APPLICATION OF BLOCKCHAIN TECHNOLOGY IN STRENGTHENING HEALTH INFORMATION SYSTEM SECURITY: A CASE STUDY OF MOUNT MERU REFERRAL HOSPITAL

Richard William Mnyawi

A Dissertation Submitted in Partial Fulfilment of the Requirements of the Award the Degree of Master of Science in Wireless and Mobile Computing of the Nelson Mandela African Institution of Science and Technology

Arusha, Tanzania

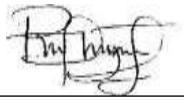
July, 2022

ABSTRACT

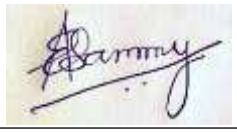
Health information system (HIS) is a digital technology used in health care data management. Through literature review, it has been observed that HIS are facing security challenges. These challenges are based on centralized system architecture creating a target for malicious attacks. Despite of the effectiveness of this technology, still HIS are suffering from a lack of data privacy and confidentiality. This research developed a blockchain-based system integrated with the Government of Tanzania Hospital Management Information System. The study employed a qualitative research method where data were collected using interviews and document analysis. Execute-order-validate Fabric's storage security architecture was implemented through private data collection. Privacy and confidentiality are attained through a private data policy. Network peers are decentralized with blockchain only for hash storage to avoid storage challenges. Cost-effectiveness is achieved through data storage within a database of a Hyperledger Fabric. The overall performance of Fabric is higher than Ethereum. Ethereum's low performance is due to its execute-validate architecture which has high computation power with transaction inconsistencies. Health policymakers should be aware of blockchain technology and make use of the findings. The scientific contribution of this study is based on; the cost-effectiveness of secured data storage, the use of hashes of network data stored in each node, and low energy consumption of Fabric leading to high performance. The system is developed in an integrated data sharing architecture in a peer-to-peer, decentralized network environment. Data sharing and information exchange are maintained without central control, with improved security and privacy of the system.

DECLARATION

I, Richard William Mnyawi, do hereby declare to the Senate of the Nelson Mandela African Institution of Science and Technology that this dissertation is my original work and that it has neither been submitted nor being concurrently submitted for a degree award in any other institution.

Richard William Mnyawi		01/08/2022
Name of Candidate	**Signature**	**Date**

The above declaration is confirmed by:

Dr. Anael Sam		01/08/2022
Name of Supervisor 1	**Signature**	**Date**

Dr. Devotha Nyambo		01/08/2022
Name of Supervisor 2	**Signature**	**Date**

COPYRIGHT

CERTIFICATION

The undersigned certify that they have read and hereby recommend for acceptance by The Nelson Mandela African Institution of Science and Technology, a dissertation titled **"Application of Blockchain Technology in Strengthening Health Information System Security: A Case Study of Mount Meru Referral Hospital**" in partial fulfillment of the requirements for the degree of Master of Science in Wireless and Mobile Computing of the Nelson Mandela African Institution of Science and Technology.

Dr. Anael Sam		01/08/2022
Name of Supervisor 1	**Signature**	**Date**

Dr. Devotha Nyambo		01/08/2022
Name of Supervisor 2	**Signature**	**Date**

ACKNOWLEDGMENTS

I thank Almighty God who can do immeasurably more than all I ask or imagine, to Him be Glory. The opportunity to pursue a Master's program at Nelson Mandela African Institution of Science and Technology (NM-AIST) was His distinguished favor. To Him be glory for His exceeding abundance of grace which enabled me to undertake this research to its accomplishment.

My special and heartfelt thanks filled with Godly love go to my wonderful, unique, and supportive supervisors Dr. Devotha Nyambo and Dr. Anael Sam. The generation of this research idea started from the concept of note preparation which was not easy, but due to your collaborative ideas and efforts, we finally made it. Thank you for spending much of your time for guidance with all patience and encouragement to make sure this work comes to the end. You have been a blessing to me. I also extend my sincere thanks to Dr. Cleverence Kombe for his support and constructive ideas which lead to the achievement of this study. Thanks to my course instructors at NM-AIST for their valued and remarkable inputs for this study. Dr. Jema Ndimbwile you have imported information system security knowledge that saves a great asset to this study, Dr. Elizabeth Mukoba thank you for your support, guidance, and valuable input to this research work.

My sincere gratitude goes to my lovely wife Hellen, my children Baraka, Emmanuel, Glory, and my entire family of the late Bishop William Mnyawi. Your heart-filled love with maximum support and encouragement during the whole period of my studies made it possible to finalize and mark this milestone.

Thanks to the management of the Nelson Mandela African Institution of Science and Technology (NM-AIST), the School of Computational and Communication Sciences and Engineering (CoCSE), and the whole NM-AIST community, the Ministry of Education, Science and Technology (MoEST) for funding my study.

DEDICATION

To my parents, the late Bishop William Mnyawi and his wife Rehema Ramadhani Mpombo.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS AND SYMBOLS

CA	Certificate Authority
CIA	Confidentiality, Integrity, and Availability
DSR	Design Science Research
GoT-HoMIS	Government of Tanzania Hospital Management Information System
HIS	Health Information System
MSP	Membership Service Provider
MTUHA	Mfumo wa Taarifa za Uendeshaji wa Huduma za Afya
NoSQL	not only SQL
PKI	Public Key Infrastructure
POC	Proof of Concept
PO-RALG	President's Office - Regional Administration and Local Government
SQL	Structured Query Language
VANET	Vehicular Ad-hoc NETwork

CHAPTER ONE

INTRODUCTION

1.1 Background of the Problem

Health Information System (HIS) plays a vital role in health care service delivery, control, and management. The system has facilitated quality service delivery with the provision of health data collection, and storage, increased revenue collection, improved health care facilities management, and lessened their control. This has greatly increased the adoption of HIS due to improved productivity, quality, and efficiency of health services. However, the centralized client-server architectural nature of HIS has exposed health systems to security threats, attacks, and vulnerabilities. HIS security challenges have threatened the integrity, confidentiality, and availability (CIA triad) of health data (Chuma & Ngoepe, 2021). Data security challenges become difficult to handle due to either the centralized nature or third-party management of HIS (Kombe *et al.,* 2018; Mahore *et al.,* 2019; Nagasubramanian *et al*., 2020). Therefore, integrating centralized HIS with blockchain technology is the best approach to mitigate security threats to achieve security goals.

Blockchain technology is one of the current trending technologies due to its remarkable impact on security improvements. It is a first-generation technology introduced in 2009 with the help of Bitcoin applications, a public ledger used for online and digital currency transactions. Its second-generation application was through information exchange which provided a programming platform, and smart contracts deployed for running blockchain systems. The technology has come with much attention due to its security, immutability, transparency, and decentralization. Much consideration of the technology is on its integrity verification and decentralized environment of blockchain. A decentralized peer-to-peer network environment has facilitated information exchange without a central authority. This has addressed problems of data storage security for centralized information systems (Wang & Zhang, 2019).

Blockchain technology is an immutable digital public ledger with a distributed database secured using cryptography. Information is stored in blocks data structures where data encryption, time stamp shows event occurrence time, and distributed consensus (Wang & Zhang, 2019). Blockchain's key characteristic features are decentralization, immutability, privacy, verifiability, transparency, and audibility (Sun *et al*., 2019). These led to its gradual

adoption of an innovative solution to information system security challenges (Sun *et al*., 2019; Tsoulias *et al.*, 2020; Wang & Zhang, 2019).

Blockchain is a highly fault-tolerant database technology due to its peer-to-peer decentralized network architecture for running a blockchain system. The technology is regarded as a data store in a data management context (Tsoulias *et al.*, 2020). It is therefore implemented for reliable records management with the provision of maximum security levels (Toapanta *et al.*, 2020). The new data block is added to the ledger based on consensus algorithms with the network members. Once the block is added to the chain, the information is immutable and transparent to all. The transactions are non-recursive once validated in a block (Fan *et al.*, 2020).

The main two categories of blockchain are public and private blockchain systems (Desai *et al.*, 2019; Kombe *et al.*, 2018). Public blockchains are also known as permissionless while private blockchains are also referred to as permissioned blockchains. Public blockchain gives data access to every network member. Member nodes can see transactions, verify and participate in the consensus process of transaction validation. Bitcoin and Ethereum are examples of public blockchains. The private blockchain is a restricted network with identity management intended for the protection of private data (Rennock *et al.*, 2018). Hyperledger Fabric is an example of a private blockchain with privacy and transaction confidentiality where only involved entities can see transactions.

The same categories were also discussed by Morkunas *et al.* (2019) and Niranjanamurthy *et al.* (2019) who added a third category, consortium blockchain; Fekih and Lahami (2020) came up with the fourth category. Hybrid blockchain is the fourth category that combines both attributes of private and public blockchains. Consortium blockchains are a variant of the private blockchain operating under a leadership group. It is a network of a privileged group where data sharing among participants can either be open or private (Fekih & Lahami, 2020; Morkunas *et al*., 2019; Niranjanamurthy *et al*., 2019). Hyperledger and R3CEV are examples of consortium blockchains.

The key pillars of blockchain network security are data integrity, confidentiality, and availability. Security aspects of blockchain are based on principles of cryptography, decentralization, and consensus mechanisms (Kombe *et al.*, 2019). Privacy is attained through cryptography which ensures that transactions are authenticated and verified. System security is

hardened and made more difficult for the security breach. Even though cybercrime is getting better at hacking, techniques to combat them are also improved.

It is not possible to deal with the attacker's motive for hacking the network system since it is an inbuilt character. It is also not possible to prevent system attackers from possessing the means to braking system security. Hardening system security is the best way of removing opportunities for attacks. Hence the focus of this study is to harden the system security of HIS using blockchain security technology to remove attack opportunities.

1.2 Statement of the Problem

Several studies conducted in developing countries revealed weaknesses in HIS data security. Challenges of security threats to shared data storage security and privacy in Mauritius, South Africa, Rwanda, and Kenya were discovered by Mtey and Dida (2019). This was also observed by Gordon and Catalini (2018), and Kamau *et al.* (2018) that these challenges are due to the lack of security and privacy of HIS. These led to a lack of transparency, accountability, integrity, and privacy of patients' data. The same phenomenon was revealed by Ndayizigamiye and Dube (2019) on threats of HIS to security breaches, attacks, and vulnerabilities.

Health Information Systems in Tanzania are faced with security challenges regarding data privacy and confidentiality in information and data sharing (Kajirunga & Kalegele, 2015; Kombe, 2020; Nehemiah, 2014). Kombe *et al*. (2018), Mtebe and Nakaka (2018), and Mtey and Dida (2019) discovered the same challenges on data security and privacy facing HIS. These security challenges are also reported by Tanzania Computer Emergency Response Team (TZ-CERT) from 1st January 2019 to 9 August 2021, that several system attacks resulted from malicious IPs, malware, and web attacks (Team, 2019-2021). The report includes all client-server-based systems including HIS. These challenges are based on centralized system architecture creating a target for malicious attacks (Kombe *et al*., 2018).

Mount Meru Referral Hospital is among of public hospitals in Arusha using the Government of Tanzania Hospital Management Information System (GoT-HoMIS). The system has centralized architecture vulnerable to system attacks. Sensitive data can be accessed and easily manipulated (Khubrani, 2021; Nagasubramanian *et al.*, 2020).

1.3 Rationale of the Study

The study aims at developing a secured blockchain-based system that is not only fault-tolerant but also resistant to data modification. The system will be able to handle hacking and attacks more effectively due to its trusted relationship among system users in the network. The system will be decentralized for data storage and a tool to provide security solutions. The central point of attacks and third-party data management will be removed to improve system security and data sharing.

The developed system will also enable secured interactions among system users with the same objective of accessing shared data. The danger of a suspicious entity injecting malicious programs will be diminished. This is made easy because network entities are easily identified through identity management. Blockchain will record all actions of the network member including their transactions, network reconfiguration, and application of smart contracts. Since blockchain is append-only, this makes the technology suitable for data fraud prevention due to its flexibility in tracking transaction history.

1.4 Research Objectives

1.4.1 Main Objective

The main objective of this research is to develop health information system data storage security using blockchain technology.

1.4.2 Specific Objectives

The specific objectives of this research are:

(i) To identify data security weaknesses of the current health information system deployed at Mount Meru Referral Hospital.

(ii) To design and develop a blockchain-based health information system that will be integrated into the existing system for data storage security.

(iii) To validate the developed blockchain-based health information system.

1.5 Research Questions

(i) What are the data security weaknesses of the current health information system deployed at Mount Meru Referral Hospital?

(ii) Which methodologies will be used to design and develop a blockchain-based health information system to be integrated into the existing system for data storage security?

(iii) Did the developed blockchain-based health information system meet system requirements?

1.6 Significance of the Study

The study will expose the researcher to the current and contemporary technology on the application of blockchain technology. The knowledge gained in this study will lead to a better understanding of how to address cyber-security challenges facing HIS, creating awareness of the application of the technology to the information system security to attain security goals.

1.7 Delineation of the Study

The study was confined to Mount Meru referral hospital as a case study with the GoT-HoMIS system. The researcher chose the system intending to improve existing system data security. The researcher was not given a full copy of the GoT-HOMIS system due to issues of system security and confidentiality of system data. The study developed and tested the prototype in a virtualized environment. This is due to health data sensitivity and a limited research budget. Other researchers, stakeholders, and developers can implement the prototype in a real environment.

CHAPTER TWO

LITERATURE REVIEW

2.1 Theoretical Literature Review

2.1.1 Disruptive Innovation Theory

Blockchain technology in organizations and business operations has recently grown exponentially. The growth of blockchain technology is in both academic and industrial fields leading to a new era of technology. This technology has revolutionized the digital world through its characteristic features of immutability, decentralization, reliability, transparency, and security. This made it be appropriate technology solution for business operations (Kummer *et al.,* 2020).

Blockchain has a wide range of applications including Bitcoin cryptocurrency. Other applications include voting systems, the Internet of Things (IoT), smart property, smart contracts, and security services. Disruptive innovation theory considers blockchain technology to be disruptive innovation due to its innovative solutions to industries and its adaptability to business models (Saadatmand & Daim, 2019). This theory guides this study and it is, therefore, worthy to be adopted.

2.1.2 Theory of Information Security

Blockchain technology has improved information security through support of the goals of the CIA security triad. This is a cybersecurity technology is applied to improve information security against cyberattacks (Taylor *et al.,* 2020). Its application has resulted in reduced possibilities of malicious acts and fraud leading to robust and strong information system security (Demirkan *et al.,* 2020).

The theory of information system security targets the protection of data integrity, confidentiality, and availability (CIA-triad). Blockchain technology has implemented data integrity through its immutability property of a distributed ledger, confidentiality through encryption mechanisms during transaction consensus lifecycle, and availability through its decentralized network architecture. In connection to these security goals; authentication, non-repudiation, accountability, and reliability are also integrated within the applied technology (Warkentin & Orgeron, 2020). For this purpose, it is a reasonable approach to apply blockchain

technology due to high-security requirements for critical operations of information systems (Monev, 2020).

2.1.3 Technology Acceptance Model

Several organizational theories on blockchain technology exist in connection with research studies on organizational systems' security. These theories are rarely used due to the missing link between blockchain technology, research studies, and theories on information system security. The gap led to reliance on assumptions rather than established theoretical implementations. Organizational theories should be well organized with knowledge and security insights aspects of blockchain technology. This will bridge the theories, and academic research with blockchain to address system security challenges. Bridging serves as a reference point in facilitating the linkage of organizational culture in compliance with information system security (Kummer *et al*., 2020). This theory paves the way and gives the focus of this study to the practical implementation of blockchain technology in cyber security (Taylor *et al*., 2020).

2.2 Empirical Literature Review

2.2.1 Health Information Systems in Tanzania

The growth of Information and Communication Technology (ICT) has facilitated quality health care service delivery in Tanzania through technology integration. Several systems components are connected for simplified service delivery. This growth led to the replacement of paperwork through the introduction of E-Systems leading to improved system operations. This migration to e-technology has also simplified and improved online information and data sharing. Information exchange and data sharing are confidential, but if the security of any of the system components is compromised, this results in breaching of the whole system's security (Haule *et al.,* 2019; Khamis & Njau, 2014; Kombe *et al*., 2019; Mtey & Dida, 2019).

2.2.2 General Overview of Blockchain-Based Information Systems

According to Wang *et al*. (2018), public key infrastructure (PKI) is the most useful environment for system security used in blockchain technology as it protects data modification through encryption. The study proposed a PKI certificate system based on permissioned blockchain. This method solved several issues such as the central point of failure, multi-certificate authority (multi-CA) mutual trust, and poor certificate configuration efficiency in digital certificates.

Another study by Zhang *et al.* (2018) revealed that developing Vehicular Ad-hoc NETwork (VANET) brought many security challenges. Proposed solutions to these challenges were based on a centralized system requiring a trusted party that exists as a central and single point of failure. Vehicular Ad-hoc Network architecture has no approach which will ensure information security. The study proposed a security architecture based on blockchain technology and mobile edge computing with three layers, perception layer, edge computing layer, and service layer. The perception layer was intended for the security of VANET data during transmission with the use of blockchain technology. The edge computing layer was purposely introduced for computing resources and edge cloud services for the perception layer. The service layer deployed a mix of traditional cloud storage and blockchain for data security.

The study carried out by Ndayizigamiye and Dube (2019) on blockchain technology provided a favorable environment for HIS. The information system's transactions could be traced with the transaction history. The study proved that blockchain technology contributed to strengthening the information system through accountability and transparency of patients' services. Challenges encountered in the healthcare system during operations were required to be strengthened on how the flow of information and processes were carried out.

Al Barghuthi *et al*. (2019) conducted a study aimed to solve security challenges related to the traditional electronic voting system. The study proposed a blockchain-based electronic voting system that allowed data decentralization and sharing between network participants. The system minimized costs that could be incurred for election.

The feasibility study was carried out on a concession mechanism in a blockchain, using a public blockchain peer-to-peer network based on distributed ledger technologies approach with a combination of distributed ledger technologies and traditional databases using traditional data storage methods such as SQL and NoSQL. The study aimed to demonstrate how it was difficult to have data storage concessions (Tschuchnig *et al.*, 2019).

The study on blockchain and identity-based encryption management by Khan *et al.* (2020) enabled data and identity protection against identity theft and fraud. The approaches resulted in the guarantee of user's privacy. The developed system used permissioned blockchain and identity-based encryption for user identification. The approach leveraged blockchain-based security making it impossible for attackers to hack the system through consensus and hash algorithms.

2.2.3 The use of Blockchain Technology in Health Information Systems

Zheng *et al.* (2018) analyzed the challenges of health data sharing and storage security due to centralized architecture. In this study, a conceptual design was designed for individual data sharing using blockchain technology aided by cloud storage. The main goal was to securely own, control and share health data. Ethereum was used as the design framework in which cloud storage was integrated into the data-sharing system to provide off-chain storage. Off-chain storage technique was used due to the large data set. Encryption mechanisms were applied to ensure security, privacy, integrity, and non-repudiation of the data.

