

https://dspace.org/

A feasibility study for a voucher management system (VOMS) as part of intelligent network (IN) architecture for telecommunications operations in a changing customer service market

Collender, Jonathan Ffyner

2016
http://hdl.handle.net/10539/20633

Downloaded from a DSpace repository. https://dspace.org/

CHAPTER 3

RESEARCH DESIGN AND METHODOLOGY

3.1 INTRODUCTION

In this chapter the research design of the study and the methodology that was applied are elucidated. The theoretical viewpoints on the research design have been established and presented. In the study use was made of an experimental design, and a comparative analysis was done. The methodology employed to collect the data was that of laboratory experiments. This is now taken further with an explanation of the methodology and the procedures followed in conducting the research.

The methodology included the following steps:

- A literature review and analysis of the effects of a changing consumer market
- A literature review of engineering concepts
- Integration of engineering products to deliver the VOMS emulator in a laboratory
- Testing of a number of scenarios to prove the attainment (or not) of the objectives of the investigation.

After a literature review (*cf*. Chapter 2) had been conducted on the individual aspects of optimal system design and high service availability structures, enhancements that were required to create an enhanced VOMS system were discussed. These components were analysed and, based on the attributes of those concepts, the new VOMS system was designed.

The purpose of this study was to determine if it would be possible to design a VOMS system that would have an application service availability extending beyond five 9's by using commodity hardware with high-availability techniques. Second, the study strived to determine whether this new design of such a system would be economically feasible to implement as compared with an existing VOMS system.

Following is a discussion on the construction of a newly proposed VOMS system; the breakdown of the components used, the resilience under failure conditions; and the laboratory setup will receive attention.

The reliability and validity and ethical considerations applicable to the study also will be discussed.

3.2 LITERATURE REVIEW

The first study method that was employed to collect data was the literature study. The purpose of the literature review was to offer an overview of the topic in order to develop the context of the research problem, to establish the need for the study and to indicate the knowledge acquired about the various high-availability technologies.

The literature review focused on customer experience, prepaid offerings and high-availability techniques, as discussed in Chapter 2.

3.3 INTELLIGENT NETWORKS (IN)

VOMS as a stand-alone system cannot function without the IN or Service Control Point (SCP) system. IN is described as a significant part of a modern mobile operator's network. As described by Zuidweg [2], the purpose for an SCP, or more commonly known as IN, is to detach the call logic handling from the switching network and to pass it on to IN. A System Information Block or SIBs are used to develop a prompt service creation in an IN architecture. VAS offerings are provided by IN; these include prepaid services and other functions such as call forwarding, along with call screening. IN standards and the evolution of IN are published as Capability Sets (CS). In the scope of this research report, the term IN describes the service system that exists within the prepaid environment. In particular, this partition of the IN is used to provide appropriate billing for specific tariffs, a form of dynamic billing [2]. In this research report, the context of VOMS is the mechanism required to add services on the IN system.

While this system was not within the direct scope of the project, it was none the less a downstream system to which the VOMS would connect. It thus became important with regard to the architecture and how VOMS connected and interfaced with an IN system.

3.4 VOUCHER MANAGEMENT SYSTEM

The Voucher Management Systems (VOMS) that exist today are implemented differently across mobile operators and there are many vendors who offer such solutions. Many vendors offer VOMS as part of their PrePaid offering to mobile operators, in which case it usually will be part of their IN or SCP architecture.

Due to the complexity within the SCP, most mobile operators use vendors to supply them with a VOMS solution. VOMS, in essence, is not a complex system and could be customised if needed.

Certain vendors were rigid in their offerings and in the event of changes or promotions that were required by mobile operators, vendors would usually have set deployment time frames and additionally would charge hefty premiums for their patches and deployments. VOMS was an independent system and could be developed as such.

VOMS formed part of the PrePaid service and was seen as a function of IN. It is primarily responsible for generating the voucher pins for mobile operators. Within this context, the same system was used to redeem and validate the vouchers and provide the corresponding amount to be added to a subscriber account on the IN [1]. This system was the entry point for subscribers to add money to their PrePaid accounts. If subscribers were unable to recharge their accounts, they could still make calls until their money on the SCP ran out, and then be unable to load additional funds onto their account. If subscribers were unable to make calls, due to an outage because of a VOMS, this could have an impact on a mobile operator's subscriber base and would lead to possible customer dissatisfaction, affecting loyalty (*cf.* 2.3).

VOMS could be configured to provide a charging or recharging service for its subscriber base. A recharging system configuration was a system where vouchers were redeemed and money was added to an IN or SCP account. When a charging function was required, a subscriber would perform a purchase, and through the transaction funds would be transferred from their IN or SCP account.

For physical vouchers, the primary function of a VOMS was to generate voucher PIN numbers, which would be distributed by a mobile operator. Such a process began at the requestor, usually another department within a mobile operator, which would receive a request(s) from retail shops for specific fixed denominations to be generated. Note that physical vouchers have a fixed denomination. The request is sent to the VOMS to generate the voucher PINs and link them to a specific denomination. While the most popular types of recharges are those in the form of physical PIN numbers which are randomly generated, PIN-less vouchers are available to such entities such as banks. In this scenario, the banks create a facility for subscribers to perform a recharge by transferring funds directly from a subscriber's bank account and to purchase airtime from most mobile operators. In these cases the banks do not require a mobile operator to provide them with physical voucher numbers when subscribers want to purchase airtime. The banks have a direct connection to mobile operators. Various interface protocols exist which banks may use to perform these recharge requests, including the most common, the PostBridge interface, which makes use of the ISO 8583 standard [6].

When the PrePaid system was introduced first, the only function on VOMS was voucher generation and voucher redemption. This was all that subscribers were offered, which were voice calls in essence. As VAS matured and data were introduced with faster and faster access speeds, VOMS could be expanded for further use.

Subscribers could redeem vouchers through various methods, including the internet through a web site, and by logging on to a mobile operator's web page, it was possible to redeem vouchers and recharge a subscriber's account. Furthermore, due to the VAS component, subscribers were able to perform various other transactions offered by mobile operators, including methods to purchase Value Added Services (VAS) products, which included Short Message Service (SMS) bundles, data bundles, Multimedia Messaging Service (MMS) bundles, and other promotional products for sale. Methods or channels used to redeem physical vouchers were the internet, SMS, Unstructured Supplementary Service Data (USSD), Interactive Voice Response (IVR) and Wireless Application Protocol (WAP) [6], [40].

These other channels listed all have connections to VOMS and are able to interface with VOMS for the subscribers to redeem their vouchers. All mobile operators had their own

unique short code offered to their subscribers for voucher redemption by means of this channel.

Depending on a mobile operator and architecture of such a VOMS implementation, it may be used for keeping a database of subscribers.

When provisioning new subscribers on a mobile network for the first time, various implementations are possible, depending on the size of a mobile operator. For smaller mobile operators, it would have been possible to pre-provision subscribers on a VOMS. A mobile operator's sim packs for a PrePaid offering will have those same numbers printed on them. When a subscriber activates the sim card, Home Location Register (HLR) at some point in the activation process become populated with the new sim card's Integrated Circuit Card Identification (ICCID) and the subscriber will already be on the VOMS or PrePaid database. The IN or SCP could also be pre-provisioned in advance and this process leads to a very quick activation for a subscriber.

