



[library@chuka.ac.ke](mailto:library@chuka.ac.ke); [www.chuka.ac.ke](http://www.chuka.ac.ke)

## DIGITAL TRANSFORMATION FOR DEVELOPMENT

### AGENT-BASED ONLINE SECURE DISKS USING NASD MODEL APPROACH

*Osero, B.O.*

*Department of Computer Science, Chuka University, P. O. Box 109-60400, Chuka*

*Email: [bosero@chuka.ac.ke](mailto:bosero@chuka.ac.ke)*

**Citation:** Osero, B.O. (2016). Agent-based online secure disks using NASD model approach. In: Isutsa, D.K. and Githae, E.W. *Proceedings of the Second Chuka University International Research Conference held in Chuka University, Chuka, Kenya from 28<sup>th</sup> to 30<sup>th</sup> October, 2015.* 300-310 pp.

#### ABSTRACT

Increasing performance and decreasing cost of microprocessors make it feasible to move processing power to the data source. This allowed us to investigate new methods of storage delivery and management not plausible in the past. Our architecture, inspired by agent-based techniques and active disk technology, promotes an open storage management framework that embeds functionality into storage devices. We use local agents to implement self-control with automated capability that can be dynamically adapted to meet storage management through improved search capabilities in the virtual environment and also retain capabilities like security, performance and availability. This paper reports experimentation on how storage virtualisation of network attached devices improves performance by using NASD model, explains attempts in building mobile agent environments on network attached disks, and demonstrates proposed framework that employs mobile agents to supplement the existing NASD model.

**Keywords:** Data management, Microprocessors, Network attached disks

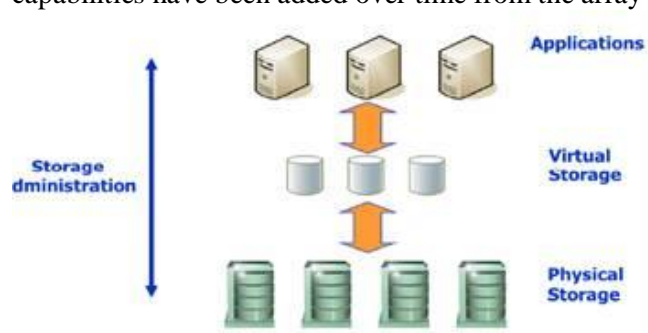
#### INTRODUCTION

Abstracting the physical components of computing resources optimizes the way in which other systems, applications, or end users interact with them. This definition summarizes two very important characteristics of virtualization: It should reduce the complexity when compared with managing devices discretely, as well as add greater capability to improve services — a bit like a one-plus-one-equals-three equation. When described in terms of storage, the Storage Networking Industry Association (SNIA), the independent body of the storage industry, has a more specific definition (two of them, in fact) for storage virtualization, from its technical tutorial on virtualization:

1. *The act of abstracting, hiding, or isolating the internal functions of a storage (sub)system or service from applications, computer servers, or general network resources for the purposes of enabling application and network independent management of storage or data.*
2. *The application of virtualization to storage services or devices for the purpose of aggregating, hiding complexity, or adding new capabilities to lower-level storage resources.*

Think of virtualization in the context of using a telephone. When you pick up the handset or initiate a call on your mobile phone, as the caller you are completely unaware of how you are connected to the other party. The complexity of establishing communication is completely invisible; you are oblivious as to the number of hops across the telephone network or the type of communication involved (such as satellite or underground cable), or even if there is a failure in the network. This is virtualization technology at work, using available resources as efficiently and intelligently as possible to deliver a service [1].

Virtualization is a powerful feature that plays a role in the current success of storage arrays. By design, virtualization manages where data is located and controls access to data for users and applications. The value of storage has moved from disk drives to the array controller as more features and data protection capabilities have been added over time from the array to the point of virtualization [19].