Shahnaz *et al.* (2019) discussed on the implementation of technology in electronic health records (EHR). The framework came with the provision of storage security through access rules. All technology issues related to scalability, interoperability, information asymmetry, and data security risks were addressed. System scalability mainly referred to the big data challenge and how this challenge was solved through the use of off-chain storage of electronic health data. Information asymmetry in health systems refers to data centralization in which hospitals have access to patients' records. Only one part, healthcare controls information making it difficult for a patient to access his records. Interoperability focused on how different health systems can communicate in a given system setting. Data breaches described how hospitals became a point of cyberattacks hence requiring a secured platform. The suggested solution to these challenges was the use of blockchain, an Ethereum platform that is secure, and transparent with the data integrity of patients' records.

Another four-layered framework for health data sharing by Akkaoui *et al.* (2020) used blockchain technology for the privacy and security of health data sharing. The framework handled data generated from patients using IoT devices and body sensors. Blockchain technology was adopted for privacy and security in data sharing. The choice of the Ethereum platform used solidity smart contracts using an on-chain storage technique that had no size limitations. Both off-chain and on-chain techniques were implemented to avoid security challenges, but off-chain was purposely used for the storage of data sets.

According to Mahore *et al*. (2019), data security-related challenges became difficult to manage due to centralized or third-party management of traditional health systems. Electronic health system management has not implemented data ownership and availability, at the same time interoperability has caused difficulties for patients' information linkage in different systems

leading to wastage of time and resources. To escape from these challenges, a patient blockchain-centered system together with cloud storage was implemented to ensure data security, availability, and system interoperability. The output of the system gave patients full control of their data which enabled researchers to access data directly from individual patients. Security issues were addressed through shared information among entities to achieve data integrity and confidentiality. Patients' data were divided as sensitive and non-sensitive to avoid exposition. Hyperledger fabric platform was implemented to develop the model which used permission architecture. Patients, researchers, and hospitals were the entities that participated in one channel. In this network, patients were able to share their data if requested and hospitals were responsible to maintain the ledger.

Madine *et al*. (2020) described the existing personal health records (PHR) systems which failed to provide secure information sharing trust and access to patients due to centralization creating vulnerability to a single point of failure. A solution to these problems was to integrate a blockchain with a personal health data system. System architecture should employ smart contracts with cryptography for automation hence securing medical information sharing and accessibility of PHR systems. This was implemented using the Ethereum platform and smart contracts were written in solidity language.

Security and privacy of patients' information is a challenging task (Biswas *et al.*, 2020). A solution to this challenge is to migrate existing traditional systems to a blockchain model. This solution facilitated a smooth process of secure data exchange among entities. This simplifies access to individual patients' medical information when required. The network-connected patients and health systems without changing their operations. Off-chain storage was used and patient-centered channels-controlled data access.

2.2.4 Deployment of Blockchain Smart Contracts

A smart contract is one of the computation aspects of blockchain technology. It is a system-defined business rule which is executed automatically in the entire blockchain network based on the used consensus protocol. In Linux foundation Hyperledger blockchain, smart contracts are termed as chain codes, a group of related smart contracts. Chain codes govern the packaging of smart contracts and the mode of deployment. They are program codes, stored procedures coded in go language and its execution environment is docker. They define the logic of a transaction in an application that is specific to a particular platform. Concession among peers

for transactions should be agreed upon to maintain the ledger state (Hajdu *et al*., 2020; Liao *et al.*, 2017; Sato & Himura, 2018)

Although blockchain systems are regarded as secure enough to attack, the contract instructions must be defects-free. The quality of a written script will help to avoid software bugs that risk system to vulnerabilities (Hajdu *et al*., 2020). Smart contracts should map business processes and operations. Program codes should be written to reflect business operations to avoid disasters due to incorrect codes (Liao *et al*., 2017).

2.2.5 Deployment of Consensus Protocols in Blockchain Networks

It is an obligation of a decentralized peer-to-peer (P2P) network that trustless peers in the network reach a consensus agreement during data entry. The protocols make the core functionality and a synchronized blockchain network environment. It is a foundation in the development of blockchain systems. Several consensus protocols differing in cost of computation, security, and consensual efficiency are used in blockchain technology. They define rules and procedures that all network members come to a consensus agreement on the state of the ledger. The mechanism creates a trusting environment among peers whereby any new block added to the system should be the one agreed upon by all participants.

2.3 Research Gap

With respect to the previous studies on various concepts and theories based on HIS security, more has been done on data storage and little on the security of data storage. Most of the studies focused on the combination of traditional database storage with distributed ledger technologies, while others used traditional cloud storage and blockchain for data security. Other studies focused on cloud storage integrated with off-chain storage which has security risks associated with it. Data security assurance through cloud storage requires renting to several service providers to avoid central storage which behaves like a centralized system leading to a single point of failure. This approach is used for maintaining data availability but it has cost implications.

Several studies deployed the Ethereum platform with solidity using Ethereum Virtual Machine (EVM), others used late versions of Hyperledger mostly v1.4. For security purposes, some studies deployed all nodes for storage purposes which led to much consumption of data storage space. Some used permissioned blockchain but it was hard to guarantee data privacy.

This study focused on the security improvement of the existing system on data storage security goals to ensure confidentiality, integrity, and availability (CIA). Data storage implementation is through private data collection to guarantee privacy. In this storage strategy, data is held within a database of a Hyperledger Fabric platform and managed with a private data policy.

Deployment of private data collection is by a combination of the actual private data which is stored in a private state database and a hash of that private data. A hash of data of every peer on the channel is written to the ledger. The hashes are used for transaction evidence provision, state validation, and audit purposes. Private data communication between peer-to-peer is enabled by gossip protocol. The gossip protocol facilitates the configuration of CORE_PEER_GOSSIP_EXTERNALENDPOINT on each peer in a channel. This protocol bootstraps the network to ensure communication between nodes.

Deployment of private state and channel state leads to a guarantee of data privacy. Nodes in a network are decentralized with blockchain only to avoid storage complications. This solved other related security challenges which are centralized in nature. Hyperledger Fabric v2.3.2 platform deployed smart contracts embedded in chain codes. The system was virtually integrated with the current health information system, GoT-HoMIS.

2.4 Conceptual Framework

This is the logical view of the study from the beginning to the end of the study. The section shows the logical flow of the proposed solution to the problem. The study was guided by the conceptual framework (Fig. 1).

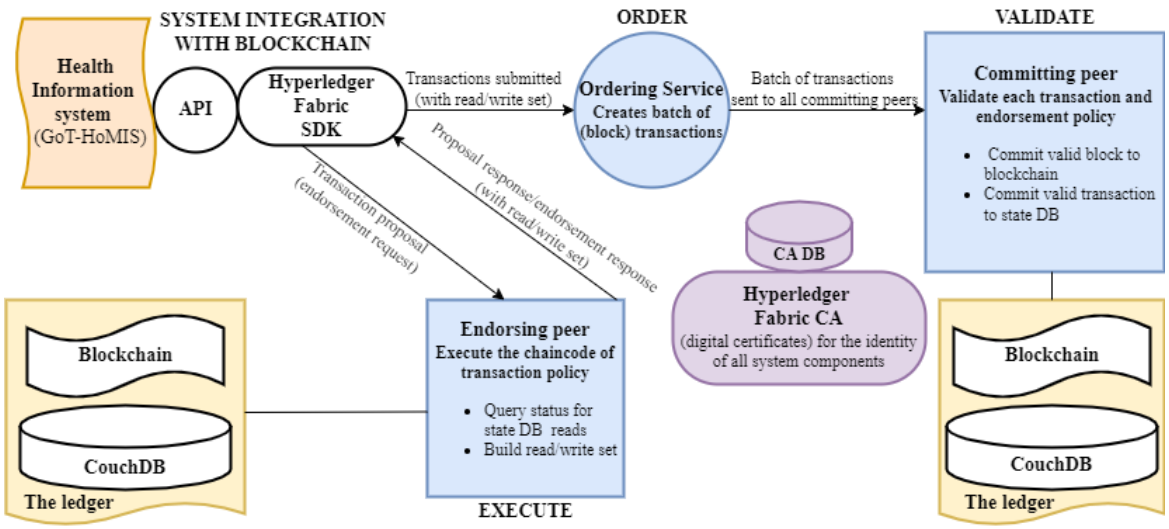


Figure 1: Research framework

The conceptual framework involves the main four parts of integrated system components in its architectural design namely; health information system (SYSTEM INTEGRATION WITH BLOCKCHAIN); transaction execution (EXECUTE); ordering service (ORDERER); transaction validation (VALIDATE).

2.4.1 System Integration with Blockchain

This is the first part of the proposed system which integrates the health information system (GoT-HOMIS) with blockchain through an application program interface (API). The API facilitates the translation of data from GoT-HoMIS to the blockchain. Hyperledger Fabric SDK is an application program. The program initiates the transaction processing cycle from endorsing peer to ordering peer and finalizes with validating peer for committing valid data to the ledger.

2.4.2 Transaction Execution (Endorsing Peer)

This is the first step of transaction execution. Hyperledger Fabric SDK sends transaction proposals to endorsing peers. It is the message that requests the endorsement of a transaction. Endorsing peer queries status of the database reads from the ledger and build read/write set and execute the chain code of transaction policy. After execution of the transaction, the peer gives the endorsement response (proposal feedback) with the read/write status of the ledger.

2.4.3 Ordering Service (Orderer)

Orderer is the dedicated peer within the transaction processing cycle which creates a batch of transactions for the creation of blocks. The peer receives submitted batch transactions with read/write set from Hyperledger Fabric SDK and orders them according to consensus. These transactions will finally be submitted to committing peer for validation.

2.4.4 Transaction Validation (Validating Peer)

This is the last part of transaction processing in which committing peers validate each transaction with reference to the endorsement policy. The peer commits a valid block to the blockchain and commits a valid transaction to the state database (the ledger).

2.4.5 Hyperledger Fabric Certificate Authority

This is the membership service responsible for identifying all entities involved in the system. This provides digital certificates for the identity of all system components and stores them in the Hyperledger Fabric database (Certificate Database).

CHAPTER THREE

MATERIALS AND METHODS

3.1 Area of Study and Scope of the Study

3.1.1 Study Area Justification and Scope

The study took place in the Arusha region, the northern part of The United Republic of Tanzania. Mount Meru Hospital is a referral public hospital located in Arusha city center. It is a large hospital within Arusha where all small public and non-public district hospitals including Manyara region hospitals refer their patients. The study spent twelve months of study from January 2021 to December 2021.

3.1.2 Details of the Existing System

GoT-HoMIS is the current Electronic Medical Record (EMR) system running in client-server network architecture. The system is web-based with a centralized database using Windows Server 2012 hosted in a Local Area Network. The operational environment is on PHP installed with Xampp Server and MySQL database (MariaDB). Various core functional modules are incorporated into the system such as EMR, Billing and Revenue Collection, Tracking and Inventory of Medical Supplies, Laboratory Information System, Practitioner Performance Tracking and Reporting (MTUHA reports) Mfumo wa Taarifa za Uendeshaji wa Huduma za Afya (Kiswahili for Health Management Information System).

The system is integrated with other systems such as District Health Information Software (DHIS) for Health Management Information Systems (HMIS) statistics, Electronic Logistics Management Information System (eLMIS), National Health Insurance Fund (NHIF), and Government Electronic Payment Gateway (GePG) for electronic payment system.

3.2 Research Design

The study adopted Design Science Research (DSR) methodology. It is a problem-solving-oriented technique with digital innovative solutions for addressing real-world problems in system design. The DSR has iterative procedures seeking to bring a relationship between problem and solution.

The study used the DSR methodology due to its technical insights which facilitate capturing, designing, and developing innovative systems. It is the methodology that is short with definite iterative development steps compared to agile or waterfall methodology. Agile or waterfall has no defined end and does not allow revision of steps during system development. It is not possible to go back to the previous development phase to change anything once phase development is completed.

The following are DSR methodology phases:

(i) Problem awareness;

(ii) Suggested solution;

(iii) Designing and development;

(iv) Demonstration of the solution to the problem;

(v) Evaluation of the solution; and

(vi) Communication of the solution.

3.2.1 Problem Awareness

This phase started creating awareness of the system to be developed by identifying existing health system (GoT-HoMIS) security challenges. Problem identification of the current system facilitated the discovery of security challenges. The challenges were identified and documented as system requirements and they were considered during the development processes. This created problem awareness during system development to ensure all identified system security challenges were addressed by the newly developed blockchain-based system.

3.2.2 Suggested Solution

This phase facilitated the study to propose suggested solutions to system security challenges identified in the previous phase of problem awareness. These are preliminary solutions to existing system challenges obtained from interviews and document analysis. The qualitative research approach was used to collect and analyze data to come up with the best solution proposals to system challenges. This phase provided insights for the researcher to acquire knowledge on the feasibility of the solution to the existing system challenges. Suggested solutions were accommodated and considered for the development of the new integrated blockchain-based system.

3.2.3 Designing and Development

This phase provided the insights for data collection and analysis which helps in system designing and development. The phase led to the description and guidance of the study through written research questions leading to practical application in developing the proposed system. Since this research is a case study, data were collected using interviews and document analysis for reliable and valid information for the system to be developed. The phase enabled the researcher to create artifacts, constructs, models, and methods embedded with the design to develop the required system. The phase implemented the artifacts by relating them to suggested system solutions. Designing and development involved desired functionalities of the system and its architecture.

The study developed an integrated secured blockchain-based system which was suggested in the suggested solution phase. It is an integrated component of a blockchain-based system in a peer-to-peer network environment with main four parts (Fig. 9). The system components work together as a single unit in a secured data storage environment. Information sharing and data exchange are carried out without central authority.

3.2.4 Demonstration of Solution to the Problem

This phase demonstrates the functionality of the developed blockchain-based system. The primary goal of this phase is to check whether the new system addresses identified security challenges and if suggested solutions to the system challenges were incorporated. Simulation procedures were based on identified system requirements and suggested solutions to those challenges. The developed system was simulated in a virtualized environment to observe system behavior and its operations. The demonstration helped the study to carefully observe the developed blockchain-based system and see if identified centralized security challenges of the GoT-HoMIS system were addressed.

3.2.5 Evaluation of the Solution

Evaluation of the solution is the phase of determining how well the developed system prototype operates to address security challenges. The evaluation process is carried out through experimentation and simulation to check if the existing system requirements of GoT-HoMIS, with the additional requirements of the newly developed system, are met. These requirements form validation metrics for the developed system. The phase involves the evaluation of

suggested solutions to system challenges if they match system requirements through observed system operation. The purpose of the evaluation is to see whether the developed artifact solved the system security challenges that were identified. The phase gives the flexibility to go back to the designing and development phase for improvement in case system requirements are not met.

3.2.6 Communication of the Solution

This is the last phase of the methodology indicating the end of the research study and system development process. The results of the developed system are communicated to show the overall contribution of the study. It is the communication of the additional knowledge added to the research study area. This study was completed with dissertation writing and published one paper originated from the dissertation.

3.3 Research Methods

The study used a qualitative research method technique for data collection using interview and document analysis methods. These methods were used to gather information which was used to get an understanding of the system to be developed.

3.4 Data Collection Methods

The study deployed both primary and secondary data collection methods.

3.4.1 Primary Data Collection Methods

These are real data sources obtained from interviews and document analysis. The study gathered information on the existing system through document reviews, interviews, and public documents which gave a clear picture of the expected system to be developed that would meet the requirements. Interviews were directed to the technical person, the system administrator. A systematic investigation of the current system was carefully carried out to come up with a blockchain-based system to solve its security challenges. The investigation was focused on the core functionalities and operations of the existing system features.

3.4.2 Secondary Data Collection Methods

These are available data sources including research materials published in peer review journals, books, technical reports, and websites. These were secondary data sources that were used to come up with the detailed knowledge of the system to be developed.

3.5 Data Analysis

The nature of this research led the study to use the requirement engineering process as a data analysis tool since only one participant was used during the data collection process. The tool determines system requirements by considering the needs of the proposed system. The data collection process used an interview in which a technical person, the system administrator was involved.

The data analysis procedure involved four processes namely:

(i) System requirement gathering;

(ii) System requirement analysis;

(iii) System requirement verification and validation; and

(iv) System requirement documentation.

3.5.1 System Requirements Gathering

This is the first step of data collection and analysis which includes detailed study, identification, and collection of system requirements to be developed. The purpose of this step is to determine the problems to be addressed. Therefore, the data collection technique was focused requirement elicitation-based problem domain. This phase is input to system requirement analysis, while the output of this phase is a documented comprehensive list of system requirements (Sadiq & Jain, 2012; Sajjad & Hanif, 2010).

3.5.2 System Requirements Analysis

Requirement analysis is the second step in the requirement engineering process. The purpose of this phase is to analyze and model system requirements gathered from the requirement stage of the data collection process. Requirement gathering gives input to this stage and the output of this phase is a complete documented set of consistent system requirements. The main goal

of this phase is to get the right system requirements. It is not possible that all listed requirements apply to system development for its operation to meet system objectives (Sadiq & Jain, 2012; Sajjad & Hanif, 2010).

3.5.3 System Requirements Verification and Validation

This is the phase of clarifying system requirements to ensure that there is no ambiguity with system requirements. This helps to obtain complete and consistent system requirements with the final requirement specification document. The process validates each stage of the development phase to follow user requirements to meet system objectives. Documented requirements and organizational policies and knowledge serve as input to this phase. All system requirements are inspected and tested using a requirement checklist to detect any defects to get improved system requirements (Sadiq & Jain, 2012; Sajjad & Hanif, 2010).

3.5.4 System Requirements Documentation

This is the last phase of data analysis in which validated and verified system requirements are documented in clearly defined simple terms. The output of this stage is well-structured and clearly defined system requirement specifications to be used during system development. These processes enabled the study to come up with the right list of the new blockchain-based system requirements (Sadiq & Jain, 2012; Sajjad & Hanif, 2010).

3.6 Validity and Reliability of Data

Validity refers to the correlation of data collected to the system requirements. The validity of the system solution relies on data collected which were used for addressing system problems during system development. Valid data results in the right solution to the problem. Reliability refers to the level of agreement with the results. The results should be reliable in that the same findings will be obtained using the same methodology. The study deployed a qualitative research method for data collection to ensure the reliability and validity of this study. Qualitative data were obtained using interviews and document analysis. Interviews were used to gather data from the system administrator. Document analysis used data that were collected from research materials published in peer review journals, books, technical reports, and websites. These methods increased the validity and reliability of data.

3.7 Ethical Consideration

The researcher got an introduction letter from the School of Computation and Communication Science and Engineering at the Nelson Mandela African Institution of Science and Technology, requesting permission to conduct the study at Mount Meru Referral Hospital (Appendix 1). The researcher was permitted to do research work and briefly introduced the research objectives and how they could be beneficial to their system.

The researcher got another introduction letter from the office of the Vice-Chancellor of the Nelson Mandela African Institution of Science and Technology, to the Principal Secretary through Director Secretary, President's Office - Regional Administration and Local Government (PO-RALG) (Appendix 2). The letter was requesting the GoT-HoMIS database schema which was used to integrate with the newly developed blockchain system.

The study ensured ethical principles were adhered to to ensure privacy, integrity, and confidentiality during data collection processes. This included respect during interviewing sessions with the responsible personnel. This facilitated the study to obtain the correct data collected for the new blockchain system to be developed.