In larger mobile operators or a larger subscriber base, a problem that could be encountered might be that pre-provisioning could clash with a mobile operator's inactivity clause and the amount of freely available cellular numbers. Thus, if a mobile operator's starter packs remained with retailers for longer than a mobile operator's inactivity period the number would be deactivated, and that would cause a problem for the subscribers upon activation. The exact method employed varies from mobile operator to mobile operator.

3.5 AN EXISTING SETUP OF A VOUCHER MANAGEMENT SYSTEM

The internal configuration of this VOMS resembled the architecture as depicted in Figure 3.1. This configuration is a standard server architectural design. The figure shows the classic client-server relationship - where a client would perform a transaction, which would be passed on to a server [41].

The server would act on the client's requests and react with an appropriate response. The strength of this design is shown in its simplicity, in having the application and the database on the same system. All the software components were installed on one system, thus making the system administration less complex. For this comparison, the software

used in the implementation of VOMS was proprietary. This comparison was based on a system that used HP-NSK as its core system. The cost of the system was high as it was proprietary. The rationale for using this implementation was the advanced software and hardware components used. Though this was a good overall system to process transactions, problems could occur. No single system would be immune to a complete system failure.

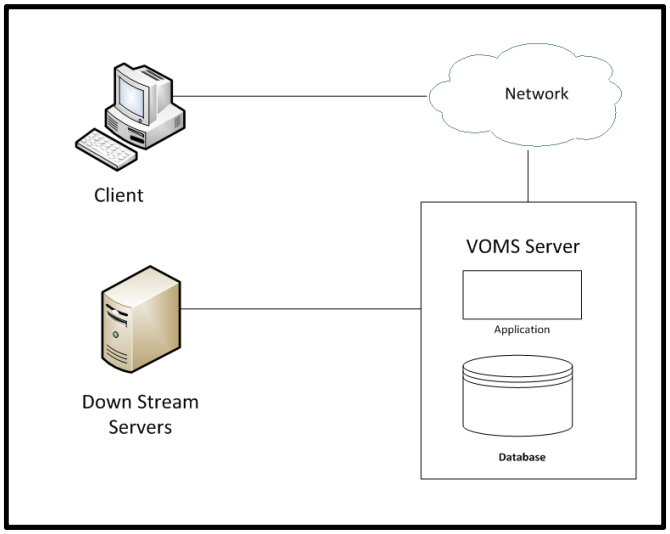

FIGURE 3.1: Layout of an existing VOMS

As seen in Figure 3.1, the database and application servers were deployed on one single system. There was no physical system separation between the database and the application. In the event of a loss of one of the components, be it the database or application, client access to the services provided could be lost. A strategy to alleviate such a situation would be to create an identical system in a separate geographical location, a disaster recovery server.

This configuration was known as a redundant architecture where an active site would process transactions until there was a loss of the primary site. The secondary, or then DR site, did not process any transactions up until there was a failure on the primary system.

This system configuration would provide a high service availability of close to five 9's. In a disaster, it would be required to cut over to the DR system. A drawback to such architecture was that it was expensive for small mobile operators to implement a system

that did nothing until there was a failure. The failover procedure could take time and the amount of time required would be determined by various factors (the response time of the support staff to intervene in the failover procedure and starting up the processes on the DR system are such examples). The longer an outage, the more serious the consequences could become to any mobile operator and its subscribers, which could include damage to their brand and a potential for revenue loss. The consequences of such influential factors were discussed in Chapter 2.

3.6 NEWLY PROPOSED IDEA FOR A VOUCHER MANAGEMENT SYSTEM

The fundamental idea behind the new VOMS was that it would be impervious to failures based on the unique architecture that it would have. Many high availability concepts would be required to construct such a system (*cf*. Chapter 2). The holistic design and use of these many techniques should ensure that the new VOMS would not stop providing its services.

The core high availability techniques used to construct the new VOMS were:

3.6.1 Hardware clustering

Blade hardware was clustered together to form a single cluster system. The virtual machines are virtualised on the blade hardware. If the virtual machines had been allocated all available physical hardware resources, more hardware blades could be clustered together. The blades only supplied the virtual machines with CPU and memory resources, if more disk space was required, more disk storage arrays could be added to the architecture. Figure 3.2 depicts such a hardware layer, with the blade and storage servers.

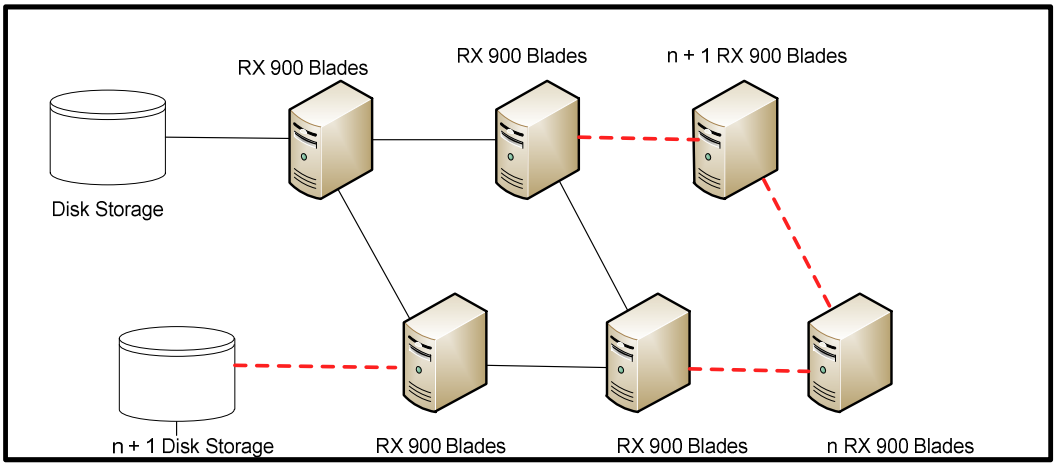

FIGURE 3.2: Clustered hardware

3.6.2 Virtual machines

Virtual machines are configured to be installed on top of the clustered hardware. Multiple instances of the application will be able to run in parallel. Virtual machines could be sized dynamically with respect to their allocated resources. More CPU and memory can be allocated per virtual machine if it were required. Figure 3.3 indicates the application front-end and the database back-end virtual machines.