**Figure 1:** SAN virtualization source [19].

Virtualization provides logical representations of physical resources while preserving the usage interfaces of those resources. Virtualization techniques can remove resource limits while improving utilization. Virtual memory and virtual networks have existed in the technology industry for years [19]. Applications that can follow mobile users when they change to a different environment, especially with the change of device and location, are in high demand by pervasive computing. Implementation of application mobility also depends on context-awareness and self-adaptation techniques [17]. The development of Storage Area Networks (SANs) has passed its experimental phase and SAN has become a mature technology that allows businesses to implement storage pooling and sharing today. SANs are high speed switched networks designed to allow multiple computers to have shared access to many storage devices [4]. Yet, implementing a SAN does not in itself solve the basic problems of meeting the requirements of the ever growing explosion of data, [4]. A decade and half later the same trends in storage requirements keep on increasing exponentially and therefore requires innovative measures to avoid being caught up in situations where we cannot handle massive data and processing requirements.

### Literature Review

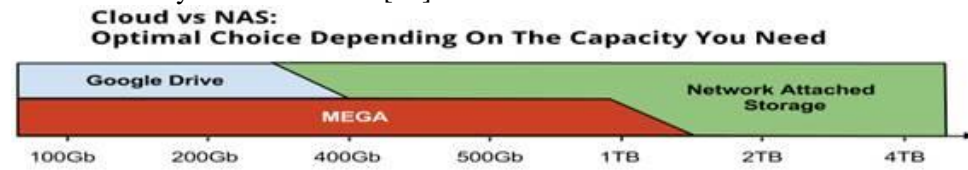
The current storage systems are characterized by store and forward file processes through the server, hence the server becomes a bottleneck. Therefore, the need to study these bottlenecks in Network Attached Disks (NAD) in relation to network storage and find a more efficient solution. [2] [6]. [5] Did research on network virtualisation and observed that virtualization can be achieved easily and cost effectively by use of open source software like Opensolaris and SUN xVM Virtualbox. [6] Indicated that it is possible for the Network to eliminate, store and forward file processes through the server and hence achieve scalability.

In Cloud, the flexibility is achieved through the virtualization of operating systems. The virtualization is the set of techniques and tools to run multiple operating systems on one physical machine to deliver a better use of resources.

The virtualization tools most used are: OpenVZ, Xen, KVM, Virtual Box, VMware and HyperV.

- OpenVZ is a virtualization solution type isolation based on the Linux kernel and allowing one physical server to run multiple instances of isolating systems operating, OpenVZ offers less flexibility in the choice of system operating. In fact, the guest operating system and host must be of type Linux.
- Xen is a free virtualization solution to run multiple operating systems on one physical machine.
- Its hypervisor is considered a solution based Para-virtualization, because guest systems must be modified to cohabit.

The cloud storage providers offer great deals and still are no match for NAS devices when it comes to large data storage. Besides, NAS also features a great number of benefits which can be valuable to many people. The cloud is available at a cheap rate for a start, but the higher cost of NAS is completely compensated by the redundancy and ease of use [12].



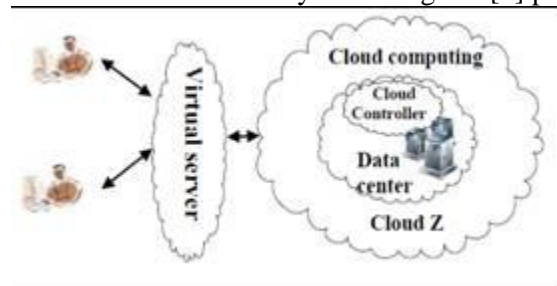
Source [12]

In their study Hitachi data systems discuss some challenges that can be encountered in a non -virtualized center one of which is data migration; data migrations can be painful in terms of cost, risk and complications. Most enterprise IT shops have to plan for storage platform end of life and migration to new storage. They must thoroughly consider the potential impacts on production environments and ensure ample support and expertise [13].

IT is looking to minimize manual efforts, create more transparent migration activity and take advantage of flexible data mobility capabilities. This way the price tag and hazards of moving data are diminished [13].

