Coventry University

DOCTOR OF PHILOSOPHY

BCFL Logging

an approach to acquire and preserve admissible digital forensics evidence in the cloud ecosystem

Awuson - David, Kenny

Award date:
2022

Awarding institution:
Coventry University

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of this thesis for personal non-commercial research or study
- This thesis cannot be reproduced or quoted extensively from without first obtaining permission from the copyright holder(s)
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 08. Jun. 2022

Coventry University
Institute for Future Transport and Cities

BCFL Logging: An Approach to Acquire and Preserve Admissible Digital Forensics Evidence in the Cloud Ecosystem

Kenny Awuson-David

A thesis submitted in partial fulfilment of the University's requirements for the degree of Doctor of Philosophy

December 5, 2021

Content removed on data protection grounds

Declaration

I understand that all materials, quotes or documents used in this research project which are from any third-party source have been acknowledged with a referencing to the author of the material. I also understand that any failure from my side to acknowledge the author's work could lead to plagiarism action been considered against me due to a failure on my part.

Name: Kenny Awuson-David

Signature:Awuson-David

Date: 22-11-20

Acknowledgement

Firstly, I would like to thank God for all his guidance, protection and strength. I also thank my wife, who was instrumental in supporting me with the structure of my thesis.

I want to thank my daughter and son for their support, advice and for inspiring me to work hard and never give up. I would also like to thank my parents for instilling in me the love of education from a very young age. My gratitude goes out to Dr Nazaraf Shah and Dr Norlaily Yaacob for their support and advice.

Dr Tawfik Al-Hadhrami - I appreciate your advice and words of encouragement, and I thank you.I could not have done my research without your selfless support.

Journals Paper

- Title: Blockchain Logging: An Approach to Maintain Forensics Evidence Integrity in Cloud Ecosystem (Accepted ELSEVIER: Special Issue on Artificial Intelligence for Cyber Defence and Smart Policing (AICDSP) Accepted.
- Journal Name: Future Generation Computer System

Conference paper:

- Awuson-David, K., Al-Hadhrami, T., Funminiyi, O. and Lotfi, A., 2019, September. Using Hyperledger Fabric Blockchain to Maintain the Integrity of Digital Evidence in a Containerised Cloud Ecosystem. In International Conference of Reliable Information and Communication Technology (pp. 839-848). Springer, Cham.

Conference Presentation:

- IRICT 2019 The 4th International Conference of Reliable Information and Communication Technology 2019.

Abstract

The on-demand nature of cloud computing technology has altered the way data and information are shared and handled online. However, organisations that continue to leverage the benefits of cloud on-demand services face severe incremental security challenges. In addition, there are well-known security, privacy and trust issues among the cloud computing stakeholders that need to be solved. These drawbacks are particularly problematic, and cloud stakeholders have struggled to solve these challenges or establish trustworthiness in the cloud environment. A novel, permissioned Blockchain Cloud Forensic Logging (BCFL) framework approach is needed, to be applied in the cloud to establish trust, traceability and admissible log evidence. Blockchain is a peer-to-peer network that uses a decentralised Distributed Ledger Technology (DLT) with a smart contract that maintains a tamper-resistant transaction ledger. It provides a promising solution for a cloud forensics acquisition. This research has designed and implemented a Blockchain Cloud Forensic Logging (BCFL) framework using the Design Science Research Methodological (DSRM) approach. BCFL operates primarily in four stages: (1) Process transaction logs using Blockchain distributed ledger technology (DLT). (2) Use a Blockchain smart contract to maintain the integrity of logs and establish a transparent chain of custody. (3) Validate all transaction logs. (4) Maintain transaction log immutability. The results from the single case study demonstrate that BCFL will mitigate the challenges and complexities faced by digital forensics investigators in acquiring admissible digital evidence from the cloud ecosystem. In addition, an instantaneous performance monitoring of the Blockchain cloud forensic logging framework was evaluated. BCFL will ensure trustworthiness, integrity, authenticity and non-repudiation of the log evidence in the cloud.

Keywords: Admissibility, Blockchain, Cloud Forensics, Design Science Methodology, Digital log evidence, GDPR, Hyperledger Fabric, Trustworthiness

Contents

1 Introduction **1**

1.1 Aim 5

1.1.1 Research Questions 5

1.1.2 Research Objectives: 6

1.2 Background 7

1.2.1 PROBLEM STATEMENT 8

1.2.2 The Identified Gap 9

1.2.3 The Impact of this Research 10

1.3 Cloud Computing 10

1.3.1 Cloud Computing Definition 11

1.4 Cloud Models 16

1.5 RESEARCH METHODOLOGY 18

1.6 Original Contributions 24

1.7 THESIS ORGANISATION 24

2 Literature Review **27**

2.1 INTRODUCTION 27

2.1.1 Cloud Computing Services 31

2.1.2 Cloud Computing Threat s and Challenges 35

2.1.3 Forensics Logging and Storage 39
2.1.4 Challenges in Cloud Forensics 43
2.1.5 Cloud Multi-Tenancy 47
2.2 Cloud Service Providers (CSPs) 51
2.2.1 Securing Cloud ecosystem 51
2.2.2 Computer Forensics 57
2.2.3 Blockchain 59
2.2.4 Hyperledger Fabric 73
2.3 Docker and Hyperledger Fabric Stacks 77
2.3.1 GDPR Compliance 78
2.3.2 Hyperledger Blockchain Types: 79
2.3.3 Hyperledger Framework Tools: 81
2.4 Summary 83

3 Methodology **84**
3.1 Methodology 84
3.2 Introduction 84
3.2.1 Design Science Research Methodology 85
3.2.2 Discrete Event System Specification (DEVS) 87
3.2.3 DESIGN SCIENCE RESEARCH METHODOLOGY 88
3.2.4 Research Main Goals 94
3.3 Summary 96

4 Framework and Implementation **97**
4.1 Introduction 97
4.2 The Impact of the Proposed Architecture 99
4.3 BCFL Blockchain Cloud Network 101

4.3.1 Distribution Ledger Technology 102
4.4 BCFL Architecture 104
4.5 BCFL Framework 107
4.5.1 Hyperledger Fabric Chaincode Process Architecture 113
4.6 Cloud Log Evidence Processes 115
4.7 Current Cloud Logging Mechanisms 117
4.7.1 Security Information and Event Management 117
4.7.2 OpenStack Cloud Logging 118
4.7.3 Blockchain 119
4.8 Supply Chain Case Study 122
4.9 Setup BCFL Virtualised Hybrid Lab 123
4.9.1 Algorithm 124
4.10 BCFL Code 126
4.10.1 BCFL Single-case Mechanism Experiments Scenario 131
4.10.2 Scenario 131
4.10.3 Performance Metrics 134
4.11 Summary 136

5 Simulation, Testing and Evaluation 139

5.1 Introduction 139
5.1.1 Second Single-Case-Study Experiment 141
5.1.2 Business single-case scenario 141
5.1.3 Overview of Insurance Scenario 142
5.1.4 Commodities Scenario 143
5.1.5 The Experimental Testing of our Scenarios 143
5.2 Blockchain Cloud Forensics Logging (BCFL) Experimental Lab 145

5.3 Traffic Flow Analysis 154
5.3.1 Metrics for Performance Analysis 156
5.3.2 Performance Metrics 156
5.4 Summary 158

6 Conclusion **160**
6.1 Conclusion 160
6.2 Limitation 163
6.3 Future Work 164

Appendices **195**

A BCFL Code **196**

B BCFL Simulation Results **223**
B.1 Initial Results 223

List of Figures

1.1 Cloud Architecture 11
1.2 Adapted from NIST Cloud Computing 13
1.3 Adapted from NIST Cloud On-Demand Service 14
1.4 Community Cloud Model 18
1.5 Research Design Process 20

2.1 Main areas of Review 31
2.2 Cloud Services Comparison 33
2.3 Challenges and Cyber Threats in Cloud Ecosystem 37
2.4 NFAT Design Goals 43
2.5 Multi-Tenancy Architecture in Cloud Ecosystem 49
2.6 Multi-Tenancy Architecture in Cloud Ecosystem 50
2.7 Blockchain Merkle tree 62
2.8 The Block Structure of Blockchain Technology 65
2.9 Blockchain Trustworthiness 66
2.10 Docker and the Proposed Hyperledger Fabric Integration 78

3.1 BCFL used Methodology (DSRM) 90
3.2 DSRM Simulated BCFL Design and Development Phases 94
3.3 Discrete Event System Specification (DEVS) 95

4.1 BCFL Architecture 106
4.2 BCFL Framework 109
4.3 API and SDK interaction with BCFL 110
4.4 Hyperledger Composer Components 112
4.5 BCFL Hybrid Cloud Chaincode Process Architecture 114
4.6 Blockchain Cloud Forensics Log Investigation Process 116
4.7 BCFL Supply-chain Process Flow Diagram 122
4.8 BCFL Blockchain Simulated Testing Diagram 125
4.9 BCFL Supply-Chain Case-study Log Evidence 133
4.10 BCFL Angular Supply chain Product Interface 134
4.11 Transaction Rate Throughput 135
4.12 Transaction Rate tps 135
4.13 Transaction Rate tps vs Latency 136

5.1 On the above image, we issued an update command to update the Ubuntu operating system with the latest tools 146
5.2 BCFL System Components Updates 147
5.3 A command (ps aux | grep) was issue to view BCFL root user account and super admin account (Awuson-David 2019). 147
5.4 The command (docker Image ls) displays all Docker Images and their container size 148
5.5 Hyperledger Fabric REST-Server API displayed participant account profile 149
5.6 BCFL Admin Participant Activates Participant Account Deleted from the REST-Server API Server 149
5.7 Display of the Log Evidence From our BCFL Blockchain Forensic Logging 150

5.8 BCFL Blockchain Cloud network was shut down by the suspect to potentially clear all logs from the Cloud network 151
5.9 BCFL restarted after a reboot. No error encountered during restart . . 152
5.10 Screenshot displaying all acquired transaction logs 153
5.11 Screenshot image showing the traffic flow of the supply chain 155
5.12 BCFL Angular API App 155
5.13 BCFL Memory usages of the performance metric 157
5.14 BCFL Memory usage and performance metric 157
5.15 BCFL CPU utilisation usage metric 157
5.16 Latency monitoring of BCFL Blockchain Cloud ecosystem 157

B.1 BCFL Supply chain Angular App Interface 223
B.2 Identification of Log evidence after System Reboot 224
B.3 User Profile Deleted 224
B.4 Log Forensic Acquisition 225
B.5 User Identification by Chaincode 226
B.6 BCFL REST-Server Supplychain Interface 227
B.7 REST Server and user activity timestamp 228
B.8 REST Server and user activity timestamp 229
B.9 REST Server User Activity 230
B.10 REST Server User Activity 231
B.11 BCFL Results 233
B.12 BCFL Results 233
B.13 BCFL Results 234

List of Tables

2.1 Worldwide Public Cloud Service Revenue Forecast (Billions of U.S Dollars, Gartner, 2020) 35

2.2 Permissionless and Permissioned Blockchain Technology Comparisons . 64

2.3 Blockchain Block Size Structure 68

2.4 Highlighting a Comparison of some of the Consensus Algorithms 69

2.5 Comparison of Using Blockchain to Secure Forensic Evidence in the Cloud Ecosystem 74

2.6 Blockchain core Components and functions 76

2.7 Hyperledger Framework Consensus Mechanisms and Types as highlighted by 80

2.8 Highlighting a Comparison of some of the Consensus Algorithms, Blockchain Types and their Use Case 82

3.1 Mitigating Digital Evidence Challenges in the Cloud Methodology Comparison 92

4.1 BCFL Framework Scenario: CHAIN OF CUSTODY 134

List of Abbreviations

ACPO ..... Association of Chief Police Officers

AI ..... Artificial Intelligence

API ..... Application Program Interface

AWS ..... Amazon Web Service

BEM ..... Boundary Element Method

BlockIPFS ... Blockchain Interplanetary File System

BPR ..... Business Process Reengineering

CA ..... Certificate Authority

CFaaS ..... Cloud-Forensic-as-a-Service

CIA ..... Confidentiality, Integrity and Availability

CLI ..... Command Line Interface

CoT ..... Cloud-of-Things

CRM ..... Customer Relationship Management

CSA ..... Cloud Security Alliance

CSP ..... Cloud Service Provider

DARPA ..... Defence Advance Research Project Agency

DEM ..... Discrete Element Method

DEVS ..... Discrete Event System Specification

DFRWS ..... Digital Forensics Research Workshop

DLT ..... Distributed Ledger Technology

DDoS ..... Distributed Denial-of-Service

Dos ..... Denial-of-service

DSRM ..... Design Science research methodology

ELK ..... Elasticsearch Logstash and Kibana

ENISA ..... European Union Agency for Cybersecurity

EOSC ..... European Open Science Cloud

EU ..... European Union

FAT ..... File Allocation Table

FBA ..... Federated Byzantine Fault Tolerance

GCP ..... Google Cloud Platform

GDPR ..... General Data Protection Regulation

HDD ..... Hard Disk Drive

IaaS ..... Infrastructure as a Service

IDS ..... Intrusion Detection System

IMP ..... Instance Message Protocol

ISO ..... International Organisation for Standardisation

ISP ..... Internet Service Provider

MIDA ..... Modified Information Dispersal Algorithm

MSPs ..... Membership Service Providers

NFAT ..... Network Forensic Analysis Tools

NIST ..... National Institute of Standards and Technology

NTP ..... Network Time Protocol

OOMS ..... Object-Oriented Modelling and Simulation

OS ..... Operating System

PaaS ..... Platform as a Service

PBFT ..... Practical Byzantine Fault Tolerance

PCI DSS ..... Payment Council Industry Data Security Standard

PFEM ..... Particle Finite Element Method

PoET ..... Proof of Elapsed Time

PoW ..... Proof of Authority

PoW ..... Proof of Work

PoS ..... Proof of Stake

P2P ..... Peer-to-Peer

RAM ..... Random Access Memory

SaaS ..... Software as a Service

SLA ..... Service-Level Agreement

SIEM ..... Security Information and Event Management

TCCP ..... Trusted Cloud Computing Platform

TCP ..... Transmission Control Protocol

TCO ..... Total Cost of Ownership

UDP ..... User Datagram Protocol

VDI ..... Virtual Disk Image

VHD ..... Virtual Hard Disk

VM ..... Virtual Machine

VMFS ..... Virtual Machine File System

VPKI ..... Vehicular Public Key Infrastructure

Chapter 1

Introduction

The affordability and straightforward approach to quickly deploy a network in a way that has never been realised before have attracted cybercriminals as highlighted by X. Han, Kheir, and Balzarotti (2015), Hooper, Martini, and Choo (2013), and Awuson-David, Al-Hadhrami, Alazab, Shah, and Shalaginov (2021). The challenges faced by cloud forensic investigators in acquiring digital evidence from a cloud platform such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) or Software as a Service (SaaS) have been acknowledged by Awuson-David et al. (2021); Pichan, Lazarescu, and Soh (2015), and Manral, Somani, Choo, Conti, and Gaur (2019). The multi-tenancy, geo-location, political and legal issues have added to the complexity of admissible digital evidence from the cloud environment, (Celesti et al., 2019), (Pichan et al., 2015) and Awuson-David et al. (2021).

However, with Hyperledger Fabric Blockchain, an investigator can easily verify the validity of the logs as Blockchain transactions are encrypted and hashed with a timestamp and there is traceability of all transactions. Forensic examiners need to acquire evidence from a cloud ecosystem that is tamperproof, free from contamination and admissible before a court of law, (Ab Rahman, Glisson, Yang, & Choo, 2016;

Awuson-David et al., 2021).

The landscape of cloud computing not only raises doubt over where data are situated, but it also gives rise to confidentiality and regulatory compliance problems (Awuson-David et al., 2021). For instance, the European Union (EU) General Data Protection Regulation (GDPR) that was introduced on 25th May 2018, requires that businesses should comply in protecting personal data. However, this compliance issue has added another layer of complexity in investigating and acquiring digital evidence from the cloud ecosystem, (Awuson-David et al., 2021; Krystlik, 2017). Blockchain technology provides auditable transaction logs, making legal disputes less likely and simpler to settle. In an increasingly cloud-oriented society, the ability to identify, obtain, preserve, and analyse potential digital evidence is increasingly essential as demonstrated by Awuson-David et al. (2021); Q. Xia, Sifah, Smahi, Amofa, and Zhang (2017).

Cybercriminals' persistent threat to use different types of techniques and malware to compromise cloud network has doubled the capacity of cloud security administrators. Integrating BCFL into the cloud platform will facilitate trustworthiness and solve the complexity and challenges digital forensic investigators to encounter during forensic acquisition in the cloud network. (Awuson-David et al., 2021). A single case study will be adopted to validate the framework through a virtualised cloud experiment. This will enable intervention at every stage of our framework design, deployment and testing. It is not only pragmatic to see what happens but also to support the Blockchain nodes with its independent logging through the use of its smart contract capability (Wieringa, 2014).

A holistic approach is utilised in the experimentation, integrating a Hyperledger Fabric as a backbone in the cloud ecosystem to establish an in-depth analysis of log evidence. Fabric's advantage in a single case study is that it is highly modular, designed for coordination across companies. The private channel feature enables secure func-

tionality that suits our experimental scenarios. The Hyperledger Fabric has a composer component API (Playground and REST Server) that facilitates configuration built as part of the system Blockchain applications as highlighted by Gaur et al. (2018), and Lone and Mir (2019). Despite cloud economic benefits and success to businesses and organisations, log evidence acquisition in the cloud is associated with well known limitations such as cloud service-level agreement (cloud-SLA), multi-tenancy, geo-location and trustworthiness as demonstrated by Ruan, Carthy, Kechadi, and Baggili (2013)

The first measure in the deployment of the experiment for this research was creating a cloud system in a virtual environment using Docker containerisation. Docker containerisation technology is more efficient than virtual machine (VM) due to being lightweight in bandwidth usage (Awuson-David et al., 2021; Pahl, 2015). Docker containers have revolutionised the software supply chain in both large and small enterprises. As highlighted by Awuson-David et al. (2021); Stelly and Roussev (2017), never before has a new technology so swiftly infiltrated the top 500 enterprises worldwide.

This virtual environment will enable BCFL to explore and verify how artefacts are preserved and secured in a Blockchain cloud ecosystem, both in transit, and storage (Awuson-David et al., 2021; Boulos, Wilson, & Clauson, 2018). The result findings will evaluate the need for using Blockchain to maintain and preserve log evidence integrity in a cloud environment. The Blockchain distributed ledger technology core mechanisms such as the smart contracts, encryption, hashing and immutability of data that Blockchain exhibit will facilitate log integrity. As demonstrated by Awuson-David et al. (2021); Mingxiao, Xiaofeng, Zhe, Xiangwei, and Qijun (2017), the goal of agreeing on a consensus and synchronising the state transversely across all peers does not require that all peers execute a chaincode. Propagating the same rule to all peers is sufficient.

The drive of this research is to accomplish the following:

- To investigate how Blockchain technology can be integrated into the cloud ecosys-

tem and also to examine how Blockchain uses its security mechanisms to maintain trustworthiness and transparency in the cloud platforms by establishing a secure communication pathway among stakeholders (Awuson-David et al., 2021).

- To evaluate related work in the area of Blockchain cloud integration and cloud computing forensic investigation. To investigate the Blockchain immutability mechanism and how it is used to preserve log evidence integrity even when the VM is deleted or rebooted (Awuson-David et al., 2021).

The purpose of the experiments undertaken for this research was to provide information on how cloud log evidence can be acquired, preserved and stored securely without being compromised or tampered with. Furthermore, DSRM was adopted for this research. A supply-chain single case study scenario was used to test and validate the experimental research results. Elasticsearch, Logstash and Kibana (ELK) was also used to capture real-time application and system performance metrics as described by Prakash, Kakkar, and Patel (2016); Son and Kwon (2017). In addition, Docker containerised nodes were evaluated to ascertain how they would handle transaction logs in a Hyperledger Fabric Blockchain environment (Awuson-David et al., 2021).

Contributions to this research are summarised as follows:

- Using permissioned Blockchain to maintain tamper-proof log evidence in the cloud ecosystem.

- Integrate permissioned Blockchain in the cloud platforms that enables evidence acquisition that enhance GDPR compliance and maintain a secured log evidence chain of custody (Awuson-David et al., 2021).

- Adding a digital forensic layer in the Blockchain Hyperledger Fabric framework that supports a secure digital forensic acquisition in the cloud.

- A DSRM was used to implement a Blockchain cloud integration that created trustworthiness among cloud stakeholder on a new pathway for a secure log evidence acquisition in the cloud (Awuson-David et al., 2021).

All the participants in the BCFL use the Blockchain DLT by taking part in a peer-to-peer network of nodes that distribute transaction among all nodes, enabling a new digital forensic acquisition pathway for the cloud. This will solve the current challenges of cloud digital evidence admissibility and shorten litigations among cloud actors (Awuson-David, Al-Hadhrami, Funminiyi, & Lotfi, 2019).

1.1 Aim

The aim of this research is to use the DLT, data immutability and hashing capability of Blockchain technology to integrate into the cloud IaaS platform. This will be deployed in a controlled innovative containerised environment that will facilitate a digital evidence scenario that evaluates the integrity of log artefacts acquired in the cloud ecosystem.

1.1.1 Research Questions

The proposed research contends that the current forensic methods are insufficient for cloud investigation and require significant changes to be made so that cloud digital forensic evidence is admissible.

The relevant questions are:

1. How can system logs (network, process and application) be extended to include additional information that will help solve recognised problems in cloud forensics, including the issues of dependency chains and time synchronisation?

2. How practical would a Blockchain approach be for capturing and storing extended cloud logs?

3. Can a BCFL framework be developed to solve several issues associated with cloud forensics investigation and evidence acquisition?

4. What relevant services are needed to use the extended logs effectively, and how can these be built?

5. What are the social and business issues that may affect the proposed Blockchain protocol, and how can they be overcome?

1.1.2 Research Objectives:

- To investigate and identify the main challenges described by the cloud security alliance (CSA) in performing digital forensics processes, where the artefact resides in a cloud ecosystem. While these challenges continue to persist in digital forensics processes, the essential characteristics of cloud computing systems enumerated in the National Institute of Standards and Technology (NIST) will be investigated (Awuson-David et al., 2021; Pichan, Lazarescu, & Soh, 2018).

- To investigate how Blockchain technology can be integrated into the cloud ecosystem and maintain trustworthiness and transparency in the cloud. Furthermore, establishing a way of communicating challenges between cloud customer, service provider, first responders, forensic investigators and system administrators. Due to the complexity in evidence acquisition in the cloud, there is a need for a trusted communication channel among cloud stakeholders (Awuson-David et al., 2021).

- To evaluate related work and the use of Blockchain consensus and smart contract mechanisms to acquire logs in the cloud ecosystem that preserve evidence

integrity, trustworthiness, transparency, and privacy. This research will demonstrate in the experimental case study scenarios, when the VM is shut down and rebooted; acquired logs evidence is still preserved and intact due to Blockchain data's immutability (Awuson-David et al., 2021).

1.2 Background

According to Hand, Ton, and Keller (2013), cloud forensics comprises a forensic method towards acquiring digital evidence that enables the examiner to ascertain the source of the crime. It comprises multiple fragmented examinations of virtual machines, traditional network and system interactions. "Logistically it includes communications among cloud stakeholders (cloud service provider, cloud customers, digital forensics investigator, and cloud security admin) for facilitating both internal and external examination of evidence (Ali, Khan, & Vasilakos, 2015; Pichan et al., 2015). Legitimately it repeatedly indicates multi-jurisdictional and multi-tenant situations". Cloud forensics is viewed as being different from computer forensics because the cloud involves several administrative domains, extensive data replication, multi-tenancy and often operates across various jurisdictions (Awuson-David et al., 2019; Hogan, Liu, Sokol, & Tong, 2011; Zissis & Lekkas, 2012).

Traditional approaches to digital forensic acquisition and evidence preservation are not practical for the cloud ecosystem. In the traditional network, the forensic investigators usually have access to network devices whereas cloud architecture is based on virtualisation. Due to the challenges mentioned above, it becomes complex for forensic investigators to have access to the devices used by an adversary as decentralised data processing and multiple third-party ownership hinders investigation. Another issue is time synchronisation. For an investigator to reconstruct an accurate timeline of

events, the correct time and time zone needs to be ascertained (Palmer, 2001). In a cloud ecosystem, establishing this information could be challenging. Public clouds will usually store evidence in a disseminated approach across various physical locations, even in more than one time zone (Palmer, 2001). Besides, virtualised services may also operate according to the time zones of their users, rather than their physically hosted locations (Wang, Chow, Wang, Ren, & Lou, 2011).

A new cloud digital forensic acquisition framework needs to be developed. This research will take a log-centric viewpoint and propose a new virtualised experimental testbed with a resilient Blockchain forensic logging mechanism that can assure trust, preserve accountability, and transparency in the cloud ecosystem (Awuson-David et al., 2021; Dai & Vasarhelyi, 2017; Lu, 2018). The proposed system will ensure the confidentiality, integrity and availability of cloud system logs in transit and storage (Awuson-David et al., 2019). The idea is built on research carried out in secure logging whereby logs are hashed at regular intervals after which changes cannot be made as highlighted by Zawoad, Dutta, and Hasan (2013). It also addresses the recognised time synchronisation challenge in the cloud.

1.2.1 PROBLEM STATEMENT

Without a resilient, secure and trusted cloud storage infrastructure with forensic logging capability, institutions such as banks, health care providers and government agencies will continue to report a data breach in their cloud storage infrastructure. Cloud service providers have persisted to research and look for a solution that can mitigate the ever-increasing threats by implementing a system that can maintain data confidentiality, integrity and availability (CIA) (Awuson-David et al., 2021). However, the question this research addresses is how to implement a forensic logging framework , a Hyperledger Fabric Blockchain in a cloud ecosystem to address the complexity of

acquiring digital evidence in a cloud ecosystem.

1.2.2 The Identified Gap

Since 2015 there has been a high increase for cloud computing services, and the surge has added another complexity in data storage , processes and transactions in the cloud ecosystem. This concept change from the stand-alone physical computer and network devices, to the cloud environment, has resulted in digital forensic investigation becoming more complicated in forensic investigation. Against the background of such significant overwhelming new complexity, forensic investigators have to perform cloud forensics investigation and acquire admissible evidence. However, in some respects, cloud forensics is in its early stages. The question remains how to establish trust between cloud user and CSP. Some CSP are solely responsible for log monitoring in the cloud ecosystem, in which case customers have limited visibility of logs which their system devices generate. With this in mind, it has become difficult for the cloud customer to continually request system security event logs from the CSP whenever there is a cyber incident. However, this approach is not forensically sound as the logs provided by the CSP could be contaminated. Integration of BCFL framework into the cloud ecosystem will bridge the research gap. However, the advent of advanced containerisation technology such as Docker allows the integration of a virtual machine which will enable us to carry out an experiment using the DSRM and evaluate our results with a case study scenario. Docker technology has helped to redefine the virtualisation in a way that has never been seen before. In this research, our BCFL framework will sufficiently reduce cloud forensics complexity by maintaining data integrity and trustworthiness both in transit and storage in the cloud.

1.2.3 The Impact of this Research

The landscape of cloud computing not only presents doubt over where data are situated, but it also gives rise to confidentiality and regulatory compliance problems(Awuson-David et al., 2021). As a result, the traditional digital forensic acquisition framework may not acquire admissible evidence in the cloud (Martini & Choo, 2012; Ruan, Carthy, Kechadi, & Crosbie, 2011). According to Johng, Kim, Hill, and Chung (2018), a holistic approach is utilised in the experimentation and integration of Hyperledger Fabric Blockchain cloud, which establishes an in-depth analysis of log preservation. Furthermore, the BCFL framework will enable cloud forensic investigators to acquire admissible log evidence with traceability with the mechanism to preserve the evidence chain of custody. The framework will enhance cloud service level agreement, and the GDPR through Blockchain DLT and smart contract mechanism. The framework uses Blockchain cryptographic and hashing mechanisms to maintain evidence integrity during the forensic acquisition processes and preserve acquired evidence with Blockchain data immutability mechanisms. Although many researchers have explored how to preserve log evidence in the cloud, their studies have not established a systematically investigative approach to demonstrate how to maintain logs' integrity in the cloud ecosystem. To this note, BCFL framework is intended to answer all our research questions.

1.3 Cloud Computing

Cloud computing technology has enabled an easy communication environment where data sharing and storage are achieved with just one stroke of a computer's keyboard. Cloud computing is a relative term to define computer resources presented as a service to clients accessible over a network infrastructure, such as a private corporation

network over the (Internet) public network. Cloud computing has facilitated market opportunities for small, medium and large businesses, transforming information technology as network resources are shared on-demand. Cloud forensic logging has its complexity and challenges concerning the admissibility of log files as evidence. Little research has been carried out in this area of forensic logging. Evidence has shown that there is a need for a more scientific approach to ensure log evidence admissibility. Figure 1.1 demonstrate a cloud enterprise architecture where the CSP manages the resource controls and distributes it to the organisation on-demand.

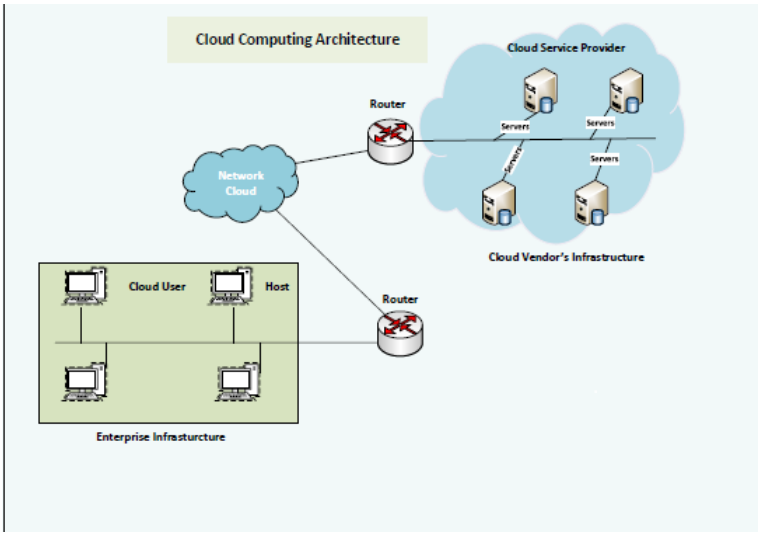


Figure 1.1: *Cloud Architecture*

1.3.1 Cloud Computing Definition

NIST (2012) provides a one-sentence definition of cloud computing as "a model for enabling businesses to share resources, on-demand network with access to a shared pool of configurable computing resources (Awuson-David et al., 2021, 2019). For instance, networking components such as servers, databases, applications, and services can be swiftly deployed and connected with little management input or service provider inter-

action (Awuson-David et al., 2021)." The NIST definition introduces the supporting concepts of three cloud service models, five essential characteristics, and four types of cloud deployments. The three service models are affectionately known as the SPI model, SaaS, PaaS, and IaaS, followed by software, platform, and infrastructure services (Awuson-David et al., 2019; Yeluri & Castro-Leon, 2014). Figure 1.2 illustrates NIST cloud computing models with a highlighting each model layer.

The Cloud Security Alliance Alliance (2009), defined cloud service models as a growing term, a method of computing in different platforms which enables efficiency through on-demand service. Cloud separates application and information resources from the fundamental infrastructure, and the mechanisms used to deliver them. Cloud enhances collaboration, agility, scaling, and availability, and offers the potential for cost reduction through enhanced and resourceful computing (Ruan et al., 2013). It demonstrates how to enable a more efficient way of networking and resource sharing that has transformed the digital landscape (Carpenter & Krutka, 2014). In addition, Chellappa (1997) identifies the cloud as the driving force behind system networks. Buyya, Ranjan, and Calheiros (2010) defined cloud computing as "A similar and distributed computing system involving a collection of interconnected or more unified computing resources based on cloud SLA, established through cooperation between the service provider and consumers".

Regardless of its difficulty and diverse nature, NIST has identified five vital features that characterise a cloud computing platform:

On-demand self-service: Consider a traditional IT environment where the financial cost and deployment complexity in the area of troubleshooting the network is a daily

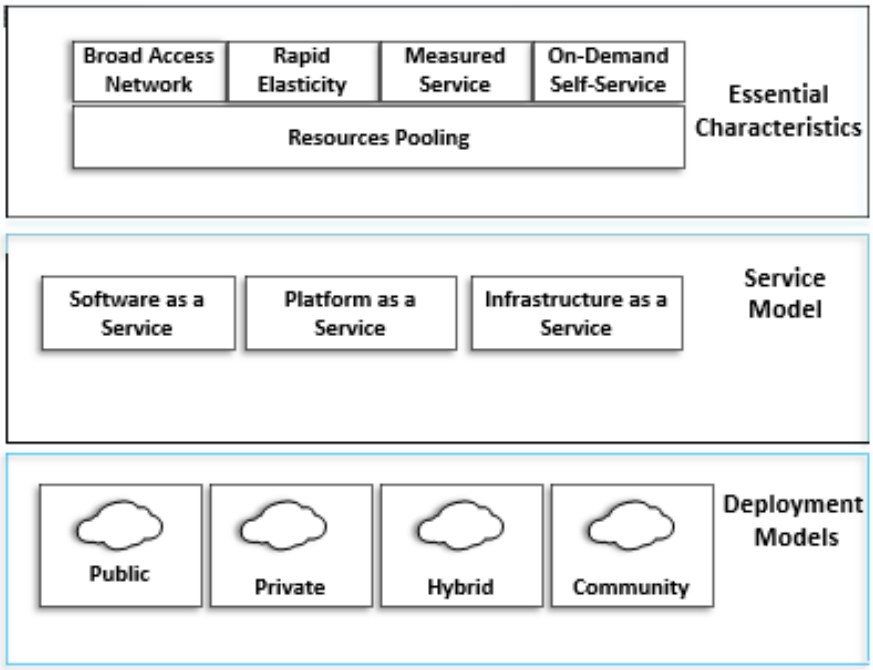


Figure 1.2: *Adapted from NIST Cloud Computing* Liu et al. (2011)

challenge (Awuson-David et al., 2019). The cloud computing ecosystem is different, as all resources are shared on-demand in an automated fashion. The cloud customer can receive or handle extra computing resources without calling the cloud customer service provider for system maintenance, application updates, and scheduling. In practice, for many CSPs, the sign-up, payment, and deployment of applications are entirely on-demand and driven from their web site without the need for any human interaction (Awuson-David et al., 2019; Samani, Reavis, & Honan, 2014). Figure 1.3 demonstrate the effortless adaptability of the cloud on-demand service by an organisation. The cloud service models have a significant benefit for organisations with its on-demand services such as broad network access, resource pooling etc. However, these platforms have added another layer of complexity in acquiring admissible evidence in the cloud ecosystem.

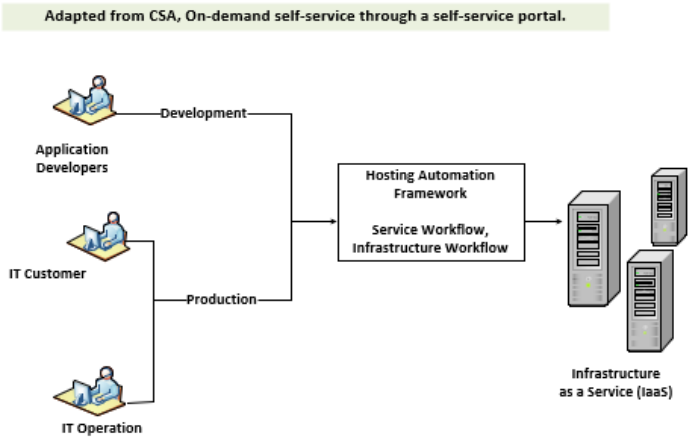


Figure 1.3: *Adapted from NIST Cloud On-Demand Service*

- Broad network access: connectivity on the cloud ecosystem is between the cloud's customer and the cloud service provider (CSP) where resource services are deployed and obtainable by the cloud customer via multiple devices. However, the resources should be accessible using standard mechanisms. According to CSA 2016, this characteristic focuses on the availability of cloud resources, without the need for any specific device or commercial software. Characteristically, the resources are available through a web browser using HTTP, HTTPS, XML, or Java.

- Resource pooling: One of the most significant benefits of cloud computing to individuals and businesses is cost-effectiveness and the ability to offer quality services at an affordable price. This is primarily driven by the characteristic defined by NIST as resource pooling, where by CSPs can utilise economies of scale to reduce the solution's overall cost. This characteristic is achieved using a multi-tenancy model. An instance of computing resources such as hardware, operating system, and database can serve different customers (tenants) but remain isolated from each other. It is also vital to note that the customer may have neither

knowledge nor control of the exact physical location of the resources within this characteristic. Subsequently, many providers allow for greater transparency and control to customers in terms of the specific location of resources as highlighted by Samani et al. (2014).

- Rapid elasticity: Sometimes, a new cloud customer faces uncertainty during times of peak demand in their business seasons. Consider the three months leading up to Christmas, when businesses face an increase in demand. Those customers wishing to buy Christmas presents for their families and friends through online shopping will, in turn, increase the website bandwidth due to high demand for web resources. Does the business buy twice as many resources just to support one-quarter of its business? Therefore, this specific feature of cloud computing seeks to address this predicament by providing cloud customers with the capability to rapidly deliver or release resources (Samani et al., 2014).

- Resource distribution: this can be tweaked as a customer requires more (or less) servers or storage. At its core, cloud elasticity entails continual reconfiguration in the network and related controls from the cloud provider. Vacca (2016) describes that NIST distinguishes two types of scaling options: horizontal and vertical, which involve launching additional services and resources, and changing the computing capacity of assigned resources, respectively.

- Measured service: Cloud ecosystem automatically control and enhance resource use by a metering capability at some level of concept suitable to the type of cloud services such as bandwidth usage and storage. Bandwidth usage can be checked, controlled, and reported, providing some level of transparency and trustworthiness for both the service provider and cloud customer (Ali, 2019).

Computer systems and cloud forensic investigation are influenced by cloud SLA

policies adopted by cloud service providers in different cloud service models. A cloud forensic investigator has no physical access to the devices, mostly the routers and server logs, including monitoring tools. The nature of cloud and virtualisation has also made it difficult for cloud forensics investigators to acquire admissible evidence as when the virtual machine is shut down all forensic evidence in that VM will be lost (Poisel, Malzer, & Tjoa, 2013; Wahyudi, Riadi, & Prayudi, 2018).

1.4 Cloud Models

In addition to the broad definition of Cloud computing, defining the various cloud models available is also essential. These models are based on two characteristics; as the cloud matures over time, there will likely be a further advancement in facilities, such as data storage and application (Leavitt, 2009). Blockchain technology can change how data is viewed in the cloud ecosystem, how programs are created and solve the problem of geo-location and cross-border forensic investigation. (Lillard, 2010). Cloud technology has redefined how data are currently stored regarding unlimited storage capability, and many businesses have taken advantage of this. Storage has moved from traditional data centres to remote servers managed by third parties, where applications can be developed, deployed, and accessed online without purchasing and installing them on computers; this on-demand process spread across multiple jurisdictions based on ownership and sharing.

Public Cloud: Typical examples include e-mail services, such as Hotmail, Apple iCloud, and Dropbox's storage services. These services are made available to the general public, who can access them via the Internet (Hosseinian-Far, Ramachandran, & Slack, 2018). The infrastructure is managed and owned by the CSP with the infrastructure located outside of the customer's premises. Besides, there are trust issues

between the cloud service provider and its customer in the contract agreements. The ISP has more control and restricts the cloud customers from managing the logging resources. Moreover, cloud customers are unaware of where the cloud resources (servers) are located and how many tenants make use of one particular server.

Private Cloud: For a public cloud, architecture is based on multi-tenancy, where different organisations share one physical server through virtualisation. The private cloud is reserved for a single cloud customer (Laili et al., 2013; Quang-Hung, Nien, Nam, Tuong, & Thoai, 2013). In a private cloud, the customer may have some level of control in the area of cloud deployment, the infrastructure, platform, or software. However, cloud service providers still have some level of managerial control. Equally, while the infrastructure may be located either on or off-premise, the customer will have control over its particular geographic location (Zissis & Lekkas, 2012). In regards to sensitive and confidential data storage, the private cloud should be a preferable model, and some government agencies have taken advantage of this.