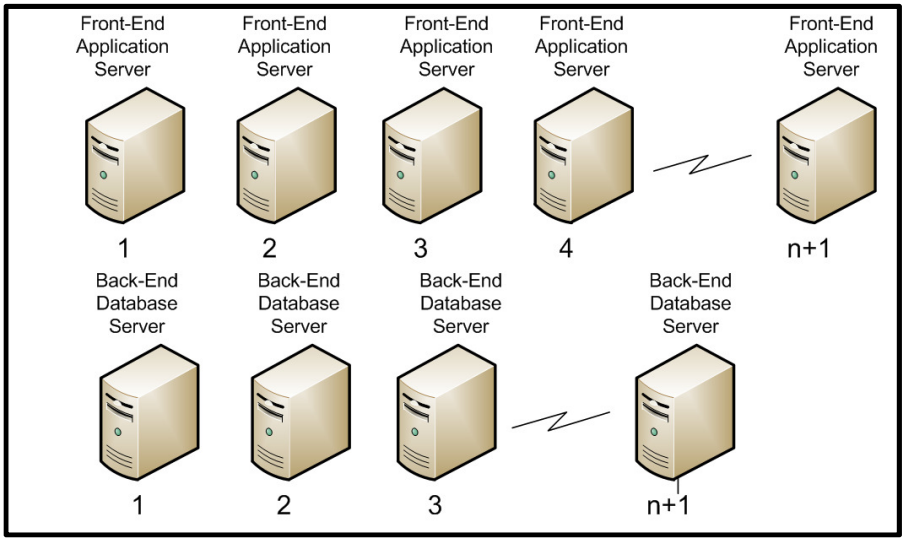

FIGURE 3.3: Virtual machines

3.6.3 Networking components

The use of a multi-layered switch and an ACE module was used in the design for various reasons, as it would provide an abstract connection layer by providing the client a Virtual Internet Protocol (VIP) address. Thus the clients would then establish a network connection with the ACE and not the front-end server directly. Furthermore, the ACE modules had the ability to interrogate the front-end and back-end servers to determine if they were available. The ACE module and switch would be the backbone of the new VOMS. Figure 3.4 illustrates the network layout for the new VOMS.

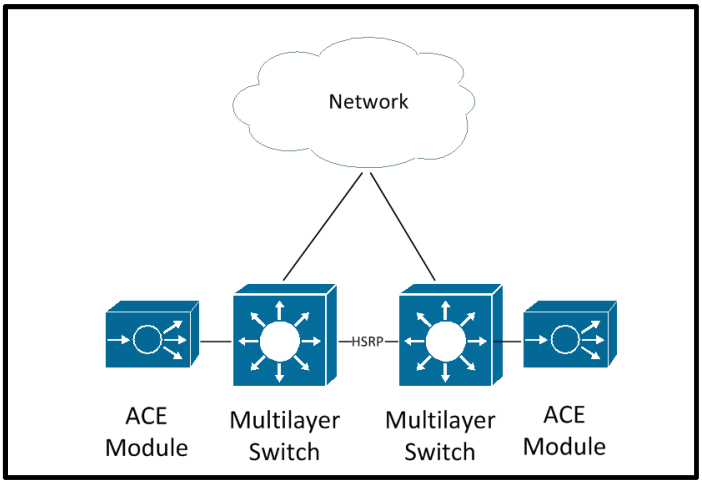

FIGURE 3.4: Network components

Figure 3.5 provides a combined view of the Virtual Machines hosted on clustered hardware along with the network layer.

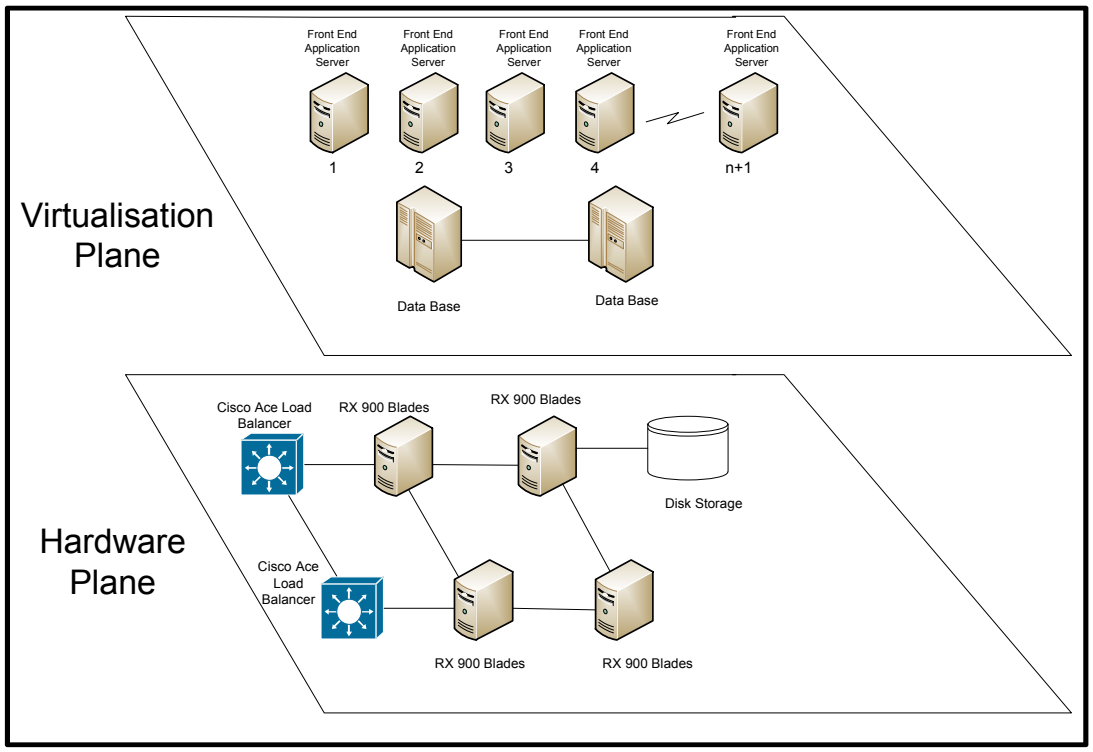

FIGURE 3.5: A breakdown of the physical and virtual planes to indicate the architectural layout

3.6.4 The newly proposed logical architecture

Figure 3.6 shows the new architecture and the arrangement of the components required to achieve the high service availability. Note that a key feature in Figure 3.6 was that the database had been physically separated from the application server. The incorporation of networking aids, that is, Cisco load balancers, had been introduced as well as the use of a database cluster. The main rationale behind the design was that should a loss of a single application server or database server occur, the service would not be disrupted.

The existing VOMS was not immune against a system failure or disaster. If the application(s) were to fail the system would not respond to client requests and the VOMS could become unavailable. The same could happen with the database, in the event of a database failure; the system could be unresponsive and unavailable. There were many such scenarios where potential threats could cause an existing VOMS to be unavailable.

The strength of the new VOMS can be attributed to its design being of such a nature that there is no single point of failure. Other factors, such as cost, are important and should be considered. A cost analysis of an existing VOMS system demonstrated that it was expensive to purchase and to maintain compared with a newly proposed VOMS. (Chapter 4, [4.8] provides details on the cost comparison of the newly proposed system which are summarised in Table 4.1.) The cost of technical staff should be less, as there is more staff with general knowledge, which is required for commodity systems as compared to the limited technical staff with proprietary system knowledge.

