?ntd=1&usp=sheets_home&ths=true	2.192	50	10	4g	PDI
00CF329014E4	28fc76e03a46	Oct 23rd 06:58	docs.google.com	/spreadsheets/u/0/	2.682	50	10	4g	PDI
033B298014E4	28fc76e03a46	Oct 23rd 06:58	docs.google.com	/spreadsheets/u/0/create?usp=sheets_home&ths=true	1.961	50	10	4g	PDI
05E8482014E4	28fc76e03a46	Oct 23rd 06:58	www.office.com	/?auth=1	1.269	50	10	4g	PDI
0C925DF014E4	28fc76e03a46	Oct 23rd 06:58	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!168&cid=32c9ae99-4e05-4acb-a12a-cdd6915e0aa9&ithint=file%2cdocx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	6.124	50	10	4g	PDI
12D32AA014E4	28fc76e03a46	Oct 23rd 06:58	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!194&ithint=file%2cpptx&action=editnew&wdTpl=TM16401379&wdlcid=2057&wdNewAndOpenCt=1603425530623&wdPreviousSession=c9f0efd4-7211-49dd-bc7d-3bfbe9057729&wdOrigin=OFFICECOM-WEB.START.TEMPLATES	4.495	50	10	4g	PDI
26AD424014E4	28fc76e03a46	Oct 23rd 06:59	docs.google.com	/spreadsheets/u/0/	2.906	50	10	4g	PDI
3F3C085014E4	8afd8e25db83	Oct 23rd 07:00	docs.google.com	/document/d/1Ttj4bVOKseE5R1wCGqr1TnPfpioXqdNsYnEGADwkKPA/edit	5.1	0	10	4g	PDI
41AD085014E4	8afd8e25db83	Oct 23rd 07:00	www.office.com	/?auth=1	1.506	0	10	4g	PDI
502B0DF014E4	8afd8e25db83	Oct 23rd 07:00	www.office.com	/?auth=1	1.094	0	10	4g	PDI
D58AE9A01752	b9e894cbd74e	Oct 26th 09:16	docs.google.com	/document/d/1PZXZwMDKFu1eFhm3yqxJYtH_bNClbLD8hYhlAtWQuKs/edit	6.257	300	0.7	3g	PDI
D8DBE9601752	b9e894cbd74e	Oct 26th 09:16	www.office.com	/?auth=1	5.971	300	0.7	3g	PDI

Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
E0705F301752	b9e894cbd74e	Oct 26th 09:17	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!168&cid=51948194-5ba0-4b59-b123-41da2beaac94&ithint=file%2cdocx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	7.861	300	0.7	3g	PDI
0EFEDC001753	88d927aee763	Oct 26th 09:18	docs.google.com	/presentation/d/1OgkEFH6LhnSGy6e_cXZylZYaGml1oi6AvbEF5BAoOlg/edit	7.66	350	1.45	3g	PDI
3B9C37301753	78574580904d	Oct 26th 09:19	www.office.com	/?auth=1	1.42	250	10	4g	PDI
3F1383A01753	78574580904d	Oct 26th 09:19	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!198&ithint=file%2cxlsx&action=editnew&wdTpl=TM16400654&wdlcid=2057&wdNewAndOpenCt=1603693175815&wdPreviousSession=a0098dfc-55a7-4a8e-acbb-64d6122de764&wdOrigin=OFFICECOM-WEB.START.TEMPLATES	8.722	250	10	4g	PDI
47138C801753	78574580904d	Oct 26th 09:20	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!200&ithint=file%2cdocx&action=editnew&wdTpl=TM00002138&wdlcid=2057&wdNewAndOpenCt=1603693194524&wdPreviousSession=0d8d6139-10b8-44f0-a57a-c71e23b12939&wdOrigin=OFFICECOM-WEB.START.TEMPLATES	4.013	250	10	4g	PDI
4C99E2D01753	78574580904d	Oct 26th 09:20	docs.google.com	/presentation/u/0/	2.564	250	10	4g	PDI
Infinity	78574580904d	Oct 26th 09:20	docs.google.com	/spreadsheets/u/0/	2.881	250	10	4g	PDI
8D7966401753	4854f0fc46d5	Oct 26th 09:21	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!198&ithint=file%2cxlsx&action=editnew&wdTpl=TM16400654&wdlcid=2057&wdNewAndOpenCt=1603693175815&wdPreviousSession=a0098dfc-55a7-4a8e-acbb-64d6122de764&wdOrigin=OFFICECOM-WEB.START.TEMPLATES	4.832	250	1.3	3g	PDI
Infinity	4854f0fc46d5	Oct 26th 09:22	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!192&cid=d358916g-0a8a-4097-aeff-5ce04c86dd65&ithint=file%2cxlsx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	5.762	250	1.3	3g	PDI

Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Search

Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
942B44401753	4854f0fc46d5	Oct 26th 09:22	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!200&ithint=file%2cdocx&action=editnew&wdTpl=TM00002138&wdlcid=2057&wdNewAndOpenCt=1603693194524&wdPreviousSession=0d8d6139-10b8-44f0-a57a-c71e23b12939&wdOrigin=OFFICECOM-WEB.START.TEMPLATES	3.668	250	1.3	3g	PDF
B8DB34301753	11aabd3d9fe8	Oct 26th 09:23	onedrive.live.com	/edit.aspx?action=edit&resid=6937FA95C77474EF!196&ithint=file%2cpptx&action=editnew&wdTpl=TM16401379&wdlcid=2057&wdNewAndOpenCt=1603693052896&wdPreviousSession=eed5b12d-c045-40a3-855e-02ef53c9dd7a&wdOrigin=OFFICECOM-WEB.START.TEMPLATES	6.906	300	10	4g	PDF
D4EA31301820	6da81201c645	Oct 27th 09:51	www.office.com	/?auth=1	1.494	250	1.6	4g	PDF
DDEB08401820	6da81201c645	Oct 27th 09:51	docs.google.com	/document/u/0/	5.621	50	10	4g	PDF
E34C71701820	6da81201c645	Oct 27th 09:51	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!192&cid=3b5a9b0e-3b8b-4020-ad1f-bb3f3c7a1611&ithint=file%2cxlsx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	8.323	50	10	4g	PDF
E60A0AD01820	6da81201c645	Oct 27th 09:51	www.office.com	/?auth=1	1.198	50	10	4g	PDF
EBA855F01820	6da81201c645	Oct 27th 09:52	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!194&cid=c61ef867-315f-448f-b901-1189c1e7ff6b&ithint=file%2cpptx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	5.603	50	10	4g	PDF
EC6A1F501820	6da81201c645	Oct 27th 09:52	docs.google.com	/document/d/13eDwCnmsYOhoEvRzDXleZtmIgMxx2M5hqSC3iJfbWLA/edit	2.903	50	10	4g	PDF
181A0D401821	edf84b3f5441	Oct 27th 09:53	docs.google.com	/presentation/d/1clFmiC5O5WGSUhkOSfEdHDZLXfKJyW7oF8SgbPd_QHw/edit#slide=id.gcb9a0b074_1_0	4.865	50	10	4g	PDF
1F9A37C01821	edf84b3f5441	Oct 27th 09:53	www.office.com	/launch/word?auth=1	1.764	50	10	4g	PDF
270357D01821	edf84b3f5441	Oct 27th 09:53	docs.google.com	/presentation/u/0/	3.132	50	10	4g	PDF
2AA1DB501821	edf84b3f5441	Oct 27th 09:53	docs.google.com	/document/u/0/	3.601	50	10	4g	PDF
2FAA4F101821	edf84b3f5441	Oct 27th 09:53	docs.google.com	/document/u/0/	3.363	50	10	4g	PDF

Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
8CAA4AD01821	f58a77ea539e	Oct 27th 09:56	docs.google.com	/presentation/d/1GW73q1C_fi1uMr4lmg_HKS8z8rlzr5l5GHTkPH2l14c/edit#slide=id.gcbga0b074_1_0	4.394	50	10	4g	PDF
90777F701821	f58a77ea539e	Oct 27th 09:56	docs.google.com	/presentation/u/0/	3.474	50	10	4g	PDF
92806ED01821	f58a77ea539e	Oct 27th 09:56	docs.google.com	/forms/u/0/	1.819	50	10	4g	PDF
961666D01821	f58a77ea539e	Oct 27th 09:56	docs.google.com	/forms/u/0/d/1d0vFJ4Qnr7cVNJljwC37Clev6FK79KPOgV7kkmR-leU/edit?ntd=1&usp=forms_home&ths=true	3.217	50	10	4g	PDF
9B4771301821	f58a77ea539e	Oct 27th 09:56	docs.google.com	/forms/u/0/	1.626	50	10	4g	PDF
7EE8C1F01822	992de6a679a6	Oct 27th 10:03	docs.google.com	/document/d/1ZOSpSwplHiAt55vKERdMdpSe-SiMJSVKs7cqpn-qHb8/edit	5.199	50	10	4g	PDF
824B9C001822	992de6a679a6	Oct 27th 10:03	docs.google.com	/forms/u/0/	1.5	50	10	4g	PDF
86A3AB801822	992de6a679a6	Oct 27th 10:03	docs.google.com	/forms/u/0/d/1O2gEe8rDY2xcQE7tVK8jj-MM13clkzdFZfy3FclBuWM/edit?ntd=1&usp=forms_home&ths=true	3.683	50	10	4g	PDF
A2CF47601822	fd652f9cee13	Oct 27th 10:04	docs.google.com	/spreadsheets/d/1Ypv7XHMg9Ql-QrDUrbu_kbQqwDWx5gNTzE-ToTONVC4/edit#gid=1386834576	5.156	50	10	4g	PDF
A5C555E01822	fd652f9cee13	Oct 27th 10:04	docs.google.com	/spreadsheets/u/0/	3.488	50	10	4g	PDF
A83719301822	fd652f9cee13	Oct 27th 10:04	docs.google.com	/presentation/u/0/	3.375	50	10	4g	PDF
AEB082101822	fd652f9cee13	Oct 27th 10:04	docs.google.com	/presentation/d/132lYgNrBWq_fOMYXIFREX96WBD89ExpxV2mAUkKFiHY/edit	6.123	50	10	4g	PDF
B1FC2AA01822	fd652f9cee13	Oct 27th 10:04	docs.google.com	/presentation/u/0/	2.829	50	10	4g	PDF
B3E7A6F01822	fd652f9cee13	Oct 27th 10:04	docs.google.com	/forms/u/0/	2.392	50	10	4g	PDF
B7245B601822	fd652f9cee13	Oct 27th 10:04	docs.google.com	/forms/d/1mD0aeg3zTefhfoEfivdCZNPhz9el040LI_HRQIT1bwc/edit	3.106	50	10	4g	PDF
C3106B801822	fd652f9cee13	Oct 27th 10:05	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!192&cid=9aef3ef1-5bfc-4dc6-8d83-70779d8eef45&ithint=file%2cxlsx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	5.594	50	10	4g	PDF
C60457201822	fd652f9cee13	Oct 27th 10:05	www.office.com	/?auth=1	14.863	50	10	4g	PDF

Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Monitor ID	Operations ID	Search Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	
CgEg0B601822	fd652f9cee13	Oct 27th 10:05	onedrive.live.com	/edit.aspx?resid=6937FA95C77474EF!196&cid=3822b402-0c55-4e04-ba19-dbe1b8667894&ithint=file%2cpptx&wdOrigin=OFFICECOM-WEB.MAIN.MRU	6.865	50	10	4g	PDF
CF78B0801822	fd652f9cee13	Oct 27th 10:05	www.office.com	/launch/excel?auth=1	1.871	50	10	4g	PDF
D37E81501822	fd652f9cee13	Oct 27th 10:05	www.office.com	/launch/word?auth=1	1.567	50	10	4g	PDF
D982E1401822	fd652f9cee13	Oct 27th 10:05	onedrive.live.com	/edit.aspx?action=editnew&resid=6937FA95C77474EF!202&ithint=file%2cdocx&action=editnew&wdTpl=TM00002138&wdlcid=2057&wdNewAndOpenCt=1603782343791&wdPreviousSession=59037d58-08f4-49aa-b636-e7f55dcd4103&wdOrigin=OFFICECOM-WEB.START.TEMPLATES	6.361	50	10	4g	PDF
FD5F06701822	a230baf49c09	Oct 27th 10:06	www.office.com	/launch/powerpoint?ui=en-US&rs=GB&auth=1	1.237	250	8.4	4g	PDF
Monitor ID	Operations ID	Timestamp	Main URL	Partial URL	Load Time (sec)	Net RTT	Net Downlink	Net ET	

Showing 1 to 162 of 162 entries (filtered from 173 total entries)

Linked Publications

Makokha, F., Chepken, C. K. & Opiyo, T. O. (2021). End User Centric Quantitative Trust Model in Cloud Computing. *American Journal of Computer Science and Engineering*. Vol. 7, No. 1, 2021, pp. 1-7.

Makokha, F., Chepken, C. K. & Opiyo, E. (2020). A Comparative Study of a Client Based Vendor Neutral Cloud QoS Monitoring Tool and Cloud Providers' Platform Integrated QoS Monitoring Tools. *European Journal of Electrical Engineering and Computer Science*. 4, 1 (Jan. 2020).

Makokha, F., Opiyo, E. E. & Chepken, C. K. (2019). Browser Integrated Vendor Neutral Cloud QoS Monitoring System. *International Journal of Computer and Information Technology* Volume 08 – Issue 06, November 2019.

Makokha, F. & Opiyo, E. (2018). A Vendor Neutral QoS Monitoring Model for SaaS Cloud Computing Solutions. *International Journal of Computer and Information Technology* Volume 07– Issue 01, January 2018.

Makokha, F., Opiyo, E & Okelo-Odongo (2017). Challenges of QoS Monitoring in Cloud Computing Solutions. *International Journal of Computer and Information Technology* Volume 06– Issue 06, November 2017.