3.8 System Development Approach

This is the implementation of the system artifacts mapping them to the proposed blockchain-based system. The study developed a decentralized peer-to-peer blockchain to avoid central authority with centralized data management.

3.8.1 System Development Methodology

The study used a prototype system development methodology. This methodology reflects the research design of the study which has an iterative process allowing refinement of the developed prototype model. The prototype is evaluated and refined to reflect all system requirements. The actual system is developed based on a final working prototype that complies with system requirements.

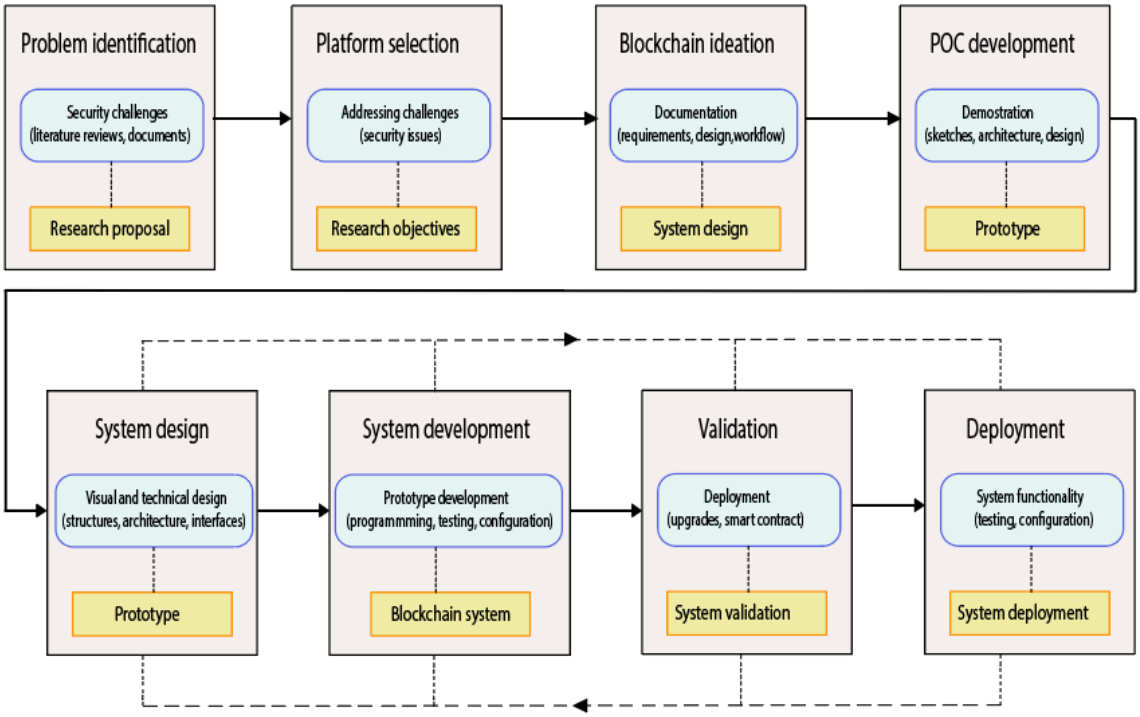


Figure 2: System development lifecycle

The components of the system development lifecycle are:

(i) Problem identification;

(ii) Platform selection;

(iii) Blockchain ideation;

(iv) Proof of concept development;

(v) System design;

(vi) System development;

(vii) System validation; and

(viii) System deployment.

(i) Problem Identification

The study identified security challenges faced by GoT-HoMIS through interviews and document analysis. The interview was directed to the system administrator. Other techniques used for problem identification were literature review, brainstorming with my fellow students, and discussion with supervisors and other professionals in the field of HIS. These ensured the study came up with a clearly defined challenge of GoT-HoMIS.

(ii) Platform Selection

Several blockchain platforms exist, therefore it was important for the study to identify and select the relevant platform to be used during system development. Platform selection depends on the consensus protocol and system security challenges to be addressed. The platform helped to build an application without creating an application from scratch.

For this study, Hyperledger Fabric was selected for developing a blockchain-based system to address existing system security challenges. The framework has distributed ledger solutions where all members of the network have known identities on permissioned networks. It is a modular architecture that increases the flexibility and resilience of the system.

(iii) Blockchain Ideations

Since the platform for developing a blockchain application is identified, the blockchain ideation phase starts. Blockchain ideation refers to the process of brainstorming ideas focusing on system requirements that the new blockchain-based system has to support to address the existing challenges of GoT-HoMIS. This phase will be able to identify which blockchain technology components will be used during development. This helps also to decide whether to use off-chain or on-chain data storage. The study used Hyperledger fabric permissioned blockchain for the system to be developed to address security challenges.

(iv) Proof of Concept Development

This is the stage of the development process showing the feasibility and viability of the new blockchain-based system to be developed. It is the assessment of the system to be developed based on the identified system requirements. It starts with a theoretical build-up of concepts and ideas on how the expected blockchain system will be developed to meet expected system security. Several cases are considered so that the final product parameters of the system are known. A prototype is designed with system architecture, design, and sketches with various system aspects after receiving feedback from the technical personnel on the system showing acceptance and commitment.

(v) System Designing

This phase deals with the creation of user interfaces for each blockchain component used in system development. This includes designs of application program interfaces to be integrated

for running the system at the back end. System designing shows how the system looks, feels, and its technological architecture. Once the designing phase is completed, the system is ready for development.

(vi) System Development

The new blockchain system was developed based on its tentative design of a private data collection channel. The system has two organizations. One organization had only one peer for private data storage and a hash of that private data. The other organization had peers which were decentralized with blockchain. The system was later virtually integrated with the existing system. The approach solved centralized network security challenges and storage space complications.

(vii) System Validation

This is the last phase in system development. It is the stage in which a developed blockchain was validated against system requirements. The validation stage aimed to check if system requirements are met. These were specifically to improve the data security of the health information system (GoT-HoMIS). The validation process ensured all identified security weaknesses of the system are addressed.

(viii) System Deployment

This phase deals with the deployment of the newly developed blockchain-based system. The system was deployed in a virtual environment and simulated with the consistent observation of its operations.

3.8.2 System Design

The developed system was virtually integrated with the existing system (GoT-HoMIS) for sharing health data (Fig. 9). Hyperledger Fabric framework was used for system development configured in a virtualized environment. Virtualization aims to create a virtual blockchain network to avoid the cost of buying several computers that were to be configured in the real physical network. The system used Ubuntu operating system 20.04.2.0, i7-9700 3.00 GHz CPU, 24 GB of RAM, and secondary storage of 1 TB installed in VirtualBox.

3.8.3 System Development

The system was developed using Hyperledger Fabric v2.3.2. Several application packages were installed as prerequisites to set up a development environment. These prerequisites include installation of tools such as; Curl version 7.68.0, Docker version 20.10.2, Docker-compose 1.29.1, node.js V10.19.0, npm 6.14.4 and python 2.7.18. JavaScript was used for the development of the smart contract. The study used Visual Studio Code version 1.55.2 for writing and editing codes.

3.8.4 System Testing

The system was tested based on the requirement specifications and its usefulness. This was done through experimentation, simulation, and scenarios to validate the proposed system. The study used V-Model testing of system development life cycle for system quality verification.

3.8.5 System Validation

The main purpose of validating the system is to check if it has fulfilled the end-user requirements. For the sake of this study, system validation was carried out based on system requirements to ensure developed system functions and operates to address system security challenges. The validation procedure followed a defined order of transaction consensus lifecycle from endorsing peer to committing peer. The process involved system execution based on system requirements, and it was carefully monitored so that it consistently conforms to the expected outputs of system security. Identity management, data integrity, data privacy, data verification, data validation, non-repudiation, and system availability were validation metrics used.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Results

4.1.1 Weaknesses of the Current Health Information System Security of Mount Meru Referral Hospital

The study found main three categories of cyber security attacks. These categories of attacks include malicious IPs, malicious software, and web attacks. Attacks through malicious IPs are caused by default and commonly used user credentials (usernames and passwords) while web-based attacks are due to centralized architecture during client-server communication. There are several malicious software attacks including Trojan horses, backdoors, and Ransomware causing downtimes in the system (Team, 2019-2021).

Database backup of the GoT-HoMIS system is on daily basis and stored in a safe place for retrieval in case of any emergency or data loss. System administrators are also obligated to send the backup after every week to the PO-RALG ICT department (Office, 2021). This is a very local data security management approach and a tedious job for addressing data security.

(ix) Existing System (GoT-HoMIS) Requirements

Table 1 shows the existing system requirements of GoT-HoMIS.

Table 1: Existing system requirements of GoT-HoMIS

S/N	Item	System requirement
1	User identification	The system should be able to identify system users.
2	Reporting	The system should be able to store the history of the system user.
3	Data access	The system should be able to give data access to the authorized user.
4	Data modification	Any data modifications in a database like an insert, delete, or update, should be executed only by the system administrator.
5	Availability	The system should be available only to authorized users whenever needed.
6	Data back up	The system should be able to offer automated data backup.

(ii) Proposed System Requirement Specifications

These are requirements to be taken care of during system development. They include functional and non-functional requirements of the system specifying various aspects of the system as a whole. Functional requirements focus on the specific system behavior or what a system or a sub-system must perform. Nonfunctional requirements refer to system aspects that do not interfere with system operations. These are system requirements specifying additional properties of the proposed system. They also describe the quality attributes of a system design and implementation to ensure system usability and effectiveness. Table 2 summarizes system requirements for the proposed system.

Table 2: Proposed system requirements

S/N	Item	System requirement
1	Identity management	The system should be able to identify its network components with their credentials for authentication to the network to avoid the injection of malicious code from an unknown malicious entity.
2	Data integrity	The system should be able to preserve data integrity to avoid data modification which leads to loss of data integrity.
3	Data privacy	The system should be able to protect data privacy. Data should remain confidential and prevent unauthorized disclosure.
4	Data verification	The system should be able to do data verification to ensure the addition of the right to the database and false data is not added to the system.
5	Data validation	The system should be able to detect malicious and non-malicious data. Validated data should be shared among system users.
6	Non-repudiation	The system should be auditable for the identification of malicious actions to the system to hold accountable the responsible entity.
7	System availability	The system should be able to do an automatic backup, authorized system users should access information, resources, and services when needed.

Other system requirements include; modularity, scalability, portability, reliability, and recoverability. Modularity is required for the improvement of system performance through transaction speed while separating transaction workflow. This leads to system scalability with

a high level of trust. Scalability is the system's ability to handle operations without any restriction. The system can manage easily the growing capacity of network users by scaling up and down as well. The system can run from one environment to the other with the same framework in a web browser. It can also run-on desktops and mobile platforms without effects to the system. This property is known as system portability resulting from pluggable architecture.

Due to decentralized system architecture, the proposed system has an availability which is also known to be system reliability and it is possible to recover it in case of any failure. This property is referred to as system recoverability.

4.1.2 System Development

A blockchain is a linked list of blocks with pointers (Fig. 3), formed by transactions bundled together in a specified period (Kombe *et al.*, 2018). A block created after the first block contains the hash of the previous block's data. Blocks store information validated by cryptographically secured nodes. The linked lists (blocks) are encrypted using hashes and digital signatures based on public/private key encryption algorithms. The hash of the previous block creates a chain of blocks making a blockchain secure.

Figure 3 illustrates a chain of five blocks and each block with its hash and the hash of the previous block. The fifth block points to the fourth block, the fourth block points to the third block, the third block points to the second, and the second one points to the first block respectively. Because it is the first to be created, the first block is also known as the genesis block, and it does not point to any previous blocks.

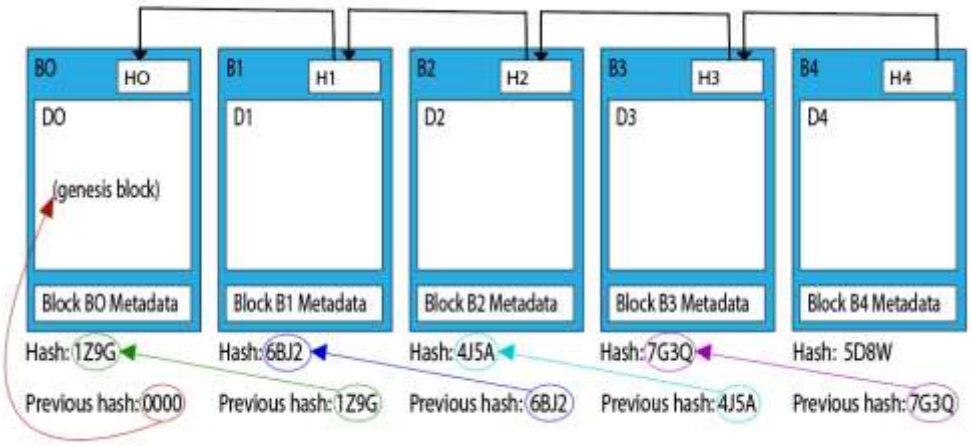


Figure 3: Formation of blockchain linked lists

A fabric ledger consists of a blockchain and a world state, each one with a set of raw facts relating to business operations. A world state is a database with the values of ledger states which are in key-value pairs. Blockchain is a log of transactions that has a record of all changes from the existing world state using the metadata. All transactions are within the blocks, and the blocks are added to the blockchain for auditing purposes (Androulaki *et al*., 2018; Nguyen *et al.*, 2019).

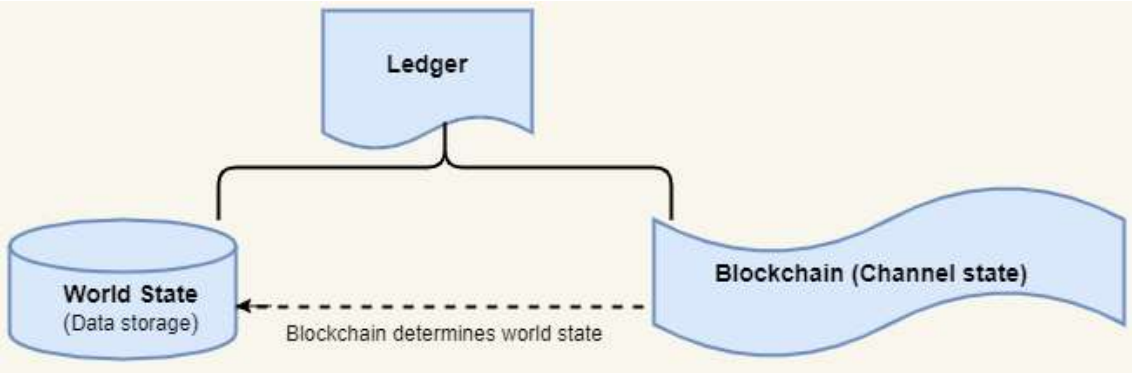


Figure 4: The two components of a Ledger, blockchain and world state

(i) Proposed Blockchain-Based System Architecture

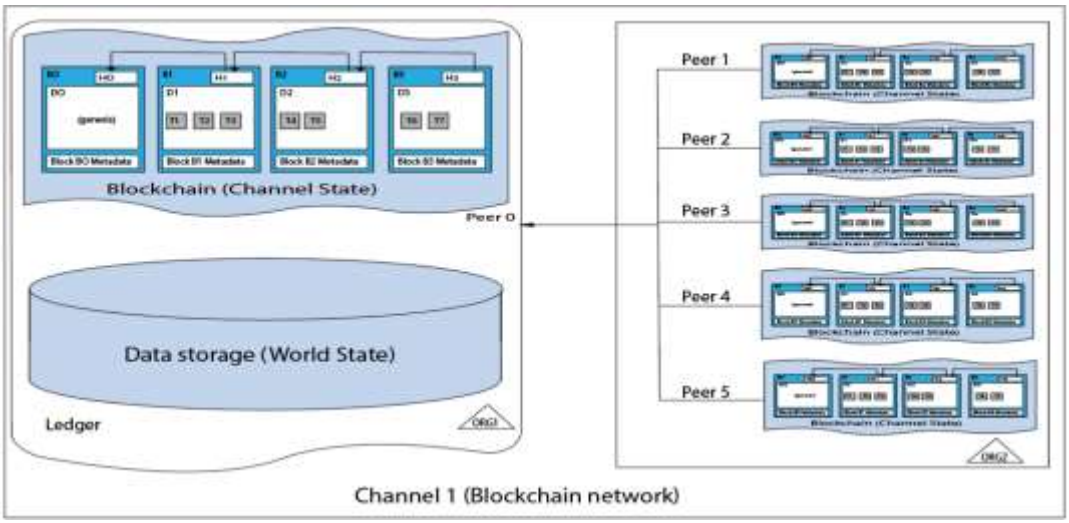


Figure 5: Diagram of the proposed system

The proposed system is a Hyperledger fabric composed of two related distinct features, the database storage and a hash of the data stored in a database. This makes the system have one channel (network) with two organizations; the actual private data (world state) and a hash of the data (blockchain). The actual private data stores the data in a private state database which holds current values (key values pair) of ledger states and is accessed through chain codes. A hash of data stored is written to each node in the network to serve as proof of the transaction.

It is this hashing that is used for validating the ledger state and for audit purposes as well. Organization one is the server with Channel State (blockchain) together with Private State (world state) while organization two has several peers with only channel state (blockchain).

(ii) Channel State

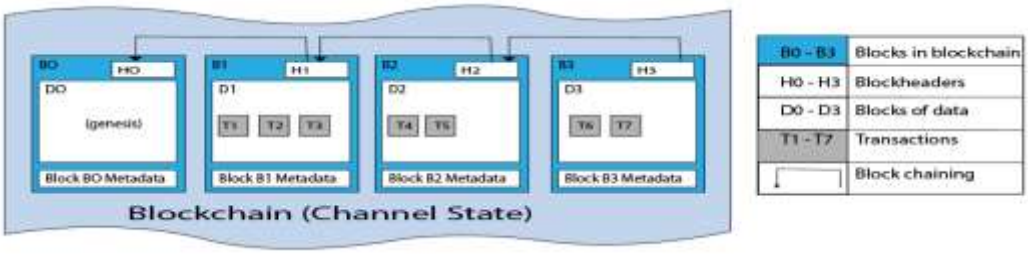


Figure 6: A blockchain (channel state) containing blocks B0 to B3

This is also termed a blockchain with interconnected blocks. It is a log of records of all changes resulting in the current world state. Each block contains transactions appended to the blockchain helping understand the history of changes as a result of the current world state. Each block has a block header with a cryptographic hash of all transactions and it is linked with the hash of the previous block. The blockchain is the same for all peers of the network. The data structure of the blockchain is different from the world state because can not be changed due to immutability. Hashing and linkages guarantee the security of ledger data. The block is in a form of a file while the world state uses a database.

(iii) World State

It records data of the business functions and it is implemented as a database. The state change frequently as states can be deleted, and updated and they can also be created. The application program invokes smart contracts which use simple ledger APIs to *get*, *put* and *delete* states.