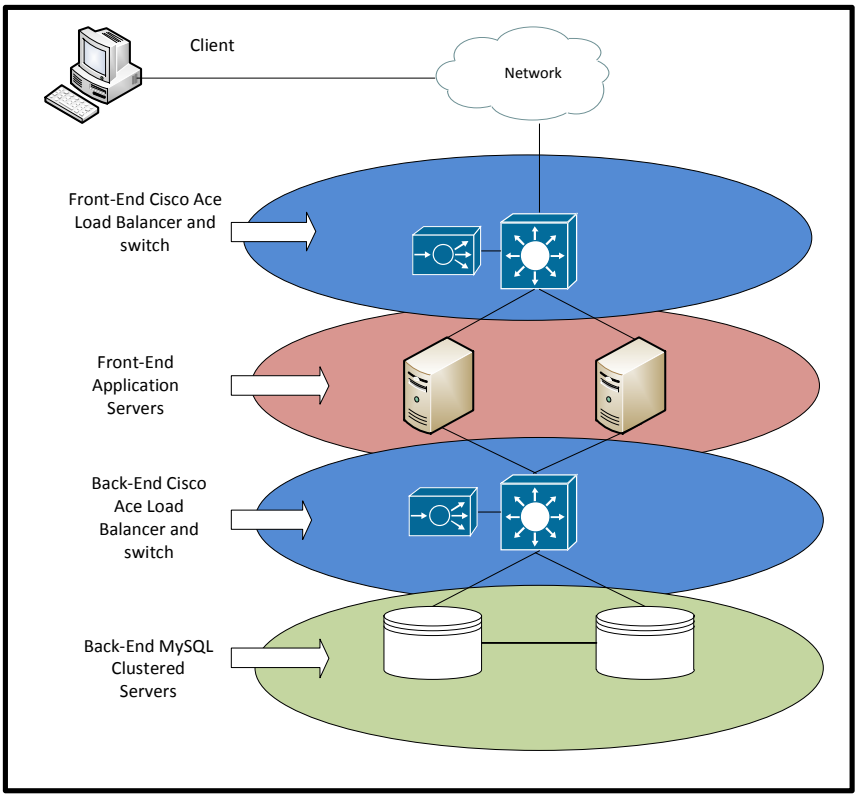

FIGURE 3.6: Newly proposed logical architectural layout for VOMS

In the following section each component used in the new design, as depicted in Figure 3.6, will be elucidated.

3.6.5 Front-end Cisco ACE Load Balancers

The front-end Cisco ACE load balancers provided access for clients to the front-end servers for the services that were offered by the applications. The clients were all given (Virtual Internet Protocol) VIP addresses. The VIP addresses were a single IP address that each client could use. The benefit of this setup would be that clients would not need to know all the various IP addresses of all the application servers.

The ACE was configured to perform health checks against the application servers. The health checks could determine if the servers were up and available; and respond in a timeous manner toward ACE. In the event that the application server did not respond to any health checks they would be marked as unresponsive and be taken out of operation by the ACE. The ACE would then no longer send traffic to that specific application server.

In the configuration for the ACE, a path would be defined for the secondary application server and this would still remain operational. Traffic would only be sent to the responding server and not to the server that was not responding. The health checks would persist and continuously check to determine when the failed server came back into service. When this happened, and depending on the configuration of the ACE, it would bring the server back into service and start directing traffic toward that server again. Appendix A shows the configuration required.

3.6.6 Front-end Application Servers

The front-end application servers held the application programs and were used to provide the services that were offered by VOMS. A VOMS has various executable programs that provide services. The VOMS had grown with its service offerings and did more than just generate and redeem vouchers. This system was used for various functions, which included querying, provisioning and promotional offerings. When new PrePaid subscribers were provisioned on a mobile operator's network they were provisioned on VOMS as one of the many systems required for a subscriber to have the ability to make a call.

Internal and external clients gained access to the services on VOMS via the front-end Cisco ACE load balancers.

The application servers in this design ran independently from each other. All application servers ran the exact same programs and were configured to be clones of each other.

3.6.7 Back-end Cisco ACE Load Balancers

The ACE in this context was configured to perform health checks against the MySQL servers and could determine if they were up and running, and responding in a timeous manner to the ACE.

In the event that the MySQL servers did not respond to the health checks, they would be marked as unresponsive and be taken out of operation by the ACE. The ACE would then no longer send traffic to that specific MySQL server. In the configuration for the ACE, a path for the secondary MySQL server would be defined and this would still remain operational. Traffic would only be sent to the server that was responsive and not to the server that was not responding.

The health checks would persist and continuously check to detect when the failed server would come back into service. When this happened and depending on the configuration of the ACE, it would put the server back into service and start directing traffic toward that server again. Appendix A shows the configuration required for the setup of such a system.

3.6.8 Database

The database would be required to be resilient; the choice for a clustered database was made based on the literature survey undertaken. MySQL was chosen as the database engine. The justification of clustering was described in detail in Chapter 2. Having access to both databases would improve the throughput of transactions as it would be possible to transact on both systems concurrently.

Table 3.1 shows a comparison between an existing VOMS and the newly proposed VOMS with regard to their differences.

Table 3.1: A comparison of the current VOMS and the newly proposed VOMS

Attributes	Current VOMS	New VOMS
Architecture	Simple	Complex
Cost of hardware	Expensive - Carrier grade	Fairly priced – commodity hardware
Operating System	Guardian (Proprietary)	Linux (Open source)
Database	SQLCI MP/MX (Proprietary)	MySQL (Non-proprietary)
Scalability	Limited system size to 16 CPUs	Theoretically unlimited
Ease of scaling up	Not scalable	Hot scaling (Could be done while the system was live)
Availability	Active-Passive - Redundant	Active-Active – very high service availability
Operating costs	Expensive (Proprietary)	Fair (Non- proprietary)

3.7 THE THEORETICAL EXPECTATION OF THE EXPERIMENT

There were various components that were required for the formation of the new VOMS, based on the literature review; all the techniques discussed in Chapter 2 were applied.