In the work of [10], they studied the Xen virtual environment and they proposed a trust model based on security agents for it, which are simple mobile agents that provide security at the virtual machine and the entry point of the network cloud to cloud customers and service providers to manage their resources and data safely and efficiently. These mobile agents not only provide security measures, but also ensured the accounting and monitoring activities in the virtual machine if it is malicious or normal state, so that the client is kept informed of the data. If alarming conditions, the client is informed and can take the necessary measures required.

In their advances to study mobile agents [3] proposed the following model:



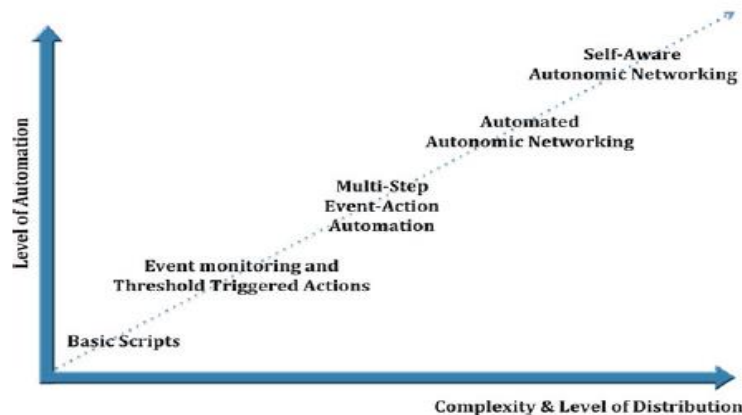
**Figure 1:** A trust model based on security agents [3].

The Figure 1 presents an architecture, which is based on the concept "mobile agent" for cloud computing, which divided into two main layers interact: First the virtual server layer and the second is the cloud computing layer [3]

Mobile agents are autonomous programs that can travel from computer to computer in a network, at times and to places of their own choosing. The state of the running program is saved, by being transmitted to the destination. The program is resumed at the destination continuing its processing with the saved state. They can provide a convenient, efficient, and robust framework for implementing distributed applications including mobile applications for several reasons, including improvements to the latency and bandwidth of client-server applications and reducing vulnerability to network disconnection. Although not all applications for distributed systems will need mobile agents, many other applications will find mobile agents the most effective technique for implementing all or part of their tasks [15].

Mobile agent paradigm has been evolved from transitional client/server paradigm and it has many advantages such as mobility, autonomy, intelligence, adaptability, cooperation, etc. Especially, the mobility is the most remarkable property of the mobile agent paradigm. Thus, the security issue of mobile agents and mobile agent systems is recognized as one the most important problems [14].

The security issues of the mobile agent paradigm are classified into two key issues generally- mobile agent security and mobile agent system security. The security issues of mobile agent systems are related with managing of mobile agent system resource, protecting from malicious agents or malicious hosts, and so on [14]. The figure below indicates the Opportunities of Network-Embedded Application scenarios

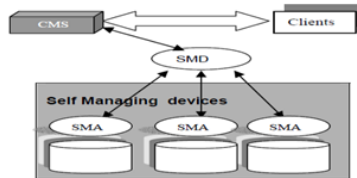


IBM has developed Predictive Failure Analysis (PFA), a self-monitoring approach for their servers that indicates potential outages proactively. For example, in hard disks, the PFA code monitors read/write errors, the height of the disk head above the platter, and the amount of power used to keep the drive spinning at a constant speed. If any of these exceed a predefined threshold, an alert is generated, allowing the operator to replace components before an outage occurs [15].

Both Grid system and mobile agent technology are common on the way that they both are being explored in a distributed environment. Both have their advantages and limitations when they work as community or group. As a community, Grid has limitations such as Grid system cannot anticipate and diagnose the changes to the state. In the same way Agent system has also limitations such as Agent framework does not provide support for secure interaction and use many assumptions. So there is a need to integrate both technologies to get true and full benefits of both technologies. [9].