Community Cloud: Community cloud incorporates a private cloud to integrate various clients into the cloud network that have a collective concern. Besides, it has some comparable features to that of the private cloud. The community cloud is mostly deployed in the healthcare environment that enables efficient communication within the ecosystem. There is a partnership of participants with shared awareness, and through the approval process, only trusted participants are provided access to the data as highlighted by Samani et al. (2014). Figure 1.4 demonstrates the architecture of the community cloud model and the multi-tenancy where three enterprises share the same server environment in a visualised multi-tenancy (Goyal, 2014; Marinos & Briscoe, 2009).

Hybrid Cloud: A hybrid cloud falls in between the private and public cloud architecture. It takes care of the solution where the business intends to combine public and

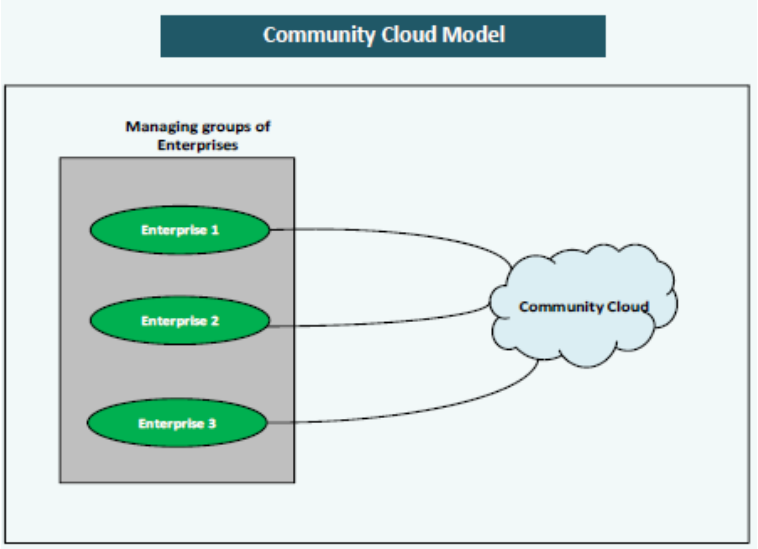


Figure 1.4: *Community Cloud Model*

private cloud solution in order to achieve an aim. Many businesses such as Facebook and Overleaf adopted this type of cloud model that facilitated user communication within their domains.

1.5 RESEARCH METHODOLOGY

The DSRM adopted will prove a detailed description of the defined problem through a set of phases and steps. This step-by-step methodology provides a backbone to design and verify the model (Broy, Feilkas, Herrmannsdoerfer, Merenda, & Ratiu, 2010). Our research methodology focuses on the BCFL framework that integrates a Blockchain into the cloud ecosystem to solve the challenges faced by cloud forensic investigators in acquiring admissible evidence in the cloud (Awuson-David et al., 2021). However, in the testing phase, the designed virtualised testbed was used to simulate the BCFL framework and capture any issues in the design with feasibility to correct any problem found. DSRM will support a systematic transition from simulation models to system realisation with reliable results that validate our framework and answer our research

objectives and questions. It guarantees that no weakness or susceptibility is ignored and that all crucial evidence is collected before designing the model. The Design Science research methodology is a model developed by A. Hevner and Chatterjee (2010) and involves the production of exceptional new and real knowledge. It consists of the output of an artefact, which in computer science research could be in the form of an adaptation, invention, improvement, or routine design (Mullarkey & Hevner, 2015).

There are different implementations of DSRM in the literature. Given the nature of this research, it was challenging to find a suitable research methodology. However, for this research, we adopted the model developed by Peffers, Tuunanen, and Niehaves (2018). The DSRM process developed by Peffers et al. (2018) divides a computers science research into the following stages; identify the problem and motivate, define objectives of the solution, design and development, demonstration and evaluation, and finally communication. This research focuses on the novelty and contribution of using tamper-proof Blockchain distributed ledger logging capability to ensure evidence admissibility in a cloud ecosystem (Awuson-David et al., 2019).

Furthermore, this service is a hybrid platform that ensures evidence integrity, trustworthiness and immutability in the cloud. Previous studies have mostly ignored the importance to note the difficulties in maintaining the admissibility of log artefacts in the cloud ecosystem, both in transit or storage (Awuson-David et al., 2019). While forensic examiners can use many automated analysis tools for examination purposes, they are not satisfactory as it is not a foolproof process for discriminating between forged traffic craft by cybercriminals and genuine traffic. Forensic examiners need to perform acquisitions in the cloud ecosystem, identifying the adversary's techniques used to compromise the network (Awuson-David et al., 2019). Figure 1.5 highlights all the processes this research will take.

As shown in the diagram, there are three phases. Each phase describes the step the

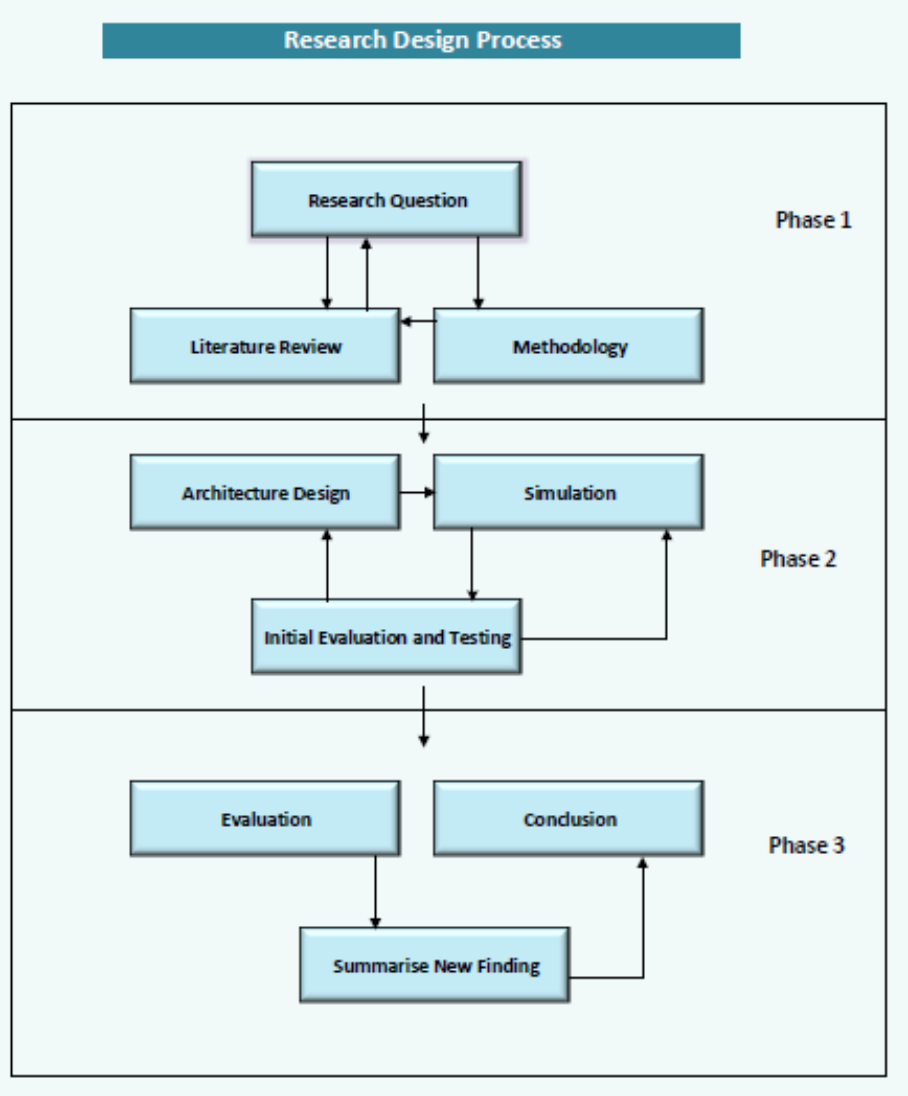


Figure 1.5: *Research Design Process*

research will take to answer the research questions and accomplish the objectives.

Phase 1: Problem Identification and Motivation: In this phase, there will be an overview of forensic investigators' current problems and challenges in acquiring admissible log evidence from a cloud environment. The research question part will

help this project to have a clear insight through literature review and methodology to answer the question of how digital investigators will acquire admissible log evidence from the cloud (Awuson-David et al., 2021).

Methodology: The methodology was the initial phase of the research work. It started with evaluating current and associated literature for the defined problem and then studying and synthesising information and literature relevant to the research area. Significant studies of the current methodologies of Blockchain integration in the cloud ecosystem were also undertaken, and a Blockchain cloud methodological comparison table was also evaluated. The main goal was to gain knowledge about securing log evidence and trust issues related to the cloud ecosystem and to identify feasible solutions Awuson-David et al. (2019).

A Design Science Research Methodology is used after extensive research on various methods available, and DSRM emerged as a suitable method to use for this research as it can support a simulated scenario that enables a scientific experiment to be conducted in a controlled virtualised lab environment. It gives the ability to go back during the design process in case of any defects. The research topics under investigation are current trends and security issues in the cloud. Hyperledger Blockchain architecture was adopted due to its capabilities as a permissioned Blockchain. There is a need to create trustworthiness between cloud stakeholders and maintain tamper-proof log evidence in the cloud ecosystem.

Literature Review: After gathering and studying the literature, the objectives are formulated and linked to the fundamental aspects of digital evidence trustworthiness, transparency and cloud security. Each system objective was achieved by evaluating the deployment models and digital forensics challenges in the cloud (Awuson-David et al., 2019). This literature review also highlighted the challenges forensic investigators faced in acquiring digital admissible evidence in the cloud. This was an essential step because

every aspect of the cloud ecosystem is subject to an attack by malicious adversaries. After reviewing and analysing the current literature, a research gap identified the need to find a solution that will support forensic acquisition in the cloud environment. The current forensic acquisition toolkit is not designed for the cloud, but rather it is designed for the traditional network where devices are physically located and are accessible at all times. The nature of the cloud and its on-demand virtualised process has made it difficult to use the standard network forensic toolkit to acquire admissible evidence. The information gathered to support this research mostly consisted of scientific papers, journals, conference proceedings, ISO standards and textbooks (Awuson-David et al., 2021).

Phase 2: Design and Development of System

Architecture: This phase will look at the three-main architecture approved by the National Institute of Standard (NIST) public, private and hybrid cloud. However, the hybrid cloud will be adopted in our architecture. The justification is for us to evaluate different cloud models that suit our experimental scenarios. The research will adopt two scenarios (Supply chain and Car Insurance) to evaluate our BCFL. The hybrid cloud architecture for the research will start with our permissioned Blockchain algorithm with an integration of Elasticsearch, Logstash and Kibana (ELK) system monitoring capability to measure endpoint visibility and real-time and support live forensic analysis. The research will explore the capability and components of Hyperledger Blockchain technology. The smart contract, hashing, channelling, data Immutability and encryption mechanisms through BCFL integrated program codes activate these functionalities. These Blockchain components will aid and support our innovative design architecture in such a way that enables an accurate evaluation and verification of our experiment results.

Simulation: According to Raghuram and Bangalore (2016) highlighted mecha-

nisms needed to deliver unified authentication to security controls, settings, and functioning states in a cloud's virtualisation and hardware layers for audibility and at the lower infrastructure layers of the cloud security providers. The measured evidence allowed organisations to conform with security policies and structured data ethics and controls such as the Federal Information Security Management Act (FISMA) and European GDPR. A real-world Blockchain cloud will be used to simulate our scenario-based experiment in a controlled innovative virtualised lab. In the virtual environment, the Blockchain technology will be hosted on a Linux Ubuntu operating system.

Initial Testing: Initial testing will be conducted to check how the innovative BCFL is generally operating as a cloud PaaS and to answer the research question and objective. Each Blockchain node establish connection and communicating with other nodes. As the architecture runs on the Blockchain cloud ecosystem, a consensus, and a smart contract with log immutability will be tested. The log evidence is anchored in a tamper-resistant Blockchain server that maintains evidence confidentiality, integrity, availability and privacy, as Blockchain has the hashing and encryption capability. Strong authentication and the least privilege principle will be implemented for the management of Blockchain nodes. The virtualisation management system will be restricted to authorised administrators from the participating parties, possibly with separation of duties. A strongly encrypted authentication mechanism will be adopted. A supply chain scenario will be used to test and evaluate the designed framework and identify how the system reacts to a forensic incident caused by insider mistake or complex transaction. The findings in the digital forensic incident phase will be to ascertain how the system would maintain the immutability of logs.

Phase 3: Evaluation and Testing:

Evaluation: The evaluation process occurs in almost every stage of the design and development process of this research. This phase also enables us to validate the frame-

work with various cyber-related incidents that required digital forensic investigation in the cloud. It is important to note that this phase will uncover the current and hidden challenges a forensic investigator could encounter while carrying out forensic investigation in the cloud. This phase formally ensures the new protocol satisfies its specification and functions soundly via an innovative real DSRM based simulated cloud forensic investigation process.

1.6 Original Contributions

Working methods have changed as many of us are now working from home. This has added another layer of complexity in the cloud. There is a need to re-architect the forensic investigation process in the cloud to fully unlock evidence admissibility and mitigate the challenges faced by forensic examiners (Awuson-David et al., 2019; Jasmine, 2019; Yin, Zhang, & Dong, 2020). However, this comes with risk, threats and challenges identified by the CSA cloud threats list as will be highlighted in the coming chapters (Awuson-David et al., 2021; Damenu & Balakrishna, 2015; Yan & Yu, 2015).

1.7 THESIS ORGANISATION

The rest of this thesis is organised as follows, mostly in accordance with the research methodology.

Chapter 2:Literature Review

This chapter introduces background material on cloud computing, digital forensics, Hyperledger Blockchain and cloud forensics (Awuson-David et al., 2019). Furthermore, it discusses the cloud platform and services and provides a clear explanation of digital forensics logs acquisition processes. The challenges associated with the acquisition and

preservation of digital evidence in the cloud ecosystem was discussed. The literature was reviewed to identify the research gaps, the complexity of the forensic investigation, and evidence admissibility. Finally, the chapter provides a brief discussion on the Hyperledger Fabric Blockchain family.

Chapter 3: Methodology and Analysis

This chapter will present a general idea on cloud architecture and current techniques used by forensics investigators to acquire digital evidence in the cloud ecosystem. It will also examine a Designed Science Research Methodology and why it was chosen for the project as the design architecture comprises of components that interact to produce system behaviour. A comparison table will evaluate other methodologies and justify why DSRM was chosen for this research. In the justification analyses, we will examine the concept of the DSRM and explain why it suits our research and also evaluate the integration process of DSRM with our BCFL framework with a clear diagram that highlights the integration process.

Chapter 4: BCFL Design Architecture, Framework and Initial Testing

This chapter describes the architecture process of the BCFL framework and demonstrates how the research objectives will be achieved. The proposed framework design will consist of five phases, namely system design, components installation, algorithm programming, deployment on the virtual environment and testing. Clear identification of each phase and analysis of how admissible evidence will be acquired in the cloud ecosystem will be demonstrated. This chapter will also elaborate on the details of Blockchain Hyperledger composer business network and development components, including implementing models, transaction logic, access control, and query definitions. We will set up a development environment. The chapter will simulate the framework's initial testing phase and look at the core components that facilitate Hyperledger Blockchain. The chapter will highlight the comparison of different Blockchain types

and demonstrate why the chosen DSRM was adopted for the BCFL framework.

Chapter 5: Evaluation, Conclusions, and Summarise New Findings

This chapter presents an evaluation and summary of the proposed Hyperledger Blockchain distributed ledger technology concepts in a cloud ecosystem. It will convey the necessary knowledge and fundamental technical DSRM approach of the proposed BCFL ecosystem and evaluate how forensic investigators will apply these technologies through real-world use cases before concluding the research findings. The chapter further emphasises the adoption of Hyperledger REST Server and angular app developed algorithm to test and simulate the cloud environment. An evaluation of how our BCFL framework integrates with the software development kit (SDK) client will also be demonstrated. A chaincode algorithm will be developed to establish a clear communication peer-to-peer pathway of all the BCFL virtualised containers. Scientific findings discovered in this chapter regarding a new forensic investigation approach in the cloud ecosystem will help summarise the concepts and new findings. We will explore the key components, including channels, Membership Service Providers (MSPs), the ordering service, and Fabric Certificate Authority (CA) (Davenport, Shetty, & Liang, 2018). The analysis presented in this chapter demonstrated how BCFL maintains immutability mechanisms that will facilitate the admissible log evidence in the cloud ecosystem.

Chapter 6: Conclusions

This is a concluding chapter which highlights Blockchain's limitations and future works. It investigates the integration of Artificial Intelligence (AI) and Blockchain technology to enhance and secure the digital evidence acquisition process.

Chapter 2

Literature Review

This chapter describes cloud computing, cloud forensics, digital forensics, Blockchain technology and Hyperledger Fabric Blockchain. By examining them, one will be aware of the challenges forensics investigator face in acquiring admissible evidence from a Cloud ecosystem. This chapter introduces some background knowledge of Blockchain Cloud Forensic Logging (BCFL). Furthermore, it discusses how the current research addresses the complexity and challenges of acquiring a secure and trusted log in the cloud ecosystem. Finally, the chapter provides a brief discussion on Blockchain Cloud Forensics Logging and the most recent log forensic acquisition framework in the cloud.

Our social norms are evolving away from the storage of personal data on a computer hard drives to the retention of that information in the "Cloud," on servers owned by internet service providers, Constantine (2011).

2.1 INTRODUCTION

In 1997, Professor Ramnath of Emory University, USA, noted cloud computing for the first time, calling it an essential new "computing model where the limitations of computing will be ascertained by economic validation rather than technical limits alone (Chandramouli, Iorga, & Chokhani, 2014). The evolution of cloud computing and its

economic impact on businesses, has preceded its increase in an unprecedented rate. As describe by Eurostat, 24 % of large enterprises made use of public cloud computing services in 2014 to 2015 (Budņiks & Didenko, 2014) and according to Makhlouf (2020) Gartner predicts the strongly marked public cloud growth in the EU will continue, with an expected 25% increase to "almost $ 250 billion by 2017, including cloud advertising, (Anderson & Gaston, 2013).

The European Union (EU) has set up its own European Open Science Cloud (EOSC) system that will facilitate cloud storage mechanisms to enable research institutions in the 27 member states to tap into the benefits cloud technology offers as highlighted by Budroni, Claude-Burgelman, and Schouppe (2019). This innovative idea will enable EU-based institutions to independently carry out their own technological and scientific research without the interference of the EU commission.The EU want to ensure that its research institutions are well-positioned and supported to integrate their current network into the cloud. The initiative was launched in 2020 as a transformation of the European strategy to support the speed-up of innovative ideas by adopting cloud ecosystem as demonstrated by Giannoutakis and Tzovaras (2016) and Yang, Huang, Li, Liu, and Hu (2017). The main objective of this strategy is to prepare Europe and its citizens for the significant challenges of technology, storage and meeting the need of institutions using the cloud as a technology to facilitate scientific innovations as emphasised by Almeida, Borges, and Roque (2017). The vast majority of significant businesses have now moved their critical infrastructure to the cloud due to its many benefits in adding value to the business, for example, productivity, storage, memory space, CPU usage and cost reduction. The U.S. Federal Government has also recognised the power of cloud computing, exemplified in the Federal Cloud Computing Strategy (2019), which was designed as an outline for the adoption of cloud services by the government itself. Cloud computing has revolutionised busi-

ness communication and has made a significant impact in supporting business using its on-demand platform and software-based pay-as-you-go model, as highlighted by Coutinho, de Carvalho Sousa, Rego, Gomes, and de Souza (2015); Kulkarni, Agrawal, et al. (2014).

The European Commission agreed to implement Europe's cloud computing strategy (European Commission, 2012), which was last updated on February 27, 2015. The strategy itself is the final product of policy, technology and regulatory landscape analysis, and stakeholder consultation. The strategy aims were to improve European GDP by 1% by 2020 as well as to create 2.5 million jobs in the EU by way of cloud adoption across a wide range of sectors. The strategy focuses on three main actions: (1) cutting through the jungle of standards, (2) safe and fair contract terms and conditions, and (3) establishing a European cloud partnership to drive innovation and growth from the public sector (European Commission, 2012).

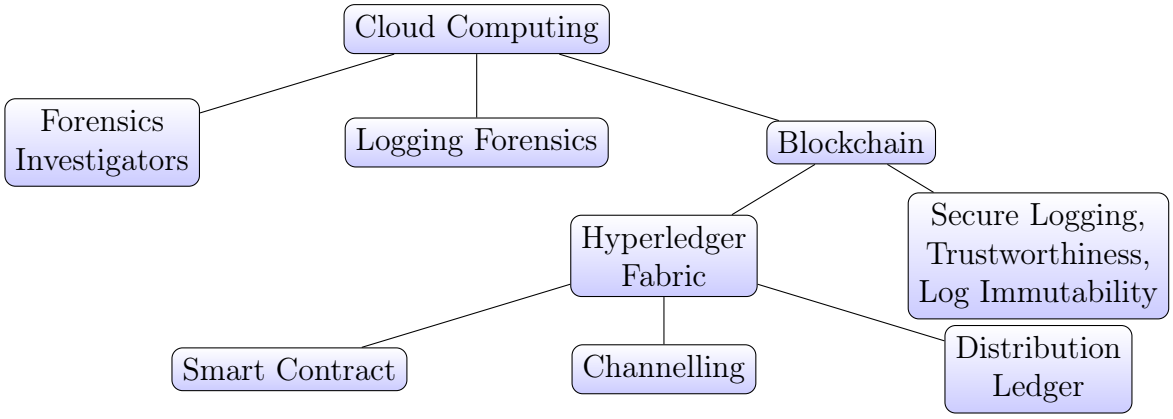
The main advantages of cloud computing are accessibility and productivity in the area of cost and storage. CSPs focus on the service they offer, such as on-demand services, spinning up an operating system, storage, and software services, including security in most cases. For example, the elasticity, multi-tenancy, pooling and on-demand capabilities have made the cloud a magnet for businesses to implement. These capabilities have their challenges and security issues in forensics' acquisition of digital artefacts in the cloud ecosystem. It is clear that the current forensics acquisition process in the cloud ecosystem is not working and a comprehensive new approach is required (Awuson-David et al., 2021). Besides, the growth of cloud technology has made the technology a primary target for cybercriminals who explore its vulnerabilities, while also leveraging its resources for anti-forensics purposes, particularly in terms of its ability to store and process evidence (Wani, AlZahrani, & Bhat, 2020).

This research sets out to examine whether existing forensic tools can be used in the

cloud ecosystem and whether the evidence acquired with the use of these tools can be admissible in court. After a comprehensive review of relevant literature , it has become clear that a new evidence integrity and trustworthiness approach is needed. As cloud forensic logging has its complexity and challenges regarding the admissibility of logs evidence in a court of law, little research has been conducted in the area of forensic logging. The chapter will concentrate on the literature review of related works in cloud computing, and cloud forensic as underscored by Patrascu and Patriciu (2014); Potey and Nikumbh (2013); Zawoad et al. (2013). The first phase of this chapter will look at cloud computing in general and various cloud services followed by a comprehensive review of all the risks, threats and challenges highlighted by the cloud security alliance (CSA), including recommended approaches to take. The second phase will focus on cloud computing and logging with relevant literature. The third phase will focus on cloud forensic logging and traffic flow analyses as an approach to acquiring admissible evidence in cloud computing.

As mentioned in Chapter 1, cloud computing has transformed how organisations, businesses and modern society communicate and store data. This was achieved by virtualising multi-million-pound data centres into a service to rent to organisations in a self-automation fashion, according to their resources and need, at an affordable rate. While these resources benefit numerous users, there is also evidence that the cybercriminal can leverage the same system benefits to commit a crime as described by Kaufman (2009) and Galante, De Bona, Mury, Schulze, and da Rosa Righi (2016). Figure 2.1 describe clearly the main areas of review of this research. The diagram also demonstrates the interconnection and review of cloud computing, logging forensics, and forensics investigation. Additional, the Blockchain and the Hyperledger Fabric permissioned Blockchain were all reviewed in the following sections.

Figure 2.1: *Main areas of Review*



Traditional network infrastructures and cloud computing vulnerability have persisted due to the Advanced Persistent Threat (APT) in e-crime. Cybercriminals have been ahead of the game by creating and deploying payloads such as Heartbleed, Internal Blue and ransomware with ease.

2.1.1 Cloud Computing Services

Bhardwaj, Jain, and Jain (2010) and Kavis (2014) demonstrate how cloud computing technology in today's World has redefined the communication landscape. Businesses have benefited from automated cloud platforms by having the choice of on-demand services such as SaaS, PaaS and IaaS according to their needs. Cloud customers have implemented the cloud storage model in gigantic numbers, with Gartner forecasting immense growth in the area, stating that end-users will be storing a third of their data in the cloud emphasised by Quick, Martini, and Choo (2013). However, many organisations have remained reluctant in deploying their data into the public cloud storage. The global community has a lack of unified standardisation with a single voice on how data control and security should be managed in the cloud environment. For instance, an organisation who fails to conform with the GDPR may face administrative,

civil, and criminal sanctions as stressed by Team (2020); Voigt and Von dem Bussche (2017). The European Union Digital Agenda Commissioner, acknowledges the need and benefit of cloud technology across Europe and stated that a resilient cloud would facilitate communication between businesses. In addition, the Commissioner stated the need for a digital health economy across Europe and sounded a warning that failure to implement a robust cloud system would impact on the already vulnerable European digital economy. I want to make Europe not just "cloud-friendly", but "cloud-active" described by Kroes (2010); Sandre (2015).

infrastructure as a Service (IaaS) is generally referred to as a version of the public cloud, like IaaS. A platform where user do not need to be concerned about significant networking components such as the server, database system and data storage solution including bandwidth. Figure 2.2 below highlights the three primary types of cloud services and their difference from a traditional network. The diagram demonstrates the responsibilities of the CSP and its customers on ownership rights of the services they provide.

The diagram's simple logic is that cloud customers have limited user ownership rights compared to the physical network, where the organisation manages all devices and resources. However, in the cloud ecosystem, the service providers have more rights regarding how the cloud resources are distributed, including the cloud architecture's logs and security. Cloud users only need a basic understanding of the network in order to manage the on-demand resources on the IaaS ecosystem. Besides, the cloud customer can also spin the provider operating system, application stack, or database as they see fit.

Platform as a Service (PaaS):) is where the cloud customer rents a network resource on-demand from a cloud service provider . This resource mostly includes the operating system (OS), bandwidth, data storage, and the software stack vital for

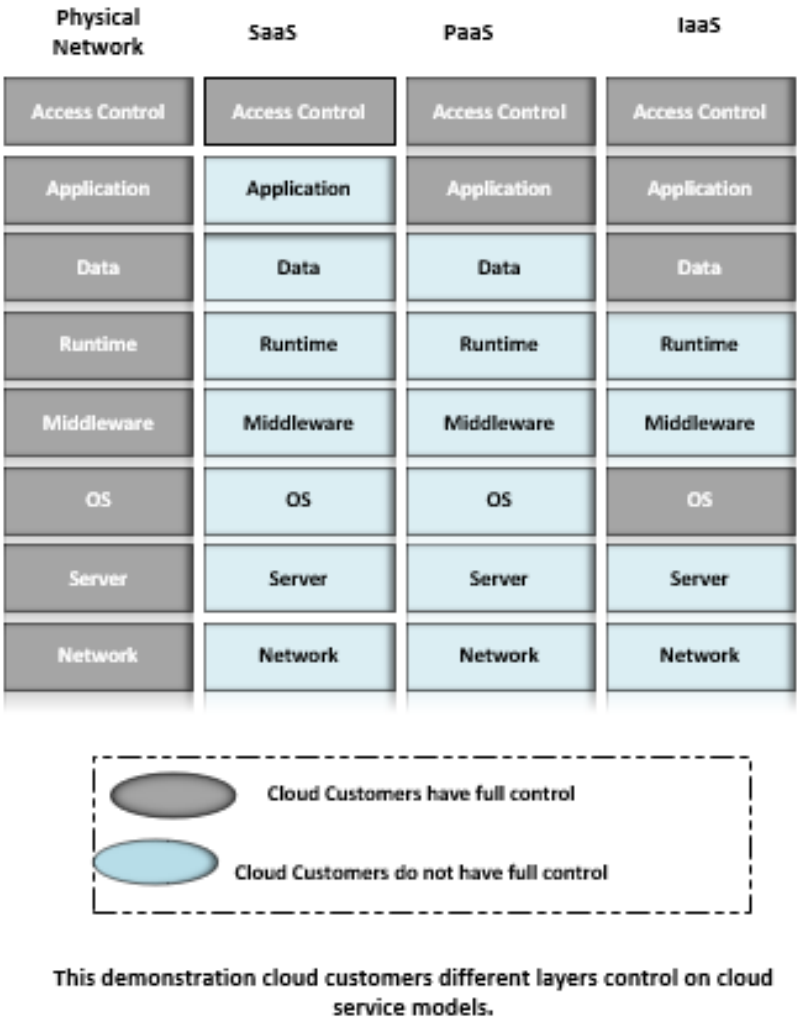


Figure 2.2: *Cloud Services Comparison*

the organisation's operational need. The customer can then install their applications and solution in line with the CSP service level agreement. They will then manage all the resources. As demonstrated by the CSA (2016), more resources ownership rights are provided to IaaS customer by the CSP, while some of the resources rights are not provided in the PaaS ecosystem. For instance, within the PaaS ecosystem, the CSP manages storage, operating system, and network, but the customer can build or develop customised applications. Examples of PaaS platforms include AppEngine from Google, Microsoft Azure, WaveMaker, SimpleDB, S3, DynamoDB, SQS, and SQF from

Amazon (CSA, 2016).

Software-as-a-Service (SaaS): Espadas et al. (2013) describes a SaaS service model as applications that run at the service provider or delegate services under their control as a third party. Users can access the cloud on-demand resources and applications through a browser, thin client, or mobile device. Examples of SaaS are Google Docs, Gmail, and MySAP and many more solutions are being developed in this ecosystem. The cloud service providers have full control of all resource in the SaaS. In contrast, a cloud customer has no rights in application maintenance, security, configuration or system update.

Traditional IT: In the traditional IT ecosystem, all computing components are deployed and managed on-premise. The organisation buys its servers. Its administrators troubleshoot the system when needed and manage the hardware, software, and application security. For instance, the organisation's IT department acquires network hardware or software of their choice and manages the acquired solution, including configuration change management. In addition, they also install the operating system and deploy the customer relationship management (CRM) application on the server and client computers. Backups are managed by them on schedule, as is the expansion of the capacity. Table 2.1 demonstrates a public cloud technology growth forecast conducted in 2020 by Gartner. It shows an evident, steady growth of the public cloud from the year 2018 to 2022.

As pointed out by Gartner's projection to 2022, the cloud digital economy has grown on changeable rate relate to IT services . Similarly, they pointed out that the upsurge request for cloud services such as PaaS, IaaS and SaaS has increased significantly. In addition, Gartner's 2019 survey emphasised that business sees cloud technology as an investment of top importance. The survey also emphasised an upsurge in the digital

	2018	2019	2020	2021	2022
Cloud Business Process Service (BPaaS)	45.8	49.3	53.1	57.0	61.1
Cloud Application Infrastructure Services (PaaS)	15.6	19.0	23.0	27.5	31.8.
Cloud Application Services (SaaS)	80.0	94.8	110.5	126.7	143.7.
Cloud Management and Security Services	10.5	12.2	14.1	16.0	17.9.
Cloud System Infrastructure Services (IaaS)	30.5	38.9	49.1	61.9	76.6.
Total Market	**182.4**	**214.3**	**249.8**	**289.1**	**331.2.**

Table 2.1: *Worldwide Public Cloud Service Revenue Forecast (Billions of U.S Dollars, Gartner, 2020)*

cloud market by the end of 2019 as described by, Coyle and Nguyen (2019); Mohamed and Ali (2018).

2.1.2 Cloud Computing Threat s and Challenges

This section focuses on the security challenges pointed out by the CSA. Firstly, in 2010 they highlighted seven top security threats to cloud computing, as discussed in this section. Furthermore, in 2013 they increased the cloud security threat level to nine and further increased it to twelve in 2016 as emphasised by Ramgovind, Eloff, and Smith (2010); Rong, Nguyen, and Jaatun (2013); Subramanian and Jeyaraj (2018). They then highlighted more cybersecurity incidents in the cloud environment, such as data breaches, identity fraud, credential details and access management, and vulnerable authentication measures. Furthermore, Singh and Chatterjee (2017) also pointed out that application program interfaces (APIs), system vulnerabilities, account hijacking, malicious insiders, advanced persistent threats, data loss, lacking due diligence, exploitation and nefarious use of cloud services, Denial of service (DoS) and user access right issues, are also important cybersecurity threats currently faced by cloud technology.

As mentioned earlier, the cloud has changed the landscape of communication through

its on-demand service where business can easily carry out maintenance and upgrades without re-cabling or rebuilding the entire IT infrastructure. Moreover, using the cloud will allow companies to take advantage of the best and latest technology since they will not have to disassemble and rebuild their entire IT infrastructure in order to upgrade (Ruparelia, 2016). With the number of users using cloud resources increasing exponentially, the reality is that unscrupulous cybercriminals will also form part of that cloud community. The steady increase of organisations moving their IT infrastructure into the cloud has added a layer of complexity and hurdles to be overcome for the digital forensic investigators charged with investigating cloud-related cyber incidents (Brown, 2015; Martini & Choo, 2012).

In Control Engineering (2008), Gartner reinforces this view "If you cannot get a contractual commitment to support specific forms of investigation – along with evidence that the vendor has already successfully supported such activities – then your only safe assumption is that investigation, and discovery requests will be impossible" (Popović & Hocenski, 2010). The position is such that Fred Cohen & Associates, (2009) pertinently ask "When you run a forensic analysis, how are you going to prove that it was correct and repeatable when you cannot even tell me what operating system and software was doing the analysis?" This chapter also describes the challenges and complexity of evidence acquisition, preservation, and custody chain in the cloud ecosystem. It is essential to point out that there is very little case law on cloud forensics . This provides another layer of complexity due to the absence of legal precedent for forensics examiners. In Figure 2.3, the CSA pencilled down twelve cybersecurity threats after conducting a comprehensive attack threat pathway survey in the cloud as mentioned earlier. The cloud security challenges are very complex irrespective of cloud services and deployment models. The nature of the cloud and its explosion in market growth has increased cyber-attack threats as noted by the CSA.

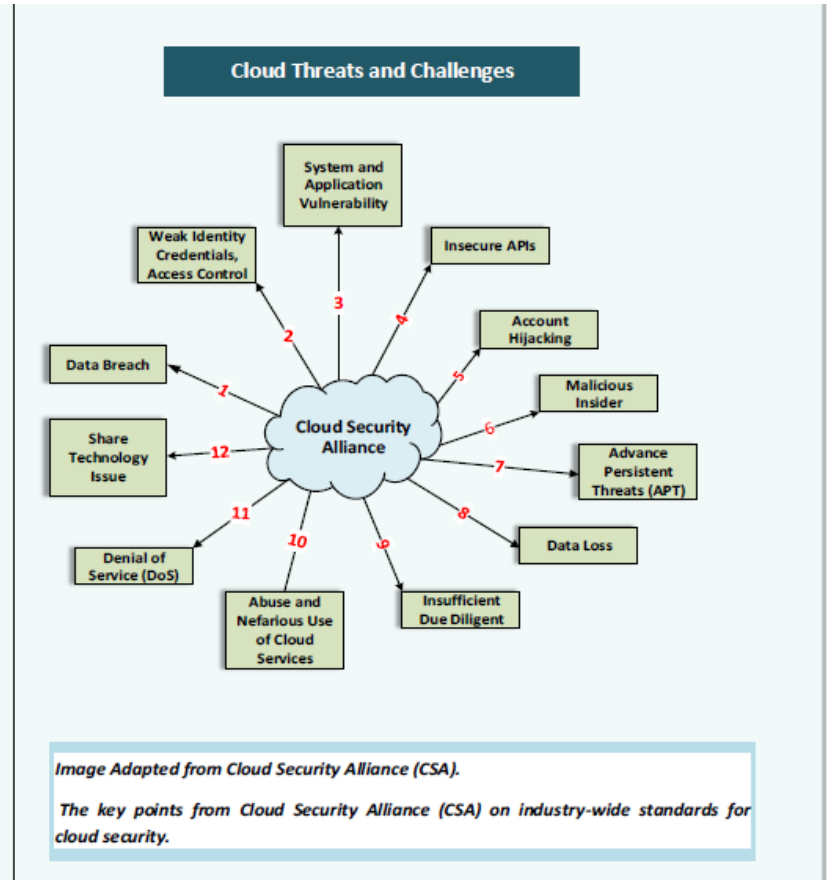


Figure 2.3: *Challenges and Cyber Threats in Cloud Ecosystem*

The CSA has emphasised the security challenges cloud computing faces as cybercriminals continue to exploit cloud weakness due to the nature of its design. As the cloud ecosystem continues to grow, research has shown that more businesses are now moving their IT system into the cloud (Marston, Li, Bandyopadhyay, Zhang, & Ghalsasi, 2011). The cybersecurity threat level has increased in the cloud as cloud adversaries continue to target the infrastructure by exploiting its weaknesses, and vulnerabilities (Fernandes, Soares, Gomes, Freire, & Inácio, 2014; Jansen, 2011).

By understanding the cybersecurity threat challenges in the cloud computing ecosystem, we first need to define the word "threat" And what it means in the cloud computing ecosystem. First let us look at some of the threat definitions. According to the Interna-

tional Organisation for Standardisation 27001 Information Security Standard, threat is defined as "a possible reason for an unsolicited occurrence, which may result in damage to a system". Under the Payment Council Industry's Data Security Standard , (PCI DSS) a threat is described as "a condition or activity that may cause information or information processing resources to be intentionally or accidentally lost, modified, exposed, made inaccessible, or otherwise affected to the detriment of the organisation" (Wei, Wu, & Chu, 2018).

Data Breach: This research will be adopting the data breach and threat model during the testing and evaluation phase of this project. Recent media report media publication by Security Boulevard have indicated an increase in cloud computing data breaches (Loureiro, 2021). This persistent threat has become a significant challenge facing institutions and corporation network infrastructures around the globe. Data breaches in the cloud ecosystem are associated with many factors, ranging from system vulnerability, misconfiguration, inside attack and malicious software, and including unsafe security practices within a corporation.

Research by Ponemon Institute Shirazi, Seddighi, and Iqbal (2017) has highlighted that cloud network users have to face more unprecedented data breaches than non-cloud users . The current complexity in cloud networking has made it difficult for a forensic investigator to respond to real-time cloud incidents due to the cloud architecture's nature, where management of cloud security significantly depends on the CSP. According to research carried out by the Ponemon Institute and Symantec , the average cost to businesses relating to data breach in 2013 was $5.4 million. The corresponding cost of lost business came to about $3 million. As described by Chen, Paxson, and Katz (2010) and Ab Rahman and Choo (2015), any security incident analyses in cloud will affect the overall resilience and performance of the cloud ecosystem. Their report highlighted what is new about cloud computing security and indicated the need for

more efficient incident response mechanisms. According to Y. Zhang, Juels, Reiter, and Ristenpart (2014), the shared resources environment is exploited to permit side and covert channels between collocated virtual machines. In contrast, the challenges of cloud multi-tenancy, such as the forensic investigator's ability to acquire admissible digital evidence in the multi-tenancy environment without interfering with the other shared tenancies, is very complicated. There has been evidence of innovative research that focuses on the security issues, and digital forensics investigation complexity associated with the cloud multi-tenancy environment, the fundamental source of cloud-specific risks (AlJahdali et al., 2014; Jasti, Shah, Nagaraj, & Pendse, 2010).