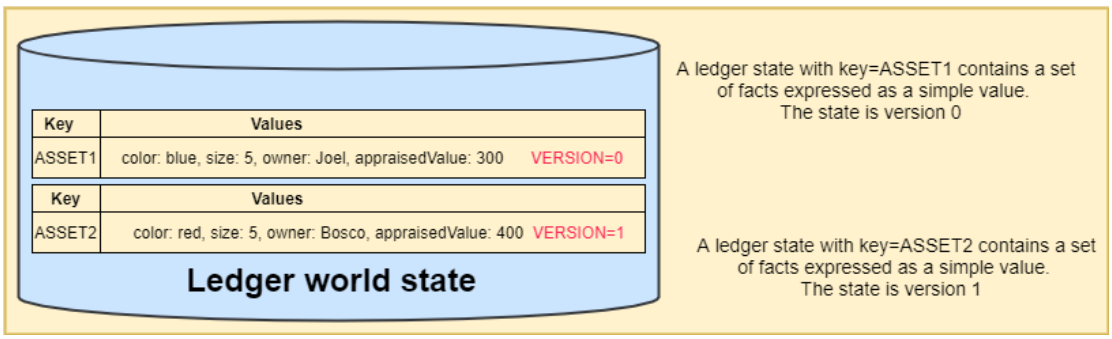


Figure 7: A ledger world state containing two states of two different versions

(iv) Details of Blocks in a Blockchain

Figure 8 gives detailed information about a block in a blockchain. A block contains a block header with a block number, the current block hash, and the hash of the previous block header to ensure that each block is chained to its neighbor leading to immutability. Block data has a record of all transactions listed according to their arrival. Block metadata has block information such as its creation time and certificate, signature, and the node's public key which created the block.

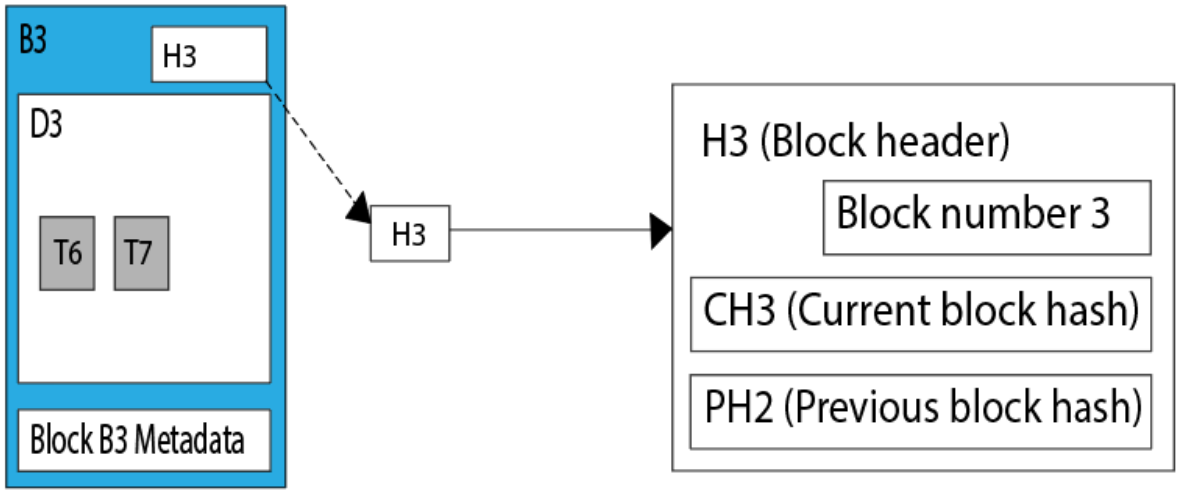


Figure 8: Details of a block in the blockchain

(v) System Components and Environmental Setup

The system is comprised of two organizations connected in a distributed environment (Fig. 5). Each organization has member nodes reflecting the number of operational departments to be connected on a network. Organization one (ORG1) is a ledger for storage purposes while organization two (ORG2) has a decentralized blockchain with five peers keeping track of data storage through metadata. Version 6.0.16 of VirtualBox and Linux operating system were used for system setup. It is a virtual network environment where peers of ORG2 were installed with Hyperledger Fabric with blockchain, and the peer of ORG1 was installed with the Hyperledger Fabric for blockchain and database storage as well.

(vi) The Design of Proposed Decentralized System Architecture

The system is composed of four main parts (Fig. 9). The first part is the GoT-HoMIS system which is the current e-health record system. The second part is the application programming interface (API) which deals with the conversion of records from GoT-HoMIS to decentralized

Hyperledger Fabric ledger transactions. The third part of the system is Fabric SDK, the Hyperledger system development kit dealing with the execution of chain codes. The GoT-HoMIS records are processed and stored in the ledger.

The last part deals with the security of data storage in the decentralized ledger. The storage environment is composed of the world state and blockchain. The records are in the form of NoSQL database format and therefore CouchDB database was used because it is fast and occupies low memory. The CouchDB is for the storage of transactions while blockchain stores the transaction history in an immutable data file structure.

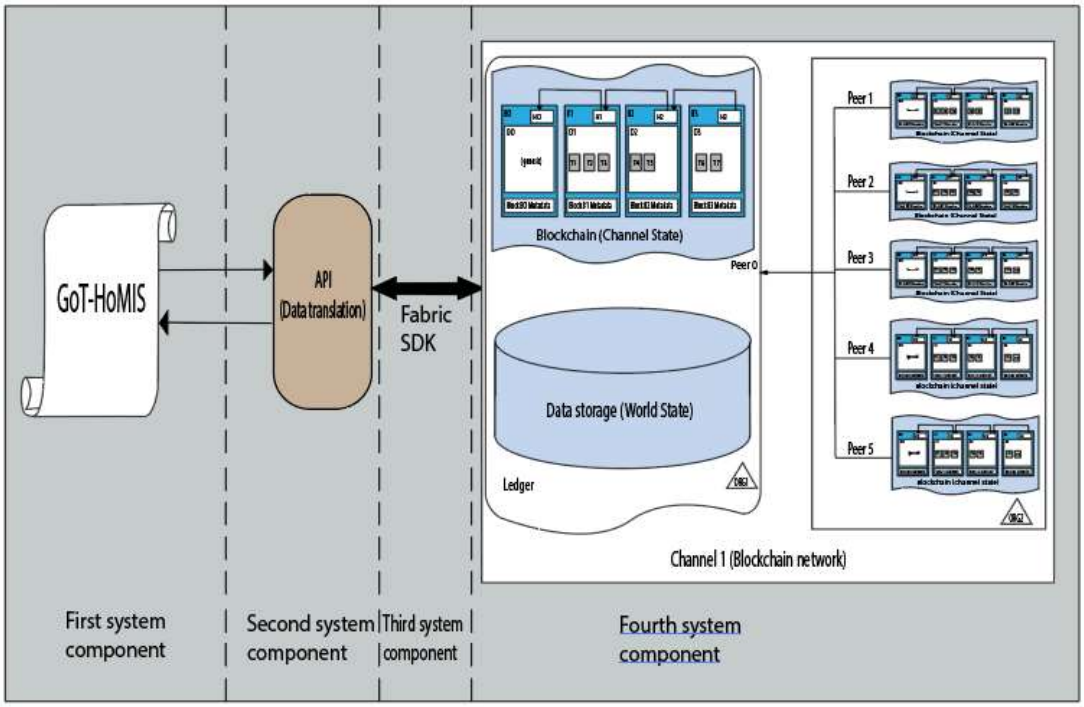


Figure 9: Parts of the system components

(vii) Integration of GoT-HoMIS With Blockchain System

The current system was integrated with the developed system (Fig. 10), through API to enable system functionality for data security. The integration enables the transfer of data from GoT-HoMIS in form of the structured query language (SQL) format to the blockchain network in form of no structured query language (NoSQL) format and vice versa.

(viii) Integrated System Components and Interactions

Figure 10 shows the main parts of the integrated system with the sequence of interactions among the system components. Records from the GoT-HoMIS system are submitted to API

for data translation and conversion from SQL database to NoSQL database. Converted records in the key-value database are sent to Fabric SDK for chaincode execution. After chain code processing, data will be stored in private state data collection in the ledger. The hashes of a private state are stored in a channel state for data integrity verification. The same interaction process can be followed in the exchange of data from private state data collection (CouchDB database) to the MySQL database of GoT-HoMIS and vice versa.

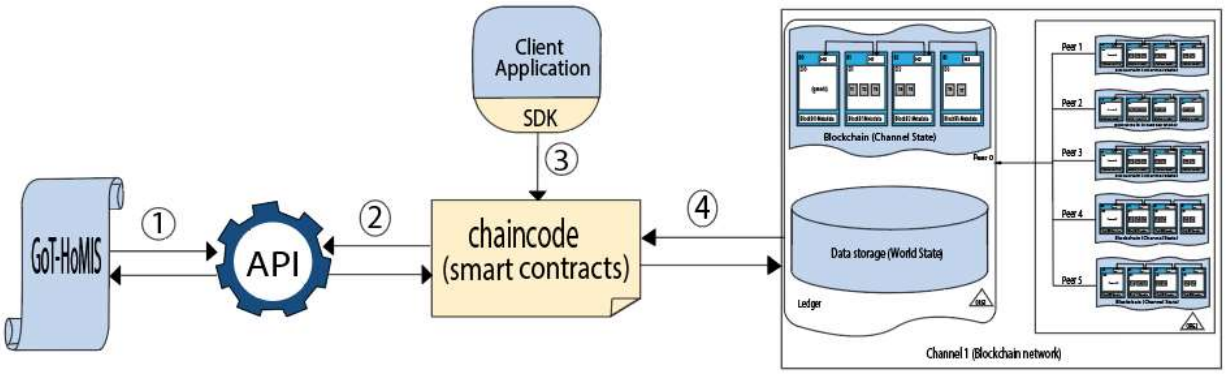


Figure 10: System components interaction and workflow

Figure 11 demonstrates how records of attributes are selected from RDBMS relations (MySQL database) through the API query. The attributes are configured in SQL query by the application program interface. The records are therefore converted to the key-value database (NoSQL) format. From NoSQL which has key-value records, records and transactions are executed in chaincodes for storage in the ledger through Fabric SDK API. Records are stored in a world state (database) and the blockchain. The metadata of the records has a version number attribute for capturing the most current record's version in a ledger (Fig. 7).

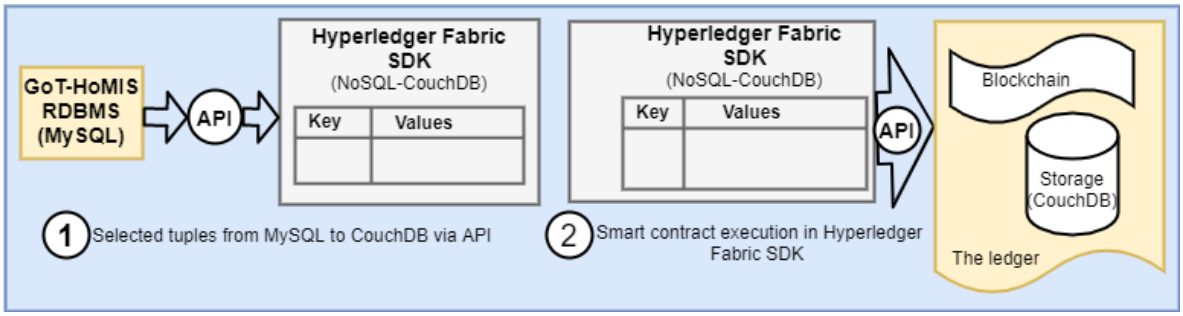


Figure 11: No.1-selection of tuples from different tables of RDBMS (MySQL database) through API query. No.2-Fabric SDK smart contract records and transitions execution for ledger storage

(ix) Hyperledger Fabric SDK

This is a client SDK with a package of important development tools. It makes it easy to develop applications within the programming framework of packages. It consists of programming languages such as Go, Java, or JavaScript among others with the flexibility to use the language of your own choice. Fabric SDK provides API to client applications for interaction with a Hyperledger Fabric blockchain network.

```
root@richard-VirtualBox:~/fabric-samples# curl -sSL https://bit.ly/2ysbOFE|bash -s

Clone hyperledger/fabric-samples repo

===> Changing directory to fabric-samples
fabric-samples v2.3.2 does not exist, defaulting main

Pull Hyperledger Fabric binaries

===> Downloading version 2.3.2 platform specific fabric binaries
===> Downloading:  https://github.com/hyperledger/fabric/releases/download/v2.3.2/hyper
ledger-fabric-linux-amd64-2.3.2.tar.gz
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   649  100   649    0     0    120      0  0:00:05  0:00:05 --:--:--   157
100 73.5M  100 73.5M    0     0   228k      0  0:05:29  0:05:29 --:--:--  240k
==> Done.
```

Figure 12: Installation of Hyperledger Fabric platform

```
===> List out hyperledger docker images
hyperledger/fabric-tools     2.3       a206a1593b4c   2 months ago   448MB
hyperledger/fabric-tools     2.3.2     a206a1593b4c   2 months ago   448MB
hyperledger/fabric-tools     latest    a206a1593b4c   2 months ago   448MB
hyperledger/fabric-peer      2.3       85c825d4769f   2 months ago   54.2MB
hyperledger/fabric-peer      2.3.2     85c825d4769f   2 months ago   54.2MB
hyperledger/fabric-peer      latest    85c825d4769f   2 months ago   54.2MB
hyperledger/fabric-orderer   2.3       7cad713cbfea   2 months ago   37.8MB
hyperledger/fabric-orderer   2.3.2     7cad713cbfea   2 months ago   37.8MB
hyperledger/fabric-orderer   latest    7cad713cbfea   2 months ago   37.8MB
hyperledger/fabric-ccenv     2.3       627c556b15ca   2 months ago   514MB
hyperledger/fabric-ccenv     2.3.2     627c556b15ca   2 months ago   514MB
hyperledger/fabric-ccenv     latest    627c556b15ca   2 months ago   514MB
hyperledger/fabric-baseos    2.3       e50ea411d694   2 months ago   6.86MB
hyperledger/fabric-baseos    2.3.2     e50ea411d694   2 months ago   6.86MB
hyperledger/fabric-baseos    latest    e50ea411d694   2 months ago   6.86MB
```

Figure 13: Installed Hyperledger Fabric tools in Docker containers

```
richard@richard-VirtualBox:~$ sudo apt install docker.io
[sudo] password for richard:
Reading package lists... Done
Building dependency tree
Reading state information... Done
docker.io is already the newest version (20.10.2-0ubuntu1~20.04.2).
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
```

Figure 14: Docker installation in Ubuntu 20.04.2

Prerequisite installation and configurations were done before the installation of Hyperledger Fabric 2.3.2. These were Curl 7.68.0, Docker 20.10.2, Docker-compose 1.29.1, node.js V10.19.0, npm 6.14.4, python 2.7.18, and Visual Studio Code 1.55.2, freely available as open-source code. After the installation of these prerequisite applications, the configuration of the network and consensus protocol followed.

(x) Attributes of a Transaction During Chaincode Execution

Transactions capture the changes in the database storage implemented through smart contract executions. Smart contracts were developed using JavaScript which is among the programming languages that can be used in smart contract development. Figure 14 gives brief detailed major fields and attributes of a transaction in a block resulting from chaincode execution.

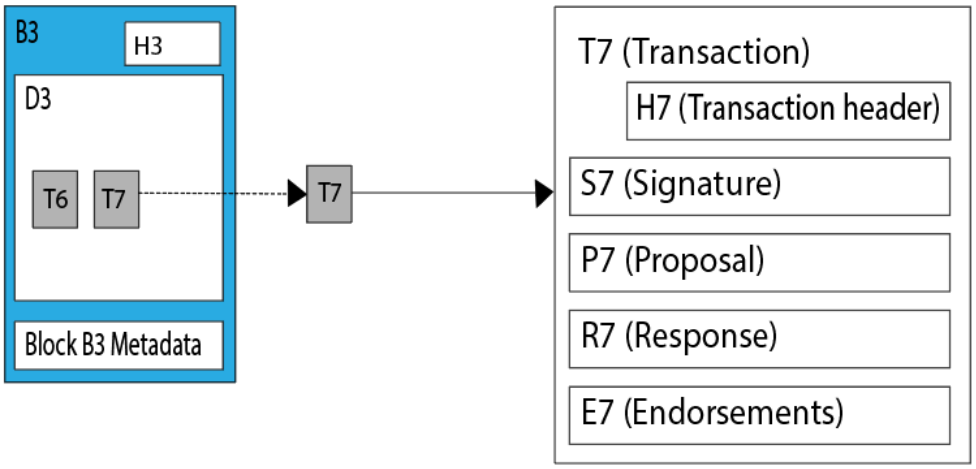


Figure 15: Details of a transaction during chaincode execution in a block of data

The following are detailed descriptions of transaction attributes:

(i) Transaction header (H4) has important metadata for the transaction such as the name of the chaincode and its version;

(ii) Transaction signature (S4) created a cryptographic signature for checking the details of a transaction whether it has been tampered with or not;

(iii) Proposal (P4) with encoded input parameters to the chaincode for ledger update;

(iv) Response (R4) recording keeping of the world state database, before and after storage values; and

(v) Endorsement (E4) records the storage of signed transaction responses of nodes involved in the endorsement and validation of the transaction.

(xi) Demonstration of Security Improvements of the Proposed System

Hyperledger Fabric is a permissioned modular architecture blockchain with a flow of transactions that follows the execute-order-validate model (Fig. 16). Its architecture consists of different types of nodes such as peers, orderers, and clients with identities provided by Fabric CA (Certificate Authority). Processing of a smart contract starts with the generation of a transaction proposal from a client to endorsing peer. The proposal is endorsed and submitted back to the client. The client gathers all information about the endorsed transaction and submits them to the ordering node service. The ordering service receives a batch of transactions for ordering and submits them to committing peer for execution. A block is generated from the ordered batch of transactions, validated, and committed to the ledger (Javaid *et al.*, 2019; Manevich *et al.*, 2018; Nguyen *et al.*, 2019).

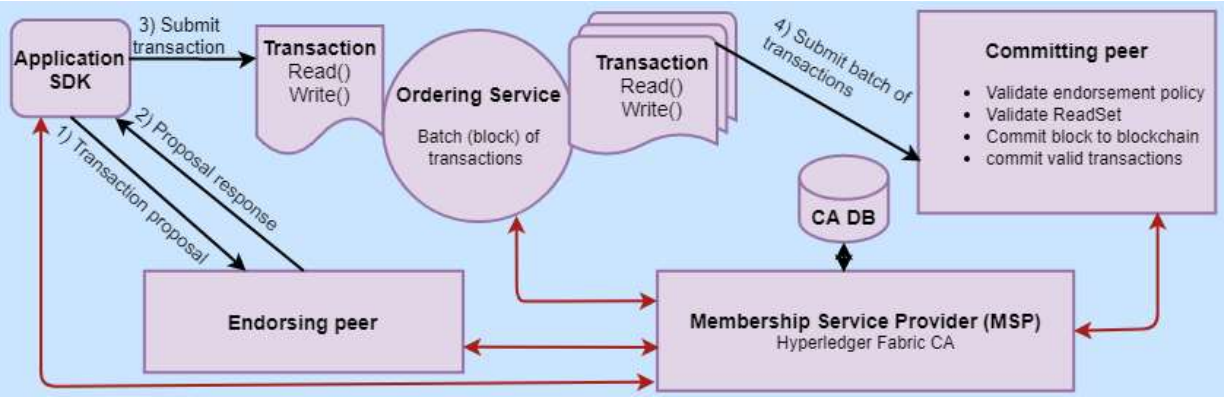


Figure 16: Security view of the execute-order-validate Fabric architecture

(a) Cryptographic Identification

Cryptographic identification provides security trust through the authentication of entities to the network (Ismail & Materwala, 2019; Nguyen *et al.*, 2019). Their identities are secured by a private key and a public certificate (Saad *et al.*, 2019; Sankar *et al.*, 2017). This mitigates spoofing attacks which use impersonation techniques to tamper with trusted source credentials. Spoof attacks compromise the communication identity of an authorized user in a network and redirect to a malicious source. Cybercriminals use this attack type in combination with other attacks, such as IP address spoofing, in combination with SYN flood attacks. This exposes the network to attacks through opened connection. Permissioned Hyperledger Fabric mitigates this risk by generating unique X.509 digital certificates for all its network members, revoked certificates will be denied system access.