### Self Managing Storage System (SMSS)

*Active Disks* are next-generation disk drives that provide an environment for executing application code directly at individual drives. Partitioning processing across clients, servers and storage devices can reduce the amount of data crossing the network and exploit the processing cycles available at the storage device. Also, agent-based communication paradigms have shown enormous potential for operating in unpredictable, metamorphic environments. With the intelligence built into it, the agents can keep track of their own conditions dynamically and thus autonomously react when the faults and/or attacks occur in their subsystems. [20]



### Self Managing devices [20]

**Figure1:** Architecture of the Self-Managing Storage System

A Self-Managing storage device is much like an active disk, which could exploit general-purpose processing power at the device itself. Thus the new storage device operating as an independent entity is tasked not only with the responsibility of storing data, but protecting it, performing backups, and expediting recovery. This device allows management policies to be downloaded to the device [20].

Figure 1 shows the main modules of the self-managing storage architecture. The Storage Management Driver (SMD) module launches Storage Management Agents (SMA) onto the storage devices to enable the manageability tasks on the storage devices [20]. Self-Managing Storage Devices offer many benefits as systems based on NASD (Network Attached Secure Disks) [5, 6] and active disks while providing additional benefits. Self-managing systems look beyond transactions towards prediction and decision making capabilities. There are several applications that can benefit from a self managing architecture. Examples include: providing good security as we show in next section, automatically allocating resources for multiple jobs according to demand, clustering which updates software and manages workload across massive clusters with a single operation, etc. Users can define their own requirements and CMS could launch the necessary functionality onto the SMA [20].

### Advantages claimed for mobile code systems

So far, there have been many advantages claimed for mobile agents, these claims are usually in the form of qualitative assessments but unfortunately, very few quantitative measures exist to support these claims [8]. However, a summary of some of the more frequently quoted and accepted claims are described in the following sections.

### Bandwidth Savings

Distributed systems by their nature are required to communicate over the network. This communication can sometimes be in the form of multiple consecutive interactions between two components, for example, a query client and a database. This type of data querying can result in heavy network traffic. Mobile agents are able to overcome this problem by relocating to the host of the database. Instead of shipping data back and forth across the network, they are able to migrate the required business logic to the data source. Once in situ, they can perform any required queries and process the returned information without saturating the network. After processing, they are able to continue with their work, transporting merely the result to a new host, if it is in fact needed [8].

### Reducing Latency

Many manufacturing and robotic systems must be controlled in real time. Controlling these systems through a factory wide network can be affected by latency and data timeliness. Mobile agents are able to overcome this problem by migrating to be local to the process and control it in real time, thus bypassing the problems of latency [8].

### **Disconnected Operation**

As the amount of Internet traffic increases, the response from the telecommunications companies in installing new carrier infrastructure is immense. Nevertheless, this effort may still not be enough to satisfy the expanding base of users. Moreover, many users will not have access to the high-speed bandwidth available to wealthy corporations. Currently, most home users in the UK still connect via a modem and copper telephone lines. The proliferation of mobile devices, such as palm top computers, which employ wireless networks, implies that many users and devices will be extremely limited in the bandwidth available to them. This disparity in quality of connection means that performing tasks that require a continuous connection to the network will be probably not be feasible financially, if not technically.

Mobile agents are a solution to this problem. A particular task can be encapsulated within a mobile agent. The agent is then dispatched to a host that is part of the network backbone, and enjoys massive bandwidth access. Once there, the mobile agent is able to carry out its task in the resource rich environment before returning home. A further advantage of this paradigm is that since the mobile agent is now independent of the device, the device can go offline, or even be switched off, before again connecting later for the agent to return with the results. [8].

### **Increased Stability**

One of the major problems with distributed systems is failure, and the identification of the particular type of failure. Traditional distributed systems are built with the philosophy that the network is permanent, and any failure is unexpected. When it does happen it is very difficult to tell whether the network has failed, the machine that was hosting the component you were communicating with has died or the component itself has frozen. One of the underlying philosophies behind mobile agents is that the network is not a permanent resource. By building software with mobile agents, distributed systems can be less dependent on the network, since the underlying tenet is local interaction. Discovering the nature of a failure in a local context is a much easier proposition, and so systems built this way can be more stable. Mobility can also be used to achieve replication for fault tolerance, and support robust distributed systems. If a host is being shut down, or experiencing problems, an agent is able to react to this by migrating to a new host where it can continue with its operations. [8].