2.1.3 Forensics Logging and Storage

There should be a standard framework in the cloud ecosystem that makes provision for cloud customers to have access to security logs timely delivery of cloud audit logs during a forensic investigation (Jasti et al., 2010). However, it is essential to note that CSP's should make log evidence available to their customers who, by their legitimate right, should have access to the virtual machine's logs (Kalaiprasath, Elankavi, & Udayakumar, 2017). A cloud customer remains responsible for performing a risk assessment to identify all the security requirements for their cloud-based service And also selecting the appropriate security and privacy controls before selecting a cloud provider and or brokers, CSA (2016).

As demonstrated by Zawoad et al. (2013), critical challenges of acquiring admissible log evidence in the cloud include use of a secure evidence preservation mechanism that focuses on decentralisation of logs among several servers and several layers of cloud architecture. Furthermore, Zawoad et al. (2013) pointed out the weaknesses in the log API and cloud management console. There has been no concrete evidence of these methods providing logs and also preserving the user's privacy. Earlier research

does not offer a secure way of revealing the logs while maintaining user privacy which, along with secure data transmission, is vital to make the cloud more cyber resilient. Zawoad et al. (2013) describes the benefits of the cloud, especially within businesses. For example, organisations can save 37% of their cost just by moving their IT infrastructure from an outsourced data centre to Amazon's cloud. However, the potential issue of malicious insiders still stands because the investigators have to depend on the CSP for potential logs evidence. The BCFL will solve the complexity of logging in the cloud. It will create trustworthiness between cloud stakeholders and facilitate a secure cloud ecosystem. BCFL will maintain forensic readiness and enhance cloud forensic acquisition processes and maintain acquired evidence integrity. However, the cloud user privacy challenges are relevant as they highlight log management approaches and the vital need for a secure logging mechanism. Cloud logging is considered an essential means of security control which helps investigators in identifying, answering, and precluding operational issues, incidents, violations, and fraudulent activities, (Lemoudden, Bouazza, & Ouahidi, 2014). Logging is mainly used in monitoring systems to collect data that aid forensics investigators to ascertain adversaries' motives.

However, Patrascu and Patriciu (2014) offer a new way of monitoring activities in cloud environments and data-centres by providing an approach that will holistically examine the methods of how logs are managed and maintained in the cloud ecosystem. The researchers highlight the security issues within the cloud ecosystem and point out how to adopt a more practical approach in managing logging in the cloud. Their paper explains the general forensics architecture stating that it is a modular architecture; the cloud computing framework contains two main layers, namely the virtualisation layer and the management layer. In the virtualisation layer, some workstations host the virtual machines and have virtualisation enabled hardware. In the management layer, there are modules responsible for enabling the entire operations specific to the cloud,

as emphasised by Khan et al. (2016) who reviewed the current state of cloud forensic logging (CFL) and the challenge of acquiring admissible evidence in the cloud ecosystem. Their research also pointed out a more practicable implementation of logging mode and the adoption of cloud log-as-a-service to mitigate some of the challenges.

Muthurajkumar, Ganapathy, Vijayalakshmi, and Kannan (2015) adopted an algorithm to implement secure logging mechanisms that encrypt the log on storage in order to preserve its integrity. A traditional network digital forensic investigation is carried out within the network infrastructure where the investigator will have full access to the crime scene. In most cases, this will help to maintain the evidence chain of custody (Montasari, 2017; Simou, Kalloniatis, Kavakli, & Gritzalis, 2014). However, there is no access to a physical computer in cloud environments, and forensic investigators must then decide how they will acquire the potential evidence from the cloud. Furthermore, the NIST Cloud Computing Forensic Science Working Group has defined a practical approach on how digital forensic investigators should process digital evidence in the cloud. However, the approach did not go far enough in addressing cross-board investigation and precise identification of cloud incident crime scenes. Nonetheless, it highlighted some of the challenges investigators could encounter during evidence acquisition in the cloud (Simmon, 2018). Many organisations have multiple data centres across the globe but the country that hosts the data centre might not have a forensic investigation legal framework in place and it becomes difficult to ascertain the admissibility of acquired log evidence in this situation. There is a need to ensure the admissibility of acquiring log evidence in the cloud; however, our proposed BCFL framework will ensure this and provide a solution to evidence acquisition challenges in the cloud. The proposed BCFL framework will achieve this by integrating Blockchain distribution ledger technology into the cloud ecosystem as part of the integration. Potey and Nikumbh (2013) briefly explained the risks in cloud computing and proposed a framework which employs ac-

countability by generating logs for each user. The researchers explained the benefits of the cloud system and the opportunities it holds—for instance, reducing business risks and creating economic benefit. In order to increase trust in the cloud, we need to make the cloud transparent and accountable. Potey and Nikumbh also reviewed the research gap which their paper intends to solve, including pointing out how logs can be secured in the cloud. However, the limitation of this paper's proposed framework is how the acquired logs' integrity could be maintained. For example, when the virtual machine instance is deleted, the investigator will lose all the logs, however, our BCFL framework uses Blockchain immutability and distributed ledger technology to maintain acquired evidence integrity. The scalability, of cloud computing with its raw processing power and unlimited storage, has attracted large organisations and government institutions. Recent media reports, surveys and statistics have indicated that more than half of US companies already use a cloud computing service. Figure 2.4 demonstrates the digital forensic investigation process that considers forensic investigation's environmental impact. It also highlights how the environmental impact could be managed and, at the same time, support the admissible evidence acquisition process. As shown in Figure 2.4 the products technologies quadrant requires the open-source and vendor communities to develop court-admissible and scientifically repeatable network forensics analysis tools (NFAT) for the examination of any network-based crimes.

The conception of a future NFAT solution is fundamental to understand, whether employed as a single monumental device or as a sequence of devices. It offers six efficient design goals , as shown in Figure 2.4. The solution could be a small portable network forensic evidence collection device built using inexpensive hardware and open-source software, also capable of functioning across several modes of operation for different network evidence collection scenarios. The six design goals are Environment Assessment

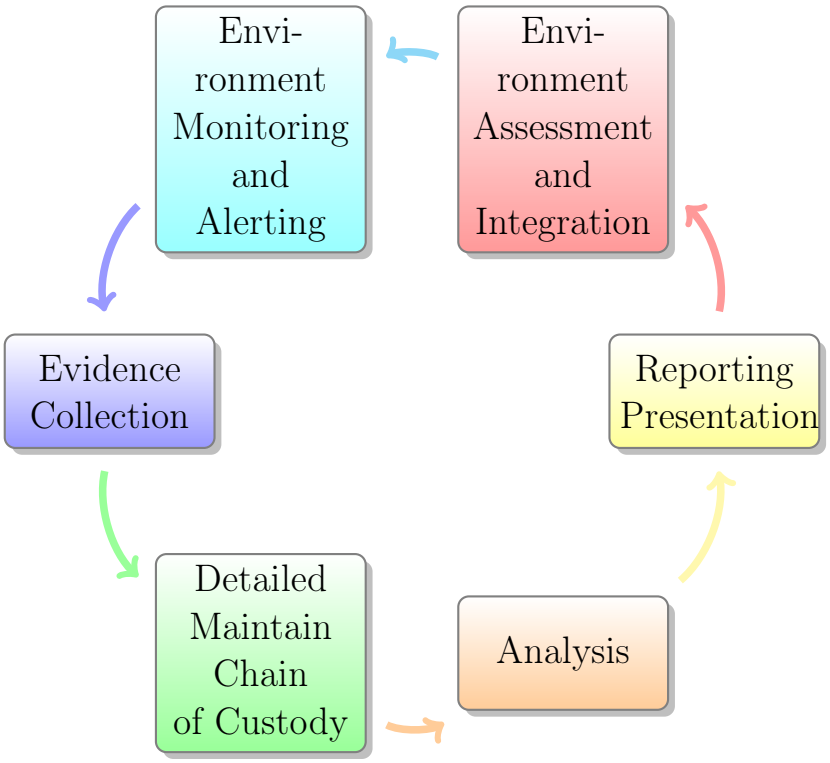


Figure 2.4: *NFAT Design Goals*

and Integration, Environment Monitoring and Alerting, Evidence Collection, Detailed Maintain Chain of Custody, Analysis , Reporting and Presentation . Each of the design goals is meant to represent a stage in the lifecycle of a network forensics investigation (Garrison, 2010).

2.1.4 Challenges in Cloud Forensics

Multi-tenancy, geo-location, elasticity and on-demand service in the cloud ecosystem have added considerable complexity and challenges in cloud forensics investigation compared with the traditional digital forensics processes. A cloud forensics investigator faces challenges in acquiring, preserving , analysing and presenting digital evidence in a form that is admissible before a court, which have led to the issue of contamination of cloud evidence acquisition. First responders cannot acquire evidence in the cloud

ecosystem on arrival at the crime scene due to limitation of access and where the evidence is stored in a multi-tenancy environment. This creates limitation in documenting the crime scene and further investigation due to the lack of specific acquisition tools. Many other issues such as geo-location, legal issues, privacy issues, data encryption, and elasticity also limit access to the cloud crime scene.

According to Simmon (2018), the main aim of first responders and forensic examiners may be the same in the cloud environment and a traditional large-scale network. Elements of the structural nature of cloud ecosystems such as recognising accountability among the stakeholder and inability to acquire network logs from the load balance or routers, multi-tenancy, and rapid elasticity has provided unique scenarios for digital investigations according to Alqahtany, Clarke, Furnell, and Reich (2015). These researchers described that the implementation of conventional techniques to conduct forensics in the cloud will not be successful due to the cloud's highly decentralised nature. The NIST report of 2015 also detailed the challenges faced by digital forensics investigators in the cloud computing ecosystem. It went on to highlight these challenges as significant apprehensions in acquiring admissible digital evidence in the cloud ecosystem. What is striking in their report is that the challenges highlighted by them are also issues in a traditional forensics investigation. As mentioned, they detailed the following issues:

- Time: It is often a vital issue related to time synchronisation and the possible loss of evidence if not found swiftly. Zimmerman and Glavach (2011) stated that once the digital evidence is acquired, all associated entities have time-synchronised via a consistent time source such as Network Timing Protocol (NTP), which will facilitate chain of custody and evidence preservation. The research could have provided more clarity on how time synchronisation could lead to evidence being contaminated due to improper time documentation and chain of custody. Few

of the researches reviewed have indicated that client side log files should match time stamps on the cloud service provider log files. It will also be challenging to establish evidence admissibility if these do not match. In addition, if the evidence is not found quickly enough, it may be overwritten or lost in some other manner as noted by Freet, Agrawal, John, and Walker (2015); Group et al. (2014), and Manoj, Bhaskari, et al. (2016).

- Locating: Forensic artifacts from social media application and contents can be a time-consuming process relating to cloud forensic investigation process. A skilled knowledge of network components will support a forensic investigator in identifying physical locations of VMs, and it is vital to identify where the backup and redundant storage are located. Acquiring the evidence can be difficult as pointed out by Poisel et al. (2013); Simou et al. (2014). Their research highlighted the need to define the crime scene before proceeding with a full investigation. Locating the cloud's digital crime scene is very complicated as the virtual servers could be outside the investigator jurisdiction.

- Sensitive Data: There has been a significant increase in data breach cases in the cloud. According to the U.S. Commission on Intellectual Property (CIO.com) estimates, there are over $300B in annual losses to U.S. companies due to theft. The pervasive use of cloud computing by employees for personal use could heighten insider theft risk given the low-cost storage arrays available and low-cost, high-speed bandwidth to move data, (Group et al., 2014; Vincze, 2016).

- Cloud Architecture: Architectural challenges such as diversity, provenance, multi-tenancy, data segregation including dealing with variability in cloud architectures between CSP and its customer are quite complicated during resource provisioning on the issue of the cloud SLA. SLA plays a vital role in defining a legal agree-

ment between cloud customers and the CSP. The SLA intends to establish a good working relationship of solving disputes between the two parties (cloud customer and the CSP) concerning service delivery. However, this has not been the case as the quality of service delivered to the customer with access to logs are not verifiable. In addition, cloud customers are not given access to logs in a multi-tenancy environment. This is why the agreement is seen as a one-sided document that only suits the CSP's (Ibrahim, Kliazovich, & Bouvry, 2016). Furthermore, the CSP should ensure that the security of their service is guaranteed to a reasonable level to their customer's. For instance, privacy and application control issues arise from third parties, including data integrity and verification. Data loss and theft should all be mitigated by the CSP (Subashini & Kavitha, 2011). The proliferation of systems, locations, and endpoints can store data accurately and secure provenance for maintaining and preserving the chain of custody. Infrastructure to support cloud resources seizure without disrupting other tenants is challenging (NIST, 2016).

- Evidence Analysis: There are complexities in analysing some of the processes in the cloud that associate with traditional forensic investigation such as correlation, reconstruction, time synchronisation, logs, metadata, and timelines. Presenting digital evidence in the cloud ecosystem in an admissible state is problematic as establishing the incident crime scene is challenging, and reconstructing it and acquiring VM images, metadata and accurate timeline. Analysing of log evidence, including synchronisation of timestamps, can prove challenging. With all these challenges and complexities in mind, our BCFL research framework approach has answered these challenges. BCFL distribution ledger mechanisms will be used to mitigate the difficulties in identifying the crime scene in the cloud ecosystem and maintaining the chain of custody. With the Blockchain integration in our

framework, geo-location issues will be mitigated as every node in the consensus mechanisms have a copy of the transaction ledger.

2.1.5 Cloud Multi-Tenancy

A key component of cloud computing is its "multi-tenancy", a capability, which is referred to as a "shared pool of resources" in the NIST definition (Brender & Markov, 2013). Services usually cannot give access to raw log data as it contains records of multiple users. Forensic investigators having access to these logs could compromise the privacy of other customers (ENISA, 2012: 45). Therefore, features synonymous with clouds (storage) services such as multi-tenancy, data security, file encryption, and communications encryption need to be addressed as part of a digital forensics investigation(Awuson-David et al., 2021). As suggested by various other researchers including, Chung, Park, Lee, and Kang (2012) multi-tenancy can be defined as cloud services that support on-demand resources or applications by multiple users. Different users ('tenants', hence the term 'multi-tenancy') may thus independently create ('instantiate') and run their own VMs (applications) within one physical server. They can terminate their VMs when no longer required, install their firewalls, and manage their virtual networks. Multiple users may share the use of common physical infrastructure; this enables resource consolidation, economies of scale, and involves more efficient resource utilisation than dedicating separate physical machines to different users. Also, different tenants within a multi-tenanted environment can only gain access to their own information within the service (AlJahdali et al., 2014; Y. Zhang, 2015). Multi-tenancy, therefore, maximises the resources by allowing shared access to resources. This is demonstrated by Tsai, Shao, Sun, and Elston (2010) with "72,500 customers supported by 8–12 multi-tenant instances in a 1: 5000 ratio." For example, data might be located in multiple national jurisdictions. Simultaneously, the identifi-

cation of both evidence and perpetrators, along with the acquisition of artefacts, can be problematic in a multi-tenant environment (Almulla & Yeun, 2011; Grispos, Storer, & Glisson, 2012; Taylor, Haggerty, Gresty, & Lamb, 2011). These issues suggest that there is a requirement for specific methods and techniques in digital forensics that can be applied in the cloud in order to obtain evidence in a way that will not affect the potential admissibility of the gathered evidence (Ruan et al., 2013).

According to Weissman and Bobrowski (2009), "multi-tenant outsourced services typically cannot give access to the log evidence as it contains artefacts of multiple users and thus would compromise the privacy of other cloud customers". Consequently, structures identical with cloud ecosystem services such as multi-tenancy, data security, file , and encryption mechanisms need to be applied as part of a digital forensics examination. Many applications require robust and tamper-proof evidence acquisition systems, for example, electronic voting or bank information systems (Cucurull & Puigalí, 2016). In the presence of the untrusted host, some threats could compromise data security. An adversary or insider could compromise a cloud network using payload such as man-in-the-middle or distribution denial of service attack (DDoS) (Hoque, Bhattacharyya, & Kalita, 2015). A log tamper-proof evidence approach is needed to mitigate the challenges faced by the cloud forensics investigator (Pichan et al., 2018). However, researches have shown an approach to acquired log evidence in the cloud system (Alqahtany, Clarke, Furnell, & Reich, 2016; Khan et al., 2016). Meanwhile, these approaches do not take confidentially and log evidence's integrity into consideration as acquired evidence needs to be admissible. However, a virtual machine is hosted on a physical server in the data-centre. That means that the investigators need to retrieve the logs from the physical server in addition to the virtual machine's logs. Then the question is, where is the server hosted? Do the investigators have access to the server (loss control of the crime sense)? It is essential for forensic investigation to prove a

cybersecurity incident case beyond a reasonable doubt by reconstructing the processes taken to acquire the evidence. It will be a complex task for forensic investigators to reconstruct the crime scene in the cloud environment as evidence could be stored in jurisdictions with rules and laws that make data retrieval for the purpose of forensics difficult (Brown, 2015; Pichan et al., 2015).

Figure 2.5 emphasises the nature of a multi-tenancy ecosystem where different organisations and individuals rent an on-demand service from the CSP.

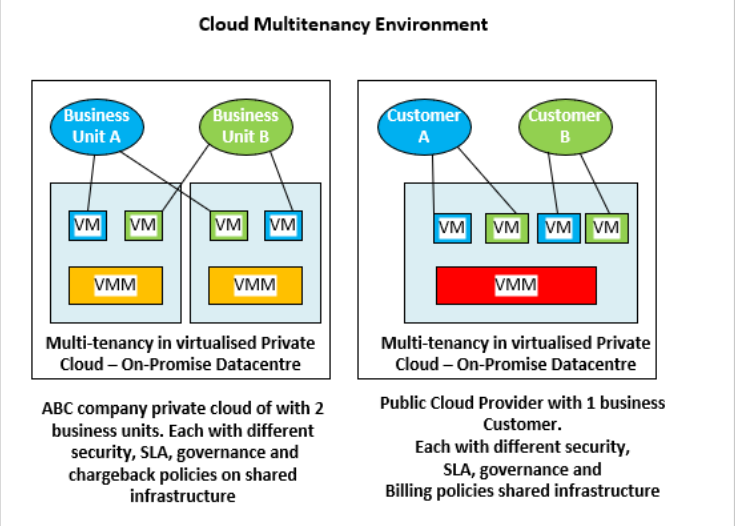


Figure 2.5: *Multi-Tenancy Architecture in Cloud Ecosystem*

However, Figure 2.6 demonstrates how the CSP stores clients' applications in a single virtualised server environment where each client application is partitioned inside the server. The complexity in this scenario is that a cyber incident that affects one client in this ecosystem can also find its way to the other clients in the same shared virtual server. With all this in mind, it is quite a challenge for the forensic investigator to acquire admissible evidence in the multi-tenancy environment. The BCFL framework is designed to mitigate the above mentioned challenges in the multi-tenancy ecosystem, one of the research gaps this research will fill, and answers questions 2 and 3 of this research.

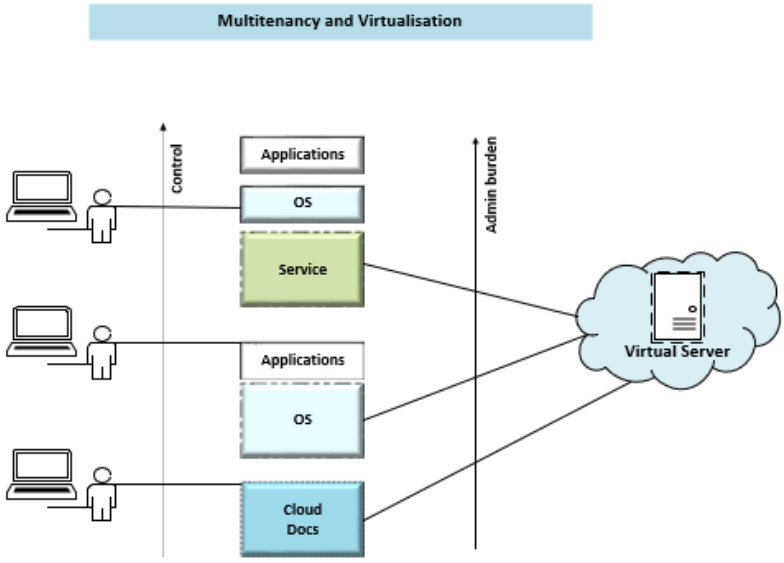


Figure 2.6: *Multi-Tenancy Architecture in Cloud Ecosystem*

Ruan et al. (2010), described the following organisational challenges and presented the significance of an efficient organisational structure to conduct forensic analysis in a faster way.

- **Investigation officers:** Every cloud service provider hosting organisation data should have certified forensic investigation officers.
- **Technical staff:** This term covers all the networking professionals, security team members, administrators, hacking experts, cloud security experts and all other support staff members who help the investigation team with different sets of information.
- **Incident Reporting:** These staff members report any suspicious event such as unauthorised system or data access, unwanted data leakage or loss, detection of internal exploit or some malware script.
- **IT Lawyers:** Cloud service providers must have some legal advisors, who can handle the complicated cases related to the multi-jurisdictional and multi-tenant

scenario.

- **Chain of Dependencies:** Cloud service providers prefer to have external assistance for conducting a forensic investigation.

2.2 Cloud Service Providers (CSPs)

A Cloud Service Providers is a company that provides different services to cloud users by giving access to on-demand cloud resources. Cloud customers access the resources without having in-depth knowledge or details of their location and ownership. Cloud consumers are only charged based on cloud resource utilisation, and such a phenomenon is known as pay-as-you-go in cloud computing (Armbrust et al., 2010). One resource can be used by many users to increase efficiency and throughput and also reduce the idle time of the resources in cloud computing. These CSPs provide their services to cloud customers in three main service categories, which are also known as service models, for cloud computing, namely (a) IaaS, (b) PaaS, and (c) SaaS (Kumar, Raj, & Jelciana, 2018). However, these services have increased over time with others such as logging-as-a-service and many more. Some of the most attractive benefits for cloud computing involve a customer's ability to receive services from a broker or provider, and expand their requirements at scale; the burden of scaling is placed on the broker or provider and becomes transparent to the user (Manvi & Shyam, 2014).

2.2.1 Securing Cloud ecosystem

As the adoption of cloud computing by organisations continues to increase year after year and businesses continue to tap into the benefits it provides, so too, the vulnerability of its platform increases. The cloud computing age has the added complexity of digital accounting, record keeping, communications, funds transfer and many more as-

sociated activities due to multi-tenancy and geolocation issues (Singh, Jeong, & Park, 2016). This degree of complexity has made evidence acquisition in the cloud difficult, of which many cybercriminals have taken advantage. Specifically, online banks, e-commerce and supply chains are seeing a rise in fraud, and retailers' databases are being attacked and robbed of their credit card information. Besides, "identity theft is currently the fastest-growing white-collar crime", (Fafinski, 2006). There is a need to keep the cloud ecosystem secure, trusted, and transparent to enhance evidence acquisition in its domain. There are many names to cover these requirements, from information security to information assurance. The overriding set of principles are those of C I A (Confidentiality, Integrity, Availability), which are considered by many to be the tenets or principles of computer security. "Attempts to break into a system from outside will get much publicity" (Armbrust et al., 2010; Duncan & Whittington, 2015; Rahman, Daud, & Mohamad, 2016). Firstly, confidentiality prevents unauthorised disclosure of cloud data. That means only those who should have access to data in cloud storage will gain that access. Access means not only reading but also viewing, imaging, cloning, snapshot, printing, or simply knowing that a particular asset exists. Secondly, integrity prevents unauthorised modification. In this context, modification includes writing, changing status, deleting and creating (Barona & Anita, 2017). Finally, availability means that data or assets in the cloud ecosystem are accessible to an authorised user at appropriate times. In other words, if users have legitimate access to a particular set of resources, that access should not be denied. Furthermore, cloud computing works in highly dynamic and distributed environments and requires protection mechanisms to prevent intentional or unintentional violation of security policies (Cheng, Liu, & Yao, 2017). Intruders can often circumvent the access control mechanisms, exploiting the virtual machine application that has vulnerabilities or flaws in its design. Furthermore, Ruan et al. (2013), stated that cloud actors include service

providers, cloud consumers, cloud brokers, cloud carriers, and cloud auditors. Garfinkel (2010) emphasised that cloud computing, in particular, may make it impossible to perform necessary forensic steps of data preservation and isolation on systems of forensic interest. The proposed BCFL framework focuses on the preservation of log evidence and ensuring the admissibility of acquired logs in cloud environments (Awuson-David et al., 2021). However, ignoring the challenges is not helpful to any cloud stakeholder. We need a framework that introduces a concept of a trust boundary in the cloud ecosystem, and the much-needed framework can then enhance the integrity of cloud log evidence. The BCFL framework will ensure the integrity of log evidence acquired in the cloud ecosystem and enable cloud stakeholders to create forensic endpoint visibility of the cloud ecosystem (Awuson-David et al., 2021).

Ruan and Carthy (2012) proposed a framework describing cloud forensics as the application of digital forensic science in cloud environments. Technically, it consists of a hybrid forensic approach (e.g., remote, virtual, network, live, large-scale, thin-client, thick-client) towards the generation of digital evidence. Organisationally it involves interactions among cloud actors (i.e. cloud provider, cloud consumer, cloud broker, cloud carrier, cloud auditor) to facilitate both internal and external investigations (Ruan et al., 2013). Legally it often implies multi-jurisdictional and multi-tenant situations (Awuson-David et al., 2019). However, the proposed approach ignored the issue of evidence, integrity and admissibility of the evidence by not ensuring a more transparent evidence chain of custody. Santos, Gummadi, and Rodrigues (2009) proposed an approach for a trusted cloud computing platform (TCCP) that enables IaaS service such as Amazon EC2 to provide a closed-box execution environment. TCCP assures confidential execution of guest VMs and allows users to attest to the IaaS provider and determine if the service is secure before they launch their VMs. This approach is a big step towards securing data integrity (Santos et al., 2009). However, there is a

limitation on how data in storage or transit can be secured. A similar approach is also proposed by F. Zhang, Huang, Wang, Chen, and Zang (2008), where they introduced PALM, a VM live migration framework. This strategy consists of three modules for the privacy and integrity protection of the sensitive data, the metadata, and the live migration process itself that are highly evidential in regards to digital evidence. Metadata plays a crucial role in digital evidence source as it has faced numerous integrity tests in forensic acquisition. Furthermore,Agrawal, Bawa, and Bayardo (2008) took a different direction in the area of access control. They adopted a fascinating approach to private indexing by introducing a distributed access control enforcing protocol. More recent work deals with the privacy problem from a different perspective by empowering the users to gain better control over the indexes (Z. Xia, Wang, Sun, & Wang, 2015). In detail, they proposed a three-tier data fortification framework to help mitigate the privacy issues, which provides robust, medium, and low protection according to the data owner's needs (Ullah, Xuefeng, & Feng, 2013; Verginadis et al., 2017). However, as stated by Coutinho et al. (2015) and Ko, Lee, and Pearson (2011), cloud computing's promise of elasticity empowered by virtualisation introduces several new complexities in accountability related to the ability of processing and monitoring multiple virtual and physical resources connected in a highly dynamic fashion. Additionally, log evidence acquisition becomes a challenging task, and the risk of possible evidence contamination is relatively high due to the remote nodes on the network (Verginadis et al., 2017). Therefore, if an instance comes under suspicion, isolating it before commencing investigations will prevent accidental or unavoidable access to other instances. Dykstra and Sherman (2012) highlight a framework for acquiring digital evidence from Amazon's Elastic compute Cloud (EC2) in adopting a process of using traditional forensic toolkits supported by Amazon export features. The researchers went on to implement Eucalyptus which operates correspondingly as EC2 but from the customer endpoint.

However, in most cases, using traditional tools such as Guidance Software Encase and AccessData Forensic Toolkit can successfully acquire digital evidence in EC2 and Eucalyptus. However, the challenges of the admissibility of such acquired evidence remain. While Marturana, Me, and Tacconi (2012) conducted a study on how to acquire digital evidence stored in Windows 7 cloud app through a web browser such Google Docs and Dropbox, however, the complexity of the acquisition process remains unclear about how to avoid contamination of such acquired evidence.

Katilu, Franqueira, and Angelopoulou (2015) emphasised the need to fill the research gap in maintaining the confidentiality of data and provenance tracking of data that aid visibility across the cloud ecosystem. Also, there is a need to maintain this visibility through data confidentiality and to bridge the gap between the existing system security mechanisms and the provenance aware system. The research demonstrated that if secure provenance mechanisms are applied, it can support digital forensic examiners to trace back the source of crime and maintain admissible chain of custody. Again, it also helps to facilitate evidence reconstruction. The admissibility of evidence provenance data is due to the design nature as it is difficult for the data owners to dispute ownership. This means there will be clear visibility in the investigation as acquired evidence can be linked to a suspect in a recording time. However, there are limitations if malware could tamper with data ownership or Metasploit payloads used to manipulate data could result in changing the data ownership while in transit or storage in the cloud ecosystem as the research has not answered these questions of malware data manipulation attacks. However, the provenance tracking system could enable easy establishing of the crime scene in the cloud ecosystem and evidence acquisition but issues such as multi-tenancy and geo-location cloud have not been considered.

Martini and Choo (2012) and R. Han, Guo, Ghanem, and Guo (2012) proposed a four-step cloud digital forensics investigation model that will enable digital forensic

examiners to acquire admissible evidence in the cloud ecosystem. According to their proposed model, the first step in cloud digital forensic investigation is planning and pencilling down the aim and objectives of their investigation on how to achieve the investigation goals. The second stage is to understand the type of cloud technology and services (IaaS, PaaS or SaaS) that will be investigated. Thirdly gaining a background knowledge of the technology and tools that will be used to carry out the investigation. The fourth step is fragmented into three groups representing the client-side of cloud technology, the server-side and the developer-side, each with further actions to ensure an admissible evidence acquisition process. The proposed model considers the three issues that affect cloud forensics acquisition, the service type, the technology and the sources of evidence. All of these regulate the tools and the acquisition methods that are considered suitable for the investigation. However, one of the downsides of this model is the ordering of the processes as, questionably, evidence source identification should come before the assurance of the background technology. David Schoorman, Mayer, and Davis (2016) and Rousseau, Sitkin, Burt, and Camerer (1998), defined trust as "the willingness of a party to be vulnerable to the action of another party based on the expectation that the other will perform a particular action important to the trusting party, irrespective of the ability to monitor or control the trusted party." This definition does not fully capture all the undercurrents of trust, such as the likelihood that the trustee will accomplish a specific action and will not involve in unethical conduct (Adjei, 2015).

The solution of Roschke, Cheng, and Meinel (2009) demonstrates the integration of IDS mechanisms in VMs that may reduce the specific threats of cyber incidents in the cloud ecosystem. It will ensure evidence admissibility and preservation of evidence throughout the investigation lifecycle. In the digital forensic investigation process, the investigator must prove beyond any reasonable doubt the admissibility of the acquired

evidence and its preservation, including a secure chain of custody (McKemmish, 1999). Therefore, there is a research gap that BCFL will fill by providing a solution to solve the challenges that the above papers have ignored in transparency, acquired artefacts integrity, hashing the acquired logs, and maintaining evidence integrity throughout the acquisition process. As will be shown in the coming chapters, Blockchain demonstrates the capability to maintain log evidence integrity, privacy, and trustworthiness in the cloud ecosystem. It has achieved secure logging in the cloud through different approaches by integrating its core components such as smart contract, channelling, log evidence immutability and encryption.

2.2.2 Computer Forensics

As demonstrated by many academic publications in the area of cloud forensics remain somewhat ambiguous. Some of the published papers in the area have provided comprehensive details for the research vital in cloud forensics by stressing the challenges for the digital forensic examiner (Birk & Wegener, 2011; Pichan et al., 2015; Quick et al., 2013). Researchers with a specific methodological emphasis regularly focus on server forensic analysis providing recommendations for issues such as logging and remote extraction of data as highlighted by Marty (2011). Easttom (2016) describes forensics as mainly the use of science to process evidence and establish the facts of a case. The individual case being examined could be criminal or civil, but the process is the same evidence acquisition process. The evidence has to be examined and processed in a consistent scientific manner. However, the world is full of digital devices that have computing capability to store and retrieve data. Furthermore, acquiring digital evidence from those devices should be carried out with a standard digital forensic toolkit, and this is where the subject of computer forensics can be explored as evidence acquired should be admissible. According to Casey (2011), several forensic procedures

and principals should always be applied when dealing with digital evidence. These principals and procedures are: 1) The integrity of the evidence should not be affected by the actions undertaken when securing and collecting evidence. 2) Training should always be conducted for the personnel responsible for examining digital evidence. 3) The evidence has to be scientifically examined with approved tools in a process that can be repeatable and admissible in court (Awuson-David et al., 2019).

According to Hayes (2015) describes Locard's Exchange Principle by Dr Edmond Locard, a forensic scientist at the University of Lyon, developed a theory known as Transfer of Evidence whose premise was that whenever a criminal comes into contact with his environment, a cross-transference of evidence occurs: "Wherever he steps, whatever he touches, whatever he leaves, even unconsciously, will serve as a silent witness against him. Besides, not only his footprints but his hair, the fibres from his clothes, the glass he breaks, the tool mark he leaves, the paint he scratches, the blood or semen he deposits or collects" (Hayes 2014). However, this theory applies also in cyberspace, for example, all network devices such as servers, firewalls, intrusion detection system (IDS) and Wireshark as network monitoring toolkit keep all the logs of every user data transmitted in the cyberspace. Many social media sites such as Facebook, Twitter and Myspace have authentication records of all users who browse their websites with a timestamp of their login and who they were communicating with in the platform. In the United Kingdom (UK), there are standards and procedures computer forensics examiners should follow. The most useful guide in the UK about digital forensics evidence handling and processes is the Good Practices Guide for Computer-Based Electronic Evidence, published by the Association of Chief Police Officers (ACPO, 1999)."Principle 1: No action taken by the police or their agents should change data held on a computer or other media which may subsequently be relied upon in court. Principle 2: In exceptional circumstances where a person finds it necessary to access

original data held on a target computer that person must be competent to do so and give evidence explaining the relevance and the implications of their actions. Principle 3: An audit trail or other record of all processes applied to computer-based evidence should be created and presented. An independent third party should be able to examine those processes and achieve the same result. Principle 4: The officer in charge of the case is responsible for ensuring that the law and these principles are adhered to" Montasari and Hill (2019). This applies to the possession of and access to the information contained in a computer. They must be satisfied that anyone accessing the computer or using a copying device, complies with these laws and principles (Horsman, 2020). However, this process is complex and challenging to adhere to in cloud forensic evidence acquisition as the incident crime scene in the cloud ecosystem is fragmented in different geolocations.

2.2.3 Blockchain

There are two different types of Blockchain, permissioned and permissionless. Permissioned Blockchain such as Hyperledger Fabric is designed with confidentiality in mind. In contrast, permissionless Blockchain such as Bitcoin does not have confidentiality (Gaur et al., 2018). We are going to highlight different Blockchain cloud frameworks in this section. Blockchain, as a distributed ledger technology is a game-changer in the way we handle our digital assets online. Blockchain data immutability mechanisms can maintain trustworthiness among cloud actors and facilitate a secure forensic acquisition in the cloud. Furthermore, suppose Blockchain is viewed from a business perspective. In which case it can be defined as a secure database system with a distributed ledger technology. Suppose one views it from a technical perspective. In that case, it could be defined as the core peer-to-peer (P2P) network that is cryptographically secure, with its smart contract mechanisms that facilitate secure communication through a consensus

mechanism. This allows Blockchain to be a decentralised consensus mechanism where no single authority is in charge of the database (Niranjanamurthy, Nithya, & Jagannatha, 2019). A block is simply a selection of transactions bundled together in order to organise them logically. In addition, Blockchain technology can be viewed as a smart database system that technology can be viewed as a smart database system that uses its core digital distributed ledger technology to build a trustworthy environment that allows individuals from different geo-locations and often participants who do not trust each other to agree on a transaction and ledger updates. Blockchain asymmetric cryptography and smart contract mechanisms have maintained transaction integrity and support of ledger consistency. Blockchain technology has facilitated business growth through its core characteristics such as secure transactions, decentralisation, persistency, anonymity and digital forensic (Awuson-David et al., 2021; Fanning & Centers, 2016; Fosso Wamba, Kala Kamdjoug, Epie Bawack, & Keogh, 2020; Zheng, Xie, Dai, Chen, & Wang, 2017).

Blockchain is made up of transactions, and its size is variable depending on the type and design of the Blockchain in use (Xu et al., 2017). The genesis block is the first block in the Blockchain; however, the structure of the blocks depends on the type of Blockchain one is deploying (Pourmajidi & Miranskyy, 2018). For example, permissionless Blockchain (Bitcoin) defers from that of Permissioned (Hyperledger Fabric) (Awuson-David et al., 2021; Gai, Wu, Zhu, Xu, & Zhang, 2019). A model based on a distributed Blockchain cloud architecture is a technology which offers low-cost secure with on-demand access to the utmost inexpensive computing IoT network. This was achieved by creating a distributed cloud ecosystem, with cost efficiency and unlimited performance computing (Sharma, Chen, & Park, 2017).

Tosh et al. (2017) discussed the importance and need for Blockchain and the capability of this technology to change the way we secure digital assets and present the

current vulnerabilities in the Blockchain cloud ecosystem. Their paper went on to propose a model block withholding (BWH) attack in a Blockchain Cloud ecosystem, considering distinct pool reward mechanisms. The ProvChain mechanism is a framework that has the capability to collect, validate cloud data origin, through embedding the source data into a Blockchain transaction. The ProvChain mechanism operates in three stages: Provenance data collection, Provenance data storage and finally, Provenance data validation (Awuson-David et al., 2021, 2019; Liang et al., 2017). However, the BCFL framework is designed to maintain logs' integrity in the cloud ecosystem and facilitate evidence admissibility. It also creates trust among cloud stakeholders to support forensic investigation needs in the cloud. It achieves this by using Blockchain smart contract and distributed ledger to ensure secure log evidence that is repeatable with easy to view chain of custody (Awuson-David et al., 2019). Traditional digital forensic techniques may not be possible to capture evidence and preserve evidence as the same as in the cloud environment. According to Awuson-David et al. (2019); Hill, Chopra, and Valencourt (2018) the decentralised ledger of Hyperledger Fabric Blockchain provides auditable transaction logs, making legal disputes less likely and simpler to settle (Bashir, 2018).

In an improved cloud-oriented society, the capacity to recognise, obtain, preserve, and analyse possible digital artefacts is valuable for business continuity, security and resiliency. Whether an incident response to an adversary, data breach, or backing of litigation (Awuson-David et al., 2019). Establish an admissible evidence pathway is critical in solving the current cloud challenges (Awuson-David et al., 2019; Ryoo, Rizvi, Aiken, & Kissell, 2013; Samarati, di Vimercati, Murugesan, & Bojanova, 2016). Distributed consensus, data consistency, and immutability of processed transactions will solve the challenges of Cloud evidence admissibility (Awuson-David et al., 2019). Additionally, the distributed consensus mechanisms make it nearly impossible to tamper

with a transaction on the network (Awuson-David et al., 2019; Nakamoto & Bitcoin, 2008).

Figure 2.7 demonstrates how Blockchain technology uses encryption and hashing mechanisms as the core building block in maintaining the integrity of transaction in the Blockchain ecosystem. This was the highlight in the design of Bitcoin Blockchain where there was clear visibility of cryptographical linkage of all vital components of the ledger technology. A Merkle tree illustrates this concept where a tree structure shows how every leaf node calculated the hash of its data while the non-leaf node shows the hash of its fundamental child. Blockchain uses this concept to maintain and preserve the integrity of all its transaction within the ecosystem.