(b) Transaction Endorsement Policy

Figure 16 illustrates the endorsement process of a transaction during chaincode execution. Hyperledger Fabric network has the mechanism to guarantee the security of transactions on the network. It ensures that transactions are not compromised through endorsement policies (Nguyen *et al*., 2019). Endorsement policy ensures transaction integrity hence preventing inconsistent transactions. Transactions will be created and stored in a way that will be prevented tampering and make it easy to detect any change in a smart contract execution (Khan *et al*., 2020). A transaction proposal will be endorsed if endorsement responses are listed in the policy match to avoid unexpected results (Javaid *et al*., 2019). Endorsing peers can not be suspended because transaction proposals that need their approval can not proceed as well. Likewise, no new transactions will be committed if the endorsement is suspended. Endorsement is one of the deployed trust mechanisms to stop malicious peers in the system (Fig. 16).

(c) Transaction Verification by the Ordering Node

The main function of the ordering service is to approve the addition of transaction blocks into the ledger (Wang & Chu, 2020). Transaction verification is done through communication between the endorsing and committing peers (Ismail & Materwala, 2019). The orderer verifies all the cryptographic information of the policy and other aspects of the chaincode execution on a channel (Saad *et al.*, 2019). If the results of endorsement responses mismatch, invocation request will not be granted and the ledger will not be updated although data will be stored for audit purposes. This mechanism is implemented to avoid the injection of malicious code. If the chaincode policy is correct, then the ordering node will send the data to all peers in the channel. All peers in the network will confirm that they have a valid transaction to be appended to the ledger. Every peer will append the read/write set to its ledger to have synchronized results.

(d) Transaction Validation

During validation of read/write sets to the ledger, the ordering peer verifies the chaincode to be executed on a channel. If happens endorsement responses of all peers are not the same, then invocation requests will not be permitted due to data mismatch. The ledger state will not be updated due to suspiciousness on transaction differences possessing suspicious data which might be replay attacks (Sankar *et al*., 2017). Every node in a network is responsible for data sharing verification to make sure false data is not added and existing data is not deleted.

Member nodes have to agree on whether the new block of data is valid and eligible for the shared ledger.

Replay attacks are also compared to man-in-the-middle, where the hacker interferes with the network communication between two hosts. The attacker eavesdrops on a network and intercepts it fraudulently. The hacker gains access to data during transmission and retransmits them as if it is from an authentic source. Network resources subjected to this attack visualize the attack as a legitimate message. Data transmitted is delayed and may even be tampered with and then resent to the receiver with malicious information. Hyperledger Fabric mitigates this attack by using read/write sets for transaction validation (Honar *et al.*, 2021; Mustafa & Waheed, 2021). Transaction validation is also used to address double-spending problems. It ensures ordered execution and committing of transactions are followed and no transaction will be skipped.

(e) Mechanisms of Digital Signatures

Digital signatures play a vital role during the endorsement process of a transaction. An endorsement request is signed by the sending client application and validated by the receiving peer. Valid transactions with the same endorsement responses will be executed and committed. Non-repudiation is attained through mechanisms of digital signatures. There is no way that an entity or any system user can deny its actions including malicious activity. Entities can be held accountable because transactions created cannot be impersonated or forged. Membership service management grants auditable mechanisms that lead to accountability of individual Fabric components. Hyperledger Fabric screens the events using digital signatures to track who did what during ledger creation (Saad *et al*., 2019).

(f) Contract Confidentiality

Contract confidentiality is attained through encryption algorithms during the endorsement process. Created transactions and smart contracts are concealed to unauthorized entities at the same time ensuring their correctness. Transactions can be verified if they are legal to be invoked by the respective entity. Every entity has control over its transaction sharing hence creating user participation privacy (Sankar *et al*., 2017).

(g) Immutability of Blockchain Linked Lists

The chaining of blocks creates layered protection against cyber-security threats through encryption algorithms to maintain data integrity. The blocks are encrypted using hashes and digital signatures based on public/private key encryption algorithms. The hash of the previous block creates a chain of blocks making a blockchain secure. The chaining process of blockchain blocks hardens hacking attempts of penetrating the system. This creates data immutability and difficulty in tampering with data unless the hacker attacks the whole network at once and alters all data simultaneously which is not possible (Khan *et al*., 2020; Paik *et al.*, 2019; Sousa *et al.*, 2018).

More members on a network increase security hence reducing the possibility of hackers attacking the system. System attack is lowered due to the complexity created by several nodes in the network. Suppose a system hacker wants to tamper with the third block (Fig. 2), this will lead to hash changes of the block making block three and other following blocks invalid (Fig. 4). The reason for invalidity is because block three does not contain the correct hash of the previous block. Therefore, changes made to a hash of a single block will lead to the invalidation of all other subsequent blocks.

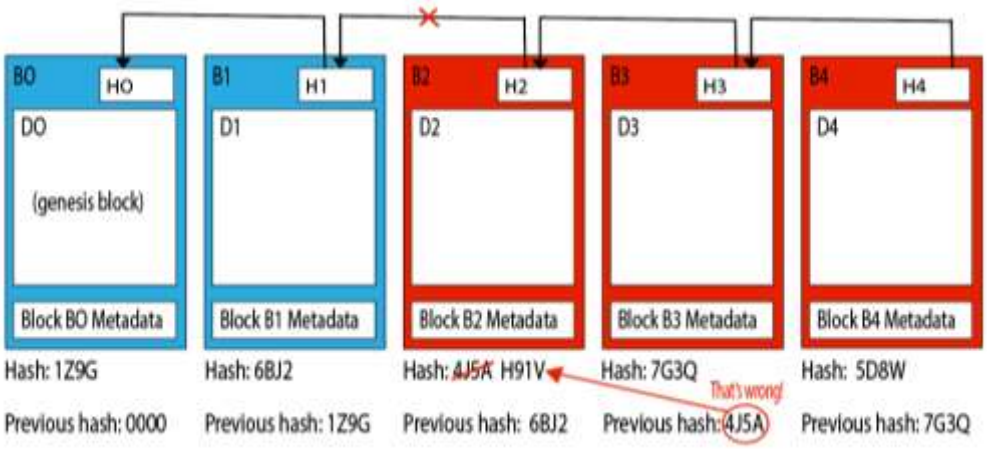


Figure 17: Demonstration of data modification detection (loss of integrity)

(h) Transport Layer Security

Fabric architecture uses Trasport Layer Security (TLS) 1.3 for data transit encryption to avoid accidental, and intentional data exposure. Transport Layer Security is a security protocol with cryptographic algorithms for privacy and data security. The protocol provides end-to-end secure communications between Fabric components. Authentication is part of TLS using credentials

created from Fabric CA to ensure authentic communications between the hosts (Saad *et al*., 2019). It is also the operator's responsibility to prevent this security breach to occur by following the best information security practices of the Fabric.

(i) Network Intrusion Detection

The blockchain is composed of blocks and each block has its data, the block's hash, and the previous block's hash (Fig. 3). A hash is a unique identification compared to a fingerprint. It identifies a block with all its contents. Hashing data, storing the hash value in a block, and the linkage of the block through hash values in the blockchain system help to detect intrusions in the network. Hashes are used to detect changes made to blocks. When a block is created, its hash is also calculated. The hash will change if there will be any changes inside the block. The block will not be the same once the hash of a block is changed. Data storage in a blockchain can be monitored through changes that will be made to blocks using unique signatures. Any attempt at data modification in chain consistency will mean a network attack leading to loss of integrity, and confidentiality, and an intruder can cause unavailability of the system. Figure 17 illustrates how network intrusion can be detected.

(j) Decentralized Blockchain Peer-to-Peer Fabric Architecture

A peer-to-peer network architecture in a decentralized environment has removed challenges based on centralized system architecture. This has eliminated the central authority of data management which leads to a single point of failure creating a specific target of malicious attacks. Data storage and control are decentralized and spread among nodes in the network. All member nodes agree and decide on which block of data is to be added to the chain. There is no "master" in charge of all nodes. Each peer has equal access to the chain. This has created system availability, with high fault tolerance and automated backup feature as the nodes in the network store the same copy of data (Fan *et al*., 2020; Paik *et al*., 2019; Tsoulias *et al*., 2020).

(xii) System Testing

The system was tested based on requirement specifications. Testing procedure used V-Model of the system development process. Every phase has a verification process corresponding to validation activity. The verification process involved a review of system requirements to find out whether specific requirements were met. The testing process was conducted through the execution of system codes.

V-Model testing has four phases which were used during the system quality verification process, namely:

- Unit testing;
- Integration testing;
- System testing; and
- Acceptance testing.

(a) Unit Testing

This is a test of business logic that is implemented through smart contracts. This test was done during the design and development phase to ensure the correct functionality of individual units of the smart contract. Consistency of transactions was checked through individual node testing on the network to ensure secured operations. Consensus algorithms and data immutability were verified and validated. Endorsement policies in the transaction consensus life cycle were adhered to ensure the proper operation of the developed system.

(b) Integration Testing

Integration testing comprised a testing combination of Fabric system components. The main goal is to make sure all connected Fabric components work together. Integrated system components were tested to ensure their smooth interaction and operation to avoid system failures. It involved application programming interface testing to evaluate system compatibility.

(c) System Testing

System testing was conducted to analyze the whole system's functionality. The test was purposely intended for the evaluation of system requirements compliance. The main purpose was to meet the goals of system security-CIA Triad. The blockchain system was tested extensively to avoid system vulnerabilities that may lead to system attacks.

(d) Acceptance Testing

This last phase involved testing the ease of use and usefulness of the system to its end users. The test was performed to measure the degree of acceptance of the system developed for solving

security challenges. The study used functional and non-functional requirements to perform user acceptance testing.

4.1.3 System Validation

System validation was carried out based on system requirements to ensure developed system functions and operates to address security challenges. The validation procedure followed a defined order of transaction consensus lifecycle from endorsing peer to committing peer. The process involved system execution based on system requirements, and it was carefully monitored so that it consistently conforms to the expected outputs of system security.

The following screenshots show proof that the system was implemented and worked smoothly without interruption or any failure during its operation.

Figure 18 shows the smart contract to be executed to create a block of data with the key 2000.07.

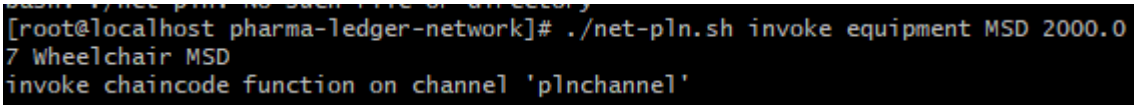


Figure 18: Invocation of a smart contract

Figure 19 shows the result of the invoked smart contract while creating a block of data.

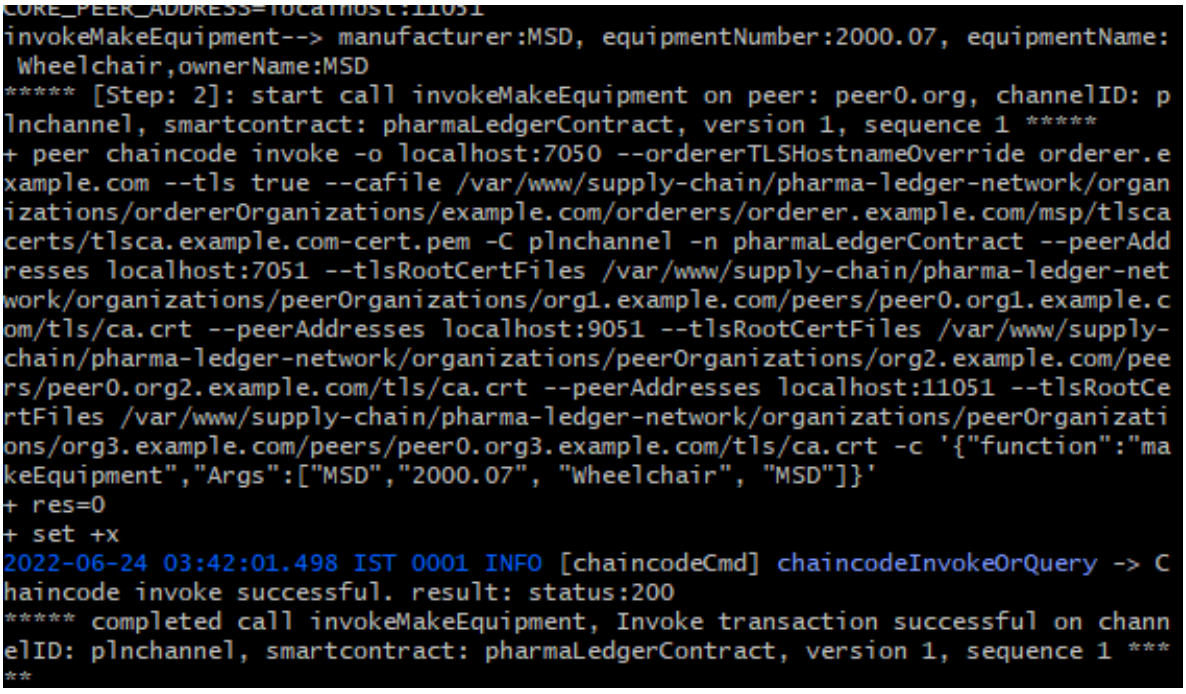


Figure 19: The outputs of a smart contract execution

Figure 20 shows systematic steps of the smart contract execution captured during block creation.

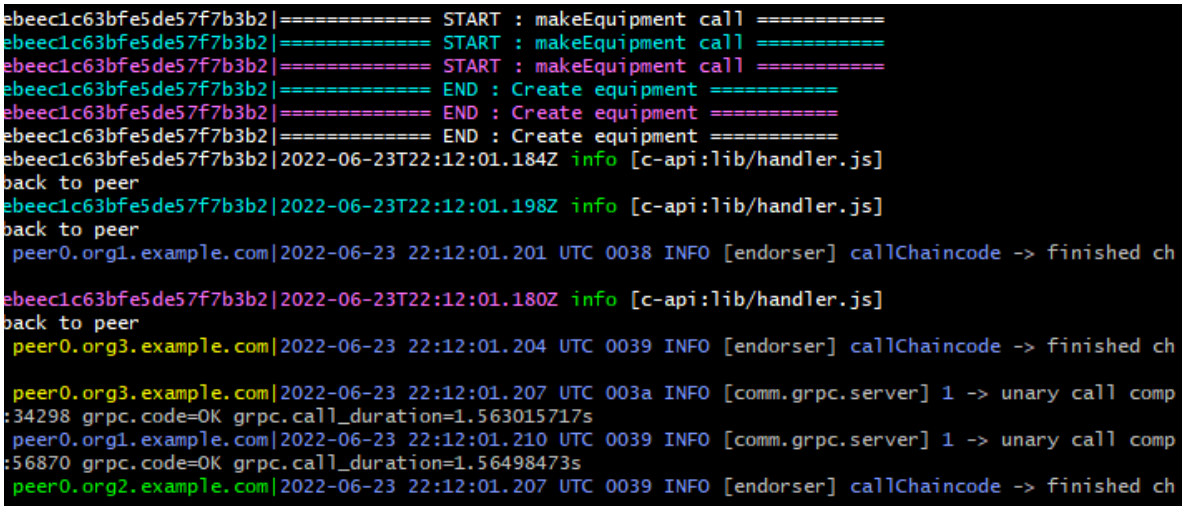


Figure 20: Execution steps of the smart contract from the start to the end of a transaction

Figure 21 shows detailed processes of block creation from endorsing peer to committing peer.

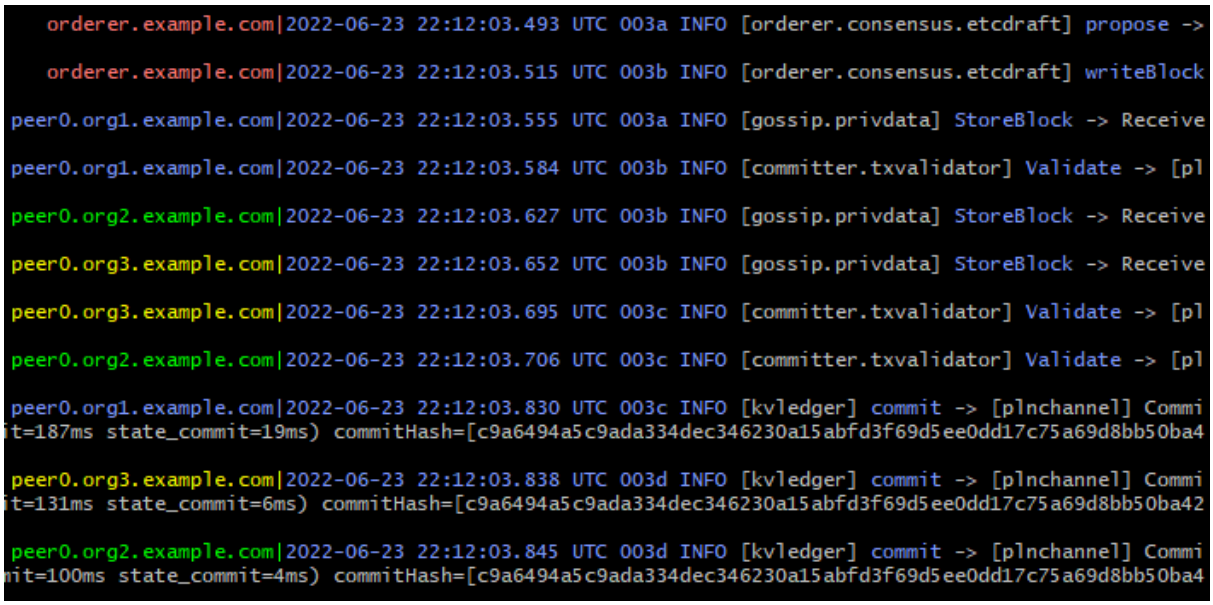


Figure 21: Continuation of smart contract execution from endorser to committer

Figure 22 shows the smart contract to be executed to view the history of a particular transaction.