### **Server Flexibility**

In contemporary distributed systems, when data is exchanged between communicating hosts, each host owns a copy of the code that is required to package outgoing and interpret incoming messages. As protocols are evolved to better support efficiency and security, the effort required to upgrade protocols becomes immense. By using mobile agents, the protocols can be encapsulated within the agents, and removed from the servers. Thus, if a protocol requires an upgrade the mobile agent population can be upgraded gradually as and when required, instead of the entire server base. Further, since mobile agents are able to carry around their own code, the distributed system can become more flexible since the mobile agent is not merely limited to the functions a server predefines. It is able to bring along new or improved code and can extend the functionality of the server in which it is executing [8].

### **Simplicity of Installed Server Base**

An additional advantage of relocating the computational logic and protocols within the mobile agent is that the installed servers become much simpler. Effectively, a server becomes merely an executing environment for hosting mobile agents. As this requires far less functionality pre-engineered into the software from the

outset, it can help with preventing legacy. Further capabilities can be added by mobile agents at a later date [8].

### Support distributed computation

Mobile agents are inherently distributed, and as such can be a fundamental enabler for distributed computation. However, they are also heterogeneous, often separated from both hardware and software dependencies by their executing environment. This means they are an ideal technology for integrating disparate legacy systems that have dependencies already [8].

### CURRENT NASD SYSTEM

Network-Attached Secure Disks (NASD), which separates store and forward copying work and management. By modifying the interface for commodity storage devices, we eliminate the server resources mainly required for data movement.

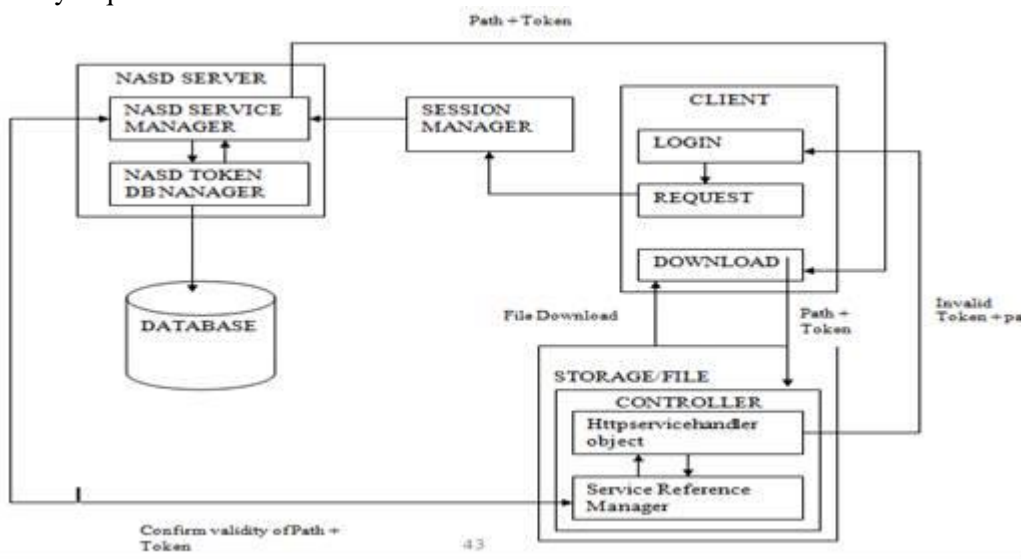


Figure 2: the major components of NASD ARCHITECTURE are outlined in the figure below [11].