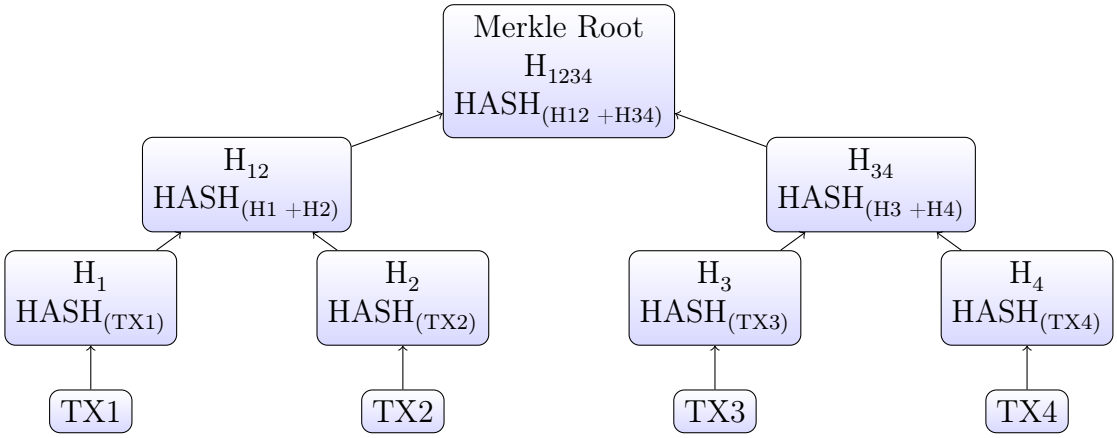


Figure 2.7: *Blockchain Merkle tree*
Bosamia and Patel (2018)

It is important to note that the Merkel tree is a cryptographic hashing tree mechanism as mentioned above, used by Blockchain to ensure that data block of every leaf node is hash and verified (Liang et al., 2017) and (Awuson-David et al., 2021; J. Li, Wu, & Chen, 2018). All non-leaf nodes in the hierarchical structure are categorised with a hash and digital signature of the child node as its input. The cryptography hash function is designed to take any input (Awuson-David et al., 2021). The cryptography hash function is designed to take any input data and produce an output built on the

algorithm in use that has changed fixed length. However, the output hash function is always a string value that is programmed in Java, Python or Go language, for instance, the SHA256 hash is a 256-bit 32byte string of the input data (Awuson-David et al., 2021). The Merkle tree structure enables Blockchain to maintain the integrity and confidentiality of all the data processes and transactions between nodes in the ecosystem. It permits Blockchain participants to remove a leaf that is considered private, but, maintains the hash algorithm, thus preserving the integrity of the tree (Awuson-David et al., 2021).

Meanwhile, with the decentralisation nature of Blockchain, each block of transactions is validated by the network. As demonstrated by Bettín-Díaz, Rojas, and Mejía-Moncayo (2018), all participants on the network receive information about transactions details, and most significantly it is not controlled by a centralised database owned by one company or institution. Once a block of transactions has been added to the Blockchain, it is difficult to reverse. Every block added on top is a confirmation that the transaction is immutable. The more blocks on top, the harder it is to reverse until it is unfeasible. 2.8 On the Blockchain network, six blocks are accepted as a confirmation that the transaction will not be inverted, (Sompolinsky & Zohar, 2015)

Nick Williamson's definition of Blockchain is as "a system that has three main complementary parts: a shared state, a set of rules for updating state via blocks and a trust model time-stamping" (Mainelli & Smith, 2015). "A Blockchain is a transaction database based on a mutual distributed cryptography ledger shared amongst all nodes participating in a system" (Smith & Kumar, 2018). Table 2.2 shows the comparison of permissioned and permissionless Blockchain technology. BCFL framework is based on Hyperledger Fabric permissioned Blockchain that maintains the privacy of all participants compared to permissionless Blockchain, where all participants can view all transactions.

]

	Permissionless	Permissioned
Public	All participants can read the transaction. All participants can validate a transaction in the block. • **Speed:** Insignificant • **Consensus:** Poor-of-Work • **Blockchain:** Bitcoin, Ethereum • **Token:** Required	All Participant can read the transaction. Only authorised participant can validate a transaction. • **Speed:** Significant • **Consensus:** Poor-of-Work • **Blockchain:** Ethereum after Casper • **Token:** Required
Private	Only authorised participants can read transaction data. Only authorised participants can validate a transaction. • **Speed:** Significant • **Consensus:** Federated Byzantine Agreement (FBA) • **Token:**Not Required	Only authorised participants can read transaction data. Only authorised participants can validate a transaction. • **Speed:** Significant • **Consensus:** Practical Byzantine Fault Tolerance Algorithm (PBFT) • **Blockchain:** Hyperledger Fabric • **Token:**Not Required

Table 2.2: *Permissionless and Permissioned Blockchain Technology Comparisons*

In Figure 2.8 highlights the design structure of Blockchain technology, the genesis block is the first block of the chain and does not have any pointer to any other block in the chain (Awuson-David et al., 2021). However, Blockchain block structure depends on the type of Blockchain, and the type of data stored in the block. For

example, Permissioned and Permissionless Blockchain use a different method to store data (Awuson-David et al., 2021). In this research, we adopted the Hyperledger Fabric Blockchain that stores all its information including the channel, its membership service certificates, the chaincode or smart contract and the consensus algorithm in the block. Moreover, the Permissionless Blockchain such as Bitcoin only stores the data

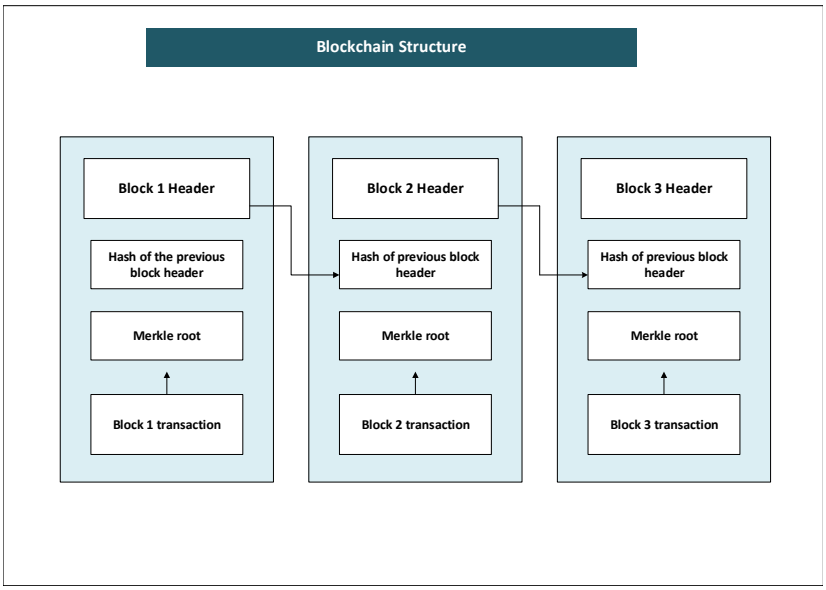


Figure 2.8: *The Block Structure of Blockchain Technology*

source and destination, including its length and size. A Blockchain hashing mechanism is a unique digest of data in a cryptographic hash algorithm. For instance, SHA 256 algorithm can generate a fixed-length hash value of the input data in the block, as demonstrated in Figure 2.8.This process enables easy identification of the blocks inevitably and ensures a tamper-proof communication channel across each block, keeping an immutable record of previous blocks which enables one to define Blockchain as a chain of hashes. All participating peer nodes receive a record copy information of the Blockchain ecosystem and most importantly only on agreed consensus that blocks are added onto the local Blockchain. All Blockchain transaction data is recorded and secure in files, and these files are called blocks. The blocks are chain on the top of

one another as highlighted in Figure 2.8. The most recent block is at the top. All the blocks in the Bitcoin ecosystem have a similar structure, and each of the blocks is chained to the most recent block. Blockchain uses a smart contract via a consensus mechanism to facilitate such data communication and preserve its integrity as defined by Szabo (1996) as "a computerised transaction protocol that executes the terms of a contract. The general objectives are to satisfy common contractual conditions (such as payment terms, liens, confidentiality, and even enforcement), minimise exceptions both malicious and accidental, and minimise the need for trusted intermediaries" (Awuson-David et al., 2021). Figure 2.9 demonstrate the four central core design concepts of BCFL that help to facilitate secure admissible evidence in the cloud. The smart contract, consensus, trustworthiness and immutability of the distributed ledger preserve the integrity of acquired log evidence in our BCFL ecosystem (Awuson-David et al., 2021).

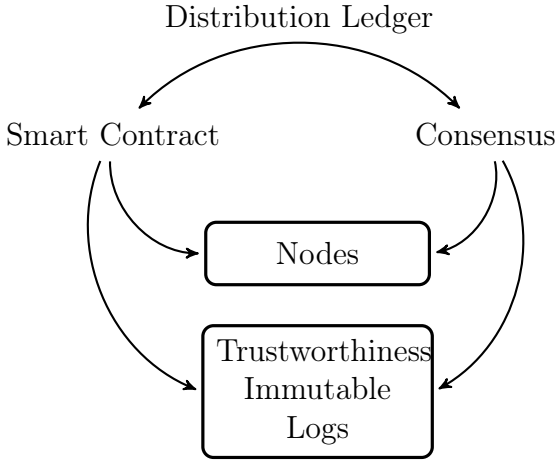


Figure 2.9: *Blockchain Trustworthiness*
(Awuson-David et al., 2021)

Ateniese, Magri, Venturi, and Andrade (2017) examines raised challenges in personal information handling in the Blockchain ecosystem and highlighted some exam-

ples. It emphasised that: "immutable data storage in Blockchain may be incompatible with legislation which requires changes to the 'official truth'." and "Even if personal data is not stored on a Blockchain, metadata can be sufficient to reveal personal information".

The European Securities and Markets Authority (ESMA) Report as highlighted by Janssen, Weerakkody, Ismagilova, Sivarajah, and Irani (2020),also raised concern regarding a permissionless Blockchain such as Bitcoin and its data storage that: "The Distributed ledger technology (DLT) initially designed for Bitcoin fashioned immutable data means that transactions, once authenticated, cannot be modified, cancelled, or withdrawn. While this immutability had clear benefits in a permissionless DLT framework, it appears ill-suited to securities markets". For example, operational errors may necessitate the cancellation of some transactions. It also failed to establish confidentiality among the participant actors.

The European Union Agency for Network and Information Security (ENISA) Report (Berberich & Steiner, 2016), also pointed out potential data integrity compromise when storing data into the Blockchain. They recommend that industries cooperate with the regulators' power to: "Define what to be kept confidential in order to remain compliant with regulatory requirements, such as General Data Protection Regulation (GDPR), as well as sector or local regulations." and "Identify or develop standard methods for removing data from a ledger." However, these concerns do not apply to this project as we adopted Hyperledger Fabric Permissioned Blockchain that gives the participants a high level of data is considered valid in reference to previous data (ENISA, 2017).

Table 2.3 shows the block sizes of Blockchain with their different fields, and the details of each block are highlighted.There are a few private Blockchain solutions out there as the technology keeps evolving. Some are cloud-based platforms as a service

(PaaS) while others run on-premises. Several organisations, such as Deloitte's Rubix and Eris Industries' Monax, offer on-demand service solutions for private Blockchain directly to businesses. Monax, for instance, offers off-the-shelf SDKs solutions to finance institutions, insurance organisations, and logistics industries. Other companies, such as Microsoft and IBM, offer Blockchain as a Service (BaaS) on their cloud infrastructure (Oracle, 2019).

Table 2.3: *Blockchain Block Size Structure*

Size of the Block	Field	Block Detailed Info
4 bytes	Magic no	Primary value 0xD9B4BEF9
4 bytes	Block-size	Bytes numbers are recoded to the end of the block
80 bytes	Block-header	Consists of 6 items in the Block
1 - 9 bytes	Transaction Counter	Positive integer V| = VarInt
-	Transaction	list of transaction

BCFL framework differs from this framework as it focuses on solving the complexity of acquiring admissible log evidence in the cloud ecosystem rather than data transaction. Table 2.4 highlights a comparison of different Blockchain algorithms. In addition, Forensic-Chain framework looks at using Permissioned Blockchain to maintain forensics evidence chain of custody (Awuson-David et al., 2021).

The main idea of forensic-chain is to solve the complexity of digital forensics evidence handling by integrating Hyperledger Blockchain mechanisms such as the data immutability, consensus, hashing and encryption to maintain evidence chain of cus-

Table 2.4: *Highlighting a Comparison of some of the Consensus Algorithms*

Fasts	PoW	PoS	PBFT
Type of Blockchain	Permissionless	Permissionless and Permissioned	Permissioned
Finality of Transaction	Stochastic	Stochastic	Deterministic
Requires Token	Yes	Yes	No
Example Usage	Bitcoin, Ethereum	Ethereum	Hyperledger Fabric

tody and handling by establishing a tamperproof environment throughout evidence processing and preservation from the crime scene to the lab (Lone & Mir, 2019).

Cebe, Erdin, Akkaya, Aksu, and Uluagac (2018) a framework based on vehicle related digital evidence acquisition in post-accident scenarios by using the acquired evidence to reconstruct what happened and who was at fault (Awuson-David et al., 2021). Their paper integrates a vehicular public key infrastructure (VPKI) and uses Blockchain consensus and smart contract mechanisms to establish privacy, traceability and trust within the ecosystem (Awuson-David et al., 2021). They looked at using Blockchain technology to secure and maintain digital evidence integrity in a smart home ecosystem. The paper's main focus is to ensure a secure evidence acquisition and preservation by integrating Permissioned Blockchain mechanisms such as smart contract, hashing, encryption, and data immutability will to ensure evidence admissibility. This high-level Blockchain architecture will mitigate the challenges and complexity of evidence acquisition and admissibility in an IoT ecosystem, (Brotsis et al., 2019). As Brotsis focuses on IoT Blockchain integration, the BCFL framework focuses on acquiring admissible log evidence in the cloud ecosystem. The BCFL framework also maintains log evidence immutability throughout the investigation life circle. All the Blockchain participants agreed on a smart contract with a consensus mechanism. Sharma et al. (2017) developed a distributed Blockchain cloud architecture

cost-effective on-demand solution for IoT network. The Blockchain secure solution enables IoT ecosystem to be more resilient against potential cyber-attack and providing a forensic readiness for the IoT ecosystem (Awuson-David et al., 2021). This was achieved by creating a distributed cloud ecosystem, with cost efficiency and unlimited performance computing.

Tosh et al. (2017) discussed the importance and need for Blockchain and the capability of this technology to change the way we secure digital assets and present the current vulnerabilities in the Blockchain cloud ecosystem (Awuson-David et al., 2019). Their paper went on to propose a model block withholding (BWH) attack in a Blockchain cloud ecosystem, considering distinct pool reward mechanisms (Awuson-David et al., 2021). ProvChain mechanism, is a framework that can collect and validate cloud data origin, by embedding the source data into a Blockchain transaction. The ProvChain mechanism operates in three stages: Provenance data collection, Provenance data storage and finally, Provenance data validation (Awuson-David et al., 2021; Liang et al., 2017). However, the BCFL framework is focused on a cloud infrastructure rather than on an IoT ecosystem. Both clearly demonstrate how a Blockchain cloud be used to maintain data integrity. However, BCFL is mainly integrating Blockchain technology into the cloud ecosystem that ensures log evidence admissibility (Awuson-David et al., 2021).

The BCFL framework mitigates the high level of evidence contamination pathways (multi-tenancy, geo-location, and cloud service-level agreement) in the cloud. In addition, it acts as a bridge that enables evidence acquisition that accomplishes GDPR compliance (Awuson-David et al., 2021). In contrast to the other related work, Provchain is based on permissionless Blockchain and requires Blockchain miners to be paid for the validation of block authenticity. In contrast, BCFL is based on permissioned Blockchain and does not require mining to facilitate its function of ensuring log

evidence acquired is admissible (Awuson-David et al., 2021).

Block-DEF is a digital forensic Blockchain-based framework used to store and preserve digital evidence information in Blockchain. However, in this framework, Byzantine fault tolerance consensus mechanisms are adopted (Awuson-David et al., 2021). Therefore, only evidence information is stored on Blockchain, while the remaining acquired evidence is then stored in a trusted platform (Awuson-David et al., 2021). The framework supports and facilitates traceability, evidence privacy, and guarantees its integrity through a multi-signature technique (Awuson-David et al., 2021; Z. Tian, Li, Qiu, Sun, & Su, 2019). Meanwhile, BCFL focuses on logs evidence admissibility by adopting a permissioned Blockchain.

Noura, Salman, Chehab, and Couturier (2020), research demonstrated how to mitigate the complexity of log compromise by cyber adversaries in the IoT ecosystem. Their research further describes how the proposed Modified Information Dispersal Algorithm (MIDA) will be used to solve the challenges of Anti-forensics techniques targeted against IoT log integrity by a cybercriminal. The MIDA ensures tamper-proof logging mechanisms by adopting a (n-t) degree based on three-layers, the cloud layer, the Fog layer, and the IoT device layer. The proposed anti-forensics countermeasure solution is to maintain the availability and log integrity in the IoT cloud ecosystem. However, the BCFL framework solves the challenges of evidence acquisition in the cloud, such as multi-tenancy issues and CSP service level agreements through Blockchain DLT.

Celesti et al. (2019) proposed to fill in the gap the CSA guidelines highlighted through a secured and resilient Cloud-to-Edge ecosystem communication that has security in mind. The proposed framework intends to improve the current cloud communication framework such as a Message Oriented Middleware that uses Instant Message Protocol (IMP) to facilitate performance in a CoT ecosystem. They proposed to use

a single case study approach to carry out an experimental test on a real testbed to justify the need for a secured middleware in the CoT ecosystem.

Putz, Menges, and Pernul (2019) proposed to solve the problem of log management in the cloud ecosystem by integrating a Permissioned Blockchain to store and maintain the log admissibility. They proposed a Blockchain logging mechanisms that enables them to evaluate the security and performance of logging. The proposed infrastructure will manage and maintain tamper-proof logs and eliminate the complexity of CSP or any third-party handling of logs in a non-secure way. BCFL framework is based on a containerised cloud environment that facilitates a security logging acquisition and evidence admissibility.

Nyaletey, Parizi, Zhang, and Choo (2019) proposed a Blockchain Interplanetary File System (BlockIPFS) approach that facilitates a more secure and improved data trustworthiness and transparent audit trail in the Cloud ecosystem (Awuson-David et al., 2021). The proposed approach will allow an incident investigation to trace back the incident's source using integrity cloud Blockchain as a Service. Table 2.5 demonstrates the novelty of this research (BCFL) in using Permissioned Blockchain (Hyperledger Fabric) to solve the current changes faced by cloud forensic investigators. The Table 2.5 compares the current work carried out by researchers in ensuring admissibility of log evidence and data security in the cloud ecosystem. The BCFL will contribute to solving the challenges faced by law enforcement agencies and digital forensic investigators in acquiring admissible evidence in the cloud ecosystem. Furthermore, it acts as a bridge that enables evidence acquisition in such a way that enhances GDPR compliance. Laurier, Kiehn, and Polovina (2018) uses REA2 to automatically create a third-party perspective that captures fairness and legitimacy of business traction at run-time. Additionally, it also maintains transparency in the supply chain ecosystem. While BCFL maintains real-time immutability, security and integrity of all transactions

logs in the cloud ecosystem.

Traditional digital forensic techniques may not capture and preserve evidence as to the same in a cloud landscape. According to Salah, Rehman, Nizamuddin, and Al-Fuqaha (2019) Blockchain transaction is stored with high integrity, resilience and trustworthiness that is tamperproof (Awuson-David et al., 2021). In an increasingly cloud-oriented society, having the ability to identify, obtain, preserve, and analyse potential digital evidence is valuable for business continuity and security (Awuson-David et al., 2021). Otherwise, whether reacting to a cyber intrusion, data breach, or in support of litigation, the ill-prepared establishment will find itself at a severe (and possibly expensive) disadvantage (CSA 2013). Distributed consensus, data consistency, and immutability of processed transactions will solve cloud evidence admissibility challenges (Awuson-David et al., 2021).

2.2.4 Hyperledger Fabric

Hyperledger Fabric Blockchain is a permissioned distributed ledger Blockchain with several Hyperledger families, as highlighted in table 2.7. Since its release in 2015, big players have adopted it in the cloud ecosystem such as AWS, Azure, IBM, Google and Oracle. An open-source permissioned Blockchain is designed with security, usability and resiliency in mind that maintains confidentiality and distributed ledger mechanisms under a consensus structure. Hyperledger mainly designs for the financial sectors, government institutions and businesses who want to maintain secure data transactions, with security and confidentiality in mind. It continues to maintain a strong world-class technical community and contribute to the development and improvement of the

Table 2.5: *Comparison of Using Blockchain to Secure Forensic Evidence in the Cloud Ecosystem*

Contribution	**C**loud Artifact Identification	**P**ermissioned Blockchain Cloud	**S**ecuring Data Integrity with Blockchain	**B**lockchain Cloud Trustworthiness	**T**amperproof Evidence	**G**DPR Cloud Forensic Challenges	Cloud Evidence Admissibility	**Scope of Literature Review**
Liang et al. (2017)	✗	✗	✓	✓	✓	✗	✗	2005 - 2017
Lone and Mir (2019)	✓	✓	✓	✗	✓	✗	✗	2004 - 2018
Cebe et al. (2018)	✗	✓	✓	✗	✗	✗	✗	2008 - 2018
Z. Tian et al. (2019)	✗	✗	✓	✗	✗	✗	✗	1997 - 2019
Putz et al. (2019)	✗	✓	✓	✗	✗	✗	✗	2002 - 2018
Noura et al. (2020)	✓	✓	✓	✗	✓	✗	✗	1997 - 2017
Celesti et al. (2019)	✓	✗	✗	✗	✓	✗	✗	1997 - 2017
Putz et al. (2019)	✗	✓	✓	✗	✓	✗	✗	1999 - 2018
Nyaletey et al. (2019)	✗	✓	✓	✓	✓	✗	✗	2014 - 2019
Our proposal framework	✓	✓	✓	✓	✓	✓	✓	2007 - 2020

(Awuson-David et al., 2021)

Hyperledger family. Hyperledger is developed in Java, Go and Node.js programming languages and uses the smart contract mechanisms known as chaincode to facilitate

trustworthiness in the Blockchain ecosystem (Androulaki et al., 2018).

It is vital to understand how the blocks and transactions are formed. In the formation, each block is arranged in a sequence and made up of established transactions. Then, the transactions are stored in a precise, controlled series (Awuson-David et al., 2019). In contrast, this is different from Permissionless Blockchain as the creation of the transaction and sequence is given, but not primarily done simultaneously or on the same computer. This is because the ordering of transactions and the execution of transactions are separated (Awuson-David et al., 2019). In Hyperledger Fabric, computers being used to operate the Blockchain can run in three different modes (node types) as follows:

Client: A client acts on behalf of users of the Blockchain and submits actions and events to the network as part of an application (Awuson-David et al., 2019).

Peer: Peers process the incoming transactions for validating and handling updating state changes as a result of transactions and chaincode execution. Once they have executed a transaction, they broadcast the result to the network so that an orderer can handle the transaction (Awuson-David et al., 2019).

Orderer: While peer nodes execute the transactions, the ordering service nodes decide the final order of events and thus determine the final set of events that will be written on to the next block of the Blockchain. It is important to understand that a single computer can support all three of these node types on a Fabric Blockchain, but this is not necessary (Awuson-David et al., 2019).

Table 2.6 highlights all the critical components of Hyperledger Fabric with its functions and how these components share responsibilities. Cloud computing offer customers on-demand shared resources in a virtual environment where human intervention is highly limited with benefits like cost efficiency, scalability, agility, convenience and elasticity as earlier stated.

Table 2.6: *Blockchain core Components and functions*

(Awuson-David et al., 2021)

Blockchain Component	Core Functions and Responsibilities
Shared Ledger	A Permissionless Blockchain such as Bitcoin enables transaction visibility by replicating a shared copy of the transaction to all participants.
Smart Contract	In a Blockchain ecosystem system, smart contract or chaincode is a business agreement programmed, embedded with the transaction record and executed with the rule defined by the business.
Privacy	Cryptography is used to maintain a secure transactions in the Blockchain Ecosystem. It facilitates a secure authentication and vitrification of all transaction. To maintain data privacy and security in the Blockchain, immutability, endpoint visibility and tamper poof logging mechanisms.
Trust	Blockchain establishes a trust mechanism by adding the ledger with appropriate confidentiality and ensures that all participants, transactions and assets are verifiable. It also maintains an immutable transaction audit trail of all events as trust is essential in any Blockchain Ecosystem.

However, this comes with the risks, threats and challenges identified by the CSA cloud threats list as highlighted in Chapter 2.1.2 (Awuson-David et al., 2019). Consequently, these challenges and threats have been taken advantage of by cybercriminals who have compounded the already existing challenges. With the steady increase of growth in the cloud computing ecosystem as many organisations have adopted this technology, there is a need to redefine a forensic acquisition process that is suited for cloud and preserved evidence privacy and integrity both on transits and storage (Awuson-David et al., 2019). This research will demonstrate that integrating Hyper-

ledger Fabric Blockchain immutable logging mechanisms into the cloud ecosystem will solve the challenges faced by a forensic investigator. However, the challenges faced by forensics investigators as demonstrated by the CSA in acquiring and preserved evidence in the cloud ecosystem has made it difficult to process cloud artefacts in a sound forensics manner, free from contamination and also admissible before a court of law (Awuson-David et al., 2019).

2.3 Docker and Hyperledger Fabric Stacks

Docker is a containerised tool developed to run the application. Its containerised environment enables the system developers to rapidly design and deploy a system application in an orderly manner by tapping into its libraries, and dependencies (Cachin et al., 2016; Tanwar, Parekh, & Evans, 2020). The Docker containerised ecosystem has reshaped the way applications and data are stored in the virtual environment, surpassing VMs with its speed, data management and lightweight of bandwidth usage. Top tech companies such as Microsoft, Apple, and IBM have tapped into the benefits that Docker brings since its release as a dotCloudplatform in 2013 under Apache 2.0 license, (Schenker, 2018).

It is useful to note that the BCFL framework uses the docker-comose-cli-YAML file to keep track of all our BCFL Blockchain node performance and log management, including viewing and querying of logs. In the BCFL framework, docker-engine was adopted to facilitate a remote connection with any choice operating system. Figure 2.10, emphasises a clear, detailed view of the Docker Hyperledger Fabric integration locally on a physical laptop. The Docker containerised image facilitates the smooth operating of BCFL (Awuson-David et al., 2021).

The Docker compose-e2e.yaml network configuration file is vital to ensure the

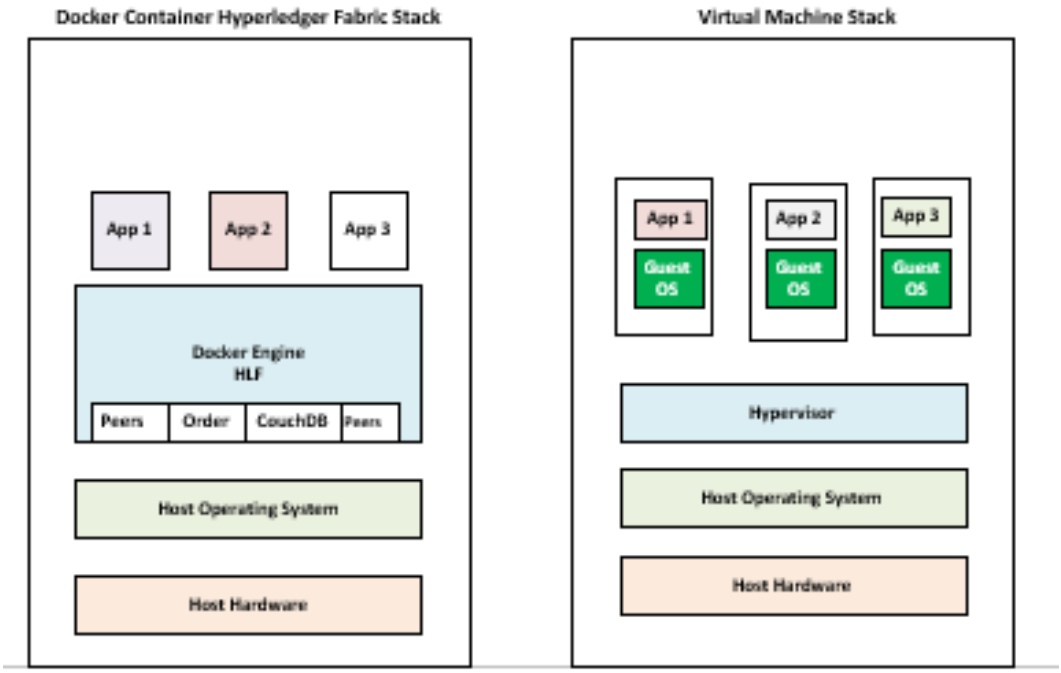


Figure 2.10: *Docker and the Proposed Hyperledger Fabric Integration* (Awuson-David et al., 2019)

startup of all the Docker containers in the BCFL environment. This file contains an essential program that uses Docker-Composer tool-sets to permit the level running of a Hyperledger Fabric-based network. The data itself depends on the statistical configured file base/peer-based.yaml and base/docker-compose-base.yaml, (Awuson-David et al., 2019; Stanciu, 2017).

2.3.1 GDPR Compliance

There is evidence that many organisations are still trying to figure out GDPR compliance regarding electronic data with organisations citing as the reasons challenges and lack of clarity due to lack of staff training. H. Li, Yu, and He (2019) demonstrate the need to support high tech organisations worldwide to reduce the complexity associated with the GDPR compliance, significantly the two leading superpowers, China and the

United States. The research highlights the need for organisations to strengthen their compliance regime and tap into the benefit of GDPR with their products and services.

2.3.2 Hyperledger Blockchain Types:

Hyperledger Blockchain: Hyperledger Blockchain: Hyperledger Fabric was designed by IBM and Linux foundation. Their main aim is to develop a Blockchain application that has modular architecture in mind. To facilitate this support, Hyperledger uses its consensus, distribution ledger, membership services, and leverages Docker containerisation mechanisms to facilitate the smart contract called chaincode. It has all the application logic of the system and supports a secure and forensic proof communication pathway through a distributed ledger mechanisms. This was the reason why Hyperledger Fabric framework was chosen for BCFL designed architecture. The next section will look at the design components of Hyperledger Fabric Architecture as demonstrated in Table 2.7.

Hyperledger Iroha: Iroha framework was designed mainly to support and facilitate Blockchain in mobile development projects. Although it is based on Hyperledger Fabric component, in addition Colu, NTT Data, Hitachi and Soramistu contributed to this development framework. The main design mechanisms of Iroha was a domaindriven C++ design as an algorithm called Sumeragi that is a new chain-based Byzantine fault-tolerant consensus.

Hyperledger Sawtooth: The major contributor to this framework is Intel, and they used mechanisms known as Proof of Elapsed Time (PoET). Their main aim of developing this framework is cross Block platform consensus mechanisms where a consensus algorithm can work in both Permissioned (Hyperledger) and Permissionless (Bitcoin or Ethereum) without any challenges. It is mainly for a diverse requirement in the Blockchain ecosystem as Sawtooth design has versatility in mind.

Table 2.7: *Hyperledger Framework Consensus Mechanisms and Types as highlighted by*

Acharya, Yerrapati, and Prakash (2019)

Hyperledg Framework	Consensus Mechanisms Type	Consensus Advantages	Consensus disadvantages
Fabric	Kafka / Permissioned Voting-base; Order is done by the leader.	Offer crash fault tolerance, good speed of finality	Not Byzantine and issue in the agreement may be vulnerable to malicious peer
Sawtooth	PoET Permissioned lottery-based, pluggable elect-ion strategy	Offers scalability and Byzantine fault tolerance	Moderate finality, might need to resolve forks
Indy	RBTF / Permissioned voting-based strategy. Only orders requested by a master instance are executed	Offers Byzantine fault tolerance, good finality speed.	All peers in the ecosystem are known and must be all connected; more peers will need more time to reach consensus.
Iroha	Sumeragi / Permissioned Server reputation system	Offer Byzantine fault tolerance, good speed of finality	All peers in the ecosystem are known and must be all connected; more peers will need more time to reach consensus

Hyperledger Burrow: Monax and Intel developed this framework in the first instance. However, it is a modular Blockchain and mainly focuses on client and is designed to support and facilitate Ethereum Virtual Machine (EVM). As a single binary, it is primarily designed with easy to deploy and configure and also speed and human engineering. Its smart contract mechanism is based on BEF consensus through Tendermint algorithm.

Hyperledger Indy: Sovrin Foundation was a core contributor to Hyperledger Indy framework primarily designed and developed to facilitate independent identity

on a distributed ledger. The primary core function of this framework was to provide digital identities rooted in the Blockchain.

2.3.3 Hyperledger Framework Tools:

Linux Foundation hosted five Hyperledger core component tools namely:

Hyperledger Explorer: One of the core web interface Hyperledger solution to which IBM, DTCC and Intel are the main contributors. The tool's primary focus is on invoke and query blocks associated with the transaction and network asset discovery details such as names, nodes, app, transaction, and chaincode with all the asset details stored in the Blockchain ledger.

Hyperledger Cello: Another Hyperledger tool that is associated with IBM as a major contributor to the project. The Cello project's main objective is to integrate Blockchain into cloud as an on-demand as-a-service model that will enable businesses to deploy Blockchain in the cloud infrastructure rapidly. This tool could support research that focuses on multi-tenancy issues in the cloud, such as mitigating the challenges the cloud forensic investigator faces to acquire admissible evidence in a cloud multi-tenancy environment.

Hyperledger Composer: This research is adopted as it has a significant component that will facilitate a secure and innovative BCFL framework that supports a tamper proof cloud forensic investigation. IBM and Oxchains are major contributors to this tool as it uses its smart contract to solve a business complex problem and increases productivity and efficiency. However, in this project, it will help reduce the complexities, risk, and threat associated with digital evidence acquisition in the cloud.

Hyperledger Quilt: Hyperledger Quilt has NTT data and Ripple as the main project contributors and is a Java application of the inter ledger protocol by Ripple, which is intended to transfer transaction values across distributed and non-distributed

ledgers (Baset et al., 2018). Hyperledger Caliper: This tool enables Blockchain participants to predefine the performance metric by using a standard case study scenario. It supports all the above mentioned Hyperledger frameworks.

Table 2.8 emphasises the comparison of permissioned and permissionless Blockchain algorithm with the highlight of their functions and the usage of each in the Blockchain platforms.

Table 2.8: *Highlighting a Comparison of some of the Consensus Algorithms, Blockchain Types and their Use Case*

Private and Permissionless	Private and Permissioned	Public and Permissionless	Public and Permissioned
Only Authorised Participants, High level access right are given to few	Closed and Only Authorised Participants	Open, but decentralised	Open, but maintain Confidentiality
Access level rights to all participants parties, but only few have high level access	Only superusers or consortium can make changes	Anyone has right to run node, read transaction	Everyone have reasonable level of user rights, but only few participant can validate and read transaction
For instance, Hyperledger Fabric	For instance, Hyperledger Fabric and R3 including Corda	Litecoin, Ethereum and Bitcoin	Ethereum
Consensus Mechanisms-PBFT	PBFT and FBA	Pow	PoS, PoA
Use Case-Cloud Forensic Logging, Supply chain, Trader, Government record	Use Case Tax returns	Use case Cryptocurrency	Use Case secure online Voting

2.4 Summary

In this chapter, a review of related literature in computer and cloud forensics was carried out, including reviewing the recent works on Blockchain cloud integration. The chapter highlights different definitions of cloud computing and reviews recent papers on cloud forensic logging challenges as pointed out by the cloud security alliance (CSA) and NIST. Furthermore, this chapter looks at the Hyperledger fabric family's key components and demonstrates why Fabric was adopted for this research project.

Chapter 3

Methodology

3.1 Methodology

The chapter presents two different methodological approaches to define the Blockchain Cloud Forensic Logging (BCFL) framework. Design science research methodology (DSRM) will be used to investigate the project contexts and artefact that associate with tamperproof Cloud forensics evidence acquisition in the cloud ecosystem. In contrast, the Discrete Event System Specification (DEVS) model approach will be used to simulate contexts and artefact in control innovative hybrid virtualised lab environment. A comparison of the proposed methodology and other related methods of Blockchain cloud-integrated were highlighted. Furthermore, integration of the DSRM and BCFL framework was analysed in this chapter.

There are two ways of acquiring knowledge, one through reason, the other by experiment (Bacon, 2010).

3.2 Introduction

The project research methodology is the process of presenting a detailed account of the distinct challenges through a set of stages and steps. This methodological step delivers flexibility in designing and justifying the forensic evidence acquisition method that will

solve the current challenges confronted by forensic examiners in the cloud ecosystem. It certifies that no evidence integrity compromise is ignored and that all critical information is is understood before designing the system. According to Zelkowitz and Wallace (1998), describes computer science approaches into four categories: scientific method, engineering method, empirical method, and analytical method (Awuson-David et al., 2019).

However, Pappa and Freitas (2009), considered computer science scientific research method to have three classifications: theoretical, experimental, and simulation. Given the nature of this research, it was challenging to find a suitable research methodology. As a result, a combination of the Freitas (2009) DEVS model approach and the Peffers, Tuunanen, Rothenberger, and Chatterjee (2007) design science research methodology was adopted to guide this research work. The next two sections will provide an overview of the design science research methodology and the implementation procedure to be used in completing this research. Design science is the design and investigation of artefacts in context. The artefacts we study are designed to interact with a problem context in order to improve something in that context (Wieringa, 2017) and (Awuson-David et al., 2019).

3.2.1 Design Science Research Methodology

Design science research methodology (DSRM), which is popular in disciplines such as engineering and architecture, focuses on creation: "how things ought to be in order to attain goals and to function" (Awuson-David et al., 2021; Simon, 2019). The design framework aims to change existing situations into preferred ones (Simon, 2019). DSRM creates artefacts: "something created by humans usually for a practical purpose" (Awuson-David et al., 2021; Dictionary, 2010). March and Smith (1995) differentiate among four different types of artefacts: concepts, models, methods, and instantiations.

Two essential characteristics of DSRM artefacts are relevance and novelty. First, an artefact must solve a significant problem: i.e., be relevant. Second, it must differentiate DSRM from routine design, Geerts (2011); Wieringa (2014) suggests that DSRM should address either an unsolved problem uniquely and innovatively or a solved problem more effectively or efficiently (vom Brocke, Hevner, & Maedche, 2020).

Evaluation of design artefacts and design theories is a central and critical part of DSRM (March & Smith, 1995), and (A. R. Hevner, March, Park, & Ram, 2004), and (Awuson-David et al., 2021; Vaishnavi, Kuechler, & Petter, 2004). In DSRM, evaluation is primarily concerned with the evaluation of design science outputs, including Information Systems (IS) Design Theories (Gregory & Muntermann, 2014) and design artefacts (Awuson-David et al., 2021; March & Smith, 1995). Together with 'build', assessment is one of two core concept that constitutes DSRM (Awuson-David et al., 2021; March & Smith, 1995). The DSRM is a model developed by Takeda, Veerkamp, and Yoshikawa (1990) and involves the production of exciting new and actual knowledge. It involves the production of an artefact, which in computer science research could be output in the form of an adaptation, invention, improvement, or routine design,(Awuson-David et al., 2021; A. Hevner & Chatterjee, 2010).

There are different implementations of DSRM in the literature, and however, for this research, we adopt the model developed by Peffers et al. (2007). The DSRM procedure established by Peffers et al. (2018) divides a computer science research into the following stages; identify the problem, define objectives of the solution, design and development, demonstration and evaluation, and communication (Awuson-David et al., 2021). In DSRM, there are two significant characteristics: designing an artefact that advances something for participants and empirically examining the performance of an artefact in a context, (Awuson-David et al., 2021; Wieringa, 2014). Wieringa described design activity and advised the importance to know the social context of stakeholders

and goals of the project, as this is the source of the research budget and the destination of useful research results (Awuson-David et al., 2019). "For the investigative activity, it is vital to be familiar with the knowledge context of the project, as one will use this knowledge and also contribute to it" (Wieringa, 2014).

3.2.2 Discrete Event System Specification (DEVS)

Bernard P.Zeigler introduced the standard Discrete Event System Specification (DEVS) in 1976 (Awuson-David et al., 2021; Zeigler, 1976). It originated from the field of systems model where it is used to model and simulate physical systems such as the cloud ecosystem. Essential to its original state is the notion of a discrete event simulation where models are updated in specified time intervals. The state changes of the model only transpire promptly at the time of scheduled discrete events in a continuous time frame. DEVS is an executable description which is based on time extended, determinate state mechanisms. The formalism also supports hierarchical, modular model construction (Praehofer, 1991; Schulz, Rozenblit, Mrva, & Buchenriede, 1998) and (Awuson-David et al., 2021; Zeigler, 1976, 2014).