3.7.1 Algorithm used for the experimentation

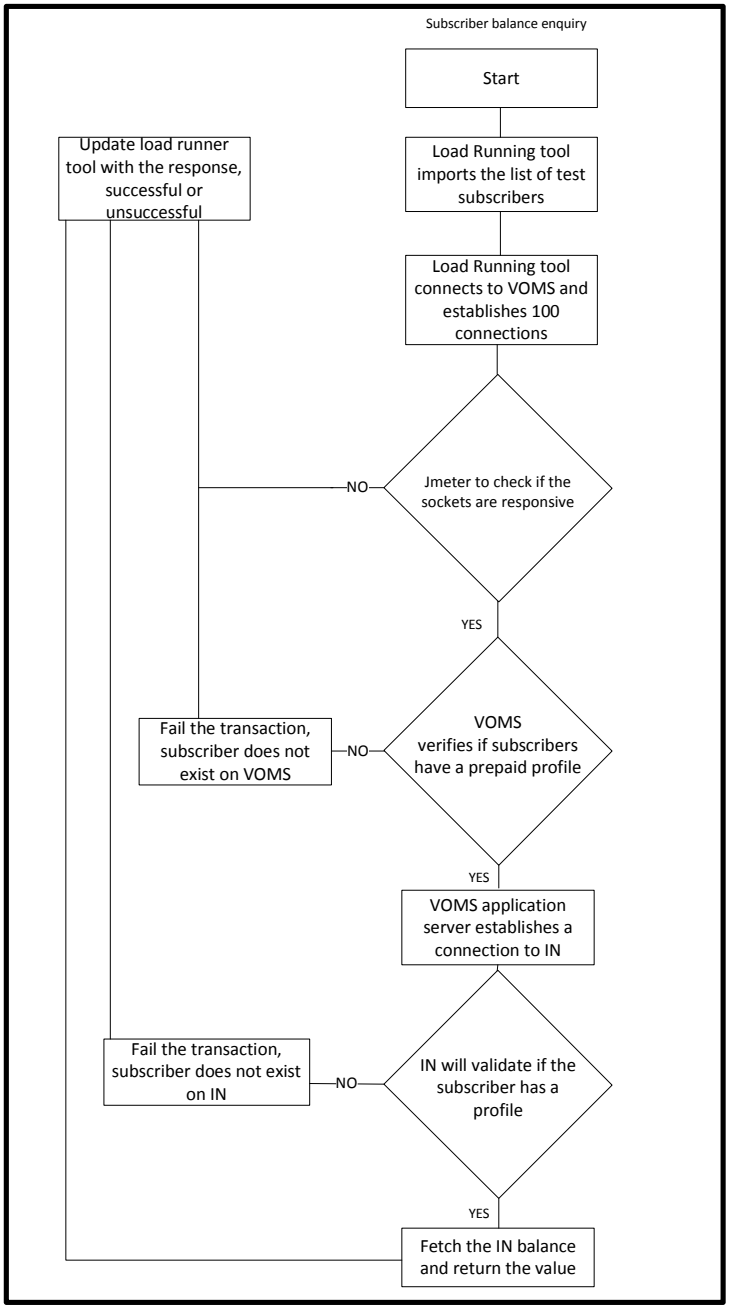

FIGURE 3.7: Transaction algorithm used for the experiment

Figure 3.7 shows the algorithm of the systems involved for the experiment. The algorithm specifically indicates how Java JMeter would view failures versus successes. Successes would only be attained if a valid response has been received from the downstream IN. Failures are noted as no responses, that is, responses where subscribers do not exist on the databases and when the responses are invalid.

3.7.2 The message flow with no failures

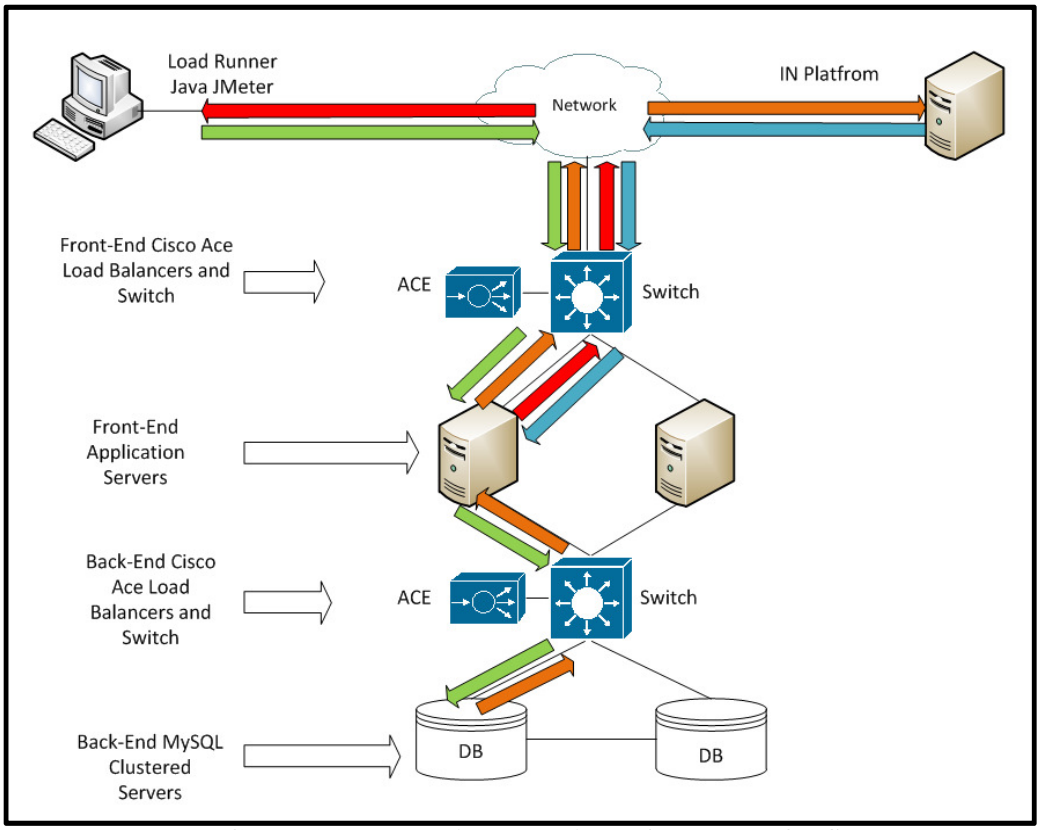

FIGURE 3.8: Theoretical behaviour of the new VOMS

Figure 3.8 illustrates the transaction flow from the load runner application, the Java JMeter. Transactions would be sent to the VIP of the new VOMS. The network would route the transaction based on the routing tables of the network to the front-end load balancers, which would forward the transaction to the front-end application server. The choice of the application server to which to route, is made by the ACE module (*cf.* Appendix A). This transaction is highlighted in green. Based on the algorithm (Figure 3.6), the application would be required to verify if the MSISDN within the transaction is valid, and perform a check against its database. The transaction would then be passed on towards the database VIP.

The response from the database back to the application (orange line) would determine if the transaction would abort if it does not exist, or continue to the IN if it does (orange line). The application server would then send the transaction to the IN. Upon reaching the

IN platform (see Figure 3.8), the MSISDN would again be verified if it does exist within the IN platform. If it does, the Rand value would be sent back (blue line). The application server would acknowledge receiving the transaction back from the IN platform and would then pass the result back to the client - in this case, the load running tool.

The load running tool will acknowledge this successful transaction. The load running tool has a graph that is created to show the successful transaction vs. unsuccessful transactions. The successful transactions are indicated in blue compared to the unsuccessful transactions which are in red.