### How the above NASD model works

NASD reduces the overhead on the file server (manager) by allowing storage devices to transfer data directly to clients. Most of the file manager's work is offloaded to the storage disk without integrating the file system policy into the disk. Most client operations like Read/Write go directly to the disks; less frequent operations like authentication go to the file manager. Disks transfer variable-length objects instead of fixed-size blocks to clients. The File Manager provides a time-limited cacheable capability for clients to access storage objects. A file access from the client to the disks has the following sequence [11]:

1. The client authenticates itself with the file manager and requests for the file access.
2. If the client can be granted access to the file requested, the client receives the network location of NASD disks and their capability.
3. If the client is accessing the disk for the first time, it receives a time-limited key for the establishment of secure communication to the disk.
4. The file manager informs the corresponding disk using an independent channel.
5. From now on, the client directly accesses the NASD disks by giving the capability it received and further data transfers go through the network, bypassing the file manager.

The above architecture shows a framework of a NASD model, when the client requests data from a storage area then the request is sent through the controller to the file-server which has the file manager network

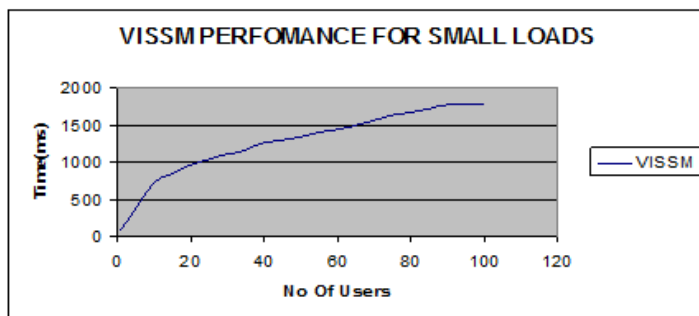
protocol and the access and security policy control semantics. If the client satisfies the security requirements then it will be allowed to get an access path from the network storage pool in the file server and then eventually download the file from the storage area directly to the client [11].

### NASD model performance compared with Store and Forward Processes (SAF)

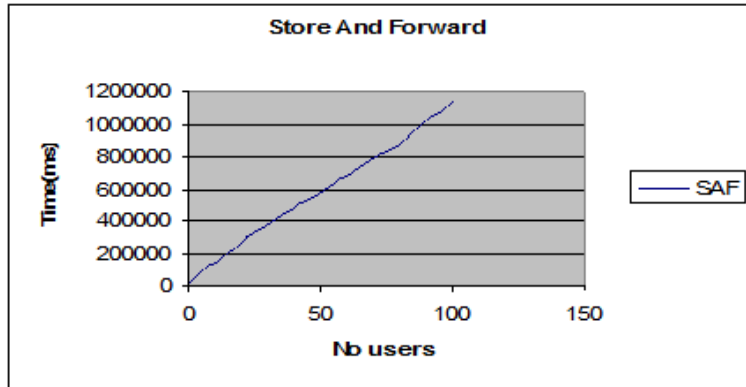
Comparing Figure 1 and 2 above at the face value it is evident that under same loads NSVM scales better than SAF. When the system load 20 users NSVM performance is approximately 1000 ms while at the while with the same number of users SAF performs at about 20000 ms which is actually 20 times higher than the store and forward (SAF) process. With 100 users NSVM performs at 1800 ms while at same of users SAF takes 1100000 ms which is 600 times slower than the NSVM. NSVM has a better performance than the SAF model. It can be noted that when there are 100 users NSVM scales at 60% while SAF scales at 10% [6].

**Table 1:** SAF processes and NASD performance [6]

No of users(1 file=167 KB)	Experiment to show performance in the disks			
	Size of file	NASD Time taken(ms)	SAF(s)	SAF(ms)
1	167	80	21.801	21801
5	835	400	88.727	88727
10	1670	701	143.646	143646
15	2505	840	205.996	205996
20	3340	972	259.383	259383
25	4175	1031	340.129	340129
30	5010	1102	382.92	382920
35	5845	1152	430.559	430559
40	6680	1262	475.644	475644
45	7515	1302	529.481	529481
50	8350	1341	574.906	574906
55	9185	1362	628.934	628934
60	10020	1471	682.04	682040
65	10855	1491	738.481	738481
70	11690	1562	784.287	784287
75	12525	1672	830.143	830143
80	13360	1672	879.384	879384
85	14195	1722	949.425	949425
90	15030	1772	1017.923	1017923
95	15865	1782	1065.071	1065071
100	16700	1793	1130.575	1130575