DEVS methodology is a scientific methodology that looks at system design and simulation which has been derived from engineering research. It offers a useful Object-Oriented Modelling and Simulation (OOMS) method for evolving, simulating and assessing composite business development models (Nidumolu, Menon, & Zeigler, 1998). The framework of Business Process Reengineering (BPR), as highlighted by Warren, MacArthur, and Crosslin (1994), suggested that system simulation can deliver the firmness to eliminate some of the presumptions from BPR. Furthermore, present a section of neutrality which simulation is suitable for tracing complex communications and circumventing of enhancing one component at the expense of overall network efficiency.

The model-based design approach uses a formal language to describe system de-

sign models. Such design models then simulated to predict the system's performance in real world scenarios (Awuson-David et al., 2021). In particular, the model-based formal design using DEVS differs from other approaches because of its use of a unique integrative framework that can address most of the issues faced in complex system design, particularly cloud computing (Awuson-David et al., 2021). The DEVS formalism has become one of the most critical Image results for modelling and simulation theory.

In this chapter, we analyse and demonstrate how DEVS as a formalised approach can aid a secured and trusted Blockchain cloud system design in a very flexible and efficient manner. Note that DEVS, real-time DEVS (RT-DEVS), and distributed DEVS will continue to be among the most powerful tools for aiding sophisticated system design, in particular, real-time distributed computer systems including VEand P2P-based systems such as Blockchain cloud ecosystem (Wainer & Mosterman, 2016).

The main contributions of the Design Science Research Methodology are to:

- Demonstrate how the design science research methodology approach will be used to investigate the project contexts and artefacts and validate the results.
- Demonstrate how DEVS can be used as a fundamental system design tool for Blockchain cloud forensic logging that aid and validate the integrity and admissibility of logs in the cloud ecosystem.
- Demonstrate how DEVS is used for design validation for complex real-time computer systems, such as distributed virtual environment, distributed real-time P2P systems, and distributed simulation systems.

3.2.3 DESIGN SCIENCE RESEARCH METHODOLOGY

In our approach to the research problem, we adopted the DSRM of Wieringa (2014) and adapted it to suit a Blockchain cloud forensic logging environment that enables ad-

missible evidence acquisition in the cloud ecosystem, as shown Figure.3.1. The DSRM model developed by Takeda et al. (1990) is a research design methodology which aims to produce original new and real knowledge (Awuson-David et al., 2021; Peffers et al., 2007). It consists of the output of an artefact, which in computer science research could be output in the form of an adaptation, invention, improvement, or routine design (Awuson-David et al., 2021; Dresch, Lacerda, & Antunes, 2015). For the investigative activity, it is vital to be familiar with the DSRM's knowledge context as it establishes this research's novelty. In our DSRM design, an initial phase is to set up a test environment that integrates Blockchain into the cloud ecosystem to simulate an experiment in a VMware virtualised environment. VMware workstation 15.0 virtualised platform was used to simulate a cloud environment that hosts a guest Linux Ubuntu 16.04 operating system (Awuson-David et al., 2021). It was considered that running an experiment on VMware application instead of Microsoft Azure, or Amazon Web Services (AWS) platform or Google Cloud Platform (GCP) would enable us to get accurate final log artefacts' experimental result as it is challenging to request log information from the cloud ISP's (Awuson-David et al., 2021).

Next, the decentralised ledger provides auditable transactions logs that maintain a high level of evidence, integrity and immutability (vom Brocke et al., 2020). The investigators can inspect all log transaction in the BCFL network terminal, which list the connections to the chaincode on the peer with timestamps. In the social context of the DSRM, the Hyperledger Fabric Blockchain was extended with the BCFL forensic layer algorithm to secure logs evidence and facilitate a new cloud digital evidence acquisition process, which represents the stakeholders' goals (Awuson-David et al., 2021). The method preserves log evidence integrity, trust and transparency in the cloud ecosystem.

The stakeholder provides funding to achieve the goal of the investigation. The in-

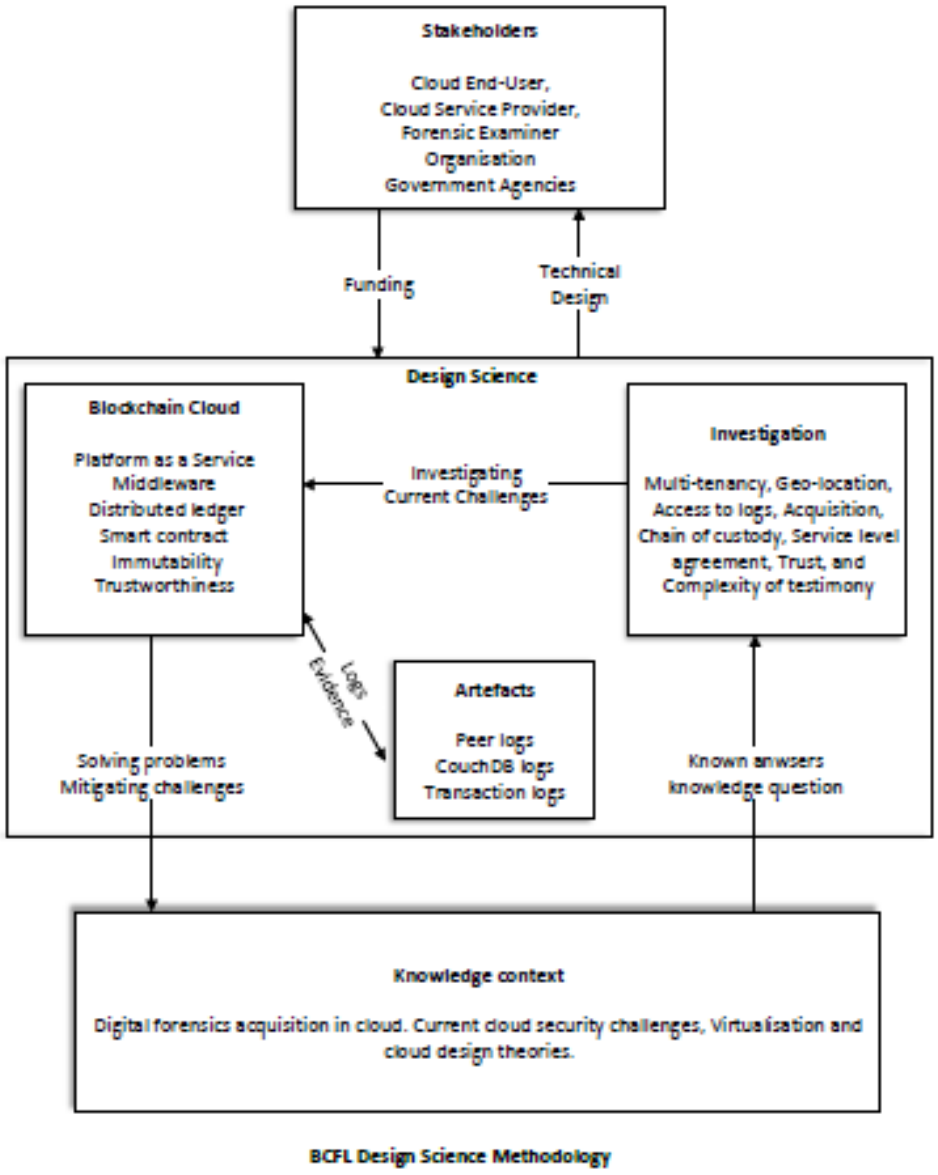


Figure 3.1: *BCFL used Methodology (DSRM)*
(Awuson-David et al., 2021)

vestigative box of the methodology captures all the challenges faced by cloud forensics investigators in acquiring admissible log evidence in the cloud ecosystem (Awuson-David et al., 2021). The knowledge context section is the main contribution of the research, where we propose a permissioned Blockchain cloud Forensics Logging frame-

work that defines a new approach of a Blockchain log digital evidence acquisition in the cloud. In addition, the BCFL section is the proposed framework that uses Blockchain mechanisms to answer and solve the cloud digital forensic acquisition challenges (Awuson-David et al., 2021). The difficulties in acquiring admissible evidence in the cloud ecosystem are immensely challenging from multi-tenancy factors; however, geopolitics add to this complexity. The main advantage of using the DSRM approach is that it can correct defects during our framework's design and testing phase. The artefacts section is where all the admissible logs can be accessed by the forensic investigator, cloud service providers and customers with user rights Awuson-David et al. (2021). Finally, the stakeholder's section addresses the design goals and budgetary issues.

Methodology Comparison

In Table 3.1, we compare different methods with DSRM that enables cloud forensic investigators to acquire admissible log evidence from the cloud ecosystem (Awuson-David et al., 2021).

In a traditional network, log evidence has played a vital role in enabling the digital forensic investigator to systematically reconstruct an electronic crime source by establishing admissible evidence to solve a case (Awuson-David et al., 2021). However, acquiring log evidence in the cloud ecosystem is problematic due to the cloud's design nature. It is also problematic to maintain evidence chain of custody in the cloud due to lack of trustworthiness among cloud actors and the designed nature of cloud as highlighted by the NIST Cloud Computing Forensic Science Working Group (NCC FSWG), and Awuson-David et al. (2021). Furthermore, the cloud hypervisors' nature has made it complicated for log evidence to be admissible in the cloud ecosystem that stakeholder could rely on. The Blockchain DSRM provides log transparency and trust-

Table 3.1: *Mitigating Digital Evidence Challenges in the Cloud Methodology Comparison*

Contribution	Methods	**B**lockchain-Based Cloud Logging	**T**ransparency and Trustworthiness	**R**ecovering Evidence Deleted from the Cloud	**C**hain of Custody	**T**imestamp Synchronisation	**M**uti-Tenancy and Geo-locations	Year
Liang et al. (2017)	Provchain:A Blockchain-based data provenance architecture	✓	✓	✓	✓	✗	✗	2017
Amato, Castiglione, Cozzolino, and Narducci (2020)	Semantic-based methodology for digital forensics analysis.	✗	✓	✗	✓	✗	✗	2020
M. Li, Lal, Conti, and Hu (2020)	Blockchain-based lawful evidence management	✓	✓	✓	✗	✓	✗	2020
Zawoad, Dutta, and Hasan (2015)	Forensics enabled cloud through secure logging-as-a-service	✗	✓	✗	✓	✓	✓	2015
Hudic, Smith, and Weippl (2017)	Security assurance assessment methodology for hybrid clouds	✗	✓	✗	✓	✗	✗	2017
BCFL	A DSRM Blockchain Cloud Forensic Logging	✓	✓	✓	✓	✓	✓	2020

(Awuson-David et al., 2021)

worthiness among the cloud actors. This has been achieved by using the distributed ledger mechanisms of Blockchain, the immutability of its logs and smart contract to preserve log evidence integrity and maintain chain of custody throughout the investigation process (Awuson-David et al., 2021). In the current cloud ecosystem, the forensic investigator depends on the CSP to access logs. There are no approved standards of approach that can validate logs provided by the CSPs which unfortunately calls into question the logs' integrity. Even when the provided logs have valuable information, it should be admissible. The BCFL approach validated the logs generated by the use case scenario (supply chain) after testing and simulation (Awuson-David et al., 2021). This simulation and testing process of the BCFL will be discussed in detail in the coming chapters.

DSRM Activity Theory Design and Development Stages

In Figure. 3.2 the simulation of the Blockchain cloud ecosystem in the first stage enables trustworthiness and transaction log integrity through the application of the Blockchain distributed ledger technology. The Fabric provides channel mechanisms that support and facilitate a more secure cloud ecosystem (Awuson-David et al., 2021). The second stage is the creation of a smart contract or chaincode that enables only authorised members with the contract to have access to application or logs. The third stage is the deployment of supply chain smart contract to all peers, and each peer then maintains a copy of the distributed ledger. This enables a more balanced agreement between cloud service providers and cloud customers (Awuson-David et al., 2021).

Furthermore, the fourth stage sees the smart contract's initialisation on the channel mechanisms that enable and maintain transaction log integrity and its immutability. The distributed ledger in the fifth stage is used to invoke and query the chaincode Awuson-David et al. (2021). It uses its chain of blocks to provide another security layer

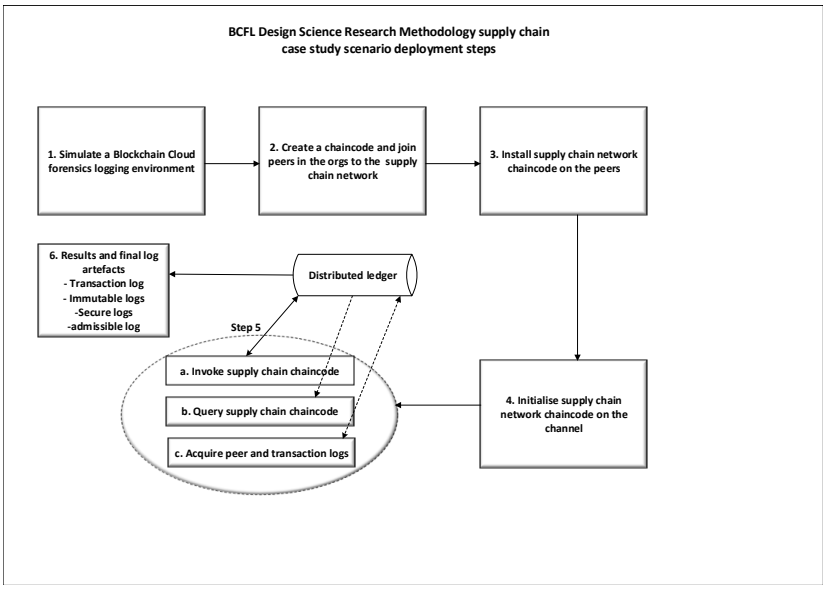


Figure 3.2: *DSRM Simulated BCFL Design and Development Phases* (Awuson-David et al., 2021)

through consensus mechanisms that facilitate transparencies and trustworthiness. At the final stage, the transaction was sent to the smart contract to update the ledger, an invocation (Awuson-David et al., 2021). Furthermore, it reads the current state of the ledger known as the query, as this will support and facilitate the acquisition of admissible log evidence in the cloud ecosystem (Awuson-David et al., 2021).

3.2.4 Research Main Goals

One of the main goals of digital forensics is to produce digital evidence that is admissible and free from contamination that can be presented to the court. It requires that the digital forensic process or techniques used are not contaminated in any way that the evidence or processes taken can be questioned. This requirement is normally described as 'forensic soundness' (Casey, 2011) and (McKemmish, 2008).

According to Casey (2001), several forensic procedures and principals should always be applied when dealing with digital evidence. These principals and procedures are

the integrity and preservation of the evidence should not be affected by the actions undertaken when securing and collecting evidence (Awuson-David et al., 2019). Figure 3.3 highlighted research project designed goal and justified the two-methodological approaches taken as to validate the BCFL framework and its architecture design in extending the Hyperledger Fabric to include a digital forensic layer capability. This methodological approach taken has answered the research objective and questions.

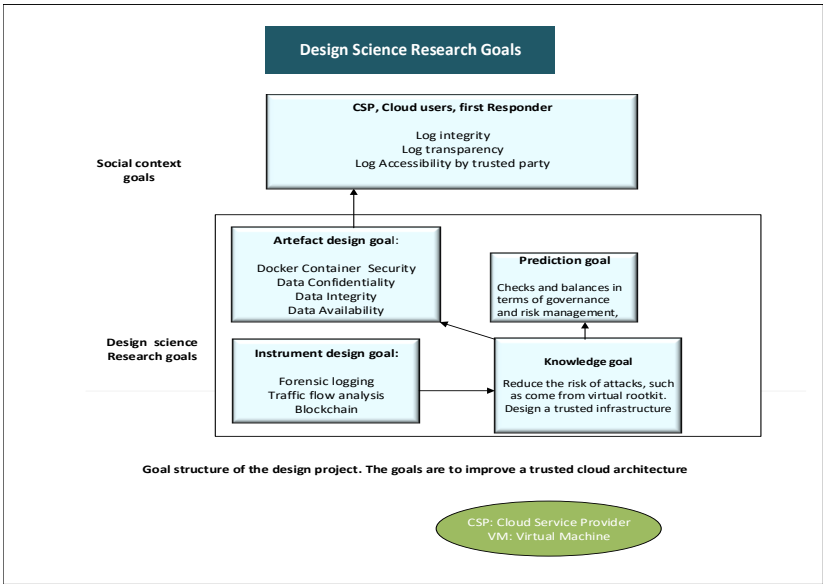


Figure 3.3: *Discrete Event System Specification (DEVS)*

Figure 3.3 demonstrates enhancement, principles and standards goals of BCFL framework. The goals are part of the validation process that facilitate Blockchain cloud integration. In this research, the high-level improvement ensures that our methodological approach answers our research question and facilitates an innovative design that preserves the integrity of log evidence in the cloud ecosystem. For instance, there will be a prediction of how an artefact will interact with a problem context and how a problem would evolve and increase the threat level if the defined prediction was resolve. A prediction is a belief about what will happen in the future, which will turn out to be either true or false, (Awuson-David et al., 2019; Wieringa, 2014).

BCFL containerised cloud simulation environment provides a real cloud test environment combined with real cloud virtualised components such as VM and containerisation. This virtualisation approach enabled us to test the experimental artefacts and ascertain an accurate result of the simulated case study, which will be discussed further in the next chapter. The approach bridges the simulation gap between conventional cloud environment managed by a third party such as Google, Amazon or IBM. The real time-simulation technique supports real-time testing of all BCFL framework features. It interacts with the real-world system and virtual systems such as REST Servers.

3.3 Summary

This chapter has looked at the two methodological approaches adopted, namely Design Science and DEVS to facilitate the design and simulation phase of our research project. The project artefacts and context were also highlighted, and the stakeholders identified in the design process. The design goal is to acquire tamperproof forensic evidence from the Blockchain cloud ecosystem in order to solve the acquisition and investigative challenges faced by cloud forensic investigators and stakeholders such as cloud service providers, cloud customers and government institutions. The knowledge goals integrating Hyperledger Fabric Blockchain into the cloud ecosystem by inserting a forensic layer and using the DLT, immutability, smart contract, and logging capability of Blockchain to acquire and preserve digital evidence in the cloud ecosystem. The next chapter will present the architecture and initial testing phase of this project.

Chapter 4

Framework and Implementation

In this chapter, we introduce and analyse the BCFL architectural framework with its components. Next, the integration and implementation of BCFL framework in the cloud ecosystem. Furthermore, an initial single case study scenario will be introduced in this chapter, followed by testing and validating the case study through real-time simulation. The BCFL system's performance will be presented in this chapter to show and validate its component metrics such as Docker containers. Initial real-time simulation and testing were conducted and will be discussed in this chapter with the initial admissible log evidence result obtained.

A system is a collection of elements that interact and form a whole. (Wieringa, 2014)

4.1 Introduction

The framework of this research defines how the BCFL was integrated into a cloud network. It also indicated consensus mechanisms, hardware, and web API interrelationships and functionalities within the designed framework. Additionally, the research architecture defined the structure of Hyperledger Fabric with a set of rules and methods of including the consensus algorithm that describes the functionality of each Hyperledger Fabric Blockchain layer. More details of this research novel contribution on

Hyperledger Fabric forensic layer is highlighted in the coming section.

However, Research has shown that many organisations are swiftly moving their IT infrastructure into the cloud ecosystem (Benlian, Kettinger, Sunyaev, Winkler, & EDITORS, 2018). Cloud digital forensic investigation incidents involve cloud service models and their application services in civil and criminal litigation. However, on many occasions, these incidents are not online related. For instance, cloud service components could be used as a vehicle to facilitate a physical crime (e.g. storage of indecent images of children, human trafficking, and materials used to carry out terrorist attacks) due to the cloud's online nature, (Farina, Scanlon, Le-Khac, & Kechadi, 2015; Manral et al., 2019) and (Cinque et al., 2018; Guo, Jin, & Shang, 2012). As demonstrated by the CSA, cybercriminals have changed their mode of attacking a cloud system. However, they have adopted a highly skilled attack pathway to compromise cloud infrastructure, and forensic investigators must play catch-up. Multi-tenancy, geo-location and political issues have made it difficult and an uphill task for forensic investigators to acquire digital evidence in the cloud ecosystem (Cinque et al., 2018; Manral et al., 2019). The literature review undertaken in Chapter 2 presented the challenges faced by forensic investigators in acquiring digital evidence from the cloud ecosystem.

Some of the evidence can span multiple geo-locations. This chapter presents the proposed BCFL framework, architecture, elements, and partial validated results followed by a discussion and analysis of the test result. This research has designed an architecture called BCFL using the Hyperledger Fabric Blockchain (Awuson-David et al., 2021). Our novelty and contribution focus on using tamper-proof logging capability as digital evidence in the Blockchain cloud ecosystem. However, we are using Hyperledger Fabric and Docker containerisation components to deploy a Blockchain Cloud Forensic as a service in the cloud ecosystem. Furthermore, this service is a hybrid platform that ensures cloud log evidence is secure, trustworthy and immutable. Due to

the distribution ledger capability of Hyperledger Fabric, all participating organisations are able to view the secured transaction log during an incident (Awuson-David et al., 2021).

The issues of multi-tenancy, politics and cloud service level agreements have made evidence acquisition more complicated in the cloud ecosystem. The literature review in Chapter 2 highlights previous studies carried out by researchers in developing a model and ensuring a tamperproof log evidence acquisition in the cloud ecosystem, (Pichan et al., 2015). However, previous research has ignored the importance of how to maintain the integrity of logs, both in transit and storage. Although forensic examiners can use many automated analysis tools for legal examination purposes, they are not sufficient as there is no foolproof method for distinguishing between fake traffic produced by cybercriminals and legitimate traffic. A human verdict is also critical because, with automated traffic analysis tools, there have invariably been chances of false positives and log tampering. Forensic examiners need to perform forensic log acquisitions in the cloud ecosystem that determine the character of an attack and then need to have the capability to use the log evidence to reconstruct events before and afterwards, (Martini & Choo, 2014; Ruan et al., 2013) and (Awuson-David et al., 2021; Zawoad & Hasan, 2013).

4.2 The Impact of the Proposed Architecture

The landscape of cloud computing not only presents doubt over where data are situated, but it also gives rise to confidentiality and regulatory compliance problems (Awuson-David et al., 2021). As a result, traditional digital forensic techniques may not be able to capture evidence, (Quick et al., 2013). According to Hill et al. (2018) and Bashir (2018), the decentralised ledger of Blockchain technology provides auditable

transaction logs, making legal disputes less likely and simpler to settle. In an increasingly cloud-oriented society, the ability to identify, obtain, preserve, and analyse potential digital evidence is increasingly important. Blockchain, by design, creates an immutable, permanent, and replicated record of the data. A holistic approach is utilised in the experimentation, integrating a Hyperledger Fabric cloud ecosystem to establish an in-depth analysis of log preservation. Fabric's advantage in a single case study is that it is highly modular, designed for coordination across companies, and the private channel feature enables secure functionality that suits our research experiment. The Hyperledger Fabric has a composer component API (Playground and REST Server) that facilitates our BCFL algorithm that architectures Blockchain applications. Despite cloud economic benefits and success to businesses and organisations, log evidence acquisition is associated with some problems such as cloud SLA Multi-tenancy, Geolocation and Trustworthiness as demonstrated by results from earlier studies.

Although a considerable amount of literature has explored how to preserve log evidence in cloud see (Chapter 2), these studies have not established a systematically investigative approach to demonstrate how to maintain the integrity of logs in the Cloud ecosystem. To this note, our chosen single case study is intended to answer the first three of our research questions. 1) How can system logs (network, process and application) be extended to include additional information that will help solve recognised problems in cloud forensics, including the issues of dependency chains and time synchronisation? 2) How practical would a Blockchain approach be for capturing and storing extended cloud logs? 3) What relevant services are needed to use the extensive log evidence effectively, and how can these be built? The purpose of the experiments undertaken for this research was to provide information on how cloud log evidence can be acquired, preserved and stored securely without being compromised or tampered with in a cloud ecosystem, in particular, how Docker containers store and maintain logs

when switched off (Awuson-David et al., 2021). The first part of the chapter focuses on the three Research Objectives, describing Hyperledger Fabric Blockchain architecture. This will then be followed by our research architecture and framework, along with our developed logging layer architecture. Thirdly, the description and components of Fabric used to maintain log integrity. Our single case study demonstrates a partial result to answer the research questions. Finally, we will examine how Docker images handle transaction logs in a chaincode Hyperledger Fabric Blockchain environment (Awuson-David et al., 2021). The discussion then begins with an overview of BCFL to determine its functions and features.

4.3 BCFL Blockchain Cloud Network

In Chapter 3, DSRM highlighted the steps taken to evaluate our BCFL framework with regard to maintaining log integrity and transparency in the cloud ecosystem. Hybrid cloud service was selected as the service type because it constitutes the basis of other cloud service types (Baranwal & Vidyarthi, 2014). A simulated cloud network was used to enable comprehensive testing of the experiment and demonstrate how BCFL facilitated the acquisition of an admissible evidence in the cloud environment. The advantage of using computer simulations is that it allows us to have a holistic view of Hyperledger Fabric logging mechanism and highlight how it maintains the immutability of log data both in transit and storage. "An excellent logging system should have flexible log collection capabilities, secure persistent storage capabilities, a highly available distributed architecture design, secure and efficient log query function, and the ability to self-repair errors" (H. Tian et al., 2019).

The first measure in the deployment of the experiment for this research was the creation of a cloud ecosystem in a virtual environment using Docker containerisation

instead of VMs (Awuson-David et al., 2021). Docker containerised technology has changed the landscape of virtualisation with speed to deploy an application (Awuson-David et al., 2021; Pahl & Lee, 2015). It uses little bandwidth to run an application compared to other virtualisation types. The container could be hosted on a personal laptop, QA cluster, or client site (Turnbull, 2014). Running a Docker on our laptop environment is considered normal as the containers will integrate with all the BCFL framework components. The most crucial point to note is that there is the capability to move fast using the adopted DSRM and correct defects with continuous delivery of our Blockchain smart contract pipeline to optimise BCFL resource and maintain transaction log integrity. According to Kaewkasi (2018) showed in their report how Docker technology has reduced the total cost of ownership (TCO) of a business virtual platform. The report highlights a 66% reduction cost of a business virtualising their system with a Docker containerised technology rather than a virtual machine.

4.3.1 Distribution Ledger Technology

Blockchain technology is designed as a peer-to-peer (P2P) ecosystem that uses distributed ledger mechanisms to facilitate a transaction on agreed consensus by the participant parties. Blockchain uses its core components such as smart contract, cryptography, hashing algorithm, and Hyperledger channel and membership service provider mechanisms to aid data integrity and trustworthiness in the ecosystem. As the distributed ledger technology, an intermediary is eliminated among the participating organisations, and preferably a consensus mechanism is used to establish trust in the ecosystem. All the above mentioned data, application, and users' security mechanisms that Blockchain provides have made it resilient against cyber-attacks and data compromise. All transactions in the ecosystem is secured, with high-level encryption, hashing and transaction ledger immutability that enables admissible evidence within

the Blockchain ecosystem.

This virtual environment will enable the BCFL framework to be studied to explore and verify how artefacts are preserved and secured in a Blockchain-cloud ecosystem, both in transit and storage (Awuson-David et al., 2021). This understanding will establish our research aim of using Blockchain encryption, hashing and immutability mechanisms to preserve the integrity and transparency of logs in the Cloud ecosystem. Meanwhile, the main objective of agreeing on consensus, and harmonising the state across all nodes does not require all nodes to activate a chaincode but ensures the same rules are applied to nodes in the ecosystem. This chapter's drive is to accomplish the first, second and third of the Research Objectives as highlighted in Chapter 1.1.2.

Despite the complexity in the cloud platforms, such IaaS ,PaaS or SaaS, forensic examiners still need to follow forensics' sound investigative procedures. Thus, there is a demand for a more generic digital forensics method that will hold up the investigations of all cloud technologies. This investigation takes the frame of a computer-simulated single case-study to help understand how Hyperledger Fabric Blockchain maintains the integrity of log cloud ecosystems.

Blockchain is designed with immutability of its data in mind, and the capability to distribute data through a consensus mechanism (Awuson-David et al., 2021). A Blockchain ecosystem based on Hyperledger Fabric will comprehend endorsement policy verification, sequential policy validation of transactions in a block, and state validation and commit (Thakkar, Nathan, & Viswanathan, 2018). These are the central states in Hyperledger Fabric that enable it to facilitate a secure and trustworthy environment (Thakkar et al., 2018). Truong, Sun, Lee, and Guo (2019), emphasised the storing of personal data on a Blockchain network, which cannot be deleted or altered both may result in data processing challenges from the viewpoint of GDPR. However, Hyperledger, a permissioned Blockchain, mitigates the data processing challenges by

using its channel capabilities. Equally, it is vital to know by whom that personal data is handled and shared, and especially with whom it was shared. The channel and the private channel mechanisms of Hyperledger Fabric offer a method for defining the objects with which data is shared. This research project helps to solve these challenges and complexities that GDPR has added to digital forensics acquisition in the cloud.

The research will fill the gap that hinders forensics investigator evidence acquisition in the cloud. However, BCFL architecture will demonstrate how the GDPR compliance is managed by integrating Blockchain cloud forensic logging as a solution for tamperproof evidence acquisition. As demonstrated by Badr, Horrocks, and Wu (2018), through a governance process, peers can determine which other peers to share their data with and have a copy of their transactions. The channel private data feature in Hyperledger Fabric can potentially provide a mechanism to store personal data off the chain, defining with whom this data is shared while maintaining the data's integrity through cryptographic hashes stored in the Blockchain (Honar Pajooh, Rashid, Alam, & Demidenko, 2021).

4.4 BCFL Architecture

The BCFL architecture has made it easy for the participating organisation to share data in a secure environment and facilitate cloud digital forensic acquisition, as highlighted in Figure 4.1. However, suppose there is a cyber incident. In that case, there is a mechanism in place in BCFL architecture to acquire tamperproof evidence with no challenges that could compromise the investigation process. In addition, BCFL proposed cloud log evidence acquisition process that will provide answers to the research objectives and questions as highlighted in chapter 1. Furthermore, BCFL architecture is based on three layers, the top layer, the middle layer and the Hyperledger Fabric

Blockchain layer.

Top Layer: The top layer demonstrates the novelty and contribution of BCFL framework that answers our research question. This layer has three main components that support our architectural design, such as the transaction identity manager. This maintains the integrity of all the transactions with an accurate timestamp of each transaction that enables endpoint visibility and forensic readiness. Another component on this layer is the organisation data sharing that uses the Blockchain distributed ledger technology within all parties involved through consensus mechanisms. It facilitates log evidence acquisition process that is admissible. Digital asset and forensics auditing are where we applied the Blockchain data immutability mechanisms that support the forensic acquisition and a transparent chain of custody (S. Li, Qin, & Min, 2019).

The middle layer of BCFL architecture is a critical layer in the overall architecture. It has the application system function codes that arrange other several phases of BCFL architecture. As demonstrated in Figure 4.1, BCFL architectural framework uses an SDK which is obtainable in Node.js, java and Golang language to facilitate and helps our application to perform functions such as channel creation and joining, registering a claim, adding or enrolling new customer and maintaining the efficiency of chaincode processes. The next layer in BCFL architectural design is the middle layer. Here we have three components, the PaaS cloud ecosystem, supply chain transaction traceability analysis model, and angular with an SDK client. The first component in this layer is the cloud PaaS used to facilitate integrating the proposed BCFL framework with DSRM. The PaaS cloud platform was used to simulate a real-time cloud environment that will validate test results' authenticity. The supply chain transaction traceability analysis model was the next component used to design a single case study. It allows us to test a real-world scenario use case to validate our design architecture. Furthermore, the SDK provides a robust mechanism to contribute to the transaction

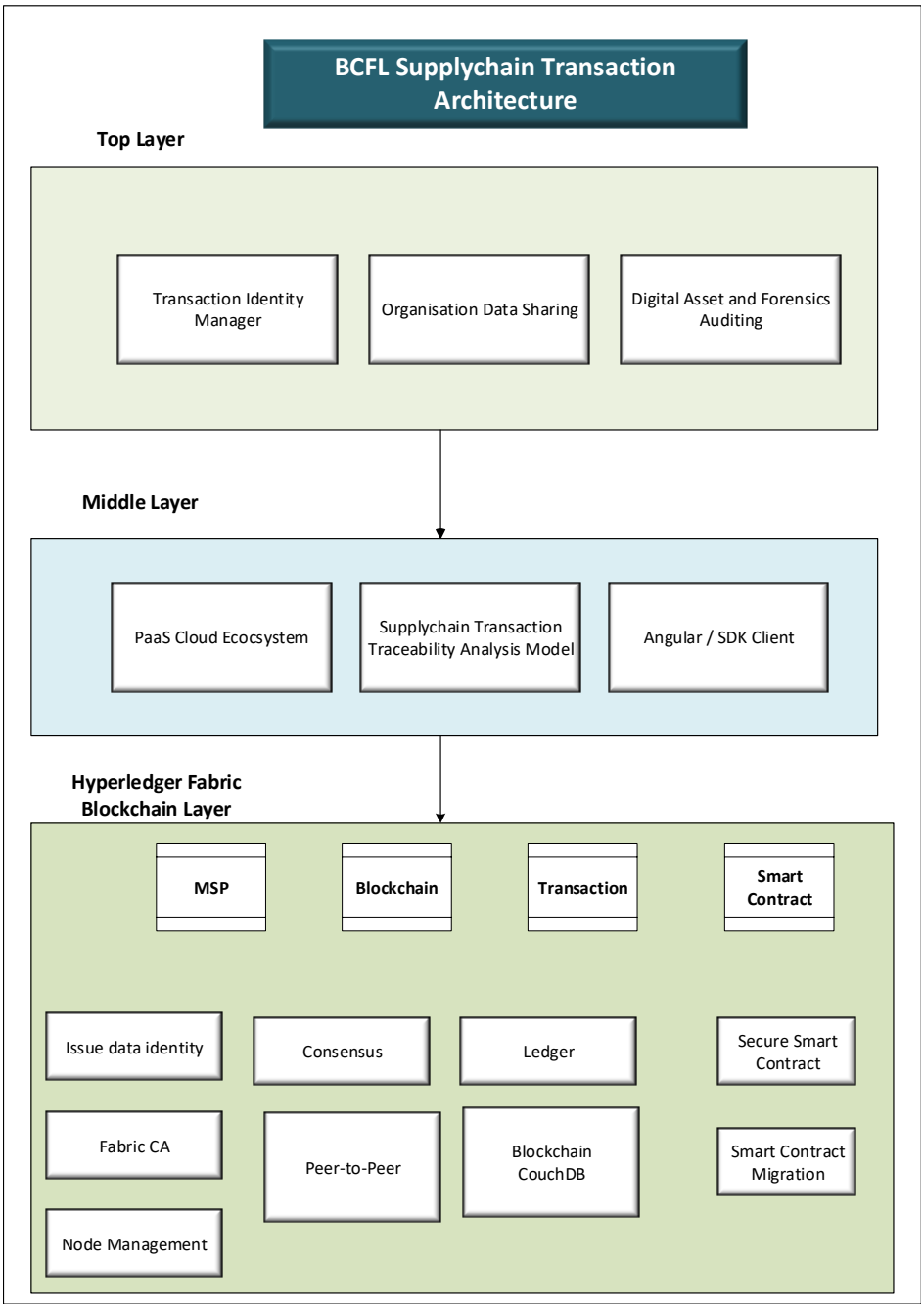


Figure 4.1: *BCFL Architecture*

and configuration-related events stemming from any part of BCFL Blockchain cloud network. The third layer of BCFL architecture is the Hyperledger Fabric Blockchain Layer, which provides data immutability, hashing, cryptographic and smart contract mechanisms that facilitate a secure data acquisition and preservation. This provides data integrity and trustworthiness in the BCFL supply chain ecosystem. The Hyperledger Fabric Blockchain layer of our architecture is responsible for the user interactions with an application that facilitates a service API comprising mostly of detailed application proficiencies, over administrative functions. For instance, the chaincode and channel operations may also be unsecured or vulnerable for system administrations. The design phase of our architecture will ensure a friendly user interface with a REST API server. As demonstrated by Baset et al. (2019) and Cui, Gaur, and Liu (2020) while any Blockchain application and the network is an agglomeration of diverse participants, this Hyperledger Fabric layer will often consist of multiple application stacks tailored to the different participants.

4.5 BCFL Framework

The Hyperledger Fabric supports a channel's concept, and a channel is a separate Blockchain that enables a secret transaction. For example, the Hyperledger Fabric Blockchain channel mechanisms will mitigate the problem of multi-tenancy in the current Cloud ecosystem (Awuson-David et al., 2021). Each Fabric client can be deployed to utilise a different communication channel in the Cloud ecosystem. The distributed ledger mechanisms play an indispensable role in resolving forensic investigators' current challenges in acquiring digital evidence in the cloud ecosystem. This is because the use of peers, who each store an immutable copy of the ledger, enhances transaction data integrity, immutability and trustworthiness at all times throughout the

transaction circle (Awuson-David et al., 2021). Another essential component of our BCFL is chaincode. BCFL Chaincode algorithm implements a business logic, which enables excellent communication between all the parties in our BCFL environment as highlighted in Figure 4.2 . The functionality is entrusted to client requests for them to invoke a transaction, provided they possess the correct Fabric membership service certificate (Awuson-David et al., 2021; Cachin et al., 2016). However, chaincode runs as an autonomous operation in the designed Docker containerisation BCFL environment, inaccessible from the Fabric network's other mechanisms.

The endorsing Fabric peer manages the lifetime of the chaincode, and the transaction requests (Androulaki et al., 2018; Awuson-David et al., 2021). In response to client requests, the chaincode queries and updates the ledger, and generates a transaction proposal using the Fabric SDK to the BCFL Blockchain cloud network. The endorser verifies the transaction and presents an encrypted signature to the fabric client back to the Fabric client. It also facilitates all the record of the read-write data block operation in the Blockchain. In addition, it has all the Blockchain records that were read or written during the transaction execution (Awuson-David et al., 2021). When the Fabric client accumulates enough transactions, it can then forward them to the orderer (Awuson-David et al., 2021). The orderer verifies the endorsement and, if successful, it then sends it to the peers. The peers view all the latest proceedings and make a decision on which ones are valid to add to our BCFL Blockchain cloud network. Finally, it informs all the Fabric clients of the current outcome of their proceedings. If there is sufficient endorsement, the transaction is added to the BCFL Blockchain cloud network (Awuson-David et al., 2021).