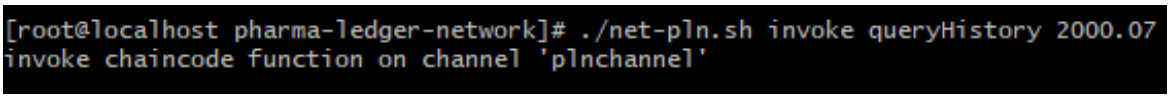


Figure 22: Execution of a smart contract to query transaction history

Figure 23 shows the transaction history of equipment number 2000.07.

```
63bfe5de57f7b3b2|getting history for key: 2000.07
63bfe5de57f7b3b2|[
63bfe5de57f7b3b2|  {
63bfe5de57f7b3b2|    equipmentNumber: '2000.07',
63bfe5de57f7b3b2|    manufacturer: 'MSD',
63bfe5de57f7b3b2|    equipmentName: 'Wheelchair',
63bfe5de57f7b3b2|    ownerName: 'MnyawiPharmacy',
63bfe5de57f7b3b2|    previousOwnerType: 'WHOLESALER',
63bfe5de57f7b3b2|    currentOwnerType: 'PHARMACY',
63bfe5de57f7b3b2|    createDateTime: 'Thu Jun 23 2022 22:12:00 GMT+0000 (Coordinated Universal

63bfe5de57f7b3b2|    lastUpdated: 'Fri Jun 24 2022 08:58:35 GMT+0000 (Coordinated Universal Ti

63bfe5de57f7b3b2|  },
63bfe5de57f7b3b2|  {
63bfe5de57f7b3b2|    equipmentNumber: '2000.07',
63bfe5de57f7b3b2|    manufacturer: 'MSD',
63bfe5de57f7b3b2|    equipmentName: 'Wheelchair',
63bfe5de57f7b3b2|    ownerName: 'XYDistributers',
63bfe5de57f7b3b2|    previousOwnerType: 'MANUFACTURER',
63bfe5de57f7b3b2|    currentOwnerType: 'WHOLESALER',
63bfe5de57f7b3b2|    createDateTime: 'Thu Jun 23 2022 22:12:00 GMT+0000 (Coordinated Universal

63bfe5de57f7b3b2|    lastUpdated: 'Fri Jun 24 2022 08:54:14 GMT+0000 (Coordinated Universal Ti

63bfe5de57f7b3b2|  },
63bfe5de57f7b3b2|  {
63bfe5de57f7b3b2|    equipmentNumber: '2000.07',
63bfe5de57f7b3b2|    manufacturer: 'MSD',
63bfe5de57f7b3b2|    equipmentName: 'Wheelchair',
63bfe5de57f7b3b2|    ownerName: 'MSD',
63bfe5de57f7b3b2|    previousOwnerType: 'MANUFACTURER',
63bfe5de57f7b3b2|    currentOwnerType: 'MANUFACTURER',
63bfe5de57f7b3b2|    createDateTime: 'Thu Jun 23 2022 22:12:00 GMT+0000 (Coordinated Universal

63bfe5de57f7b3b2|    lastUpdated: 'Thu Jun 23 2022 22:12:00 GMT+0000 (Coordinated Universal Ti

63bfe5de57f7b3b2|  }
63bfe5de57f7b3b2|]
```

Figure 23: The results of a smart contract for tracking the history of a transaction.

The following validation metrics were used; identity management, data integrity, data privacy, data verification, data validation, non-repudiation, and system availability.

(i) Identity Management

Every unit in a channel is cryptographically identified. All identities were secured by a private key and a public certificate. Each organization in a channel has its own Certificate Authority (CA), proof of trust for its members' identities. Network member identities were created through unique X.509 digital certificates (Fig. 16). This provided security trust by proving the authenticity of the entity in a network. Network members' identity is certified to join the network while denying network access to revoked certificates.

(ii) Data Integrity

The chaining of blocks preserves data integrity. Even if a node tampers other nodes will remain secure. Secured nodes will continue with data verification, keeping a record of the entire network. Any data alteration in the network is analyzed and compared to the whole chain metadata excluding those not matching. If a block in a chain tampers, hash changes of the block tampered and other subsequent blocks will be invalidated. The reason for invalidity is due to the fact tampered block does not contain the correct hash of the previous block. Tampering with the data will need to attack every single node on the network and alter all of their data simultaneously, which is not possible (Paik *et al*., 2019).

(iii) Data Privacy

System transactions and smart contracts are hidden to unauthorized nodes at the same time ensuring their correctness. Endorsement policy ensures the confidentiality of a contract by concealing it from unauthorized entities (Fig. 16). Every entity has control over its transaction hence creating privacy of user participation. Mechanisms of contract confidentiality are implemented through encryption algorithms. System confidentiality is also attained through the chaining process where encryption algorithms are implemented. The linked lists and blocks are encrypted using hashes and digital signatures based on public/private key encryption algorithms.

(iv) Data Verification

Transaction verification is carried out through communication between the endorsing and committing peers. The orderer verifies all the cryptographic pieces of information of the endorsement policy and other aspects of the chaincode execution on a channel (Fig. 16). If the results of endorsement responses mismatch, the invocation request will not be granted and the ledger will not be updated although data will be stored for audit purposes. If the chaincode policy is correct, then the ordering node will send the data to all peers in the channel. All peers in the network will confirm that they have a valid transaction to be appended to the ledger. Every peer will also append the read/write set to its ledger to have synchronized results.

(v) Data Validation

Every node in a network is responsible for data sharing verification to make sure false data is not added and existing data is not deleted. Member nodes come to a consensus on whether the

new block of data is valid and eligible to be shared in the ledger. Data validation is carried out through the endorsement process of transactions during chaincode execution. Endorsement policy ensures transaction proposal is endorsed if it matches endorsement responses that contain valid data. An endorsement request is signed by the sending application and validated by the receiving peer. New transactions will not be committed if the endorsement is suspended.

(vi) Non-repudiation

Hyperledger Fabric screens events using mechanisms of digital signatures to track who did what during ledger creation. Non-repudiation is implemented during transaction endorsement and processing (Ribeiro *et al.,* 2020). Identity management plays a big role in accountability to system users through auditability of user behavior. Membership service management grants auditable mechanisms to users which leads to accountability to individual Fabric components (Fig. 16). There is no way that an entity or any system users can deny its actions. Entities can be held accountable for their transactions because transactions created cannot be impersonated or forged.

(vii) System Availability

Decentralized peer-to-peer network architecture removed a single point of system failure of centralized data storage management. This created system availability with fault tolerance for authorized users. This created automated data backup management where nodes in the network store the same copy of data, and information is exchanged without a central authority.

4.1.4 System Performance Measurement

Performance measurement of the developed system was evaluated against the Ethereum blockchain system identified in a literature review. This measurement was purposively carried out to prove the credibility of the developed system based on the research gap. Performance metrics focused on overall performance and detailed performance of the system. Overall performance evaluated the system's throughput and latency. The detailed performance provided detailed information on the whole process of system performance to discover performance bottlenecks.

(i) Detailed Performance Evaluation

(a) Built-in Support for Data Privacy

The proposed system deployed a private data policy in which data is stored in a private state database with the hash of that private data. Private data policy uses private data collection which has two related distinct features of Hyperledger (Brotsis *et al.,* 2020). These features are the actual private data (world state) and a hash of the data (blockchain). The actual private data stores the data in a private state database which holds current values (key values pair) of ledger states and is accessed through chaincodes. A hash of the data storage is written to each node with the access rights to the private data leading to data privacy in a blockchain network. Private data is also referred to be off-chain data or off-chain transactions. Ethereum uses the same approach of private data to address system performance challenges while Fabric network uses a private data approach for addressing data privacy challenges. The storage of Ethereum's private is outside the platform while with Fabric, the storage is held within the framework.

(b) Data Storage Capacity

Data storage implementation of the developed system was through private data collection to guarantee privacy as well as minimize storage capacity. Data is held within a database of a Hyperledger Fabric platform and managed with a private data policy (Brotsis *et al*., 2020). This has led to a single storage device compared to the deployment of all nodes for data storage which has much consumption of storage space.

(c) Cost Implications

Cloud storage is one of the approaches to avoiding large data sets challenging. Using only one service provider for cloud storage service behaves like the centralized system architecture. This creates a single point of system attack and failure in case of any network vandalism to the service provider. Cloud storage requires renting to several service providers to avoid the risk of central storage. This approach is used for maintaining data availability but it has cost implications as compared to the developed system approach which used a single storage device within the Hyperledger Fabric.

(d) System Confidentiality

Hyperledger Fabric is a private permissioned blockchain system requiring its users to be granted permission to join and connect to the network. The system can be designed into sub-channels allowing the same nodes to participate in other multiple channels at the same time while guaranteeing data storage confidentiality. Ethereum is a permissionless blockchain system where anyone can join the network, interact with the system ledger, and access stored data. No data privacy since whatever is stored in the blockchain system is visible to all network members (Androulaki *et al*., 2018; Saad *et al*., 2019).

(e) Computational Power

Hyperledger Fabric network involves independent organizations for transaction validation. Transaction validation is relatively quick with low computation power resulting from the low cost and low latency of Fabric transactions (Kombe *et al.*, 2018). Transactions have low computational power consumption because they are signature-based as compared to Ethereum. Ethereum transaction validation process has much high computation power due to its protocol which involves the computation of complex mathematical calculations during the addition of a block to the chain (Manevich *et al*., 2018; Saad *et al*., 2019; Sousa *et al*., 2018). The validation process can also be reversed leading to wasteful consumption of resources if the miners fail to successfully add a block in a chain. This is contrary to Hyperledger Fabric where transactions are irreversible.

The cost of computational power is lowered in the Fabric network due to sharing of the validation process across the network rather than leaving the whole task to specific organizations. Transaction proposal is endorsed by endorsing peers and sent to ordering nodes for ordering service. After ordering the service, transactions will be validated by all peers involved in a respective transaction. The suffering of a specific single set of computation processes is removed. This causes a reduced computational or latency burden during smart contract processing (Monrat *et al.*, 2019). It is a very different process in the Ethereum network where only miners bear the whole cost of transaction processing leading to high computational power.

(f) Transaction Ordering and Validation

Ordering service in a Hyperledger Fabric is compared to miners in Ethereum, but it has much less computation power compared mining process in Ethereum. Ordering service orders batch of transactions and distributes them to the validating peers on the network. Each organization runs an ordering service to avoid the responsibility of a single organization to create and distribute blocks in a chain. The service does not either access ledger transactions or validate transactions. Its main task is to order transactions to be validated and committed to the ledger. This is an approach differentiating Fabric from Ethereum which addresses several security challenges faced by Ethereum including computation. The creation and distribution of blocks in Ethereum is the responsibility of a single organization leading to high computation power and inconsistency (Androulaki *et al*., 2018).

(g) Modularity, Plug and Play Components

Fabric's architecture enables configuration in multiple ways which lead to innovation and optimization that satisfies solution requirements. Fabric supports the use of smart contracts for general-purpose programming languages without constraining to a specific language. Pluggable consensus protocol made it to be effective for customization to specific user requirement models. Its modularity led to the high performance of consensus services and support of various database management systems. This addressed challenges such as confidentiality, performance, scalability, flexibility, and resiliency faced by Ethereum (Androulaki *et al*., 2018; Sousa *et al*., 2018).

(ii) Overall Performance Metrics

Performance measurement was facilitated by the combination of blockchain data and the consumption of computing recourses. These metrics are; transactions per second, transactions per CPU, transactions per memory speed, transactions per disk input/output, and transactions per network data. The study metrics enabled the discovery of the system's throughput and latency (Zheng *et al.,* 2018).

(a) Transaction Per Second

This metric measures the throughput of the transaction during its execution. The metric shows the number of transactions (Txs) executed in a given period from time t_p to t_q. Transaction per second (TPS) of the peer (p) is calculated by:

$$TPS_P = \frac{\text{Count}(\text{Txs from (tp,tq)})}{tq - tp} (\text{Txs}/s) \quad (1) \quad \text{(Zheng } et\ al.\text{, 2018)}$$

From this formula, the average TPS for (P) peers in a network is:

$$\overline{TPS} = \frac{\sum_p TPS_p}{P} = (\text{Txs}/s) \quad (2) \quad \text{(Zheng } et\ al.\text{, 2018)}$$

(b) Transactions Per Central Processing Unit

This is the measurement of Central Processing Unit (CPU) resource consumption during smart contract execution. The level of CPU consumption depends on the smart contract's business logic. Smart contracts with encryption and looping series consume much CPU resources. Utilization of CPU is highly noticed during transaction consensus life cycle while validation and committing of blocks. From t_p to t_q, Transactions per CPU (TPC) of the peer (p) is computed with the following formula:

$$TPC_p = \frac{\text{Count (Txs from } (t_p, t_q))}{\int_{t_p}^{t_q} F * CPU(t)} (\text{txs}/(\text{GHz.s})), \quad (3) \quad \text{(Zheng } et\ al.\text{, 2018)}$$

F stands for a CPU core, and CPU(t) denotes the use of blockchain CPU from t_p to t_q. The average usage of CPUs is computed by:

$$\overline{TPC} = \frac{\sum_p TPC_p}{P} = (\text{txs}/(\text{GHz.s})), \quad (4) \quad \text{(Zheng } et\ al.\text{, 2018)}$$

(c) Transaction Per Memory Second

Transaction per memory second (TPMS) is the measurement of memory consumption during smart contract execution. Memory utilization during the execution of transactions (Txs) from t_p to t_q is computed using this formula:

$$TPMS_p = \frac{\text{Count (Txs from } (t_p, t_q))}{\int_{t_p}^{t_q} RMEM(t) + VMEM(t)} (\text{txs}/(\text{MB.s})), \quad (5) \quad \text{(Zheng } et\ al.\text{, 2018)}$$

RMEM(t) is the physical memory utilized by the blockchain system from t_p to t_q, and VMEM is the virtual memory. The average memory utilization can be calculated by:

$$\overline{\mathrm{TPMS}} = \frac{\sum_p \mathrm{TPMS_p}}{\mathrm{P}} \ (\mathrm{txs}/(\mathrm{MB.s})), \qquad (6)$$ (Zheng *et al*., 2018)

(d) Transactions Per Disk Input/Output

The measurement of Transactions per disk input/output (TPDIO) represents the utilization of blockchain I/O resources. The processes such as block committing and contract execution consume I/O resources during ledger state maintenance.

$$\mathrm{TPDIO_p} = \frac{\mathrm{Count\ (Txs\ from\ (t_p,t_q))}}{\int_{t_p}^{t_q} \mathrm{DISKR(t)+DISKW(t)}} (\mathrm{txs/kilobytes}), \qquad (7)$$ (Zheng *et al*., 2018)

DISKR(t) shows the amount of data read from the storage disk, and DISKW(t) shows the amount of data written to the storage disk from t_p to t_q. The consumption of disk resources by all peers (P) in the network can be computed as follows:

$$\overline{\mathrm{TPDIO}} = \frac{\sum_p \mathrm{TPDIO_p}}{\mathrm{P}} \ (\mathrm{txs/kilobytes}), \qquad (8)$$ (Zheng *et al*., 2018)

(e) Transaction Per Network Data

The Transaction per network data (TPND) is the measurement of blockchain system network flow utilization from time t_p to t_q. The metric measures the flow of transactions of the network data. The TPND of the peer (p) is computed as follows:

$$\mathrm{TPND_p} = \frac{\mathrm{Count\ (Txs\ from\ (t_p,t_q))}}{\int_{t_p}^{t_q} \mathrm{UPLOAD(t)+DOWNLOAD(t)}} (\mathrm{txs/kilobytes}), \qquad (9)$$ (Zheng *et al*., 2018)

UPLOAD(t) denotes upstream network size and DOWNLOAD(t) denotes downstream size from time t_p to t_q. Average computation of the whole network flow can be obtained by:

$$\overline{\mathrm{TPDN}} = \frac{\sum_p \mathrm{TPDN_p}}{\mathrm{P}} \ (\mathrm{txs/kilobytes}), \qquad (10)$$ (Zheng *et al*., 2018)

(f) Assessment of Overall Performance Metric Results

This section assesses metric results of overall performance based on comparison experimentation between Fabric and Ethereum blockchain platforms. Peers with 8 GB RAM, Intel Core i7-4790 3.60 GHz were used with 100 smart contracts (Zheng *et al.*, 2018).

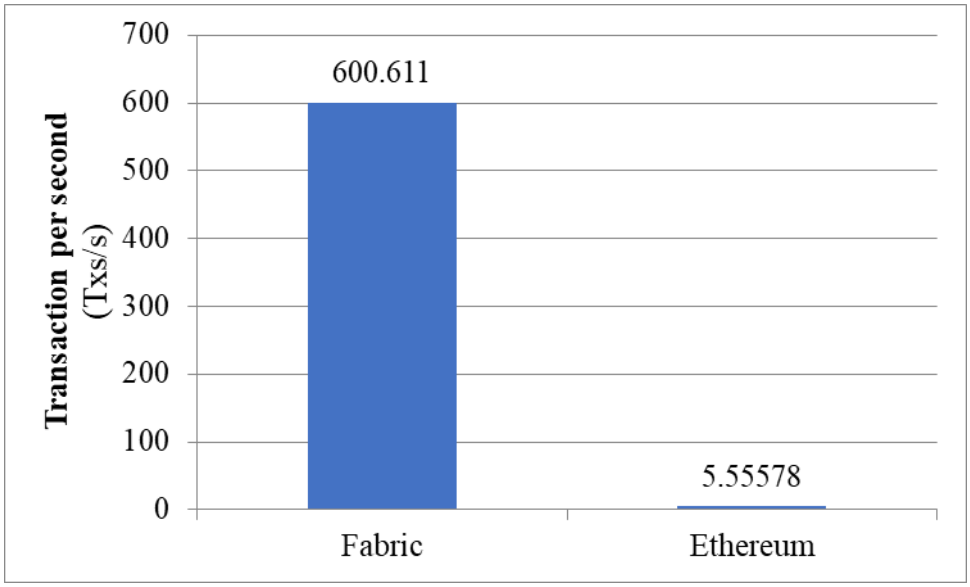


Figure 24: Computed throughput

Figure 24 illustrates the assessment transactions of a Fabric and Ethereum in a second. The results show that the average throughput of Fabric is higher than Ethereum. This is an indication that Fabric has a higher transaction rate per second compared to Ethereum.

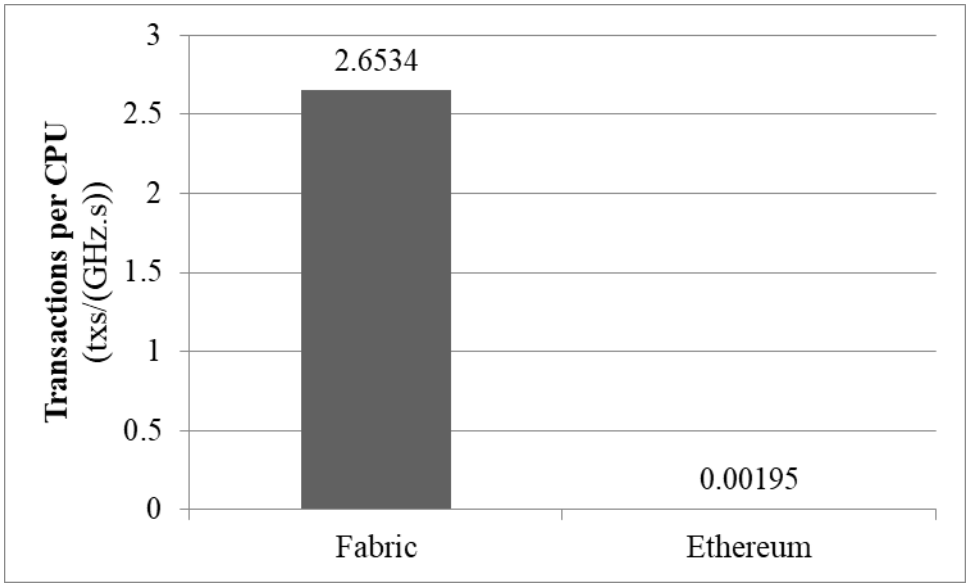


Figure 25: Computed average transactions per CPU

Figure 25 shows Fabric and Ethereum utilization of one gigahertz CPU core for each node in a second. The results show that Ethereum's utilization of processors is very low compared to Fabric.