3.7.3 The message flow with a front-end failure

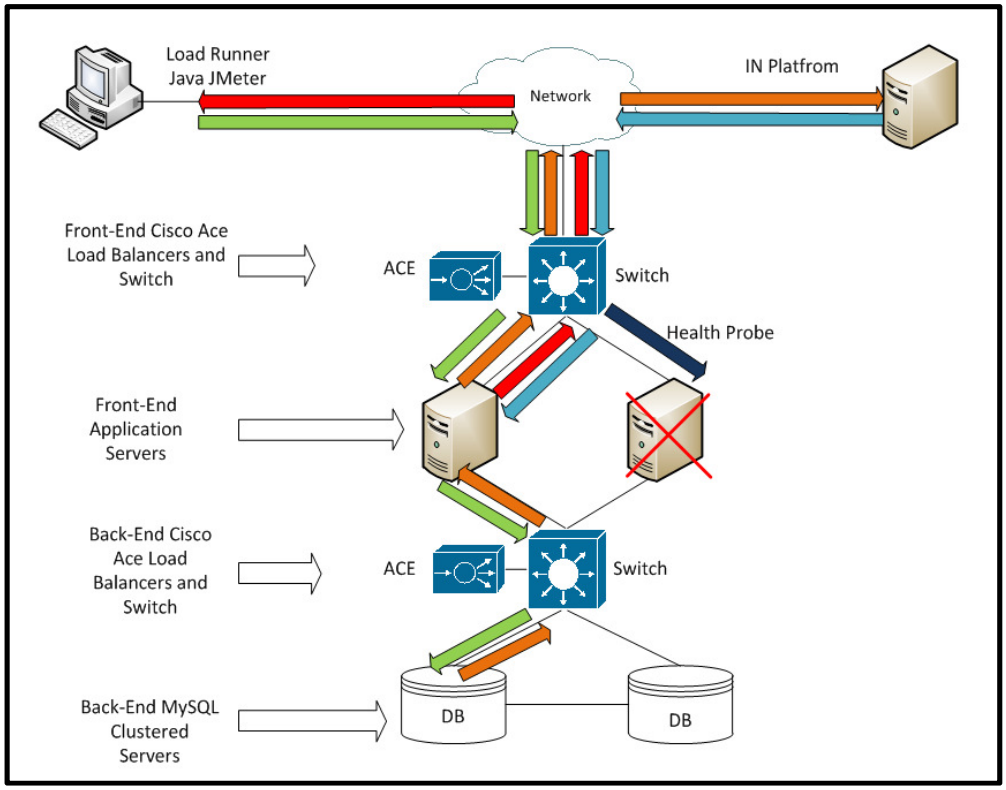

FIGURE 3.9: Theoretical behaviour of the new VOMS during a loss of the front-end application server

Figure 3.9 shows the scenario of the loss of a front-end application server. The load balancer would determine, by means of a health probe that the application server is unresponsive and thus will remove it from the pool of available servers for processing transactions. When a transaction is received from the client, the ACE modules will know

which application servers are available in the pool of servers and direct the transaction accordingly.

3.7.4 The message flow with a back-end failure

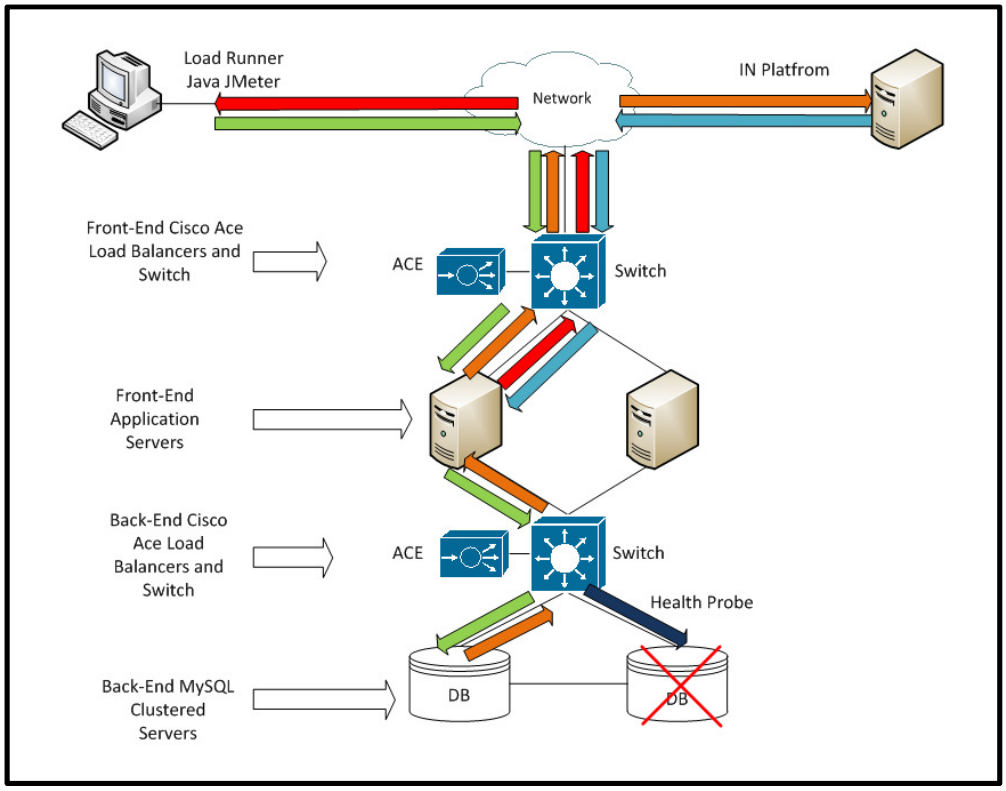

FIGURE 3.10: Theoretical behaviour of the new VOMS during a loss of the back-end database server

Figure 3.10 shows the scenario of the loss of a back-end database server. The load balancer would determine, by means of a health probe that the database server is unresponsive and thus remove it from the pool of available servers for processing transactions. When a transaction is received from the front-end application server, the ACE module will know which back-end database servers are available in the pool of servers and direct the transaction accordingly.

3.8 THE EXPERIMENT

Two months were required to configure the test equipment in a laboratory. Many issues were encountered in the configuration and setup phases. A specific application was used for the emulations to prove that the new system could handle the loss of individual components and remain available.

The scope of this research project excludes testing the resilient nature of the Cisco front-end and back-end systems, as this would not add any significance to this study and there are manuals that discuss how Cisco systems handle their system failovers [42]. Cisco load balancers were configured to be setup in HA mode. A configuration known as HSRP would control any failures experienced by the Cisco equipment. The focus thus would be on creating a high availability application and database to determine if the application could be split across and run on multiple machines independently (this was discussed in Chapter 2). Figure 3.11 depicts the layout of the design that was used to determine if it would be possible to achieve an HA VOMS system. Annotations show the individual components used to construct the newly proposed system layout.

3.8.1 The model used for the experiments

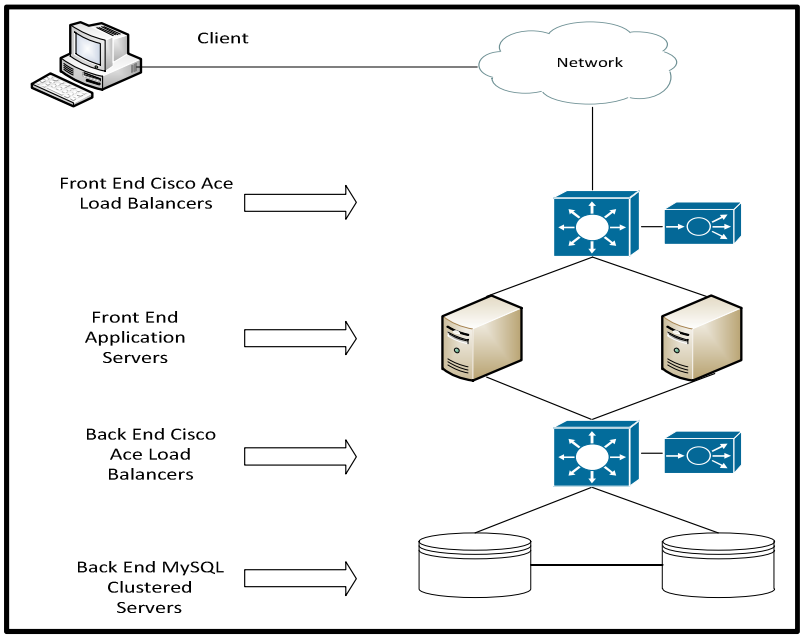

FIGURE 3.11: Physical connections for the new architecture

3.8.2 Loss of front-end application server

In Figure 3.11 a negative scenario has been formulated to demonstrate the loss of an application server.