Also, due to Fabric's decentralised architecture nature, the categorisation of transaction execution can be governed and committed differently on the different components which include the endorsers, orderers, and committees as highlighted in Figure

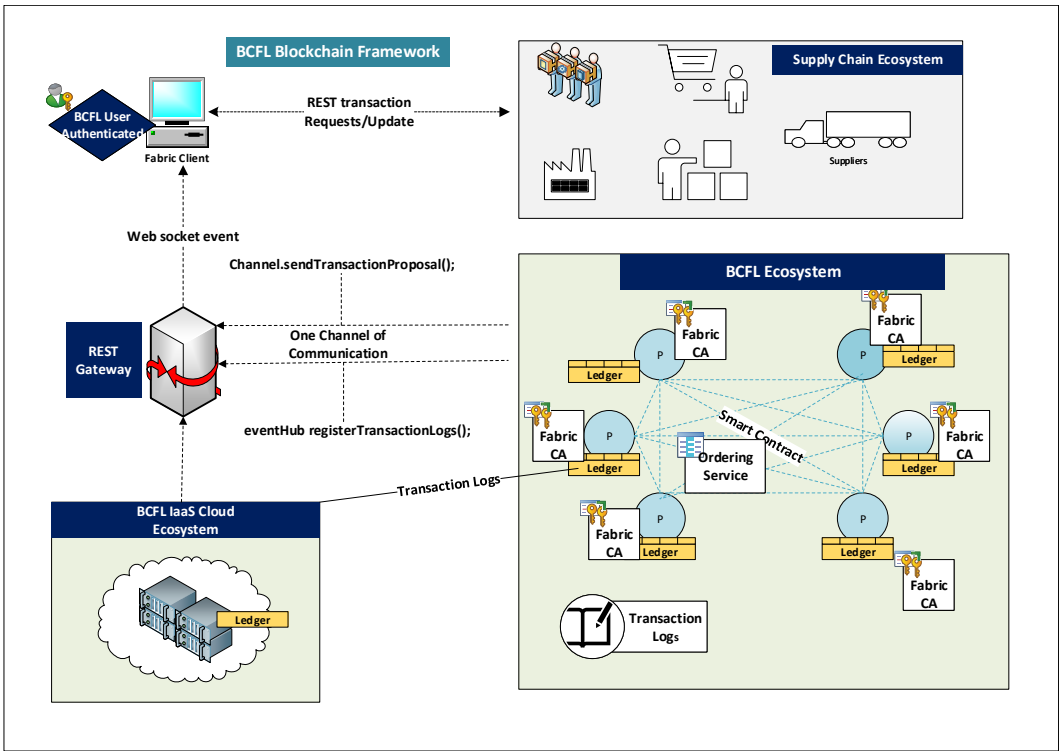


Figure 4.2: *BCFL Framework*
(Awuson-David et al., 2021)

4.3. In addition, this action then provides a timestamp between the deliberation and the assurance of the transaction, within which critical collision can happen. However, decentralisation also leaves the network in most cases, vulnerable to potential cyberattack, both intentional or accidental, for example a disgruntled employee intentionally damaging transaction sequences, or an untrained staff member accidentally changing the sequence of transactions.

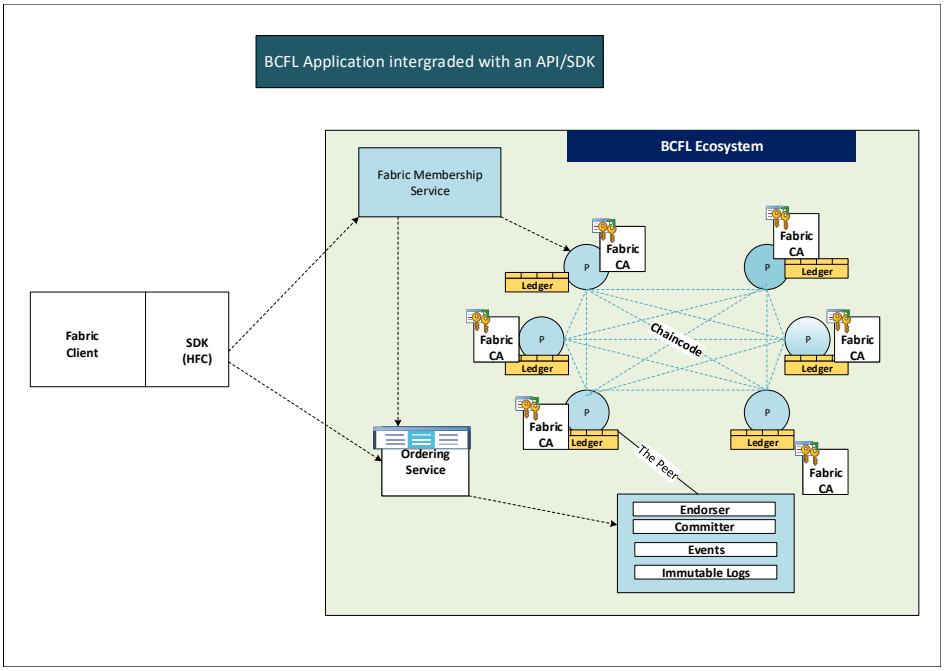


Figure 4.3: *API and SDK interaction with BCFL*

Orderer: While peer nodes execute the transactions, orderer nodes take a look at all the completed transactions and decide on the final order in which they are considered to have occurred in the Blockchain. The ordering service nodes determine the final order of events and thus select the last set of functions that will be written on to the next block of the Blockchain. As demonstrated in Figure 4.3, incoming traffic reaches the peers first, and then facilitates the transactions using Hyperledger chaincode encryption mechanisms, or smart contracts and then forwards successful transactions to the ordering service. Once received, the ordering service makes a decision on a final

order of transactions. The result of the transactions process sets is re-transmitted to peer nodes, which write the last block on to the chain for the final process cycle (Kuzlu, Pipattanasomporn, Gurses, & Rahman, 2019).

The Blockchain ledger is the structured, immutable record that records the current state of all transactions between nodes, peers and client operations on the network. It is the primary database where every single transaction that is performed on the Blockchain network is added. All channel transaction logs in our BCFL Blockchain cloud network can be accessed with a simple command-line instruction. There are two types of data in Fabric Blockchain ledger, a world state and a Blockchain transaction ledger. The state data is mutable but has a version number that keeps being incrementally updated when the state changes. The Blockchain ledger is an immutable and tamperproof piece of forensic evidence (Kuo, Kim, & Ohno-Machado, 2017; Ølnes, Ubacht, & Janssen, 2017; Wattenhofer, 2017).

Blockchain chaincode services: Chaincode is an application-level program stored on the ledger that makes use of business logic on how applications interact with the ledger. By doing so, it can run transactions that may modify the world state. As highlighted by Sousa, Bessani, and Vukolic (2018) and Bessani, Sousa, and Vukolić (2017) that transaction logic is written as chaincode (in the Go or JavaScript languages), and executes insecure Docker containers. The transaction transforms data, scoped by chaincode on the channel from which it operates.

Consensus: Consensus is at the heart of any Blockchain system. It also enables a trust system. In general, the consensus service allows digitally signed transactions to be proposed and validated by network members. In Hyperledger Fabric, the consensus is pluggable and tightly linked to the endorse-order validation model that Hyperledger Fabric Blockchain offers (Baset et al., 2018). Given the number of logs that are generated by cloud solutions, there are many tampering possibilities. While the tamper

detection solution can be used to detect any changes in the logs, we argue that the critical nature of logs calls for immutability (Gramoli, 2020; Sankar, Sindhu, & Sethumadhavan, 2017).

Figure 4.4 demonstrates the Hyperledger composer's architecture in facilitating BCFL framework integration with the cloud PaaS ecosystem. Composer is one of Blockchain Hyperledger Fabric's core components designed to enable a Blockchain developer to integrate another system component such as Docker into a given design architecture. It is mainly programmed in a JavaScript language. It is important to note that it is one of the most active tools in the BCFL framework design.

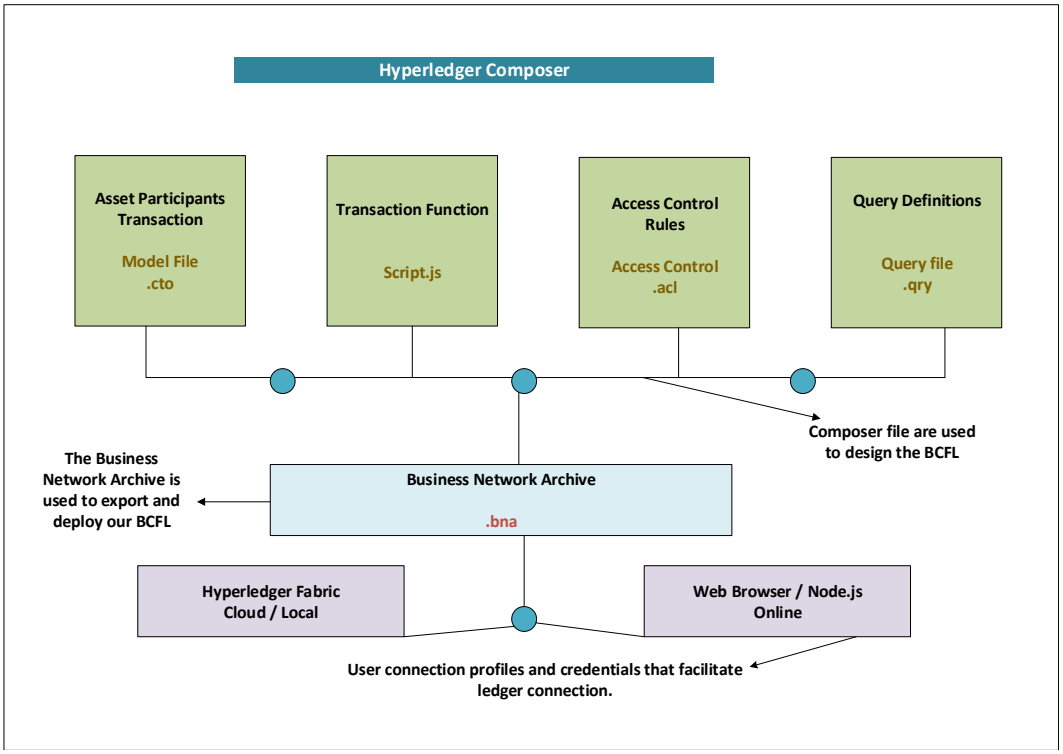


Figure 4.4: *Hyperledger Composer Components*

As mentioned earlier, that Hyperledger composer played a significant role in designing the proposed BCFL framework. The integration of composer enables us to define our use case BCFL participants, transaction type, which is acquiring admissible log in

the cloud ecosystem. To ensure log evidence integrity is maintained in the proposed BCFL framework, access-control rules were implemented. This will ensure only BCFL participants can access the log evidence according to their defined user rights. A model (.cto) file holds all of the preceding definitions in the BCFL cloud ecosystem. All BCFL user assets were defined as a real-world supply chain scenario object and participants with a unique Blockchain identity that maintained transaction traceability. In the BCFL framework, a script (logic.js) file implements log transaction logic and a package.json file containing a wealth of forensic metadata. It enables BCFL to maintain forensic readiness and solve the cloud forensic acquisition challenges mentioned earlier in chapter 1.2.

4.5.1 Hyperledger Fabric Chaincode Process Architecture

The Figure 4.5 illustrates the creation stages of our Blockchain cloud forensic logging (BCFL) application and transaction flow. The first stage is the chaincode creation which is the main engine house of Hyperledger Fabric design stage architecture. Figure 4.5 also demonstrates how Fabric peers, orderer, and CA and MSPs communicate using gRPC Remote Procedure Call as well as the procedure procreated by the peer to run the chaincode. This procedure distributes a service endpoint executing the JSON RPC 2.0 requirement for channel and chaincode operations. An instance of a Hyperledger Fabric Blockchain is referred to as a channel. This is an essential part of BCFL framework as each log transaction links to each other during chaincode creation.

The peers have a secure cryptographical mechanism that maintains the integrity of all the transaction logs. The first step in designing and deploying our framework is determining how many channels are required to build and implement our BCFL architecture for the insurance framework application. We will use one channel, which will preserve all the transaction logs' integrity and the history of the insurance claims

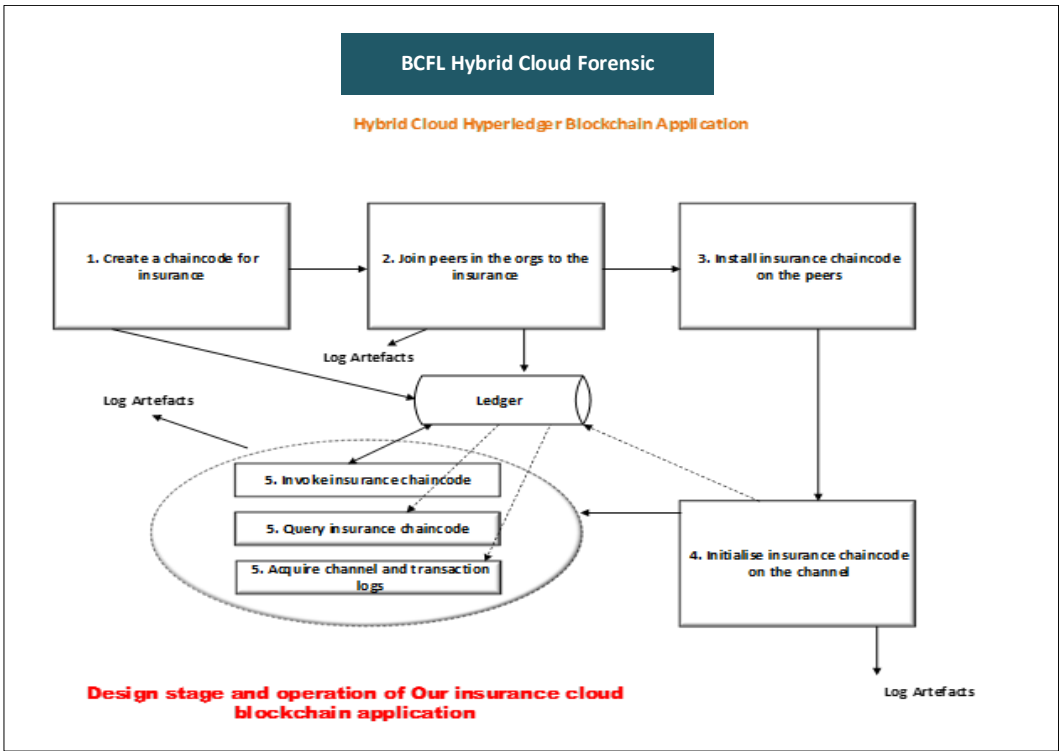


Figure 4.5: *BCFL Hybrid Cloud Chaincode Process Architecture*

carried out among the different clients or participants. As demonstrated by Androulaki et al. (2018); Cachin et al. (2016) and Gorenflo, Lee, Golab, and Keshav (2020), "a Fabric peer node may belong to multiple channels, which from the application standpoint will be oblivious to each other, but which help a single peer, run transactions in diverse forms on behalf of its owners (or clients). A channel may execute multiple smart contracts, each of which may be an independent application or linked together in a multi-contract application". In the BCFL framework, we design and implement a single-channel, single-contract framework application to test our scenario. In Chapter 5, we demonstrate it using our designed framework.

4.6 Cloud Log Evidence Processes

To acquire more information on service errors, one can easily browse log files for clues involving the specific request I.D. However, if the VM is ever shut down, then the entire system, including logs, can also be destroyed and never recovered (Lillard et al. 2010) and (Awuson-David et al., 2021). A forensics investigator uses a process to acquire digital evidence from a network by securing the crime scene such as the compromised computer or network (Awuson-David et al., 2021). Secondly, ensures to make a secured copy of the logs artefact with a continuous chain of custody to avoid contamination of the log evidence if used for legal proceedings. Thirdly, always preserves the acquired evidence in a forensically sound method that is admissible when required in the court of law. However, Figure 4.6 demonstrates how the BCFL enhances the current cloud digital forensic investigation process. These includes the planning, analyses and presentation phase of cloud Blockchain log acquisition.

System logs are an essential source of digital evidence which are accessible by digital forensics investigators in the traditional network. However, more challenging in the cloud ecosystem due to lack of ownership or full user rights between the CSP and cloud customers and this has lead to a lack of transparency, trust and integrity of log evidence (Khan et al., 2016) and (Ruan et al., 2011), (Awuson-David et al., 2021). There are different types of log evidence from the application logs, to system security logs and the audit logs play a crucial role in evidence source. For example, the system security logs can help the investigator reconstruct the crime scene and identify the particular suspect who attempted to compromise an authorised system, including his or her activity timestamps (Awuson-David et al., 2019). While the Application logs record activity created by the applications along with errors, warning and other functional faults of the applications (Awuson-David et al., 2021).

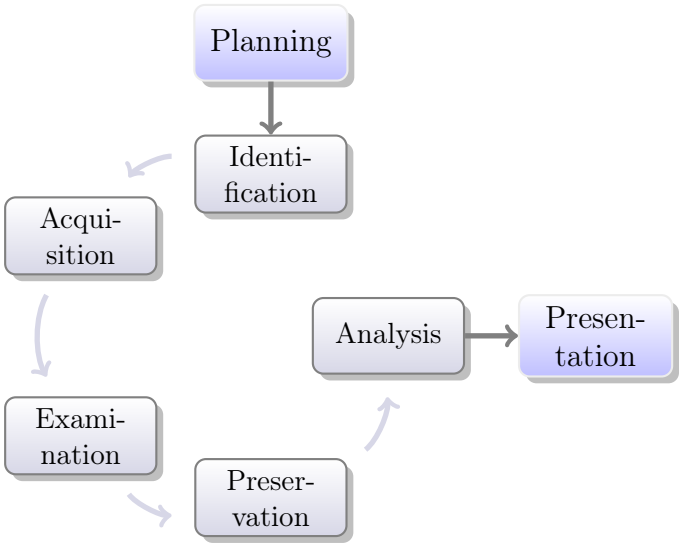


Figure 4.6: *Blockchain Cloud Forensics Log Investigation Process* (Awuson-David et al., 2021)

For instance, the investigators need ascertained imaging and chain of custody of evidence from the hypervisor or virtual machine layer. The forensics investigator uses a process to acquire digital evidence from a network by securing the crime scene such as the compromised computer or network (Awuson-David et al., 2021). Secondly, she make copies of logs, disks, other digital artefacts, and access logs as needed to support or refute the supposed criminal activity. Thirdly, she provides authenticated copies of full logs to the requesting lawyer or law enforcement as required. Fourthly, in the UK, they must adhere to the four principles of the Association of Chief Police Officers' (ACPO) digital forensic investigation guide. However, Cloud technology is different, but Blockchain has emerged as a technology to mitigate these challenges (Awuson-David et al., 2021). As demonstrated by Viriyasitavat, Da Xu, Bi, and Sapsomboon (2018) "A technology that enables immutability, and integrity of data in which a record of transactions made in Blockchain ecosystem are maintained across several distributed

nodes linked in a peer-to-peer network" (Awuson-David et al., 2021).

4.7 Current Cloud Logging Mechanisms

In this section, the current logging mechanisms in the cloud ecosystem and the role they play in securing logs in the cloud environment are explored. Two mechanisms are considered: Security Information and Event Management (SIEM), and OpenStack Cloud logging.

4.7.1 Security Information and Event Management

SIEM systems allow the CSP to grant a partial user right to the customer to manage their logging environment. However, this comes with a limitation as it depends on the type of Cloud service the customer is running. This access right allows the customer to manage event correlation to identify potential intrusions into the monitored environment. The logs include the following attributes: timestamp, user and device name including the severity of the interference. The Cloud provider also ensures that all records to be passed to a SIEM system to provide an overall view of security within the cloud ecosystem. However, customers who are managing their services in a multi-tenancy environment might not benefit fully from this tool.

SIEM systems are one of the tools we are going to be exploring, looking at how it maintains the integrity of logs both on transit and storage. This mechanism allows the CSP to grant a partial user right to the customer to manage their logging environment. The logs will include the following attributes: timestamp, user and device name including the severity of the interference. The CSP will also ensure that all records to be passed to a SIEM system to provide an overall view of security within the cloud ecosystem. However, customers who are managing their services in a multi-tenancy

environment might not benefit fully from these tools.

Where necessary, logs artefacts should be secure from servers, routers, firewalls, and other devices. Disks can be forensically captured to enable examination of any evidence that held on them. Containing an incident can be relatively straight forward as the system is entirely under the control of the affected organisation that can shut down servers or cut off network segments if required.

4.7.2 OpenStack Cloud Logging

OpenStack cloud services usually write its log files to /var/log/<service-name>/*. The OpenStack service operates system packages in scalable, performant and highly adaptive automated fashion that enables alternate log package to tar and ship old OpenStack log files, therefore circumventing unnecessary storage usage (Rosado & Bernardino, 2014). The exclusions to this default log directory are Keystone and Swift. Keystone uses Apache and consequently writes its log files to /var/log/apache2/*. Swift writes its log files to /var/log/syslog. Furthermore, if a client requests an OpenStack service's API; a request UUID is generated for the client (Abdelrazik et al., 2017; Corradi, Fanelli, & Foschini, 2014).

OpenStack architecture processes logging differently to Hyperledger Fabric Blockchain. We will demonstrate how our framework logging handling differs from OpenStack in Chapter 5 of this research project. Hyperledger Blockchain will enable cloud customers to audit the state of systems, monitor users' activity, and ensure user accountability with respect to their role and performance (Awuson-David et al., 2021), and (Awuson-David et al., 2019).

4.7.3 Blockchain

In 2009, a whitepaper called Bitcoin: A Peer-to-Peer Electronic Cash System was published by Satoshi Nakamoto to resolve the current challenges faced in the monetary market, with the main aim being to develop a technology that could allow electronic transactions from one party to another without going through financial institutions (Awuson-David et al., 2021). One of the significant challenges addressed was the double method, which is used to avoid the double-spending (a unique problem with digital currency is the risk of reproducing the same amount, even after spending). The idea of Bitcoin is to provide a digital currency that makes it easier to solve the problem of double-spending, and the technology that facilitates this is known as Blockchain (Nakamoto et al., 2008), and (Awuson-David et al., 2021; Zissis & Lekkas, 2012).

Blockchain structure can be described as an organisational structure where each department (which are the blocks) has a defined project to work on, with start and end dates (the transaction). The department then allocates these projects to individuals that work in a consensual manner where every effort is made to improve the performance of the organisation. Similarly, in a Blockchain ecosystem, each block consists of a transaction that has a header and timestamp for forensic and digital hashing that secure transaction of each node on the Blockchain network (Awuson-David et al., 2021). Each block also includes a reference that contains the information from the previous block. Blockchain security mechanisms such as cryptographic, smart contract, hashing and data immutability are the building blocks of Blockchain structure in preserving the integrity of data both in transit and storage. This technology proved itself during the introduction of the Bitcoin cryptocurrency as a permissionless Blockchain. It establishes trustworthiness between two strangers to carry out a safe transaction without a central entity such as banks (Awuson-David et al., 2021).

The distributed ledger mechanisms where specific transactions are linked to each

other, mainly through the Merkle tree, as shown in Figure 2.8 (Awuson-David et al., 2021).

Participants constitute one of the original core event components that interact with the Hyperledger Fabric Blockchain. In the current Cloud ecosystem, participants are the same as cloud customers who rent a particular service from the cloud service providers (Awuson-David et al., 2021). In the case of Blockchain, architecture participants play a specific part in the network as they control the most data. Even so, in our case, BCFL participants might not, however, know that they are interacting with our BCFL Blockchain cloud network.

The members of the supply chain interact with the Blockchain through Fabric SDK and use the HTTP protocol to access the BCFL resource. Critically, they are doing so on behalf of their organisations as they are the agents of the organisations. Likewise, when it comes to system and device participants, it is unlikely that devices will host a copy of the Blockchain ledger(Awuson-David et al., 2021). In this way, devices are a little more like individual participants. In contrast, systems in the network can act either on behalf of an organisation or in some cases, actually, represent the organisation. Blockchain participants have solved the current problem of the provider service layer agreement, which was a unilateral agreement and did not protect the interest of the current cloud customers. However, the Hyperledger Fabric membership service agreement has solved this problem as each member of the network has an equal right through a distributed ledger mechanism (Awuson-David et al., 2021).

The current supply chain Cloud ecosystem has faced challenges in the area of trust, transparency, data integrity, traceability of order and shipment (Gaur et al., 2018), and (Azzi, Chamoun, & Sokhn, 2019). The different actors that makeup the supply chain from the manufacturer, distributor, retailer and the customer have found it challenging to establish trust between their different domains in the current cloud ecosystem.

This has led to a lack of confidence in order processes as each actor is protecting their trade secrets as the current system cannot provide trust and transparency. A recent media report has highlighted that the United States (US) retail food giant, Walmart has launched its own Hyperledger Fabric Blockchain food retail ecosystem to facilitate traceability of food items (Awuson-David et al., 2021; Olsen, Borit, & Syed, 2019). Clear visibility of the supply chain ecosystem is needed to maintain forensic data integrity and transparency across all the domains, and the BCFL Blockchain framework solves this problem.

The BCFL architectural designed Docker engine was used to enable the integration of BCFL insurance and trade network use case. During the Docker code integration phase, compose-e2e.yaml is a Hyperledger Fabric necessary file in which the majority of the code editing was done, in order to facilitate a smooth start of BCFL Docker container. It is important to note that the dependence of Docker statically configured files are base/peer-base.yaml and base/docker-composer-base.yaml files, respectively. The Fabric Membership service integrates the transaction logs, BCFL users and peers including the chaincode and maintains the channel. One of the research's novelties is the transparent integration of the audit logs in more natural to read format. This ensures that log evidence is acquired in the cloud ecosystem with minimum difficulty. Finally, the REST-Server and Playground will be used by BCFL network users to access the Hyperledger Fabric Containers. However, most other log systems today do not adequately address security issues. When the cloud system is attacked, the adversary can easily cover up their intrusions by deleting logs or tampering with log records (H. Tian et al., 2019), and (Awuson-David et al., 2021; Pichan et al., 2018).

Figure 4.7 demonstrates the process flow of our supply-chain scenario, emphasising a secure transaction flow between all the stakeholders in the BCFL ecosystem. As Hyperledger offers data immutability and historic log recovery, our BCFL will maintain

the endpoint visibility and real-time forensic log monitoring used as digital evidence during an incident.

4.8 Supply Chain Case Study

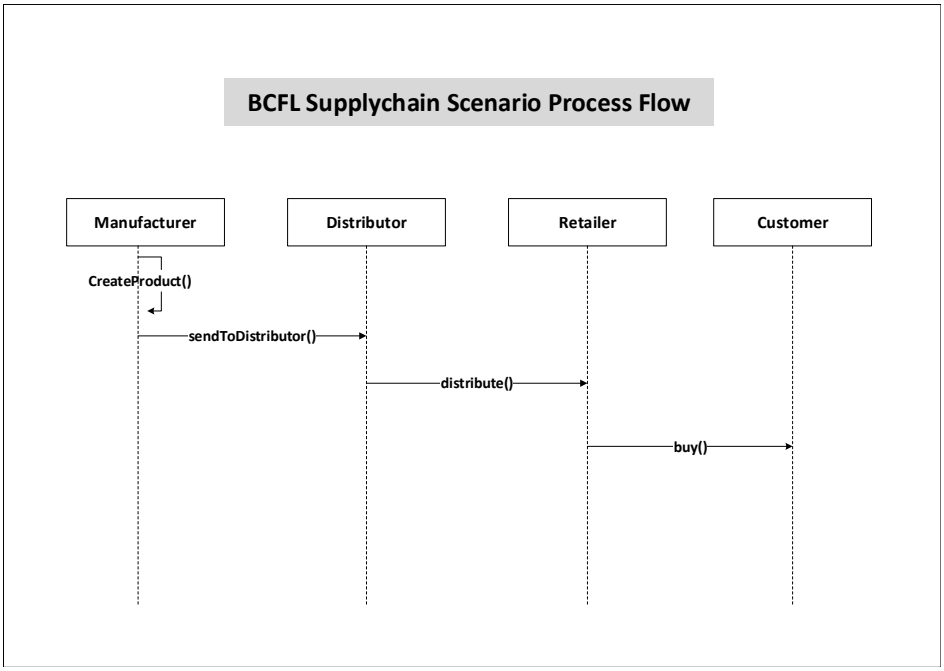


Figure 4.7: *BCFL Supply-chain Process Flow Diagram*

The BCFL decentralisation capability enables real-time tracking of orders, shipments, and the elimination of duplicate data entries and human error.

Deploying VMWARE Workstation 15.0

The virtualised hybrid lab was designed, built and deployed on Windows 10 enterprise operating system, 64-bit with 1TB HDD and 32GB RAM. The first step taken in this deployment was to login in the Window 10 operative system (OS).Then, from the Windows start menu, the directory containing the downloaded installer file was selected (Awuson-David et al., 2021). Further administrative right permission was granted to install the VMware workstation 15.0, from that point installation, and software license

agreement was accepted, and the installation directory was specified. Furthermore, all the default steps were taken to the end, with the integration of our BCFL algorithm to facilitate the configuration process to mimic a cloud IaaS platform that will enable us to perform the research experiment (Awuson-David et al., 2021). Google, Amazon or IBM cloud infrastructure did not fit the purpose of this experiment as they have limitations such as firewall rules and user service level agreements that need to be adhered to (Awuson-David et al., 2021).

Furthermore, as the installation completes, we have the virtualising cloud interface, so we then continue with the installation of Linux Ubuntu 16.04. The Ubuntu OS was adopted to allow us to deploy and configure Hyperledger Fabric Blockchain locally on our laptop. The following computer resources were allocated to the virtual Ubuntu OS, 8GB RAM and 100GB HDD from Windows 10 guest OS. These computer resources should be enough to run the Hyperledger Fabric and Docker components on top of the Ubuntu virtual machine (VM). Meanwhile, we downloaded all the prerequisites for Hyperledger Fabric installation. Finally, the chaincode, smart contracts, other programs such as logic.js and Docker.yaml were all programmed on Golan language (Awuson-David et al., 2021).

4.9 Setup BCFL Virtualised Hybrid Lab

The testing steps shown in Figure 4.8 ensure that the BCFL log acquisition was forensic proof and admissible. The testing flowchart demonstrates log evidence integrity processes, a transparent action taken to acquire the evidence with an established chain of custody. It answers the paper's questions that deleted VM instance logs are recoverable in a Blockchain Cloud ecosystem (Awuson-David et al., 2021).

The BCFL flowchart diagram has demonstrated the capability of Permissioned

Blockchain integration in the cloud and the way logs have been processed and evaluated in a secured Blockchain ecosystem. A transaction is initiated from the system participant in the diagram, then processed in the BCFL cloud network where immutable logs are preserved. In addition, all preserved logs are evaluated by the BCFL chaincode algorithm followed by a decision made by the nodes and smart contract was updated after the process (Awuson-David et al., 2021, 2019).

4.9.1 Algorithm

Hyperledger composer REST server was used to facilitate an endpoint API that interacts with the BCFL ecosystem. This enables an authorised client to have access to BCFL cloud resources through a secure REST server endpoint API (Awuson-David et al., 2021). It also ensures transaction integrity through a signed certificate. As demonstrated on the BCFL Pseudo Code RESTFUL Server Algorithm 1, a successful client authentication generates code 200 while an unsuccessful client authentication on the system will throw a 401 code error (Awuson-David et al., 2021). Additionally, the BCFL Framework testing and evaluation process is shown in Figure 4.8

Algorithm 1: Pseudo Code for BCFL RESTFUL Server Algorithm

```
Data: Connect to composer REST Server (s)
//submit a GET request
Result: generate = httpRequest.send("GET",locathost:3000);
if generate.code = = 200 then
    Successful system authentication
else
    (generate.code s = 401)
    Server authorisation error
    Update Hyperledger Fabric chaincode
end
```

(Awuson-David et al., 2021)

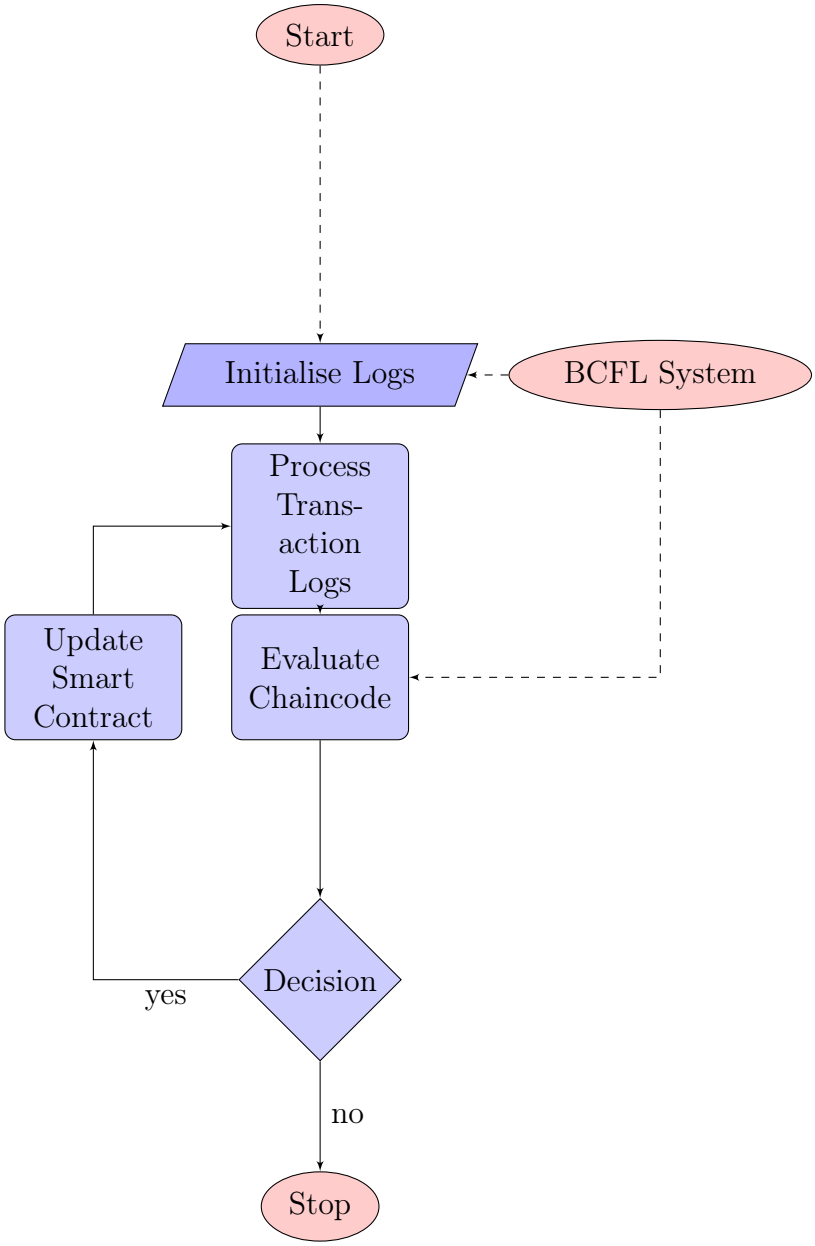


Figure 4.8: *BCFL Blockchain Simulated Testing Diagram* (Awuson-David et al., 2021)

4.10 BCFL Code

Here is the design code of our BCFL framework:

The Logic, Model, Permission and Angular App codes of our BCFL integrated with our DSRM and BCFL architecture enable a virtual test environment that ascertains the actual result of simulating a Blockchain cloud ecosystem. A supply-chain scenario was used to acquire log evidence that represented an accurate log of Blockchain cloud IaaS ecosystem. The experiment and testing also captured the true admissible logs, with immutability and trustworthiness.

The logic code is a clear demonstration of how BCFL maintains its secure environment by using a Hyperledge Blockchain ledger mechanisms facilitated by a chaincode or smart contract which ensures a consensus algorithm agrees to all Blockchain ecosystem participants.

```
%% LOGIC Code:

*/
Logic

/**
* The logic code defines how the chaincode sets the ownership of the asset,
transaction and movements of products.
* @param {org.supplychain.network.MoveProduct} moveProduct
- the trade product transaction
* @transaction
*/
```

```
async function moveProduct(moveProduct)
{ // eslint-disable-line no-unused-vars
moveProduct.product.issuer = moveProduct.product.owner;
moveProduct.product.owner = moveProduct.newOwner;
const assetRegistry = await
getAssetRegistry('org.supplychain.network.Product');
await assetRegistry.update(moveProduct.product);
}

/**
* This is the model code that define Asset, Product
and participance in the Supplychain Network
* @param {org.supplychain.network}
* @transaction
* Supply Chain Network
*/

namespace org.supplychain.network

enum ProductDesc {
o TShirt
o TechCap
o Belts
o Fleece
o Flipflops
}
```

```
enum ProductSize {
o SMALL
o MEDIUM
o LARGE
}

asset Product identified by productId {
o String productId
o String producttype
o ProductSize size
o ProductDesc description
o Double quantity
o Double unitPrice optional
o Double totalPrice optional
--> Participant owner
--> Participant issuer
}
participant Customer identified by email {
o String email
o String firstName
o String lastName
o String type
}
participant Manufacturer identified by email {
o String email
o String firstName
```

```
o String lastName
o String type
}
participant Distributor identified by email {
o String email
o String firstName
o String lastName
o String type
}
participant Retailer identified by email {
o String email
o String firstName
o String lastName
o String type
}
transaction MoveProduct {
--> Product product
--> Participant issuer
--> Participant newOwner
}

/*
The permissions are set to all participants.

*/
```

```
/*
org.Supply-chain Network Angular Permission Code:
*/
rule Default {
description: "Allow all participants access to all resources."
participant: "ANY"
operation: ALL
resource: "org.supplychain.network.*"
action: ALLOW
}
rule SystemACL {
description: "System ACL to permit all access"
participant: "org.hyperledger.composer.system.Participant"
operation: ALL
resource: "org.hyperledger.composer.system.**"
action: ALLOW
}
rule NetworkAdminUser {
description: "Grant business network
administrators full access to user resources"
participant: "org.hyperledger.composer.system.NetworkAdmin"
operation: ALL
resource: "**"
action: ALLOW
}
rule NetworkAdminSystem {
```

```
description: "Grant business network administrators
full access to system resources"
participant: "org.hyperledger.composer.system.NetworkAdmin"
operation: ALL
resource: "org.hyperledger.composer.system.**"
action: ALLOW
```

4.10.1 BCFL Single-case Mechanism Experiments Scenario

In this section, an initial experiment was simulated using a single case study scenario that enabled the framework's first initial result. Chapter 5 demonstrated more use cases and comprehensive BCFL framework testing to ascertain how BCFL acquired admissible log evidence in the cloud environment.

Single-case mechanism experiments enable simulation of the scenario presented by a model of Blockchain cloud context. The experiment was conducted in a virtual lab environment, and the scenario was simulated to capture and observe the mechanisms, evaluate, draw conclusions and view test results as demonstrated in the next section.

4.10.2 Scenario

Alice is the director of Tag Shop in a thriving high street chain with excellent online visibility. Tag Shop is running a Hyperledger Fabric Blockchain network. It has VMs workstation and server rented from the CSP for the business's day-to-day running. However, Tag Shop recently hired a new system administrator, Bob, whose responsibility is to maintain the system and update applications when necessary (Awuson-David et al., 2021). During a system update, Bob accidentally deleted a week's worth of transactions and one VM. This incident negatively impacted Alice's business in terms

of profit and reputation. Consequently, Alice hired a digital forensic company (ACMD Ltd) to investigate and recover all deleted VM transactions. As the forensic investigator analyses the Blockchain logs, they were able to reconstruct how the incident happened (Awuson-David et al., 2021). The forensic investigators looked at the following steps to solve the case:

Digital Forensic Investigation and Result: In forensics it is vital to maintain data and evidence integrity at all times. The investigation in this scenario established that Hyperledger Fabric Blockchain used hashing, encryption and immutability to maintain log evidence integrity and preserve the chain of custody (Awuson-David et al., 2021).

Defining and securing the digital crime scene in a cloud ecosystem could be problematic due to the multi-tenancy, geo-locality and GDPR compliance. However, as shown on the acquired digital log evidence in Figure 4.9, the Hyperledger chaincode maintains a secured accurate and immutable timestamp (Awuson-David et al., 2021). All Bob's transaction entries IDs were acquired by Alice's hired forensic investigators as highlighted on the evidence snapshot. The acquired digital log evidence also highlights BCFL's capability to enable effective traceability to forensic readiness mechanisms in the supply-chain ecosystem (Awuson-David et al., 2021).

Evidence Preservation and Chain of Custody: Throughout the investigation, the chain of custody was maintained. The most important part of the investigation is how BCFL enables log filtering mechanism that facilitates real-time evidence acquisition that does not interfere with Tag shop's daily business operations (Awuson-David et al., 2021). The case study has proven that integrating Blockchain in the cloud ecosystem will mitigate many of the challenges the digital forensic investigator and police first responders face in ensuring the admissibility of digital evidence in the cloud ecosystem (Awuson-David et al., 2021).