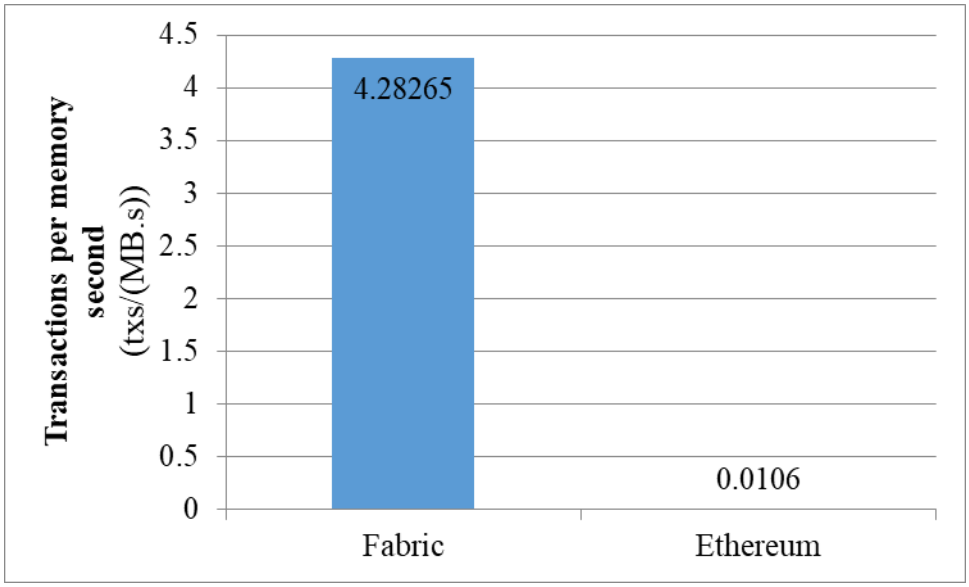


Figure 26: Computed average transactions per memory second

Figure 26 illustrates transactions of Fabric and Ethereum consumption of computer memory per unit of time. The result shows that the Fabric system consumes more than four transactions per 1 megabyte per second of a peer's memory.

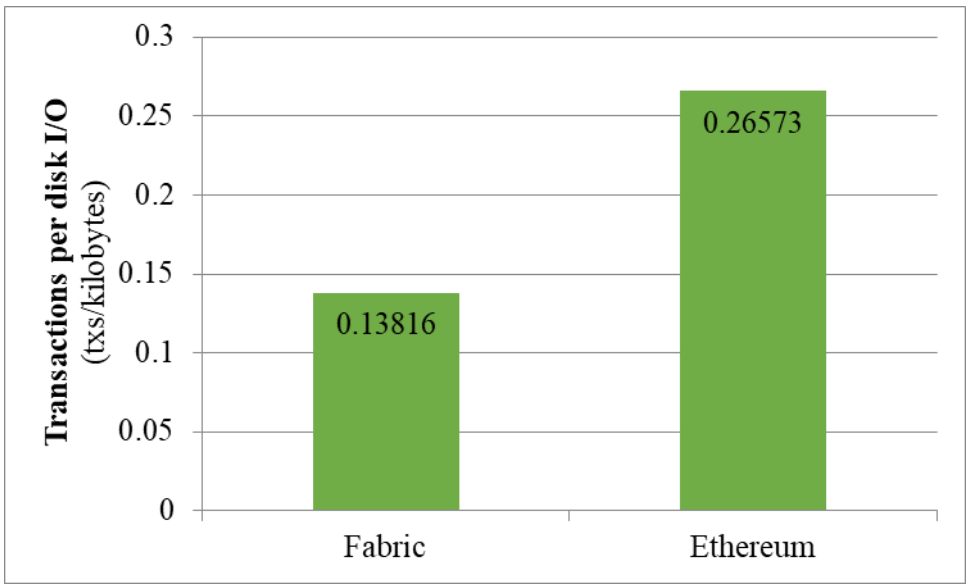


Figure 27: Computed average transactions per disk read/write

Figure 27 illustrates transactions of Fabric and Ethereum applications for read and write of 1 megabyte of a peer per unit time to/from a peer's disk storage. The result shows that Ethereum has higher writes and reads transactions for one megabyte per second compared to Fabric.

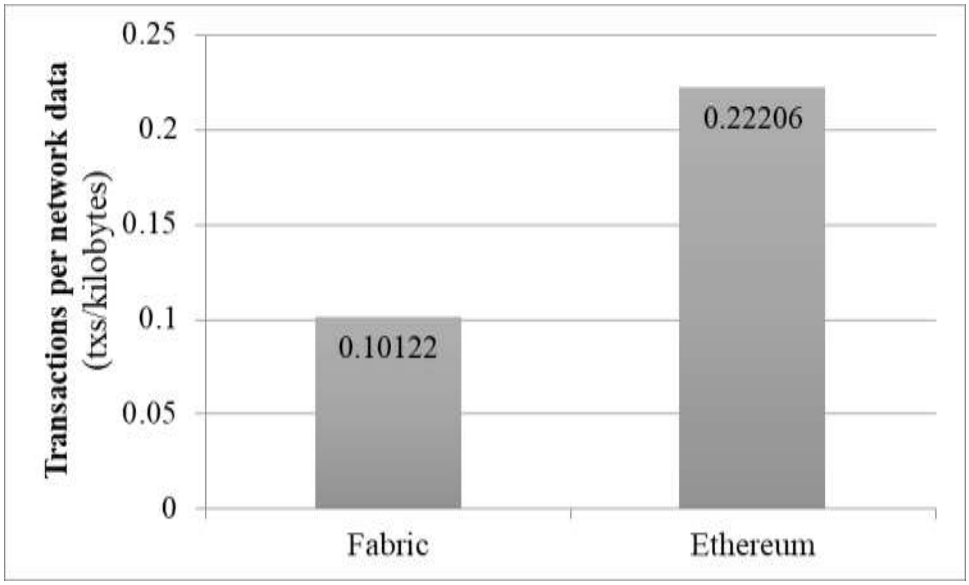


Figure 28: Computed average of transactions network data

Figure 28 displays computed transactions of a data flow in a second. The result shows that Fabric's transactions per network data are lower than Ethereum's. Ethereum consumes half the bandwidth of the Fabric system.

Results shown in Fig. 18 to Fig. 22 were used to measure the performance of the developed system. The overall performance results show that Fabric has a larger transaction per second (throughput) than Ethereum. This is because Fabric is purposely designed to be a permissioned blockchain and its consensus protocol is much faster. Ethereum's utilization of computing resources (transactions per CPU, transactions per memory second) is very low due. This is due to its consensus protocol which consumes much computing resources while computing hashes. Fabric's transactions per network data are lower than Ethereum due to high network consumption of Practical Byzantine Fault Tolerance (PBFT).

The fabric has smart contracts which stores arrays making it have high throughput but a low TPDIO. This is caused by frequent read and write in the world state storage disk. Ethereum's smart contracts have many iterations of operations leading to the lowered TPMS and TPC. This has resulted from the high consumption of computing resources through loop operations.

Therefore, the Hyperledger Fabric blockchain system shows higher overall performance than Ethereum which shows lower overall performance.

4.2 Discussion

The study developed a blockchain-based system that was virtually integrated with the existing health system of Mount Meru Referral Hospital. This case study aimed to strengthen GoT-HoMIS system security by improving its data storage security using blockchain technology.

The goal was successfully attained through the achievement of the following specific objectives:

(i) To identify data security weaknesses of the current health information system deployed at Mount Meru Referral Hospital;

(ii) To design and develop a blockchain-based health information system that will be integrated into the existing system for data storage security; and

(iii) To validate the developed blockchain-based health information system.

The first objective results show that the major weakness of the GoT-HoMIS system is within its centralized client-server architecture. The system is web-based and is accessed via a centralized database. The system is susceptible to cyber-attacks such as malicious IPs, malicious software, and web attacks. This creates the existence of system vulnerabilities. A centralized database is central storage which makes it a single point of failure. Therefore, the architectural nature of GoT-HoMIS exposes itself to security threats. This leads to system security challenges on integrity, confidentiality, and availability. Sensitive data can be accessed and easily manipulated.

System requirements were identified and guided system development for addressing security challenges inherited from GoT-HoMIS. The developed system met system requirements on identity management through Hyperledger Fabric CA, data integrity preservation through digital signature, data privacy, data verification, data validation, and non-repudiation which were maintained through the Fabric transaction consensus life cycle. System availability and data backup were implemented through decentralized peer-to-peer network architecture. Other system requirements included modularity, scalability, portability, reliability, and

recoverability. They all reflect the nature of Fabric architecture to improve system performance.

The second objective presented the designs and development of the proposed blockchain system. The system was developed based on its design of a private data collection channel for data storage. System development focused on CIA triad security to meet security goals. The developed system used the permissioned Hyperledger Fabric v2.3.2 platform with the implementation of a private data policy. Private data communication between nodes was facilitated through gossip protocol. Data storage security and privacy were maintained through private state and channel state. Member nodes in a channel were decentralized with blockchain only to avoid storage complications. This solved other related security challenges which are centralized in nature. Smart contracts were developed in JavaScript and the platform was configured in a virtualized environment.

The third objective was to validate a developed blockchain-based system, in which V-Model testing was used for system verification and validation. The developed system was tested based on the requirement specifications through experimentation, simulation, and scenarios to validate the proposed system. The performance of the developed system was evaluated against the Ethereum blockchain system identified in a literature review. System validation was purposively carried out to prove the credibility of the developed system based on the research gap. Performance metrics focused on overall performance and detailed performance of the system. Overall performance evaluated the system's throughput and latency. The detailed performance provided detailed information on the whole process of system performance to discover performance bottlenecks.

The overall performance results show that Fabric has a larger transaction per second (throughput) compared to Ethereum. This is because Fabric's transactions are signature-based. Fabric is purposely designed to be a permissioned blockchain and its consensus protocol is much faster. Ethereum's utilization of computing resources (transactions per CPU, transactions per memory second) is very low. This is due to its consensus protocol which consumes much computing resources while computing hashes. Fabric's transactions per network data are lower than Ethereum due to high network consumption of Practical Byzantine Fault Tolerance (PBFT).

The fabric has smart contracts which stores arrays making it have high throughput but a low TPDIO. This is caused by frequent read and write in the world state storage disk. Ethereum's smart contracts have many iterations of operations leading to the lowered TPMS and TPC. This has resulted from the high consumption of computing resources through loop operations. Therefore, the Hyperledger Fabric blockchain system shows higher overall performance than Ethereum which shows lower overall performance.

4.2.1 Findings from Data Collection

(i) What are the data security weaknesses of the current health information system deployed at Mount Meru Referral Hospital?

The centralized architecture of GoT-HoMIS was the major weakness of the system making it a single point of failure. Centralized architecture exposes the system to cybersecurity threats creating susceptibility to cyber-attacks. The study found main three categories of attacks namely; malicious IPs, malicious software, and web attacks. Malicious IP attacks are caused by default and commonly used user credentials (usernames and passwords) while web-based attacks are due to client-server communication. Malicious software attacks include Trojan horses, ransomware, and backdoors causing downtimes in the system. Database backup was another weakness that was revealed by this study. System administrators were obligated to do data backup on daily basis after working hours. The backup has to be sent to the PO-RALG ICT department after every week for secured storage in case of any emergency. It was an obsolete and tedious job that was addressed through peer-to-peer decentralized network architecture.

The answer to this research question was obtained through document analysis and interviews for discovering system security challenges facing GoT-HoMIS. Security challenges identified were based on centralized system architecture. The developed system was required to meet system requirements for identity management, data integrity, data privacy, data verification, data validation, non-repudiation, and system availability to address the challenges of the current system. Other requirements included modularity, scalability, portability, reliability, and system recoverability. Peer to peer Fabric architecture with transaction consensus lifecycle was designed and developed to address security challenges. Security goals were met through identity management, endorsement policy, transaction verification, and validation.

(ii) Which methodologies will be used to design and develop a blockchain-based health information system to be integrated into the existing system for data storage security?

The study used the DSR methodology during the system designing and development process. It is a problem-solving technique with digital innovative solutions. This helped to address the identified system security challenges through system designing, development, verification, and validation. A qualitative research approach for data collection using interviews and document analysis was used. Both primary and secondary data collection methods were used.

Data analysis used the requirement engineering process to come up with the right list of system requirements. The identified list of system requirements led to system development guidance that addresses system challenges. The study used a prototyping system development methodology. This methodology reflected the conceptual framework and research design of the study. The methodology allowed prototype refinement in which the actual system was developed based on the final working prototype.

The proposed system was designed and developed using permissioned Hyperledger Fabric v2.3.2. System artifacts were designed and implemented during the development process. Several application packages were installed as prerequisites to set up the development environment. These prerequisites include installation of tools such as; Curl version 7.68.0, Docker version 20.10.2, Docker-compose 1.29.1, node.js V10.19.0, npm 6.14.4 and python 2.7.18. JavaScript was used for the development of the smart contract. The study used Visual Studio Code version 1.55.2 for writing and editing

(iii) Did the developed blockchain-based health information system meet system requirements?

System validation was carried out based on system requirements to ensure developed system functions and operates to address security challenges. The validation procedure followed a defined order of transaction consensus lifecycle from endorsing peer to committing peer. The process involved system execution based on system requirements, and it was carefully monitored so that it consistently conforms to the expected outputs of system security. The developed system was validated based on the requirement specifications metrics.

The performance Hyperledger fabric system was evaluated against the Ethereum blockchain system which was identified in a literature review. Performance metrics focused on overall performance and detailed performance of the system. Overall performance evaluated the system's throughput and latency. The detailed performance provided information on the whole process of system performance to discover performance bottlenecks.

The overall performance shows that Fabric has a larger transaction per second (throughput) than Ethereum. This is because Fabric is purposely designed to be a permissioned blockchain and its consensus protocol is much faster. Ethereum's utilization of computing resources (transactions per CPU, transactions per memory second) is very low due. This is due to its consensus protocol which consumes much computing resources while computing hashes. Fabric's transactions per network data are lower than Ethereum due to high network consumption of Practical Byzantine Fault Tolerance (PBFT).

The fabric has smart contracts which stores arrays making it have high throughput but a low TPDIO. This is caused by frequent read and write in the world state storage disk. Ethereum's smart contracts have many iterations of operations leading to the lowered TPMS and TPC. This has resulted from the high consumption of computing resources through loop operations. Therefore, the Hyperledger Fabric blockchain system shows higher overall performance than Ethereum which shows lower overall performance.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The purpose of this study was to develop health information system data storage security using blockchain technology. The study developed a decentralized Fabric system that was virtually integrated with GoT-HoMIS to address data storage security challenges. The data storage architecture of the developed system deployed a private data collection channel, which is the combination of the actual private data stored in a private state, and a hash of the private data to guarantee data privacy. A private data policy was implemented in which data storage is within the Fabric framework.

System development methodology led to the goal attainment of data privacy and confidentiality where hashes of the transactions were stored on the network nodes. Smart contracts led to executed transactions that were concealed to unauthorized entities. This was implemented through the decentralization of network nodes with blockchain only for hash storage of the transactions. The separation of (blockchain) transactions and the database helped to avoid the storage challenge of big data sets of each node in the network. This was the niche occupied through this study, where cloud storage was deployed to address the challenge of big data. The Cloud storage approach leads to incurred costs of renting to several cloud computing service providers to avoid a single point of failure.

Fabric system security storage architecture is based on the execute-order-validate model. Its storage architecture has higher overall performance compared to execute-validate Ethereum systems identified in the literature review. Ethereum blockchain involves complex mathematical computations, leading to a low performance with transaction inconsistencies. Cost-effectiveness of the developed system is achieved through data storage within a database of a Hyperledger Fabric.

The newly developed blockchain-based system is designed as a security tool for data storage, highly tolerant to faults and resistant to data manipulation. The system is resistant to cyber-security attacks and solved other related security challenges including a single point of failure. Security goals including authentication, non-repudiation, accountability, and reliability were attained. Suspicious acts with fraud intentions will be easily identified and blocked. This

approach addressed other related security challenges which were centralized in nature. The study will contribute knowledge on cybersecurity issues to academicians and other researchers.

5.2 Recommendations

The government of The United Republic of Tanzania through The Ministry of Communication and Information Technology should adopt blockchain technology for the security implementation of HIS. System developers especially those dealing with the development of e-government systems should be trained and engaged in the implementation of blockchain technology. This technology can be integrated into all health systems such as asset management, procurement, and financial management systems.

Blockchain technology should be added to the education curriculum of universities and other academic institutions. This will increase knowledge and awareness among graduates who will be experts in managing system security. During the study, it was observed that system administrator was not aware of blockchain technology, and capacity building through frequent training should be given to system administrators and other system users to increase the security awareness of new emerging technologies.

The study recommends the following areas for further study:

(i) Application of blockchain technology for asset management to keep track of hospital assets and avoid unnecessary property loss;

(ii) The use of technology in hospital financial management systems, and

(iii) The use of technology in hospital procurement issues. The technology will be one of the government's initiatives to protect public properties and finance through accountability and fraud prevention.

REFERENCES

Akkaoui, R., Hei, X., & Cheng, W. (2020). EdgeMediChain: A Hybrid Edge Blockchain-Based Framework for Health Data Exchange. *IEEE Access, 8,* 113467-113486. https://doi.org/10.1109/ACCESS.2020.3003575

Al Barghuthi, N. B., Hamdan, I., Al Suwaidi, S., Lootah, A., Al Amoudi, B., Al Shamsi, O., & Al Aryani, S. (2019). An Analytical View on Political Voting System using Blockchain Technology-UAE Case Study. *2019 Sixth HCT Information Technology Trends.* 132-137. https://doi.org/10.1109/ITT48889.2019.9075074

Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., & Yellick, J. (2018). Hyperledger fabric: A distributed operating system for permissioned blockchains. *Proceedings of the Thirteenth EuroSys Conference,* 1-15. https://doi.org/10.1145/3190508.3190538

Biswas, S., Sharif, K., Li, F., Alam, I., & Mohanty, S. (2020). DAAC: Digital Asset Access Control in a Unified Blockchain-Based E-Health System. *IEEE Transactions on Big Data*. https://doi.org/10.1109/TBDATA.2020.3037914

Brotsis, S., Kolokotronis, N., Limniotis, K., Bendiab, G., & Shiaeles, S. (2020). On the security and privacy of hyperledger fabric: Challenges and open issues. *2020 IEEE World Congress on Services,* 197-204. https://doi.org/10.1109/SERVICES48979.2020.00049

Chuma, K. G., & Ngoepe, M. (2021). Security of electronic personal health information in a public hospital in South Africa. *Information Security Journal: A Global Perspective*, 1-17.