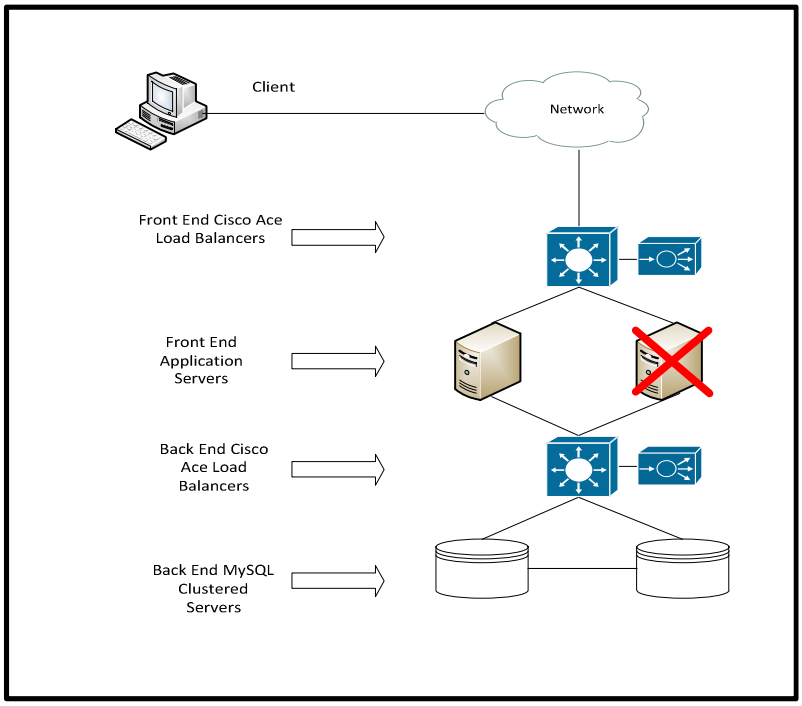

Figure 3.12: Front-end application server failure

In this scenario one of the application servers was taken out of service and became unavailable. The ACE health probes would then fail this specific server and it would be taken out of commission or service by the ACE module. The ACE would then no longer allow any connections to this application server. If traffic were to be sent to the application server during the failure, only those transactions with which it was busy would fail. The VIP and the service would still remain available. The client would receive note of the failure and would be able to resubmit that failed transaction.

3.8.3 Loss of MySQL server

Here the negative scenario had been formulated and showed the loss of a database server. Figure 3.13 shows the configuration of such a failure.

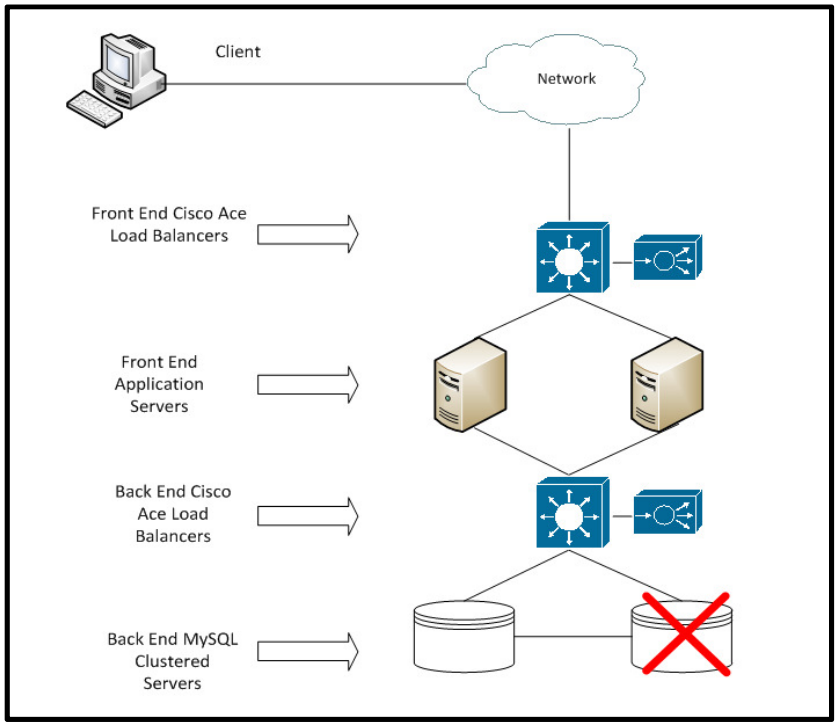

FIGURE 3.13: Loss of a single MySQL cluster node

In this scenario the back-end MySQL cluster node would fail. The ACE health probes would fail for this specific server and it would be taken out of commission by the ACE module. The ACE would then no longer allow any connections to this database server. If traffic were to be sent to the database server during the failure, only those transactions with which it was busy would fail. The VIP would still remain available. The service would still remain available. The client would receive note of the failure and would be able to resubmit that transaction.

3.8.4 The theoretical expectation from the experiment

In Figure 3.14 the graph displays the theoretical behaviour of an existing VOMS and the newly proposed VOMS architecture under a severely faulty condition. The legend indicates the two systems on the graph. When a fault occurred, an existing VOMS architecture caused client traffic to stop processing when either the application service or the database encountered a problem. The new architecture continued to process client transactions in a faulty condition, which could include a loss of the front-end application service or the loss of the database. Herein lies the strength of the newly designed architecture, namely that it would be able to withstand failure conditions and not disrupt transactions. The clients would be unaware and the process of recovery would be transparent for the new architecture. A time of four minutes was decided upon as the duration per experiment. After two minutes' time, a failure would be introduced into the system. The behaviour would then be recorded by means of a load running tool named Java JMETER. Observations could then be made and a discussion on the result be reported.

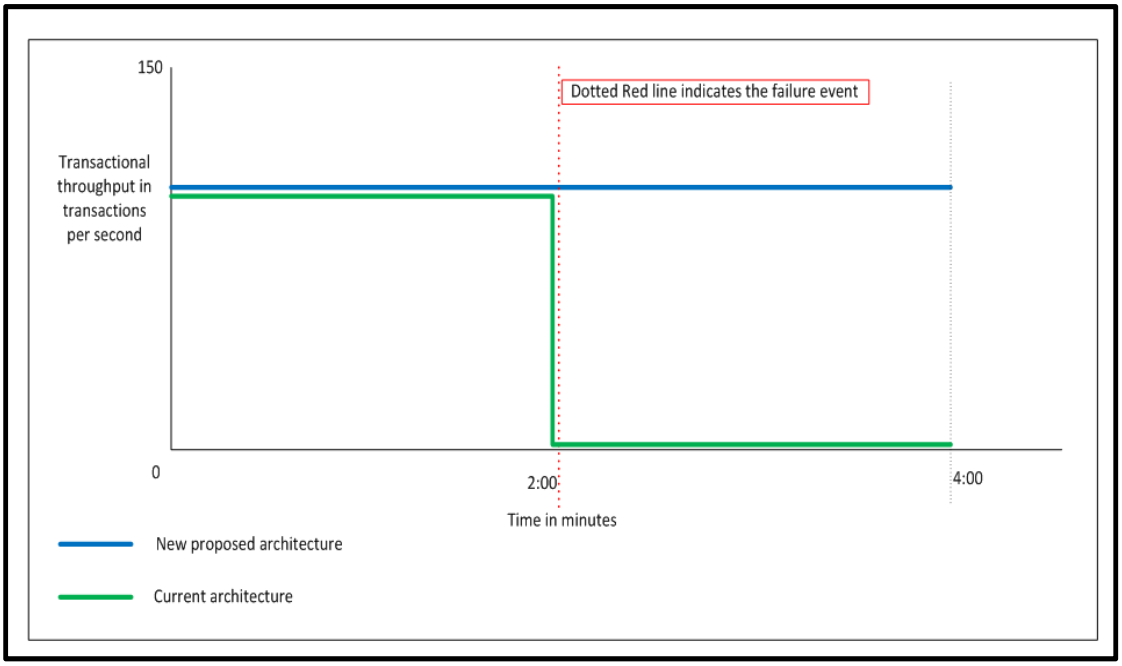

FIGURE 3.14: Theoretical behaviour of an existing and newly proposed architecture

JAVA JMeter was a load running tool used to emulate load on the newly proposed architecture. A sample set of 25 000 subscribers would be used to perform a balance enquiry on the downstream IN system. As can be seen in Figure 3.15, the number of active threads or active emulated client connections chosen was 100 for the duration of the emulations.