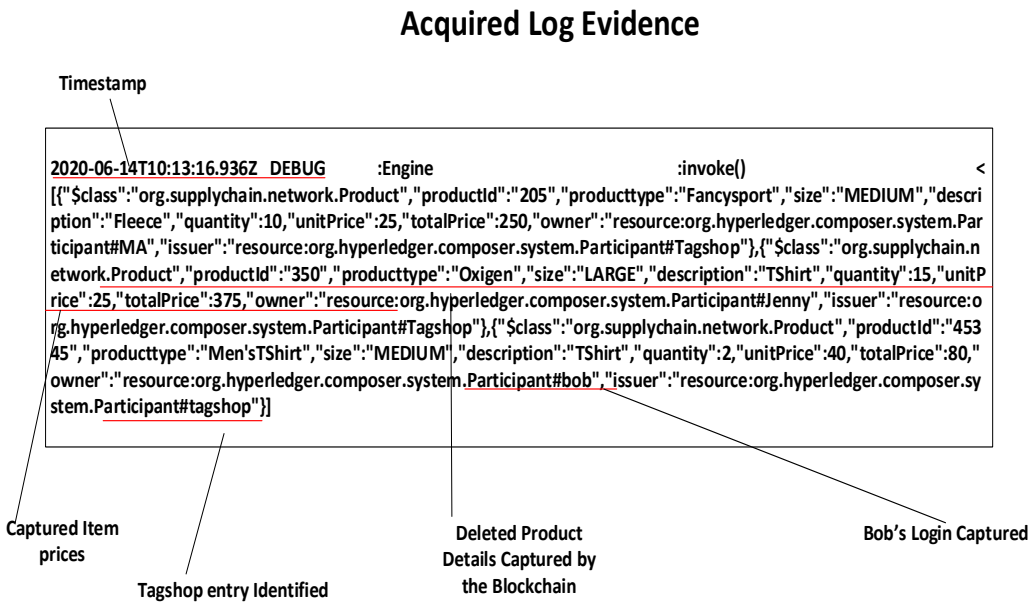


Figure 4.9: *BCFL Supply-Chain Case-study Log Evidence* (Awuson-David et al., 2021)

The Tag shop case Table, 4.1 highlights the chain of custody and how it was maintained throughout the investigation process by ACMD Ltd (Awuson-David et al., 2021).

Even when the investigation is over and evidence destroyed or returned, an entry will still be made on the chain of custody form to identify all actions taken by investigators. ACMD also included the chain of custody entries in their report which highlights all log evidence that was acquired as part of evidence reconstruction and admissibility (Awuson-David et al., 2021).

Figure 4.10 demonstrates a friendly angular interface app of BCFL framework supporting easy day-to-day administration of all participants' transactions in the BCFL cloud network (Awuson-David et al., 2021).

Table 4.1: *BCFL Framework Scenario: CHAIN OF CUSTODY*

Tracking No:	Date/Time:	From:	TO:	Reason:
1	Date:14-06-20 Time:10:13 am	ACMD Ltd John Smith Signature N/A	Name Org: Jo Brown /ACMD Ltd. Signature J.Brown	Log Evidence Seizure from Tagshop Cloud Blockchain Network
2	Date:14-06-20 Time:10:40 am	Mark Johnson/ACMD Ltd Signature M.Johnson	Name ON: Evidence locker/ ACMD Signature N/A	ACMD Cloud Blockchain Secure Storage

(Awuson-David et al., 2021)

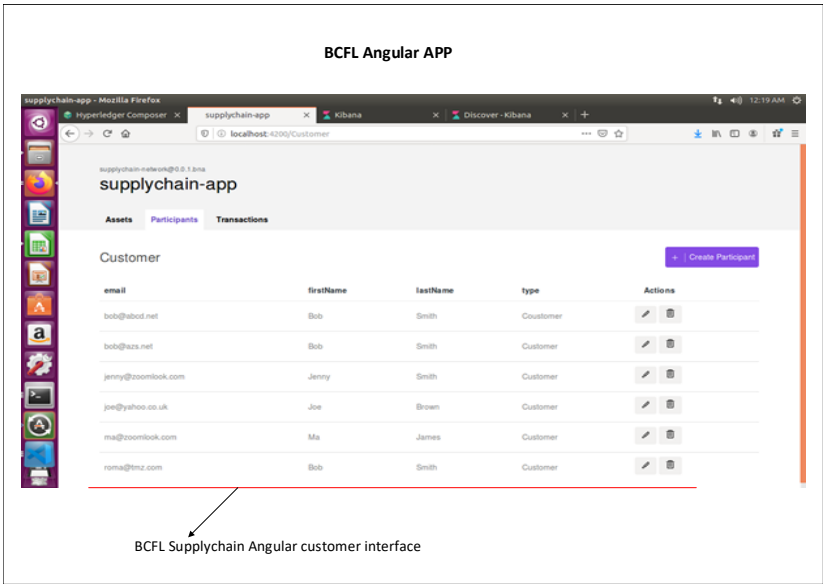


Figure 4.10: *BCFL Angular Supply chain Product Interface*

4.10.3 Performance Metrics

ELK is open source software used for real-time system monitoring and components system application performance monitoring as shown in Figure 4.11, 4.12 and 4.13 (Awuson-David et al., 2021). The ELK integration into our framework supports real-time performance monitoring and analysis. It has a data channel or pipeline mechanism

that processes data from multiple end nodes simultaneously, transforms it and forwards it to Elasticsearch for further processing (Awuson-David et al., 2021).

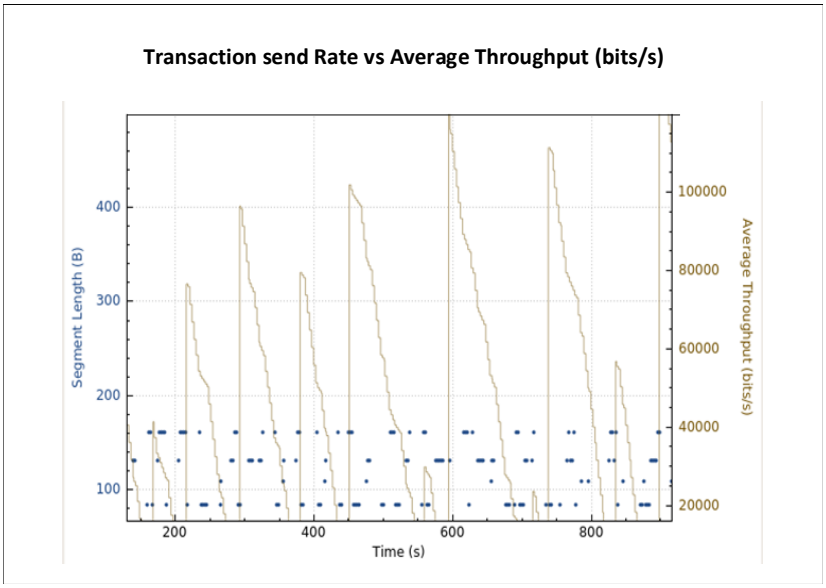


Figure 4.11: *Transaction Rate Throughput*

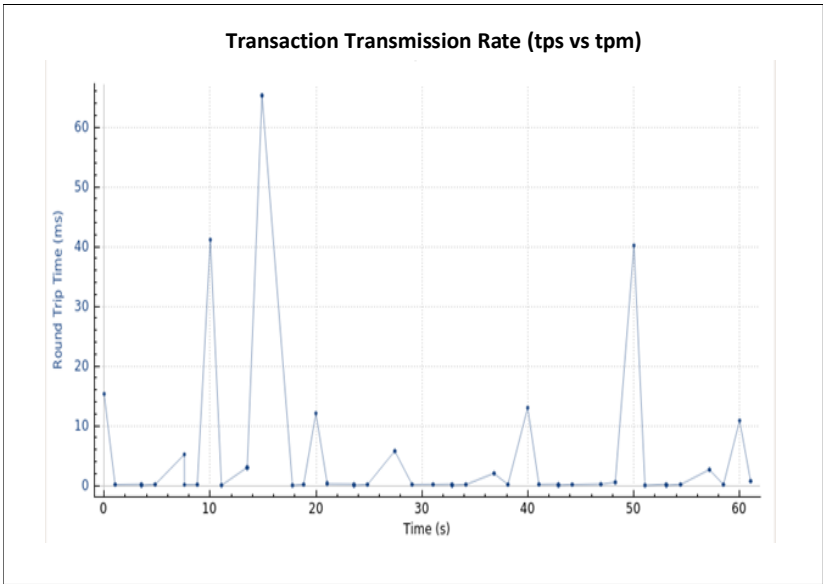


Figure 4.12: *Transaction Rate tps*

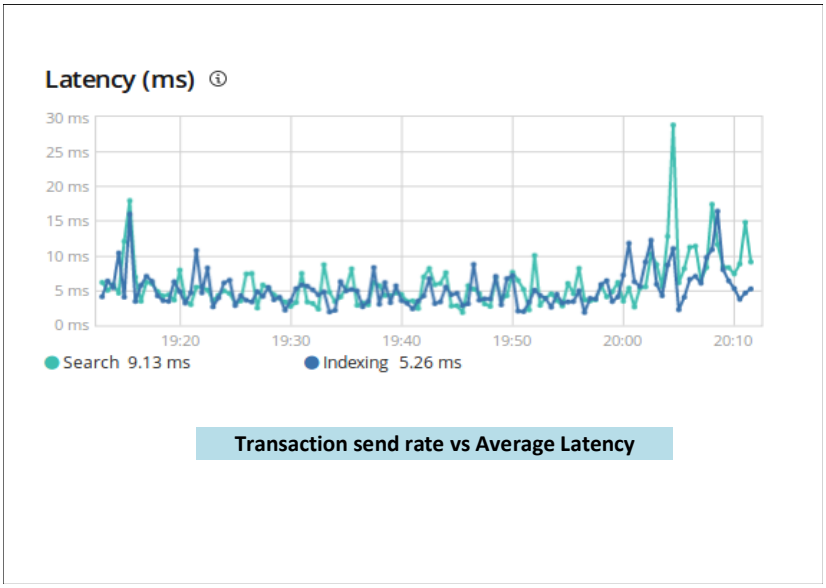


Figure 4.13: *Transaction Rate tps vs Latency*

Kibana gives us a holistic performance view of the CPU and memory usage in a graphical format. ELK performance measurement technology has, therefore facilitated our framework (Awuson-David et al., 2021). For example, suppose any peer or node in our BCFL goes down. In that case, the system administrator can activate the forensic readiness plan and respond to the cyber incident in real-time. A vital element of the incident response plan is the escalation procedures (Awuson-David et al., 2021). The average throughput performance metrics monitoring enables a more unobstructed view of how the send transaction rate in second and minutes are recorded (Awuson-David et al., 2021).

4.11 Summary

In this chapter, we introduced a BCFL architecture and framework and described initial testing of the system and explored it in detail. We explained how the framework supports an active cloud forensic investigation. The BCFL can securely capture dis-

tributed transactions in a distribute-secure log maintained by peers (Awuson-David et al., 2021). A sound Blockchain cloud as a service platform incorporating our BCFL will resolve difficulties in cloud forensic investigation and will support the investigation, including administrator in real-time forensic analysis (Awuson-David et al., 2021). The BCFL architecture establishes how participants interact with the assets in our BCFL architecture. Furthermore, it is important for the cloud forensics investigators to understand how Blockchain transaction logs facilitates digital forensic investigation in the cloud ecosystem. Cloud computing offers customers on-demand shared resources in a virtual environment where human intervention is highly limited with benefits such as cost efficiency, scalability, agility, convenience and elasticity (Awuson-David et al., 2021). However, this comes with risk, threats and challenges identified by a CSA cloud threats list. Besides, with the rapid increase of cloud computing, many organisations have started to adopt the technology. Chapter 2.1.2 reviews related academic literature in the Blockchain cloud framework that redefine a forensic process that suits the cloud ecosystem. However, BCFL differs from other reviewed academic literature. Its focus on solving the challenges of using Blockchain distributed ledger technology to ensure tamperproof log evidence. These challenges can be addressed by ensuring a secured process of digital evidence acquisition in the cloud ecosystem. The BCFL will mitigate the challenges by using the BCFL algorithm to enhance the Blockchain consensus and distributed ledger mechanisms to secure log evidence acquisition in the cloud ecosystem (Awuson-David et al., 2021).

Many researchers have focused on developing new tools in acquiring digital artefacts in the cloud ecosystem with little research on the use of a new framework to acquire digital forensic evidence in the cloud (Awuson-David et al., 2021). This is the gap this research seeks to fill, to achieve this various cloud frameworks were reviewed, and Blockchain was selected as an integrated technology in the cloud ecosystem to maintain

evidence integrity, privacy and trustworthiness (Awuson-David et al., 2021).

Chapter 5

Simulation, Testing and Evaluation

This chapter concentrates on simulation, testing and evaluating BCFL framework design. First, a single case study was used to test and evaluate each artefact by simulating the natural Blockchain cloud ecosystem. Secondly, we evaluated the process and explicitly identified each test result that answered our research objectives and questions. Finally, the simulated and tested process was carried out in five phases: A single case study and components testing demonstrated how VM instance logs could be recovered after a system is deleted, testing the admissibility of the acquired logs, and finally, testing BCFL system performance.

Huge volumes of data may be compelling at first glance, but without an interpretive structure, they are meaningless.

Tom Boellstorff

Ethnography and Virtual Worlds: A Handbook of Method, 2012.

5.1 Introduction

The purpose of this research is to investigate how to acquire forensic log evidence from a containerised Blockchain Hyperledger Fabric cloud ecosystem. We will be examining how to recover logs from a deleted Docker Container. The previous chapter described the components of BCFL architecture, advanced framework, Hyperledger

Fabric, Docker and Elasticsearch, detailing how these tools are integrated with the BCFL network. Further to justify why these tools were used for this research project in terms of preservation, trustworthiness and integrity of logs in the Blockchain cloud ecosystem. This chapter will describe two case study scenarios that will be used to test our BCFL simulated Blockchain Containerised experiments. These scenarios were designed to investigate how to integrate a Blockchain cloud system and how to maintain the integrity of logs both in storage and on transit in the cloud ecosystem.

The experiments will also verify the findings of the literature review, which states that when a virtual machine in the cloud ecosystem is shut down, logs on the system will be lost. Secondly, multi-tenancy and geolocation issues added yet more complexity in acquiring log evidence from the cloud ecosystem . This chapter will answer the research question s, including the objectives of the research. It will answer the second, third and fourth research objectives which concern how to maintain the integrity, trustworthiness and the tamperproof state of the logs. To determine how forensic investigators can quickly acquire log artefacts in the cloud ecosystem without the complexity of multi-tenancy, geolocation and the limitations, we need a system that is resilient to these challenges. As proven by researchers, VMs will lose their logs when they are powered down. However, our DSRM was adapted to suit our BCFL architecture framework. The test results from these experiments was evaluated in this chapter.

It will be compared against the measures for a secured logging mechanism in the Blockchain cloud ecosystem proposed in Chapter 3 to determine the integrity of logs both in transit and storage. The second section of this chapter will analyse in detail the BCFL innovative Blockchain cloud ecosystem virtualised lab before analysing the performance metrics of our BCFL Blockchain cloud network, in particular how the CPU, RAM and memory of the Docker containerised system is running.

5.1.1 Second Single-Case-Study Experiment

In this project, a single-case study experiment was adopted to test and evaluate the BCFL Blockchain cloud network architecture in a real-world scenario. A single-case study experiment is a test of a single case in which the researcher applies stimuli to the fact and explains the responses in terms of mechanisms internal to the case (Awuson-David et al., 2019; Wieringa, 2014). The single-case study will enable us test and integrate Hyperledger Fabric Blockchain components into the BCFL cloud environment evaluate how logs are stored in a Docker environment and then investigate how these log files are acquired between peers and then stored in a CouchDB database.

5.1.2 Business single-case scenario

This section will explore and implement an insurance claim case is the first, and the second scenario will be explore a diamond, platinum and silver trading scenario. No one wants to have to make an insurance claim , but when things do go wrong, and accidents happen, this may result in financial losses, (Xun (Brian) Wu, 2018). With the BCFL framework, the transactions recorded in the ledger are immutable, and the data current state can only be updated when all parties agree. As demonstrated by J. Li, Wu, Jiang, and Srikanthan (2020), records in the Blockchain can be shared in real-time. Insurance processes and commodities trading experience all kinds of situations that illustrate the inadequacies and doubt in real-world processes that the BCFL framework is designed to mitigate. “It is possible to verify payments without running a full network node. A user only needs to keep a copy of the block headers of the longest proof-of-work chain, which he can get by querying network nodes until he’s convinced he has the longest chain, and obtain the Merkle branch linking the transaction to the block it’s timestamped in” (Nakamoto, 2019). So, the designed BCFL framework

insurance business scenario and commodities trading scenario contain simulated versions of claim transactions carried out in the real world for the insurance scenario. According to Bashir (2017), Permissioned Blockchain technology offers organisations and businesses secured transaction mechanisms and data confidentiality. Hyperledger Fabric Blockchain uses chaincode and channelling mechanisms to provide a suitable range of cryptographic protocol and algorithms that can be adapted in any business environment. In the next phase we will have an overview of the insurance business and commodities scenario.

5.1.3 Overview of Insurance Scenario

According to Xun (Brian) Wu (2018), the traditional insurance claims process has remained the same for decades, as there are some critical problems in the process, including false claims, fraud detection, slow and complicated claims processing, human error, unpleasant customer experience, and inefficient information flow in reinsurance. The case scenario will describe a simple claim transaction. The insurance claim from the insured to the insurer and then to the broker. This scenario exhibits a simulation of an insider fraudulent transaction claim which is perpetuated by a staff member of the FCT insurance company, named Tom. A demonstration of how the fraud was carried out by a FCT staff member that has superuser access right was simulated as a validation of the BCFL framework. The staff member with trusted access on the BCFL cloud network was able to create a fake user (insure) and then put up a claim in his name. Afterwards, he deleted his account profile and the virtual instance allocated to the user. According to Bashir (2017), auditing is another requirement of a Hyperledger Blockchain, however, it is estimated that an immutable audit trail of all identities, related operations and any changes are securely persevered The following will be established in our results in this scenario:

- Highlight the current challenges faced by a forensic examiner in the current cloud ecosystem, which shows that when the server is rebooted, all logs are deleted.
- Establish the immutability and transparency of logs both in storage and transit in cloud Hyperledger Blockchain-based architecture..
- BCFL architecture will solve the problems of multi-tenancy, accessibility, and geolocation issues faced by a forensic investigator in acquiring digital evidence from the current cloud ecosystem. In this scenario, we will see how our framework mitigates all these issues.
- Use Blockchain data integrity (encryption, hashing and timestamp) mechanisms to maintain the evidence integrity throughout the investigation process and managing chain of custody.

5.1.4 Commodities Scenario

There are three groups of commodities companies that trade on Gold, Diamond, Silver and Platinum. However, one of the companies, ABC Ltd, noticed that there were some irregularities in their sales transactions. The senior Cloud administrators at ABC Ltd understood that they were running a Blockchain Cloud network that supports data integrity, immutability and trustworthiness. Due to this persistent incidence of irregular sales transactions, they decided to carry out a forensic investigation of their system by examining all their transaction logs.

5.1.5 The Experimental Testing of our Scenarios

In this section, we will provide comprehensive testing of our BCFL Blockchain network:

- End to end system component testing of the containerised Blockchain Cloud Forensic Logging. This testing will verify our first and second scenario and answer the research objectives.

- Conduct a comprehensive testing of BCFL Hyperledger Fabric containers such as the peer, order, CouchDB and Fabric Cloud network (cov-Insurance-network and trade network) in the Docker environment. Then use the Fabric Composer API's, such as Playground and REST-Server to create a transaction and view those transaction logs with their timestamp and entries. This will verify and answer the second and third research objectives.

- Testing the immutability of logs in BCFL scenarios will provide an insight into how deleted BCFL Hyperledger Fabric transactions can be recovered as tamper-proof, from our BCFL framework without compromising the forensic artefacts. This testing will answer the third research objective and provide an understanding of how Hyperledger Membership Service, Chaincode, Channel and the Smart Contract maintain a secured Cloud multi-tenancy environment. It will also be noted that in BCFL, when a Hyperledger Fabric container is deleted, it becomes invisible and there is no option within the Fabric or Docker for the recovery of the deleted container.

- Test Monitoring: System monitoring is an essential capability for addressing regulations and ensuring high accessibility, volume planning, pattern acknowledgement, and fault identification. We will conduct application and component performance real-time analysis of the BCFL Blockchain cloud Network. In addition, we will measure the performance metric of the BCFL Blockchain cloud ecosystem to answer the fourth research objective and question.

The flowchart in Figure 4.8 examines the two scenarios in this chapter to answer research objectives 1, 2 and 3 of the projects. Following the steps shown in the diagram will ensure that the BCFL log acquisition is accurate. A set of variables were simulated to obtain this result. In this flowchart, the audit log can again be used to identify the Docker container's legitimate user. Blockchain's immutability mechanism preserves all information and logs evidence of the deleted containerised VM instance, however, due to the virtualised Docker container in BCFL architecture, its rapid deployment and recovery stage is better than the current VM's. The immutability of the log will facilitate easy recovery of all dated logs.

5.2 Blockchain Cloud Forensics Logging (BCFL) Experimental Lab

In this section, we are going to bring together all the concepts we have discussed with a sample business network, involving a real-world example. Specifically, a detailed walk-through of the BCFL framework using Hyperledger composer, further helping to understand how participants, assets, transactions, and events are linked. Due to the limitation in using real cloud ecosystems such as Google cloud or IBM Bluemix, the experiment was hosted locally to get the full result of testing. We are going to use a VMware based system to host the experimental network on a laptop. This will be explained in more detail in the analysis section of this research.

BCFL Lab Setup

According to Xun (Brian) Wu (2018), who defined Hyperledger Composer as a set of JavaScript-based high-level toolsets and frameworks that build and run an application on top of a Hyperledger Fabric Blockchain, the composer tool generates a RESTful endpoint to interact with Fabric channels.Figure 5.1, above, demonstrates the update of the current components that are coded to facilitate smooth deployment of the BCFL Blockchain cloud network. These system and application updates are needed to secure a more robust Blockchain cloud architecture environment.

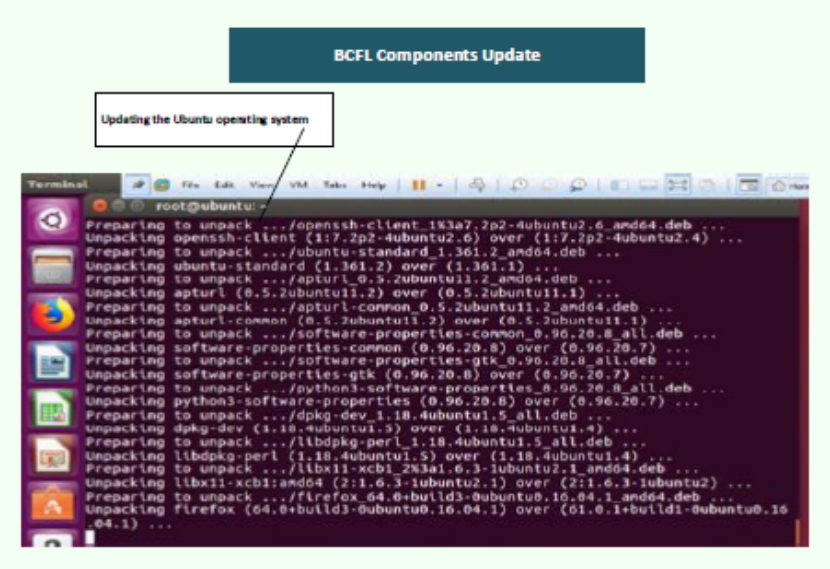


Figure 5.1: *On the above image, we issued an update command to update the Ubuntu operating system with the latest tools*

Figures 5.2 and 5.3 present the backend interface of our BCFL framework in which initial performance testing was carried out. Furthermore, the components algorithm was coded to support the deployment after the system and applications were updated.

Python libraries packages are needed to support Hyperledger Fabric deployment on Ubuntu 16.04, as this will mitigate any deployment or configuration error that might occur.

Figure 5.2: *BCFL System Components Updates*

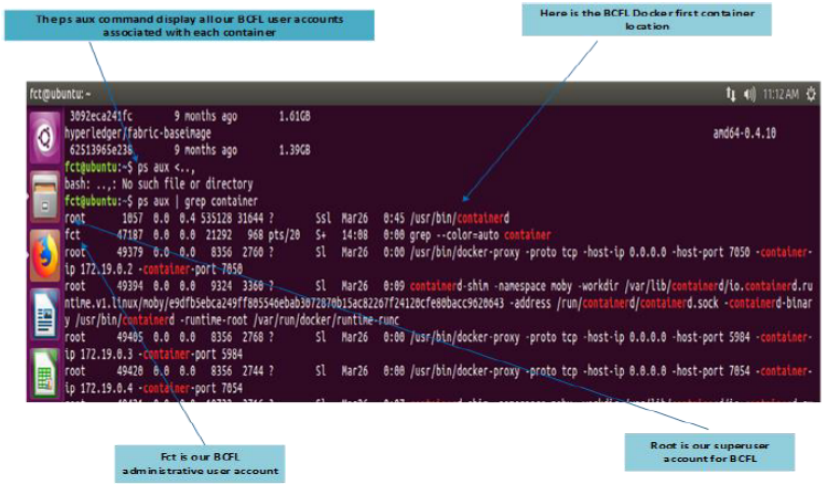


Figure 5.3: *A command (ps aux | grep) was issue to view BCFL root user account and super admin account (Awuson-David 2019).*

A ps aux | grep command was issued to verify all root and superuser accounts on the BCFL Blockchain cloud network containers. It also identifies all the containers running on the BCFL, including its storage space and CPU usage. These tests will regulate the successful implementation of Hyperledger Fabric in the local laptop, and answer our first research question. The following instruction command was issued to view all users and superuser accounts (fct and root) on the deployed BCFL Blockchain cloud network. The following command 'docker image ls' was issued to test the post implementation phase of the BCFL Blockchain cloud Network deployment as demonstrated in Figure

5.4.

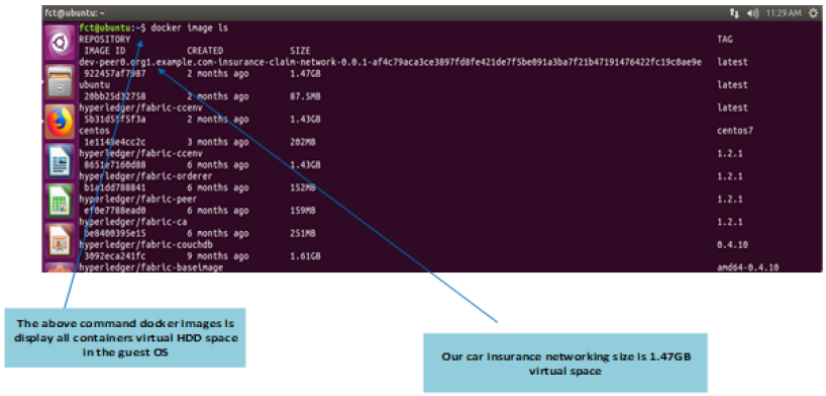


Figure 5.4: *The command (docker Image ls) displays all Docker Images and their container size*

This test was also conducted on the 'cov insurance claim network' and trade network to establish the virtual storage space that was allocated to the BCFL network. The BCFL 'cov insurance claim network' is running a 1.47GB of virtual hard drive space which is very efficient as Docker containers use less network resource such as bandwidth, compared to traditional VMs. Finally, six other Docker Containers are running on the system, and they are not part of the BCFL Blockchain Cloud Network, but an internal network of the organisations.

Figure 5.5 below, demonstrates how a fake user was created by the suspected BCFL administrator who happened to have a superuser account and have access to the cov insurance claim network Hyperledger REST-Server API web interface. Tom created a fake user account (Tim) with his superuser root access and then later he deleted the account after carrying out a fraudulent insurance claim. Here are the profile details of the user he created, named Tim, last name Matt, and a user ID of 250. We are going to view the user's details in the next section and see how the account profile was used to facilitate a crime. We have also highlighted all user profile information from our BCFL REST-Server interface. It is most important we understand this conceptual picture

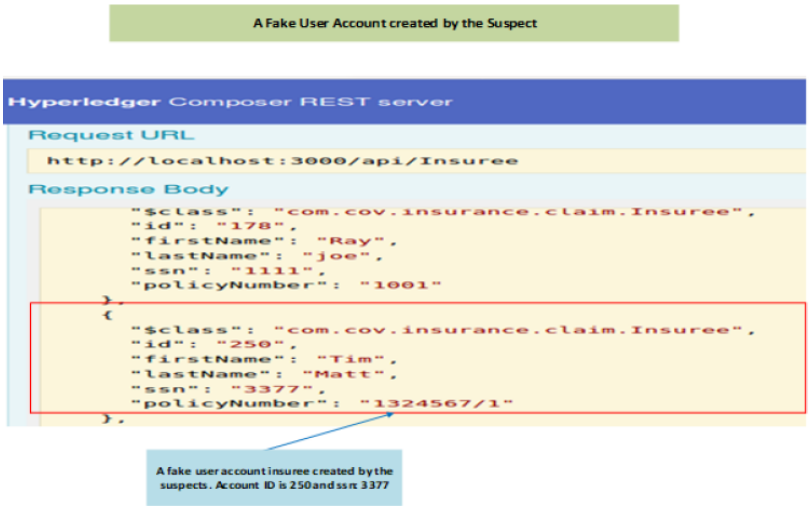


Figure 5.5: *Hyperledger Fabric REST-Server API displayed participant account profile*

relationship between the BCFL Blockchain cloud network participant, transaction and assets.

Tom's main aim was to delete the forged user virtual instance and account from the Blockchain Network as highlighted in Figure 5.6. He thought if he was to delete all log entries from the server; then all log entries would be lost when the server was restarted, as current research suggests would occur in the traditional VM environment.

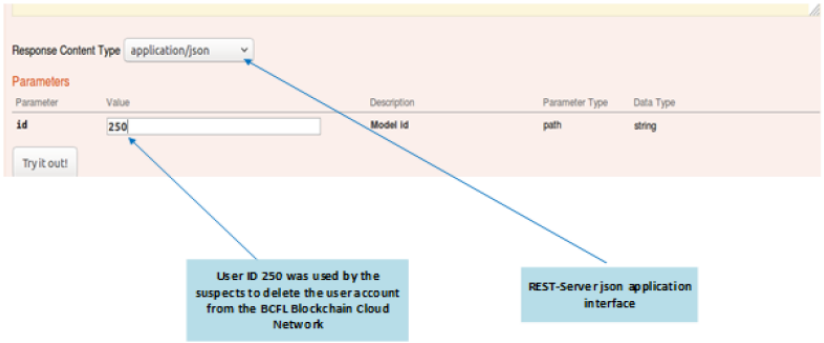


Figure 5.6: *BCFL Admin Participant Activates Participant Account Deleted from the REST-Server API Server*

This experiment will prove, Hyperledger Fabric Blockchain has immutability, data

integrity and trustworthiness capability at all times even if the system is shut down intentionally or unintentionally. As a participant, he is unaware that Hyperledger Fabric Blockchain maintains data integrity at all times, even after the data is deleted.

Figure 5.7 we demonstrate how the BCFL Blockchain cloud network logging capability was able to record all log entries after the fake participant account was deleted from the asset. It answered the second research objective, which was: To investigate how Blockchain technology can be integrated into the cloud ecosystem and used to maintain trustworthiness and transparency in the cloud ecosystem establishing a way of communicating challenges between the cloud customer, CSP, first responders, forensic investigators and system administrators (Awuson-David et al., 2021).

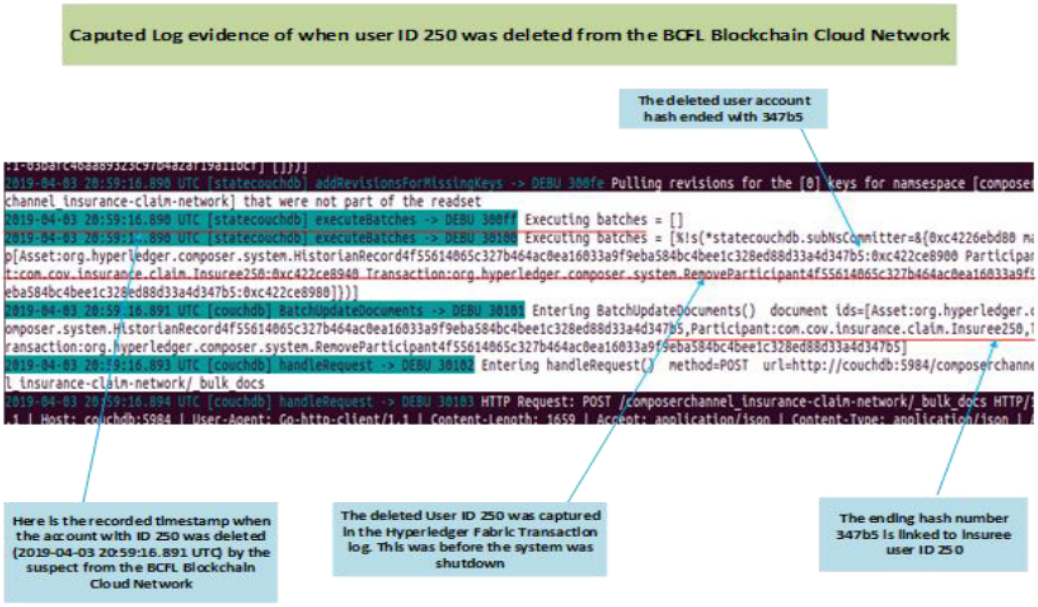


Figure 5.7: *Display of the Log Evidence From our BCFL Blockchain Forensic Logging*

Due to the complexity in evidence acquisition in the current cloud ecosystem, there is a need for effective communication between stakeholders. Blockchain is a decentralised system, and everyone in the smart contract or consensus has a participant access right and the shared ledger can view all the transaction log entries as long as he

or she has an Internet connection.

This answered the first and second research questions: How can system logs (network, process and application) be extended to include additional information that will help solve recognised problems in cloud forensics, including the issues of dependency chains and time synchronisation? Secondly, how practical would a Blockchain approach be for capturing and storing extended cloud logs? Hyperledger Fabric has the capability of a comprehensive logging mechanism that is suitable for log forensic acquisition. Hyperledger Blockchain maintains log integrity throughout the acquisition process therefore, it supports a sound chain of custody throughout the log acquisition process.

The BCFL Blockchain cloud Network system is completely shut down by Tom to facilitate his crime as he believes that once the system is shut down all the log entries will be lost, as shown in 5.8.

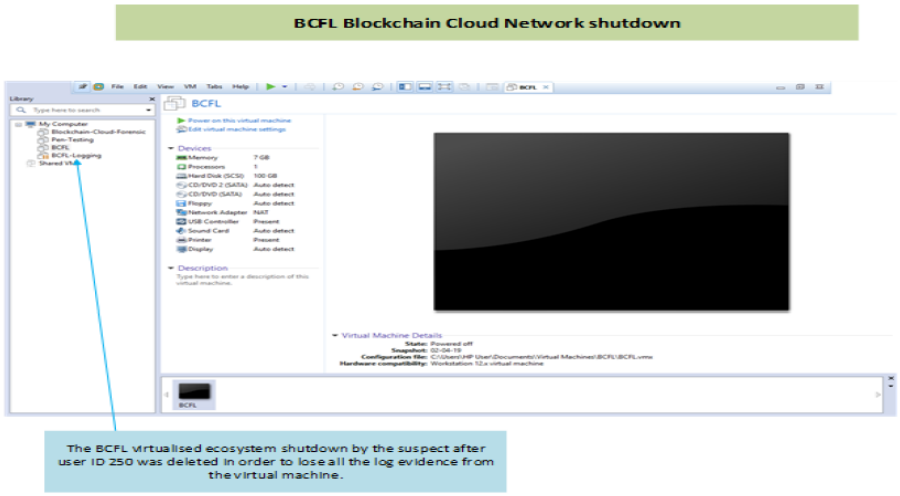


Figure 5.8: *BCFL Blockchain Cloud network was shut down by the suspect to potentially clear all logs from the Cloud network*

In the next phase, we establish how the BCFL Docker containerised virtual machine is powered and how to view the log entries to investigate if the log entries are still intact

or not.

The BCFL server established that Tom used the following command ' /fabric-tools/startFabric.sh' to start all the BCFL Docker Containers in the Hyperledger Fabric Cloud Network after reboot as shown in 5.9. Tom also used the 'watch -n 0.1 'docker ps' command to watch the container's start-up in real-time and in the next phase of the experimental testing we can view the logs after the reboot.

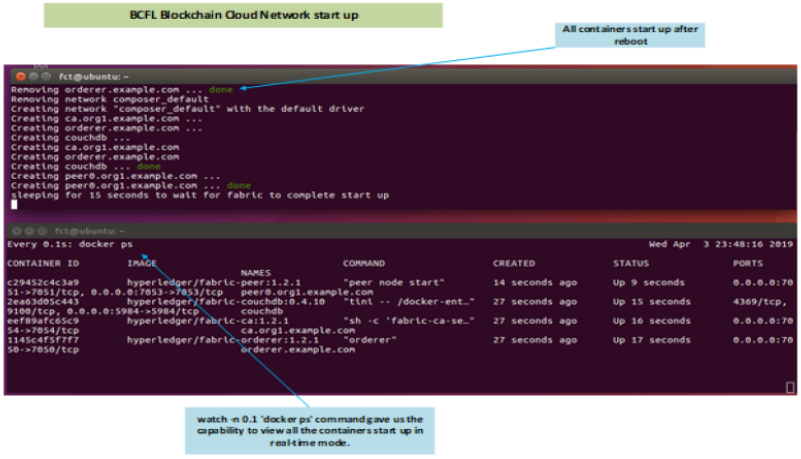


Figure 5.9: *BCFL restarted after a reboot. No error encountered during restart*

Figure 5.10 shows the timestamp of the deleted account with user ID 250 before the BCFL Blockchain Cloud Network was shut down as 2019-04-03T20:59:16.890 while after a reboot we view the log timestamp as 2019-04-03T20:59:14.511Z. The time difference between entering the transaction logs before and after the BCFL shut down is two seconds.

This experimental result has answered the third research objective; To investigate and Identify the main challenges as described by the CSA in performing digital forensics processes, where the artefact resides in a Cloud ecosystem. While these challenges continue to persist in digital forensics processes, the essential characteristics of cloud computing systems are enumerated in Section 1.3.1 according to the NIST.

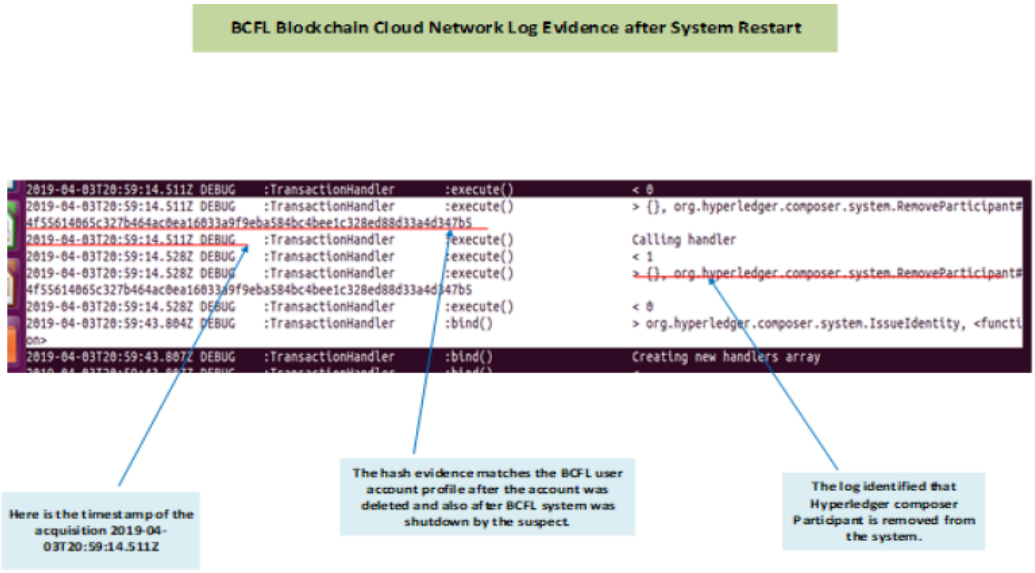


Figure 5.10: *Screenshot displaying all acquired transaction logs*

It answers the second and third research questions, how useful would a Blockchain approach be for capturing and storing extended cloud logs? Furthermore, what relevant services are needed to use the extended logs effectively, and how can these be built? Blockchain technology has established its capability and well-integrated technology to maintain, secure and managed cloud forensic log evidence both on transit and storage. Integrating Blockchain forensic as a service in the cloud ecosystem will help to solve the problems and threats that were established by CSA.