**Figure 1:** A graph showing the Showing the classical SAF [6].



**Figure 2:** A graph showing the NASD performance [6]

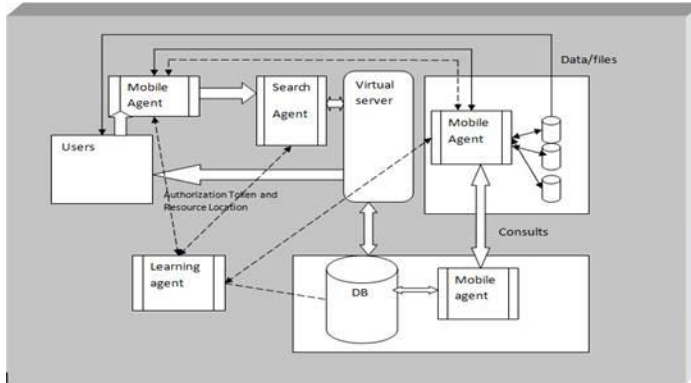
### Statement of the Problem

Network attached secure disks employ virtualisation to enable object based direct data transfer between the clients and the user, this method has proven effective according to [2],[5]. If implemented properly the NASD model will define the next generation secure and self managing online storage architecture. A good storage system should perform optimally notwithstanding its size, data can be stored and retrieved quickly, it should also be available all the time whenever required, and the system should also be able to scale seamlessly to meet growth in user requirements. Today's storage systems fall short of any of the above requirements. They are actually characterized by store and forward processes which replicate and duplicate resources thus making the process slow. The above shortcomings are currently solved by manual techniques and administrators who are ill equipped to undertake such complex optimizations and configuration maneuvers. It is worth to note that human expertise is scarce and expensive making the storage management process a costly undertaking [11].

To solve the above problem advances have been made by [2], [6], [7], [11] in coming up with secure disks and interactions in virtual environments, however both of these models fail to address the aspect of how to make the disks intelligent so as to make decisions autonomously although similar advances have been made by [2]. A study by [16] on network embedded reveals that mobile agents exhibit higher levels of autonomy than other forms of distributed system management; therefore Mobile agents allow the managed system to autonomously handle complex management tasks, thus in principle enabling scalability and fast reaction to events, which is bought by a higher complexity of the mobile agent execution environment and programs when compared to the previous approaches., our approach will follow the same route as [2] but implemented using mobile agents in the Network attached secure disks (NASD) model environment. This solution will enhance protection for critical business data and increase asset utilization, availability, and reliability levels with low management cost. To solve this inherent problem our major objective will be to study the existing virtual storage NASD model and implement and coordinate agents in this virtual environment thus making the network attached storage devices intelligent. Intelligent disks will be able to make decisions on who accesses files, when and how the files are accessed from storage area in the network and this will allow system administrators make easy and informed decisions on storage.

### Proposed solution

To address the gaps that exist between mobile agents and network attached disks that have not yet been fully exploited; a more intelligent, self managed and secure storage environment has been proposed as shown in the following model. The model introduces embedded intelligence into the network attached disks so that they are able to make important decisions on when data is being compromised, when to release data, the number of users on the system and perform load balancing. This will minimize the need for the tedious manual system configurations performed by the System administrators.



**Figure 4:** A proposed intelligent NASD system using agents

### Contributions

1. Introduce mobile agents that will be able to supplement the existing tokenized security model hence providing a more reliable and secure environment.
2. Introduce intelligence into disks through the mobile agent that will constantly be updated by the learning agent which monitors the user's behavior therefore provide easy decision support mechanisms on dynamic storage requirements and update the system administrators who will take an action.
3. Introduce a learning agent that will constantly be updating the search agent and the mobile agents by observing the log data in the database and the requests made to both the mobile agent and the search agent.
4. To increase reduce the response time between the client requests and the virtual server there was need to introduce a search mechanism that will be designed to supplement the virtual server and this will be implemented as a search agent

### Objectives

1. Design security mechanism that employs mobile agents to supplement the existing NASD security framework.
2. Demonstrate through experimentation whether mobile agents perform better than the NASD models.
3. Design a model for coordinating and implementing agents in a virtual space/environment.
4. Design and test a prototype for NASD implementing mobile agents.
5. Implement a search algorithm that can improve search in the virtual server in order to reduce the turnaround time between the client and the server.