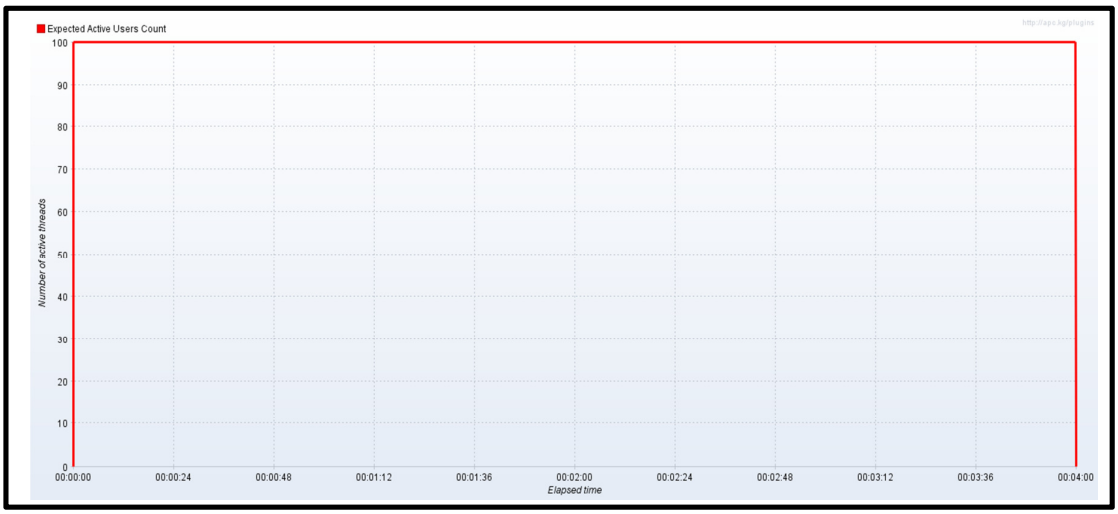

FIGURE 3.15: A graph denoting flat transaction rate for emulations

3.9 EQUIPMENT USED FOR THE EXPERIMENT

The laboratory equipment was configured and had the following hardware specifications:

3.9.1 Front-end application servers

The servers used for the emulations were a Virtual Machine (VM) built on an RX900 hardware platform which provided the CPU and Memory.

The RX900 hardware was produced by Fujitsu Siemens. The scalable hardware had multiple cores. There were as many as 80 cores per RX900. The RX900 had 1 TB RAM and 128 CPUs as standard.

The Storage provided was built on a Storage Area Network (SAN). The OS installed was Red Hat Linux for both servers. The servers were designed within the following parameters:

3.9.2 Front-end servers

- Front-end 1: 2 x QEMU Virtual CPUs @ 2266Mhz, 2048MB memory **OS:** Red Hat Enterprise Linux 6, update 4 (x86_64)
- Front-end 2: 2 x QEMU Virtual CPUs @ 2266Mhz, 2048MB memory **OS:** Red Hat Enterprise Linux 6, update 4 (x86_64)

3.9.3 Back-end servers

- Back-end 1: 4 x QEMU Virtual CPUs @ 2266Mhz, 16384MB memory **OS:** Red Hat Enterprise Linux 6, update 4 (x86_64) **MySQL:** MySQL Ver. 5.5.30-ndb-7.2.12-cluster-gpl for linux2.6 on x86_64 (MySQL Cluster Community Server)
- Back-end 2: 4 x QEMU Virtual CPUs @ 2266Mhz, 16384MB memory **OS:** Red Hat Enterprise Linux 6, update 4 (x86_64) **MySQL:** MySQL Ver. 5.5.30-ndb-7.2.12-cluster-gpl for linux2.6 on x86_64 (MySQL Cluster Community Server)

3.9.4 Ace modules both front end and back end

- Cisco ACE Load Balancer: Daughter Card: Controller FPGA Rev:1.5; NP 1: Clock Rate: 600MHz, Memory Size: 4096 MB; NP 2: Clock Rate: 600MHz, Memory Size: 4096 MB; NP 3: Clock Rate: 600MHz, Memory Size: 4096 MB; NP 4: Clock Rate: 600MHz, Memory Size: 4096 MB

3.10 RELIABILITY AND VALIDITY

3.10.1 Reliability

Reliability may be defined as the degree to which a measuring tool renders unchanged, invariable and equal results when the same measurements are done more than once under the same circumstances, that is, “if the results of a study could be reproduced under a similar methodology, then the research is considered reliable” [43].

In this study, reliability was ensured by the proof that under emulation conditions the newly constructed VOMS would have an extremely high availability during a failover scenario.

3.10.2 Validity

The Oxford English Dictionary defines validity as "the quality of being well-founded on fact, or established on sound principles, and thoroughly applicable to the case or circumstances; soundness and strength (of argument, proof, authority, etc.)" [44].

In this study, validity was ensured by consulting with senior staff members across various South African mobile operators.

3.11 ETHICAL CONSIDERATIONS

3.11.1 Approval

Permission to conduct this research and render a report to submit for an M.Sc. degree was obtained from the Faculty of Engineering, University of the Witwatersrand.

3.11.2 Right to privacy

The information collected during the study will be dealt with in the strictest confidence and responsibility under the supervision of the study leader. Only the author, study leader, and the mobile operator will be privy to the information pertaining to the study and the data.

3.12 CONCLUSION

Chapter 3 contains a discussion of the study methods used in the investigation and the operating procedures applied. This chapter described an existing VOMS and provided the context for the need for change. The rationale for the development of a new VOMS was provided and explanations were given for why its architecture proved to be more resilient against faults.

The drive for higher application service availability is important, and with the methods described in this chapter, those components mentioned would contribute towards to the new VOMS having higher application service availability.

Cost was an important consideration; the new VOMS would need to be more cost effective as compared to an existing VOMS installation. Costing will be discussed in Chapter 4.

Based on the outcomes of the literature review, it was possible to achieve service availability greater than five 9's. Careful consideration for cost and design would render this new architecture effective in providing service availability exceeding five 9's. This is based on the requirements set out by Holenstein *et al.* [39].

In the next chapter, Chapter 4, **Description and interpretation of the results**, the results of the study will be elucidated and interpreted.