Demirkan, S., Demirkan, I., & McKee, A. (2020). Blockchain technology in the future of business cyber security and accounting. *Journal of Management Analytics, 7*(2), 189-208.

Desai, H., Kantarcioglu, M., & Kagal, L. (2019). A hybrid blockchain architecture for privacy-enabled and accountable auctions. *2019 IEEE International Conference on Blockchain,* 34-43.

Fan, C., Ghaemi, S., Khazaei, H., & Musilek, P. (2020). Performance evaluation of blockchain systems: A systematic survey. *IEEE Access, 8,* 126927-126950. https://doi.org/10.1109/ACCESS.2020.3006078

Fekih, R. B., & Lahami, M. (2020). Application of blockchain technology in healthcare: A comprehensive study. *International Conference on Smart Homes and Health Telematics, 12157,* 238-276. https://doi.org/10.1007/978-3-030-51517-1_23

Gordon, W. J., & Catalini, C. (2018). Blockchain technology for healthcare: Facilitating the transition to patient-driven interoperability. *Computational and Structural Biotechnology Journal, 16,* 224-230.

Hajdu, Á., Ivaki, N., Kocsis, I., Klenik, A., Gönczy, L., Laranjeiro, N., & Pataricza, A. (2020). Using Fault Injection to Assess Blockchain Systems in Presence of Faulty Smart Contracts. *IEEE Access, 8,* 190760-190783.

Haule, A., Dida, M. A., & Sam, A. E. (2019). Towards Data Exchange between Health Information System and Insurance Claims Management System. *International Journal of Information Engineering & Electronic Business, 11*(2), 28-34.

Honar Pajooh, H., Rashid, M., Alam, F., & Demidenko, S. (2021). Hyperledger Fabric Blockchain for Securing the Edge Internet of Things. *Sensors, 21*(2), 359.

Ismail, L., & Materwala, H. (2019). A review of blockchain architecture and consensus protocols: Use cases, challenges, and solutions. *Symmetry, 11*(10), 1198.

Javaid, H., Hu, C., & Brebner, G. (2019). Optimizing validation phase of hyperledger fabric. *2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems,* 269-275.

Kajirunga, A., & Kalegele, K. (2015). Analysis of activities and operations in the current E-Health landscape in Tanzania: Focus on interoperability and collaboration. *International Journal of Computer Science and Information Security, 13*(6), 49-54.

Kamau, G., Boore, C., Maina, E., & Njenga, S. (2018). Blockchain technology: Is this the solution to emr interoperability and security issues in developing countries? *2018 IST-Africa Week Conference*.

Khamis, K., & Njau, B. (2014). Patients' level of satisfaction on the quality of health care at Mwananyamala hospital in Dar es Salaam, Tanzania. *BMC Health Services Research, 14*(1), 1-8.

Khan, S., Jadhav, A., Bharadwaj, I., Rooj, M., & Shiravale, S. (2020). Blockchain and the Identity-based Encryption Scheme for High Data Security. *2020 Fourth International Conference on Computing Methodologies and Communication*, 1005-1008. https://doi.org/10.1109/ICCMC48092.2020.ICCMC-000187

Khubrani, M. M. (2021). A Framework for Blockchain-based Smart Health System. *Turkish Journal of Computer and Mathematics Education, 12*(9), 2609–2614.

Kombe, C. (2020). A secure and interoperable blockchain-based information sharing system for healthcare providers in developing countries. *NM-AIST*. https://dspace.nm-aist.ac.tz/handle/20.500.12479/894

Kombe, C., Dida, M., & Sam, A. (2018). A review on healthcare information systems and consensus protocols in blockchain technology. *International Journal of Advanced Technology and Engineering Exploration, 5*(49), 473-483.

Kombe, C., Sam, A., Ally, M., & Finne, A. (2019). Blockchain technology in sub-Saharan Africa: Where does it fit in healthcare systems: A case of Tanzania. *Journal of Health Informatics in Developing Countries, 13*(2), 1-19.

Kummer, S., Herold, D. M., Dobrovnik, M., Mikl, J., & Schäfer, N. (2020). A systematic review of blockchain literature in logistics and supply chain management: Identifying research questions and future directions. *Future Internet, 12*(3), 60.

Liao, C. F., Bao, S. W., Cheng, C. J., & Chen, K. (2017). On design issues and architectural styles for blockchain-driven IoT services. *2017 IEEE International Conference on Consummer Electronics-Taiwan,* 351-352. https://doi.org/10.1109/ICCE-China.2017.7991140

Liao, C. F., Cheng, C. J., Chen, K., Lai, C. H., Chiu, T., & W, L. C. (2017). Toward a service platform for developing smart contracts on blockchain in BDD and TDD styles. *2017 IEEE 10th Conference on Service-Oriented Computing and Applications*, 133-140. https://doi.org/10.1109/SOCA.2017.26

Madine, M., Salah, K., Jayaraman, R., Yaqoob, I., Al-Hammadi, Y., Ellahham, S., & Calyam, P. (2020). Fully Decentralized Multi-Party Consent Management for Secure Sharing of Patient Health Records. *IEEE Access, 8*, 225777-225791.

Mahore, V., Aggarwal, P., Andola, N., & Venkatesan, S. (2019). Secure and Privacy-Focused Electronic Health Record Management System using Permissioned Blockchain. *2019 IEEE Conference on Information and Communication Technology*, 1-6. https://doi.org/10.1109/CICT48419.2019.9066204

Manevich, Y., Barger, A., & Tock, Y. (2018). Service discovery for hyperledger fabric. *Proceedings of the 12th ACM International Conference on Distributed and Event-Based Systems*, 226-229. https://doi.org/10.1147/JRD.2019.2900647

Monev, V. (2020). Defining and Applying Information Security Goals for Blockchain Technology. *2020 International Conference on Information Technologies*, 1-4. https://doi.org/10.1109/InfoTech49733.2020.9211073

Monrat, A. A., Schelén, O., & Andersson, K. (2019). A survey of blockchain from the perspectives of applications, challenges, and opportunities. *IEEE Access, 7*, 117134-117151. https://doi.org/10.1109/ACCESS.2019.2936094

Morkunas, V. J., Paschen, J., & Boon, E. (2019). How blockchain technologies impact your business model. *Business Horizons, 62*(3), 295-306.

Mtebe, J. S., & Nakaka, R. (2018). Assessing Electronic Medical Record System Implementation at Kilimanjaro Christian Medical Center, Tanzania. *Journal of Health Informatics in Developing Countries, 12*(2), 1-16.

Mtey, M., & Dida, M. (2019). Towards interoperable e-Health system in Tanzania: Analysis and evaluation of the current security trends and big data sharing dynamics. *International Journal of Advanced Technology and Engineering Exploration, 6*(59), 225-240.

Mustafa, M. K., & Waheed, S. (2021). An E-Voting Framework with Enterprise Blockchain *Advances in Distributed Computing and Machine Learning*, *127*, 135-145.

Nagasubramanian, G., Sakthivel, R. K., Patan, R., Gandomi, A. H., Sankayya, M., & Balusamy, B. (2020). Securing e-health records using keyless signature infrastructure blockchain technology in the cloud. *Neural Computing and Applications, 32*(3), 639-647.

Ndayizigamiye, P., & Dube, S. (2019). Potential adoption of blockchain technology to enhance transparency and accountability in the public healthcare system in South Africa. *International Multidisciplinary Information Technology and Engineering Conference*, 1-5.

Nehemiah, L. (2014). Towards EHR interoperability in Tanzania hospitals: Issues, challenges, and opportunities. *International Journal of Computer Science, Engineering and Applications, 4*(4), 29-36.

Nguyen, T. S. L., Jourjon, G., Potop-Butucaru, M., & Thai, K. L. (2019). Impact of network delays on Hyperledger Fabric. *IEEE Conference on Computer Communications Workshops*, 222-227. https://doi.org/10.1109/INFCOMW.2019.8845168

Niranjanamurthy, M., Nithya, B., & Jagannatha, S. (2019). Analysis of Blockchain technology: Pros, cons, and SWOT. *Cluster Computing, 22*(6), 14743-14757.

Office, N. A. (2021). Ripoti ya mdhibiti na mkaguzi mkuu wa hesabu za serikali kwa mwaka wa fedha 2019/2020, 59-61. Dodoma, Tanzania: *The National Audit Office.*

Paik, H.-Y., Xu, X., Bandara, H. D., Lee, S. U., & Lo, S. K. (2019). Analysis of Data Management in Blockchain-Based Systems: From Architecture to Governance. *IEEE Access, 7*, 186091-186107. https://doi.org/10.1109/ACCESS.2019.2961404

Rennock, M. J., Cohn, A., & Butcher, J. R. (2018). Blockchain technology and regulatory investigations. *Practical Law Litigation, 1*(2018), 35-44.

Ribeiro, V., Holanda, R., Ramos, A., & Rodrigues, J. J. (2020). Enhancing key management in LoRaWAN with permissioned blockchain. *Sensors, 20*(11), 3068.

Saad, M., Spaulding, J., Njilla, L., Kamhoua, C., Shetty, S., Nyang, D., & Mohaisen, A. (2019). Exploring the attack surface of blockchain: A systematic overview. 1-30.

Saadatmand, M., & Daim, T. (2019). Blockchain technology through the lens of disruptive innovation theory. *2019 IEEE Technology & Engineering Management Conference*, 1-6. https://doi.org/10.1109/TEMSCON.2019.8813566

Sadiq, M., & Jain, S. (2012). An insight into requirements engineering processes. *International Conference on Advances in Communication, Network, and Computing, 108*, 313-318.

Sajjad, U., & Hanif, M. Q. (2010). Issues and challenges of requirement elicitation in large web projects. 1-60

Sankar, L. S., Sindhu, M., & Sethumadhavan, M. (2017). Survey of consensus protocols on blockchain applications. *2017 4th International Conference on Advanced Computing and Communication Systems*, 1-5. https://doi.org/10.1109/ICACCS.2017.8014672

Sato, T., & Himura, Y. (2018). Smart-contract based system operations for permissioned blockchain. *2018 9th IFIP International Conference on New Technologies, Mobility, and Security*, 1-6. https://doi.org/10.1109/NTMS.2018.8328745

Shahnaz, A., Qamar, U., & Khalid, A. (2019). Using blockchain for electronic health records. *IEEE Access, 7*, 147782-147795.

Sousa, J., Bessani, A., & Vukolic, M. (2018). Abyzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform. *48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 51-58. https://doi.org/10.1109/DSN.2018.00018

Sun, Y., Zhang, L., Feng, G., Yang, B., Cao, B., & Imran, M. A. (2019). Blockchain-enabled wireless Internet of Things: Performance analysis and optimal communication node deployment. *IEEE Internet of Things Journal, 6*(3), 5791-5802.

Taylor, P. J., Dargahi, T., Dehghantanha, A., Parizi, R. M., & Choo, K.-K. R. (2020). A systematic literature review of blockchain cyber security. *Digital Communications and Networks, 6*(2), 147-156.

Team, T. C. E. R. (2019-2021). Tanzania Computer Emergency Response Team-Honeypots weekly report. Dar-es-Salaam: Tanzania Communication Regulatory Authority.

Toapanta, S. M., Quimis, O. A. E., Gallegos, L. E. M., & Arellano, M. R. M. (2020). Analysis for the Evaluation and Security Management of a Database in a Public Organization to Mitigate Cyber Attacks. *IEEE Access, 8,* 169367-169384. https://doi.org/10.1109/ACCESS.2020.3022746

Tschuchnig, M. E., Radovanovic, D., Hirsch, E., Oberluggauer, A. M., & Schäfer, G. (2019). Immutable and Democratic Data in Permissionless Peer-to-Peer Systems. *Sixth International Conference on Software Defined Systems*, 294-299. https://doi.org/10.1109/SDS.2019.8768645

Tsoulias, K., Palaiokrassas, G., Fragkos, G., Litke, A., & Varvarigou, T. A. (2020). A Graph Model-Based Blockchain Implementation for Increasing Performance and Security in Decentralized Ledger Systems. *IEEE Access, 8,* 130952-130965. https://doi.org/10.1109/ACCESS.2020.3006383

Wang, C., & Chu, X. (2020). Performance characterization and bottleneck analysis of hyperledger fabric. *40th International Conference on Distributed Computing Systems*, 1281-1286. https://doi.org/10.1109/ICDCS47774.2020.00165

Wang, H., & Zhang, J. (2019). Blockchain-Based Data Integrity Verification for Large-Scale IoT Data. *IEEE Access, 7,* 164996-165006.

Wang, R., He, J., Liu, C., Li, Q., Tsai, W. T., & Deng, E. (2018). A privacy-aware PKI system based on permissioned blockchains. 2018 *IEEE 9th International Conference on Software Engineering and Service Science,* 928-931. https://doi.org/10.1109/ICSESS.2018.8663738

Warkentin, M., & Orgeron, C. (2020). Using the security triad to assess blockchain technology in public sector applications. *International Journal of Information Management, 52*, 102090. https://doi.org/10.1016/j.ijinfomgt.2020.102090

Zhang, X., Li, R., & Cui, B. (2018). A security architecture of VANET based on blockchain and mobile edge computing. *The 1st IEEE International Conference on Hot Information-Centric Networking*, 258-259.

Zheng, P., Zheng, Z., Luo, X., Chen, X., & Liu, X. (2018). A detailed and real-time performance monitoring framework for blockchain systems. *2018 IEEE/ACM 40th*

International Conference on Software Engineering: Software Engineering in Practice Track, 134-143.

Zheng, X., Mukkamala, R. R., Vatrapu, R., & Ordieres-Mere, J. (2018). Blockchain-based personal health data sharing system using cloud storage. *2018 IEEE 20th International Conference on e-Health Networking, Applications, and Services*, 1-6. https://doi.org/10.1109/HealthCom.2018.8531125

APPENDICES

Appendix 1: Interview questions

THE NELSON MANDELA AFRICAN INSTITUTION OF SCIENCE AND TECHNOLOGY (NM-AIST)

Requirement analysis tool for system development. **This is a guide; which will lead during interviewing system administrator(s) at Mount Meru Referral Hospital**

Research title: Application of Blockchain Technology in Strengthening Health Information System Security; A Case Study of Mount Meru Referral Hospital

1. What are the functional requirements of the GoT-HOMIS system?
    a. …….
    b. ……..……
    c. ……..……
    d. ……..……
    e. ……..
2. What are the non-functional requirements of the GoT-HOMIS system?
    a. …….
    b. ……..……...
    c. ……..……...
    d. ……..……

3. How was the system designed and developed to meet functional and non-functional requirements?

   ……
   ……
   …….

4. How do you implement system security to stored data?

   ……
   ……
   ……

5. How do users of the system access stored data?

   ……
   ……
   ……

6. How do you ensure that there are no changes that can be made to stored data? (Initiatives to ensure data integrity)

   ……
   ……
   ……

7. How do you know that changes have been made to the stored data? (How do you know that there is a loss of data integrity?)

   ……
   ……
   ……

8. Do you conduct an audit and maintain audit trails of the data? Yes/No…….....

9. If yes, explain how tracing can be done to show changes made to data leading to loss of data integrity

   ……
   ……
   ……

10. How do you recover changes made to stored data when there is a loss of integrity?

……

..……..……

……..……...……..……

11. Does your system allow data sharing with other stakeholders? Yes/No…….

12. If yes;

    a. How do you share data?

……

……

……

    b. How data sharing security is implemented?

……

……

……

13. What is the mechanism to make sure that any department or any other stakeholder will not access data stored that they are not necessary for them to get access to? (Initiatives to ensure privacy and confidentiality?)

……

……

……

…….

14. Outline security challenges/weaknesses of the system.

    a. ……

    b. ……

    c. ……

    d. ……

    e. ……

Appendix 2: Introduction to Mount Meru Referral Hospital

THE NELSON MANDELA
AFRICAN INSTITUTION OF SCIENCE AND TECHNOLOGY
(NM-AIST)

School of Computational and Communication Science and Engineering

Direct Line: +255 272970001
Fax: +255 272970016
E-mail: dean-cocse@nm-aist.ac.tz

Tengeru
P.O. Box 447
Arusha, TANZANIA
Website: www.nm- aist.ac.tz

Our Ref: NM-AIST/M.009/T.19/13 Date: 02nd March, 2021

Medical Officer Incharge,
Mount Meru Referral Hospital,
P.O.BOX 3092,
Arusha,

Dear Sir/Madam,

RE: INTRODUCING MR. RICHARD W. MNYAWI

Kindly refer to the above heading.

I wish to introduce Mr. Richard W. Mnyawi with Registration No. NM-AIST/M.009/T.19, a Master Student at Nelson Mandela African Institution of Science and Technology in the School of Computational and Communication Science and Engineering (CoCSE).

As part of the requirement for Master degree, Mr. Mnyawi is undertaking a research entitled **"Application of Block Chain Technology in Strengthening Health Information System Security: A Case Study of Mount Meru Referral Hospital"**

In order to accomplish his research objectives, he would like to collect some information from your institution. The information to be collected will be used for research purposes. The research will help the student to develop a block chain based system solution which will be integrated to the existing system to improve data storage security as it states in the research objectives.

It is my sincere hope that you will assist the student in accomplishing his study.

Looking forward to your cooperation.

Sincerely,

Shubi Kaijage *PhD*
Ag. Dean - CoCSE

Academia for Society and Industry

Appendix 3: Introduction letter to General Secretary-TAMISEMI

THE UNITED REPUBLIC OF TANZANIA

MINISTRY OF EDUCATION, SCIENCE AND TECHNOLOGY

THE NELSON MANDELA AFRICAN INSTITUTION OF SCIENCE AND TECHNOLOGY (NM-AIST)

OFFICE OF THE VICE CHANCELLOR

In reply please quote:

Date: 10th December 2021

Ref. No: NM-AIST/M.009/T.19
Principal Secretary,
TAMISEMI
P.O. Box 1923,
DODOMA

Ufs: Director Secretary,
ICT, PO-RALG
TAMISEMI
DODOMA.

RE: INTRODUCTION LETTER FOR RESEARCH DATA COLLECTION

Kindly refer to the above heading.

I wish to introduce Mr. Richard William Mnyawi with Registration No. NM-AIST/M009/T19 a Master's student in Wireless and Mobile Computing (WiMC) program enrolled at the Nelson Mandela African Institution of Science and Technology (NM-AIST) in the School of Computational and Communication Science and Engineering (CoCSE).

As part of requirement for Master's degree, Mr. Richard is undertaking research titled ***"Application of Blockchain Technology in Strengthening Security of Health Information System;" A Case Study of Mount Meru Referral Hospital.***

In order to accomplish his research objectives, he would like to request the GoT-HoMIS data base schema for successful completion of his study.

The information to be collected will only be used for research purposes and will enable the student to develop a blockchain based system solution which will integrated to the existing system to improve data storage security.

It is my sincere hope that you will assist the student accordingly

Looking forward to your cooperation

Sincerely.

Prof. Emmanuel Luoga
Vice Chancellor.