### FUTURE WORK

To implement the above proposed model in an agent based platform and also develop a simulator to test the results by comparing an agent based model and the Network attached storage disks (NASD) based model. To study emerging trends in technology like internet of things (IOT) and be able to realign future research in the proposed model to suite this emerging trends. Security and integrity of stored also remains a challenge in today's fast changing and complex systems our aim will be to look for better, strong and improved security schemes and mechanisms to suit our research especially multi-factor authentication (MFA) where we will apply biometric authentication, tokenization and strong passwords.

## REFERENCES

- Adrian, M. 2010. Storage Virtualization for Dummies; Hitachi data systems edition
- Amir.2000. Scalable and manageable storage systems, CMU-CS-00-178, DEC 2000.
- Ali Alwesabi, Kazar Okba, Security Method: Cloud Computing Approach Based on Mobile Agents: International Journal of New Computer Architectures and their Applications 41:17-29 The Society of Digital Information and Wireless Communications, 2014 ISSN: 2220-9085
- Mark et al. 2000. Storage Networking Virtualization, What's it all about? IBM.com/Redbooks, International technical support IBM.
- Nikhi. G. and Apte, S.S. 2013. Mobile Agent based Communication Platform for Heterogeneous Distributed Database, September
- Osero. 2013. Network storage virtualisation and management, International Journal of Education and Research, 1 No. 12 December 2013, ISSN: 2201-6333 Print ISSN: 2201-6740 Online www.ijern.com
- Zablon, O. 2009. Multiple agent co-ordination in a virtualized environment, a thesis presented at the CSI University of Nairobi.
- Todd, P. 2000. A Doctoral Thesis "On the Structuring of Distributed Systems": The Argument for Mobility, Loughborough University February
- Muthu K. and Manickam. 2010. A Security Model for Mobile Agent in Grid Environment. In: International Journal of Computer Applications 0975 – 8887 Volume 2 – No.2,
- Priyank, S., Ranjita, S. and Mukul, M. 2011. Security Agents: A Mobile Agent based Trust Model for Cloud Computing. In: International Journal of Computer Applications 0975 – 8887 Vol. 36– No. 13
- Osero. 2010. Storage virtualization and management, MSc. Thesis at the University of Nairobi <http://www.cloudstoragecomparison.net/cloud-storage-nas/>
- Hitachi Data Systems. 2015. Storage Virtualization: How to Capitalize on Its Economic Benefits,
- Dongwon Jeong et al. 2006. A Secure Migration Mechanism of Mobile Agents Under Mobile Agent Environments, Springer-Verlag Berlin Heidelberg
- Ichiro, S. 2004. Selection of mobile agents. In 24th IEEE International Conference on Distributed Computing Systems ICDCS'2004, pages 484-493. IEEE Computer Society,
- Alexander, C. and Ralf, W. 2013. Network-Embedded Management and Applications: Understanding Programmable Networking Infrastructure, Springer Science and Business Media New York
- Ma et al. 2006. (Eds.): UIC 2006, LNCS 4159:648-657 Mobile Agent Enabled Application Mobility for Pervasive Computing, Springer-Verlag Berlin Heidelberg
- EMC2. 2008. Where information lives Virtualization: Current Benefits and future potential, Technology Concepts and Business Considerations, January 2008.
- Randy et al. 2012. SAN virtualization evaluation guide, 2011 evaluator group.
- Srinivas et al. 2015. Self-managing storage system: Design and evaluation [cl.ece.arizona.edu/projects/old/smp/smppaper.pdf](http://cl.ece.arizona.edu/projects/old/smp/smppaper.pdf), last downloaded on 30/9/2015, 11:43 am