BCFL will address the issue of acquisition in multi-tenancy in cloud because it is peer to peer and encrypted (a hashed network which means that each tenant will maintain its segmentation). Figure 5.10 above, has established the immutability of Blockchain logs as the logs are still intact even after the user account and virtual instance were deleted, and the Docker virtual container shut down. The system administrator is still able to view logs as it were before the reboot.

Forensic acquisition of cloud network is an uphill task faced by the digital forensic investigator as described in the literature review Chapter 1.2 of this research however,

the Hyperledger Fabric Blockchain technology has gained trust in the digital forensic community to mitigate or solve this evidence acquisition problem fundamentally. The experimental testing of the BCFL Blockchain cloud network will scientifically prove how trust can be added to forensic log acquisition in the Blockchain cloud ecosystem. This trust will mitigate the risk and threats described by CSA and NIST in Chapters 1 and 1.2 of the project.

Furthermore, institutions and businesses such as government agencies, the financial sectors, logistics, and supply-chain can benefit from the immutability and shared ledger mechanisms where trust can be achieved through transactional timestamps as mentioned in the initial testing in Chapter 4. This digital interaction does not depend on trust alone but also guarantees that the source of the transactional data preserves a permanent track record of communication between the different organisations in the consensus. The trust and log immutability will reduce risks and threats as described by CSA and NIST in Chapters 1 and 1.2 of this project. In the BCFL cloud network logging mechanisms we are using Hyperledger Fabric log levels such as Critical, Error, Warning, Notice, Info and Debug to ascertain log type for the BCFL Network.

5.3 Traffic Flow Analysis

Figure 5.11 captured all the traffic flow of each BCFL Blockchain node transaction.

System monitoring and application performance was part of the BCFL architectural integration to detect anomaly system behaviour and security threat. This was a risk assessment to ensure that 360 degree security is maintained on the BCFL REST API and Angular app.

Figure 5.12 provided the BCFL participants with a Blockchain API interface that enabled the forensic investigator to view all the transactions made in BCFL using the

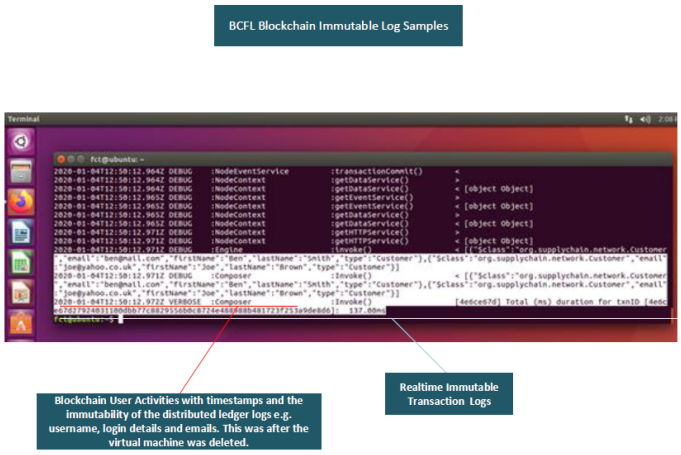


Figure 5.11: *Screenshot image showing the traffic flow of the supply chain*

integrated Angular API app.

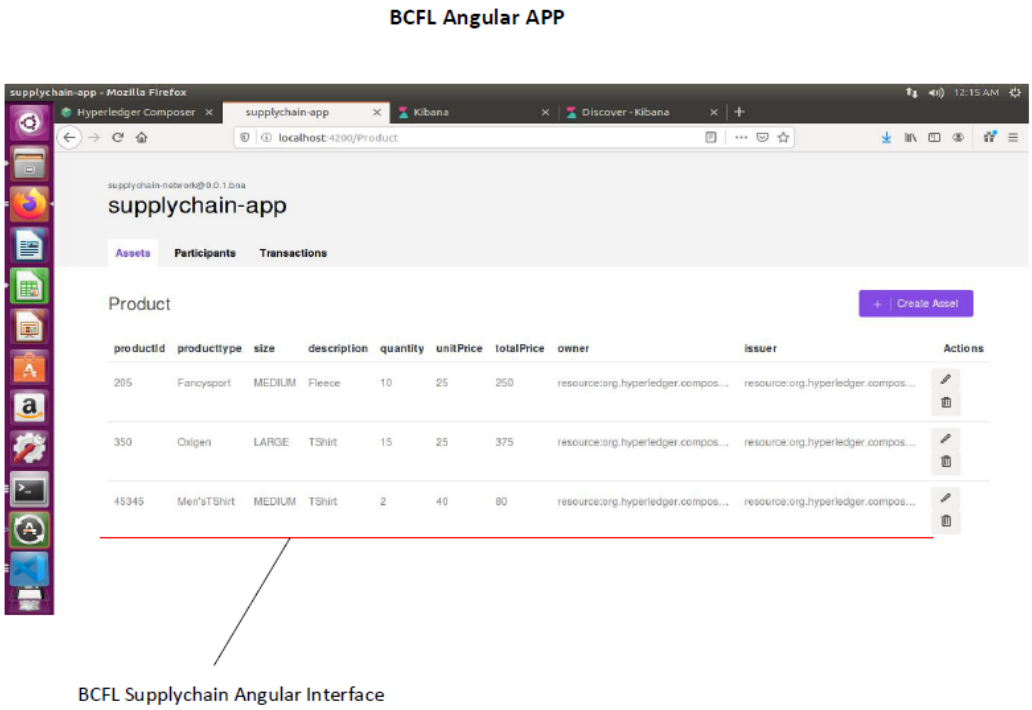


Figure 5.12: *BCFL Angular API App*

Elasticsearch components such as packetbeat and metricbeat were enhanced and integrated into the BCFL framework to facilitate this. In addition, performance moni-

toring testing was carried out with the enhance Elasticsearch components (metricbeat and packetbeat) to view BCFL network containers' overall health. The central processing unit (CPU), system memory, and running processors, including the TCP and UDP packets, were monitored in real-time, therefore, increasing understanding of how BCFL Blockchain Cloud system performance will add clarity in the testing process.

Any disruption in performance will invalidate the test result as the system is not in a fully functional state. Fabric Blockchain application has peers maintaining a shared replicated ledger as the equivalent of a database (Dinh et al., 2018).

5.3.1 Metrics for Performance Analysis

Blockchain network also faces some of the critical issues that a traditional DBMS-base transaction faces. Figure 5.13, and 5.14 are an indicator of the overall application performance as stated earlier. These can range from real-time CPU usage, memory usage, disk I/O speeds, network bandwidth, latency, throughput, and traffic flow. In Blockchain, these performance indicators are essential in the case of all the Docker containers that are running peer, order, CA and CouchDB . Furthermore, Blockchain is a massive processing system which demands more memory usage and more significant storage capacity to maintain efficient application performance.

5.3.2 Performance Metrics

We are going to adopt the two main performance indices of BCFL application performance metrics as shown in Figures 5.13, 5.14, 5.15 and 5.15:

Throughput: A Wireshark statistical throughput performance capability was adopted to measure all the transaction from different stages. This throughput mechanism enhances the capacity to view the internal working of the system from the time

a participant constructs a transaction proposal for endorsement, up to the time when an event indicating ledger commitment is received. It also enables the measurement of the orderer and CouchDB storage throughput, which in turn provided the overall picture of the BCFL Blockchain cloud application performance.

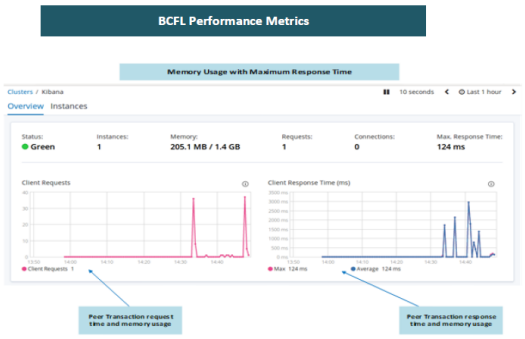


Figure 5.13: *BCFL Memory usages of the performance metric*

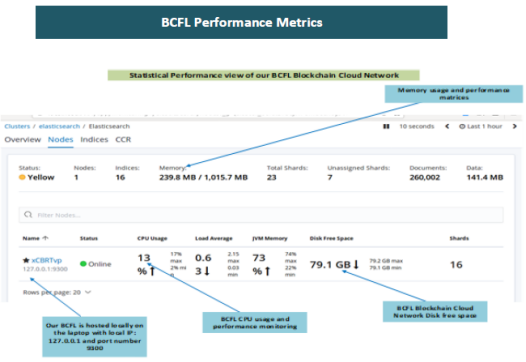


Figure 5.14: *BCFL Memory usage and performance metric*

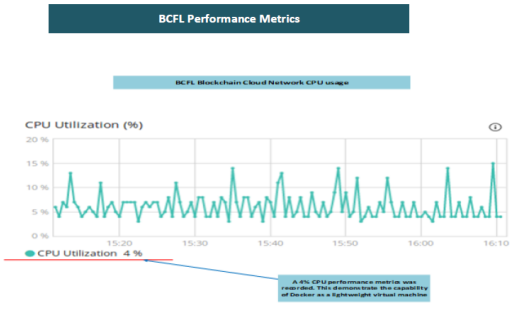


Figure 5.15: *BCFL CPU utilisation usage metric*

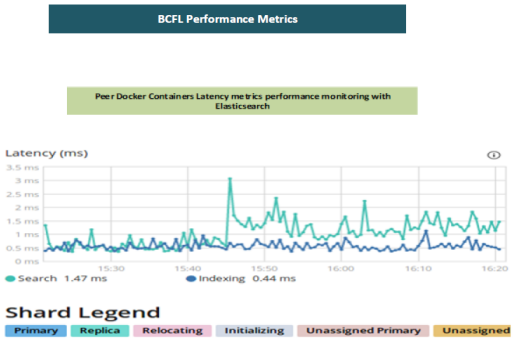


Figure 5.16: *Latency monitoring of BCFL Blockchain Cloud ecosystem*

Latency: In BCFL Hyperledger Fabric Blockchain ecosystem latency performance metrics is essential as to determine the traffic flow of all transaction. CPU usage and other hardware components performance metrics are crucial however, latency traffic flow metrics will facilitate understanding of how each transaction flow from source to destination. It will also enhance the capability to measure chaincode execution and endorsement, ordering and block creation, transaction validation and ledger commitment in real-time (Dinh et al., 2018).

5.4 Summary

The reviewed literature in Chapter 1.2 has established that log evidence in cloud can be altered due to the one-sided relationship between the cloud customer and their service providers. A significant number of experiments have been designed and tested to verify the existing literature on Blockchain cloud architecture and to document the architecture of BCFL. Although the test results are presented, each phase of the experiments answered the research objectives. The system that was designed, built and deployed for this experiment was established on a 64-bit Windows 10 Pro, Enterprise, VMware 15.0.4 workstation. Additionally, on top of the VM workstation a guest OS, Ubuntu 16.04 LTS, 100GB HDD and 8GB RAM were installed for the Hyperledger Fabric Docker container to use which gives the overall system more flexibility and reliance as described in the Methodology, Chapter 3, of the research. Docker version 18.0 and Docker Compose were installed to facilitate a clean installation of Hyperledger Fabric deployment prerequisite. Then BCFL Hyperledger Fabric Cloud network was fully established and running.

Digital forensic stakeholders are looking for a way to secure digital evidence in the cloud ecosystem however, this has become an uphill task. There is no global standard

framework on how digital evidence that is tamperproof can be forensically acquired in the cloud. This chapter introduced the results of the BCFL architectural framework also enables cloud forensic investigators to acquire tamperproof log evidence. BCFL facilitates a secure chain of custody and logs evidence preservation (Awuson-David et al., 2021).

Chapter 6

Conclusion

There is a need for cloud stakeholders to consider Blockchain technology adoption to ensure trustworthiness in the cloud ecosystem. BCFL will improve transactional record by facilitating immutable log evidence between all cloud actors. This chapter introduces the conclusion of this research, and highlights limitations of Blockchain technology. In addition, it emphasises the need for future work in the area of Blockchain Artificial Intelligence integration in the cloud ecosystem that enhances digital forensic acquisition.

6.1 Conclusion

The research adopted Hyperledger Fabric permissioned Blockchain as earlier mentioned in Chapters 1 and Section 1.2. The novelty of this research is the addition of a forensic layer in the Hyperledger Blockchain architecture with the BCFL framework. Table 2.2 and 2.7 explain different types of Blockchain and why Hyperledger Fabric was adopted for this research. BCFL framework was built and designed to enable cloud forensics investigators to acquire admissible digital evidence from the cloud network. Chapter 1 reviewed the definition and types of cloud computing as emphasised by global standard organisations such as CSA, ENISA and NIST. Furthermore, Subsection 2.1.2

also reviewed the three main types of cloud: public, private, and hybrid.

In order to understand a comprehensive view of cloud architecture, the cloud SPI model the SaaS, PaaS, and IaaS, were all reviewed. Cloud computing technology offers its customers on-demand services such as availability, multitenancy, agility, scalability, and security as earlier mentioned in Chapter 2.1.2. However, no framework demonstrated how digital forensics investigation should be conducted in the cloud platforms. The research reviewed and analysed the relationship between CSP, cloud customers and cloud forensic investigators. In Chapter 3, the DSRM methodology was used to support the BCFL framework and define how digital forensics investigation should be conducted in the cloud.

DSRM shows how trust was achieved between the cloud stakeholder, as demonstrated in Figure 3.1. There was clear evidence that highlights lack of trust between the CSP and their customers. These complex relationships have made it challenging to acquire admissible evidence in the cloud. The CSP is responsible for managing security at the physical infrastructure level.

The BCFL framework in Chapter 4 demonstrated Blockchain cloud network integration that facilitated the acquisition of admissible evidence in the cloud ecosystem through trust between all the participants in the BCFL network. It also used the Blockchain smart contract to maintain the integrity of logs and preserved the acquired evidence. This has answered question 1 to 5 of our research questions shown in Subsection 1.1.1.

Chapters 4 and 5 demonstrated that the BCFL framework have solved the challenges (multi-tenancy, geolocation, SLA, evidence tampering and insider threat) of acquiring admissible evidence in the cloud. It also answered all the research questions and objectives in Chapter 1 Subsection 1.1.1 and 1.1.2 by using Blockchain to build trust, transparency, and immutability log evidence in the cloud environment. Chapter

4 Figure 4.9 highlighted the initial result of the BCFL cloud forensic acquisition of admissible log evidence. Additionally, Table 4.1 of Subsection 4.10.2, also in Chapter 4, demonstrated how cloud forensic investigators maintain the acquired admissible evidence chain of custody throughout the investigation process. BCFL framework forensic process has answered the research question and objectives and the research aim in Section 1.1.

The integration of Blockchain into the cloud with Blockchain DLT removes the risk of log tampering in a cloud multitenancy environment. Additionally, Chapter 5 also shows that when the virtual machine is switched off as shown in Figure 5.8, BCFL still maintains the logs' integrity and ensures that the logs are not deleted. Figure 5.10 in Chapter 5 also demonstrated the result of the BCFL acquired Blockchain transaction log with an explicit immutable timestamp that maintains the integrity of acquired logs.

Cloud technology has redefined the way organisations and individuals communicate by eliminating the long delay in cabling and updating the network infrastructure as in a traditional network. However, cloud technology still comes with its own risk that has been exploited by cybercriminals. The disadvantage from digital forensic investigation viewpoint is that the nature of the technology has added a layer of complexity in investigating the cyber-related incidents. Many frameworks, as pointed out in the literature review in Section 1.2, have been developed to solve the challenges of digital forensic acquisition in the cloud ecosystem, but the admissibility of the acquired evidence has not been fully exploited by previous research.

However, little attention has been paid to GDPR compliance directives where digital forensic investigators such as first responders and law enforcement tread carefully in securing the crime scene. Also, maintain chain of custody and have the GDPR compliance to adhere to. In this research, BCFL methodology provides a framework that mitigates the challenges faced by forensic investigators in acquiring admissible log

evidence from the cloud ecosystem, as earlier mentioned (Awuson-David et al., 2021).

The future work will focus on developing an innovative framework that integrates Blockchain and artificial intelligence (AI) in the cloud that enhances the logging and monitoring of anonymous behaviours in the cloud (Awuson-David et al., 2019).

6.2 Limitation

Permissioned and permissionless Blockchain technology is its infancy. It can facilitate decentralised smart logging mechanisms that ensure data security in transit and storage compared to the traditional centralised cloud logging mechanisms. Although at this early stage of Blockchain technology it is useful to highlight its limitations as demonstrated by Gomber, Kauffman, Parker, and Weber (2018); Morkunas, Paschen, and Boon (2019), and Awuson-David et al. (2021); Hughes et al. (2019); Nawari and Ravindran (2019) before deploying the technology in the operational environment.

- Operational Environment: This research experimental case study was carried out on a virtualised cloud network and has not been tried in an operational cloud environment. However, the BCFL implementation mimics the real operational environment with a real permissioned (Hyperledger Fabric) Blockchain technology and a VMware cloud platform (Awuson-David et al., 2021).

- Legal Framework: There is no global standard framework for adopting Blockchain into organisation existing network infrastructure. Lack of standards has delayed potential businesses to secure data logging, monitoring, and storage, including data decentralisation mechanisms that Blockchain offers as highlighted by Saberi, Kouhizadeh, Sarkis, and Shen (2019), and Awuson-David et al. (2021); Janssen et al. (2020).

- Confidentiality: Permissionless Blockchain does not have user privacy, as all participants in the ecosystem can view each other's transactions. For example, in the Bitcoin permissionless Blockchain, each participant transaction can be viewed by all other participants through a consensus, including the miners Peng et al. (2020), and (Awuson-David et al., 2021; Henry, Herzberg, & Kate, 2018). However, in the Hyperledger Fabric, Blockchain confidentiality is integrated into its implementation.

- Security Mechanisms: Blockchain technology maintains data integrity through public-key encryption for authentication and transaction execution across all nodes (Varshney, Sharma, Kaushik, & Bhushan, 2019). However, Blockchain uses a secured process to facilitate security and data immutability. When a participant in the Blockchain network loses or forgets their private key, the system has no mechanism to recover the password for an authorised participant (Awuson-David et al., 2021).

- Limited Flexibility: Blockchain immutability mechanisms ensure all transactions are tamperproof within the Blockchain consensus mechanisms. The immutability mechanisms prevent legitimate use cases that need some level of changes to the transaction data.

6.3 Future Work

In the future, it will be more efficient and practical to adopt a Blockchain AI integration into the cloud ecosystem. The Blockchain AI integration will maintain clear visibility of all transactions and logging, including capturing cloud anonymous behaviours activities and logs. The immutable captured logs will then be preserved for forensics. The AI will

maintain the integrity of the Blockchain database and the DLT mechanisms throughout the transaction process.

The most unambiguous way to simulate and validate this approach is using a hybrid virtualised python anaconda (conda) development environment. The reason behind considering this design and development python anaconda platform is the ability to use the developed AI Blockchain algorithm to interact with the artefacts (digital evidence) and knowledge context (the digital crime scene) (Singla, Bose, & Katariya, 2018), and (G. Zhang, Li, Li, Hui, & Jin, 2018), and (Awuson-David et al., 2021; Bertino, Kundu, & Sura, 2019; Krittanawong et al., 2020).

This thesis has answered all the research objectives and questions, as demonstrated in Chapters 4 and 5 by validating the BCFL framework through the various case and testing phases. The research has extended the capability of permissioned (Hyperledger Fabric) Blockchain by adding a forensic layer that facilitates trustworthiness and digital evidence integrity in the cloud.

Appendix B highlights all the results, screenshots and evaluation of DSRM and BCFL framework. It shows the designed architecture and valid test results with an answer to all the research questions and objectives.

References

Abdelrazik, A., Bunce, G., Cacciatore, K., Hui, K., Mahankali, S., & Van Rooyen, F. (2017). *Adding speed and agility to virtualized infrastructure with openstack.* Tech. Rep.

Ab Rahman, N. H., & Choo, K.-K. R. (2015). A survey of information security incident handling in the cloud. *computers & security*, *49*, 45–69.

Ab Rahman, N. H., Glisson, W. B., Yang, Y., & Choo, K.-K. R. (2016). Forensic-by-design framework for cyber-physical cloud systems. *IEEE Cloud Computing*, *3*(1), 50–59.

Acharya, V., Yerrapati, A. E., & Prakash, N. (2019). *Oracle blockchain quick start guide: A practical approach to implementing blockchain in your enterprise*. Packt Publishing Ltd.

Adjei, J. K. (2015). Explaining the role of trust in cloud computing services. *info*.

Agrawal, R., Bawa, M., & Bayardo, R. J. (2008, August 5). *Uniform search system and method for selectively sharing distributed access-controlled documents.* Google Patents. (US Patent 7,409,406)

Ali, M. (2019). Cloud computing at a cross road: Quality and risks in higher education. *Advances in Internet of Things*, *9*(3), 33–49.

Ali, M., Khan, S. U., & Vasilakos, A. V. (2015). Security in cloud computing: Opportunities and challenges. *Information sciences*, *305*, 357–383.

AlJahdali, H., Albatli, A., Garraghan, P., Townend, P., Lau, L., & Xu, J. (2014). Multi-tenancy in cloud computing. In *2014 ieee 8th international symposium on service oriented system engineering* (pp. 344–351).

Alliance, C. (2009). *Security guidance for critical areas offocus in cloud computing v21.* CSA.

Almeida, A. V. d., Borges, M. M., & Roque, L. (2017). The european open science cloud: A new challenge for europe. In *Proceedings of the 5th international conference on technological ecosystems for enhancing multiculturality* (pp. 1–4).

Almulla, S. A., & Yeun, C. Y. (2011). New secure storage architecture for cloud computing. In *Future information technology* (pp. 75–84). Springer.

Alqahtany, S., Clarke, N., Furnell, S., & Reich, C. (2015). Cloud forensics: a review of challenges, solutions and open problems. In *2015 international conference on cloud computing (iccc)* (pp. 1–9).

Alqahtany, S., Clarke, N., Furnell, S., & Reich, C. (2016). A forensic acquisition and analysis system for iaas. *Cluster Computing*, *19*(1), 439–453.

Amato, F., Castiglione, A., Cozzolino, G., & Narducci, F. (2020). A semantic-based methodology for digital forensics analysis. *Journal of Parallel and Distributed Computing*, *138*, 172–177.

Anderson, K., & Gaston, K. J. (2013). Lightweight unmanned aerial vehicles will revolutionize spatial ecology. *Frontiers in Ecology and the Environment*, *11*(3), 138–146.

Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., ... others (2018). Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the thirteenth eurosys conference* (p. 30).

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., ... others (2010). A view of cloud computing. *Communications of the ACM*, *53*(4), 50–58.

Ateniese, G., Magri, B., Venturi, D., & Andrade, E. (2017). Redactable blockchain–or–rewriting history in bitcoin and friends. In *2017 ieee european symposium on security and privacy (euros&p)* (pp. 111–126).

Awuson-David, K., Al-Hadhrami, T., Alazab, M., Shah, N., & Shalaginov, A. (2021). Bcfl logging: An approach to acquire and preserve admissible digital forensics evidence in cloud ecosystem. *Future Generation Computer Systems*, *122*, 1–13.

Awuson-David, K., Al-Hadhrami, T., Funminiyi, O., & Lotfi, A. (2019). Using hyperledger fabric blockchain to maintain the integrity of digital evidence in a containerised cloud ecosystem. In *International conference of reliable information and communication technology* (pp. 839–848).

Azzi, R., Chamoun, R. K., & Sokhn, M. (2019). The power of a blockchain-based supply chain. *Computers & industrial engineering*, *135*, 582–592.

Bacon, R. (2010). On experimental science. *Bacon R. Opus Majus. URL: http://www. fordham. edu/halsall/source/bacon2. html*, *14*.

Badr, B., Horrocks, R., & Wu, X. B. (2018). *Blockchain by example: A developer's guide to creating decentralized applications using bitcoin, ethereum, and hyperledger*. Packt Publishing Ltd.

Baranwal, G., & Vidyarthi, D. P. (2014). A framework for selection of best cloud service provider using ranked voting method. In *2014 ieee international advance computing conference (iacc)* (pp. 831–837).

Barona, R., & Anita, E. M. (2017). A survey on data breach challenges in cloud computing security: Issues and threats. In *2017 international conference on circuit, power and computing technologies (iccpct)* (pp. 1–8).

Baset, S. A., Desrosiers, L., Gaur, N., Novotny, P., O'Dowd, A., & Ramakrishna, V. (2018). *Hands-on blockchain with hyperledger: building decentralized applications with hyperledger fabric and composer*. Packt Publishing Ltd.

Baset, S. A., Desrosiers, L., Gaur, N., Novotny, P., O'Dowd, A., Ramakrishna, V., ... Wu, X. B. (2019). *Blockchain development with hyperledger: build decentralized applications with hyperledger fabric and composer*. Packt Publishing Ltd.

Bashir, I. (2017). *Mastering blockchain*. Packt Publishing Ltd.

Bashir, I. (2018). *Mastering blockchain: Distributed ledger technology, decentralization, and smart contracts explained*. Packt Publishing Ltd.

Benlian, A., Kettinger, W. J., Sunyaev, A., Winkler, T. J., & EDITORS, G. (2018). The transformative value of cloud computing: a decoupling, platformization, and recombination theoretical framework. *Journal of management information systems*, *35*(3), 719–739.

Berberich, M., & Steiner, M. (2016). Blockchain technology and the gdpr-how to reconcile privacy and distributed ledgers. *Eur. Data Prot. L. Rev.*, *2*, 422.

Bertino, E., Kundu, A., & Sura, Z. (2019). Data transparency with blockchain and ai ethics. *Journal of Data and Information Quality (JDIQ)*, *11*(4), 1–8.

Bessani, A., Sousa, J., & Vukolić, M. (2017). A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform. In *Proceedings of the 1st workshop on scalable and resilient infrastructures for distributed ledgers* (pp. 1–2).

Bettín-Díaz, R., Rojas, A. E., & Mejía-Moncayo, C. (2018). Methodological approach to the definition of a blockchain system for the food industry supply chain traceability. In *International conference on computational science and its applications* (pp. 19–33).

Bhardwaj, S., Jain, L., & Jain, S. (2010). Cloud computing: A study of infrastructure as a service (iaas). *International Journal of engineering and information Technology*, *2*(1), 60–63.

Birk, D., & Wegener, C. (2011). Technical issues of forensic investigations in cloud computing environments. In *2011 sixth ieee international workshop on systematic*

approaches to digital forensic engineering (pp. 1–10).

Bosamia, M., & Patel, D. (2018, 08). Current trends and future implementation possibilities of the merkel tree. *INTERNATIONAL JOURNAL OF COMPUTER SCIENCES AND ENGINEERING*, *6*, 294-301. doi: 10.26438/ijcse/v6i8.294301

Boulos, M. N. K., Wilson, J. T., & Clauson, K. A. (2018). *Geospatial blockchain: promises, challenges, and scenarios in health and healthcare.* BioMed Central.

Brender, N., & Markov, I. (2013). Risk perception and risk management in cloud computing: Results from a case study of swiss companies. *International journal of information management*, *33*(5), 726–733.

Brotsis, S., Kolokotronis, N., Limniotis, K., Shiaeles, S., Kavallieros, D., Bellini, E., & Pavué, C. (2019). Blockchain solutions for forensic evidence preservation in iot environments. In *2019 ieee conference on network softwarization (netsoft)* (pp. 110–114).

Brown, C. S. (2015). Investigating and prosecuting cyber crime: Forensic dependencies and barriers to justice. *International Journal of Cyber Criminology*, *9*(1), 55.

Broy, M., Feilkas, M., Herrmannsdoerfer, M., Merenda, S., & Ratiu, D. (2010). Seamless model-based development: From isolated tools to integrated model engineering environments. *Proceedings of the IEEE*, *98*(4), 526–545.

Budņiks, L., & Didenko, K. (2014). Factors determining application of cloud computing services in latvian smes. *Procedia-Social and Behavioral Sciences*, *156*, 74–77.

Budroni, P., Claude-Burgelman, J., & Schouppe, M. (2019). Architectures of knowledge: the european open science cloud. *ABI Technik*, *39*(2), 130–141.

Buyya, R., Ranjan, R., & Calheiros, R. N. (2010). Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. In *International conference on algorithms and architectures for parallel processing* (pp. 13–31).

Cachin, C., et al. (2016). Architecture of the hyperledger blockchain fabric. In *Workshop on distributed cryptocurrencies and consensus ledgers* (Vol. 310, p. 4).

Carpenter, J. P., & Krutka, D. G. (2014). How and why educators use twitter: A survey of the field. *Journal of research on technology in education*, *46*(4), 414–434.

Casey, E. (2001). *Handbook of computer crime investigation: forensic tools and technology*. Elsevier.

Casey, E. (2011). *Digital evidence and computer crime: Forensic science, computers, and the internet*. Academic press.

Cebe, M., Erdin, E., Akkaya, K., Aksu, H., & Uluagac, S. (2018). Block4forensic: An integrated lightweight blockchain framework for forensics applications of connected vehicles. *IEEE Communications Magazine*, *56*(10), 50–57.

Celesti, A., Fazio, M., Galletta, A., Carnevale, L., Wan, J., & Villari, M. (2019). An approach for the secure management of hybrid cloud–edge environments. *Future Generation Computer Systems*, *90*, 1–19.

Chandramouli, R., Iorga, M., & Chokhani, S. (2014). Cryptographic key management issues and challenges in cloud services. In *Secure cloud computing* (pp. 1–30). Springer.

Chellappa, R. (1997). Intermediaries in cloud-computing: A new computing paradigm. In *Informs annual meeting, dallas* (pp. 26–29).

Chen, Y., Paxson, V., & Katz, R. H. (2010). What's new about cloud computing security. *University of California, Berkeley Report No. UCB/EECS-2010-5 January*, *20*(2010), 2010–5.

Cheng, L., Liu, F., & Yao, D. (2017). Enterprise data breach: causes, challenges, prevention, and future directions. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *7*(5), e1211.

Chung, H., Park, J., Lee, S., & Kang, C. (2012). Digital forensic investigation of cloud

storage services. *Digital investigation*, *9*(2), 81–95.

Cinque, M., Russo, S., Esposito, C., Choo, K.-K. R., Free-Nelson, F., & Kamhoua, C. A. (2018). Cloud reliability: Possible sources of security and legal issues? *IEEE Cloud Computing*, *5*(3), 31–38.

Constantine, D. (2011). Cloud computing: The next great technological innovation, the death of online privacy, or both. *Ga. St. UL Rev.*, *28*, 499.

Corradi, A., Fanelli, M., & Foschini, L. (2014). Vm consolidation: A real case based on openstack cloud. *Future Generation Computer Systems*, *32*, 118–127.

Coutinho, E. F., de Carvalho Sousa, F. R., Rego, P. A. L., Gomes, D. G., & de Souza, J. N. (2015). Elasticity in cloud computing: a survey. *annals of telecommunications-annales des télécommunications*, *70*(7-8), 289–309.

Coyle, D., & Nguyen, D. (2019). Cloud computing, cross-border data flows and new challenges for measurement in economics. *National Institute Economic Review*, *249*(1), R30–R38.

Cucurull, J., & Puiggalí, J. (2016). Distributed immutabilization of secure logs. In *International workshop on security and trust management* (pp. 122–137).

Cui, Y., Gaur, V., & Liu, J. (2020). Blockchain collaboration with competing firms in a shared supply chain: Benefits and challenges. *Available at SSRN*.

Dai, J., & Vasarhelyi, M. A. (2017). Toward blockchain-based accounting and assurance. *Journal of Information Systems*, *31*(3), 5–21.

Damenu, T. K., & Balakrishna, C. (2015). Cloud security risk management: A critical review. In *2015 9th international conference on next generation mobile applications, services and technologies* (pp. 370–375).

Davenport, A., Shetty, S., & Liang, X. (2018). Attack surface analysis of permissioned blockchain platforms for smart cities. In *2018 ieee international smart cities conference (isc2)* (pp. 1–6).

David Schoorman, F., Mayer, R. C., & Davis, J. H. (2016). Empowerment in veterinary clinics: The role of trust in delegation. *Journal of Trust Research*, *6*(1), 76–90.

Dictionary, M.-W. (2010). *Tolerance. dari http://www. merriamwebster. com/dictionary/tolerance.* Diunduh.

Dinh, T. T. A., Liu, R., Zhang, M., Chen, G., Ooi, B. C., & Wang, J. (2018). Untangling blockchain: A data processing view of blockchain systems. *IEEE Transactions on Knowledge and Data Engineering*, *30*(7), 1366–1385.

Dresch, A., Lacerda, D. P., & Antunes, J. A. V. (2015). Design science research. In *Design science research* (pp. 67–102). Springer.

Duncan, B., & Whittington, M. (2015). Company management approaches stewardship or agency: Which promotes better security in cloud ecosystems? *Cloud Comput*, 154–159.

Dykstra, J., & Sherman, A. T. (2012). Acquiring forensic evidence from infrastructure-as-a-service cloud computing: Exploring and evaluating tools, trust, and techniques. *Digital Investigation*, *9*, S90–S98.

Easttom, C. (2016). Applying graph theory to evidence evaluation. *Research Gate*.

Espadas, J., Molina, A., Jiménez, G., Molina, M., Ramírez, R., & Concha, D. (2013). A tenant-based resource allocation model for scaling software-as-a-service applications over cloud computing infrastructures. *Future Generation Computer Systems*, *29*(1), 273–286.

Fafinski, S. (2006). Access denied: Computer misuse in an era of technological change. *The Journal of Criminal Law*, *70*(5), 424–442.

Fanning, K., & Centers, D. P. (2016). Blockchain and its coming impact on financial services. *Journal of Corporate Accounting & Finance*, *27*(5), 53–57.

Farina, J., Scanlon, M., Le-Khac, N.-A., & Kechadi, M.-T. (2015). Overview of the forensic investigation of cloud services. In *2015 10th international conference on*

availability, reliability and security (pp. 556–565).

Fernandes, D. A., Soares, L. F., Gomes, J. V., Freire, M. M., & Inácio, P. R. (2014). Security issues in cloud environments: a survey. *International Journal of Information Security*, *13*(2), 113–170.

Fosso Wamba, S., Kala Kamdjoug, J. R., Epie Bawack, R., & Keogh, J. G. (2020). Bitcoin, blockchain and fintech: a systematic review and case studies in the supply chain. *Production Planning & Control*, *31*(2-3), 115–142.

Freet, D., Agrawal, R., John, S., & Walker, J. J. (2015). Cloud forensics challenges from a service model standpoint: Iaas, paas and saas. In *Proceedings of the 7th international conference on management of computational and collective intelligence in digital ecosystems* (pp. 148–155).

Gai, K., Wu, Y., Zhu, L., Xu, L., & Zhang, Y. (2019). Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks. *IEEE Internet of Things Journal*, *6*(5), 7992–8004.

Galante, G., De Bona, L. C. E., Mury, A. R., Schulze, B., & da Rosa Righi, R. (2016). An analysis of public clouds elasticity in the execution of scientific applications: a survey. *Journal of Grid Computing*, *14*(2), 193–216.

Garfinkel, S. L. (2010). Digital forensics research: The next 10 years. *digital investigation*, *7*, S64–S73.

Garrison, C. P. (2010). *Digital forensics for network, internet, and cloud computing: A forensic evidence guide for moving targets and data*. Syngress.

Gaur, N., Desrosiers, L., Ramakrishna, V., Novotny, P., Baset, S. A., & O'Dowd, A. (2018). *Hands-on blockchain with hyperledger: Building decentralized applications with hyperledger fabric and composer*. Packt Publishing Ltd.

Geerts, G. L. (2011). A design science research methodology and its application to accounting information systems research. *International journal of accounting*

Information Systems, *12*(2), 142–151.

Giannoutakis, K. M., & Tzovaras, D. (2016). The european strategy in research infrastructures and open science cloud. In *International conference on data analytics and management in data intensive domains* (pp. 207–221).

Gomber, P., Kauffman, R. J., Parker, C., & Weber, B. W. (2018). On the fintech revolution: Interpreting the forces of innovation, disruption, and transformation in financial services. *Journal of Management Information Systems*, *35*(1), 220–265.

Gorenflo, C., Lee, S., Golab, L., & Keshav, S. (2020). Fastfabric: Scaling hyperledger fabric to 20 000 transactions per second. *International Journal of Network Management*, *30*(5), e2099.

Goyal, S. (2014). Public vs private vs hybrid vs community-cloud computing: a critical review. *International Journal of Computer Network and Information Security*, *6*(3), 20.

Gramoli, V. (2020). From blockchain consensus back to byzantine consensus. *Future Generation Computer Systems*, *107*, 760–769.

Gregory, R. W., & Muntermann, J. (2014). Research note—heuristic theorizing: proactively generating design theories. *Information Systems Research*, *25*(3), 639–653.

Grispos, G., Storer, T., & Glisson, W. B. (2012). Calm before the storm: The challenges of cloud computing in digital forensics. *International Journal of Digital Crime and Forensics (IJDCF)*, *4*(2), 28–48.

Group, N. C. C. F. S. W., et al. (2014). *Nist cloud computing forensic science challenges* (Tech. Rep.). National Institute of Standards and Technology.

Guo, H., Jin, B., & Shang, T. (2012). Forensic investigations in cloud environments. In *2012 international conference on computer science and information processing (csip)* (pp. 248–251).

Han, R., Guo, L., Ghanem, M. M., & Guo, Y. (2012). Lightweight resource scaling for cloud applications. In *2012 12th ieee/acm international symposium on cluster, cloud and grid computing (ccgrid 2012)* (pp. 644–651).

Han, X., Kheir, N., & Balzarotti, D. (2015). The role of cloud services in malicious software: Trends and insights. In *International conference on detection of intrusions and malware, and vulnerability assessment* (pp. 187–204).

Hand, R., Ton, M., & Keller, E. (2013). Active security. In *Proceedings of the twelfth acm workshop on hot topics in networks* (pp. 1–7).

Hayes, D. R. (2015). *A practical guide to computer forensics investigations*. Pearson Education.

Henry, R., Herzberg, A., & Kate, A. (2018). Blockchain access privacy: Challenges and directions. *IEEE Security & Privacy*, *16*(4), 38–45.

Hevner, A., & Chatterjee, S. (2010). Design science research frameworks. In *Design research in information systems* (pp. 23–31). Springer.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS quarterly*, 75–105.

Hill, B., Chopra, S., & Valencourt, P. (2018). *Blockchain quick reference: A guide to exploring decentralized blockchain application development*. Packt Publishing Ltd.

Hogan, M., Liu, F., Sokol, A., & Tong, J. (2011). Nist cloud computing standards roadmap. *NIST Special Publication*, *35*, 6–11.

Honar Pajooh, H., Rashid, M., Alam, F., & Demidenko, S. (2021). Hyperledger fabric blockchain for securing the edge internet of things. *Sensors*, *21*(2), 359.

Hooper, C., Martini, B., & Choo, K.-K. R. (2013). Cloud computing and its implications for cybercrime investigations in australia. *Computer Law & Security Review*, *29*(2), 152–163.

Hoque, N., Bhattacharyya, D. K., & Kalita, J. K. (2015). Botnet in ddos attacks: trends and challenges. *IEEE Communications Surveys & Tutorials*, *17*(4), 2242–2270.

Horsman, G. (2020). Acpo principles for digital evidence: Time for an update? *Forensic Science International: Reports*, *2*, 100076.

Hosseinian-Far, A., Ramachandran, M., & Slack, C. L. (2018). Emerging trends in cloud computing, big data, fog computing, iot and smart living. In *Technology for smart futures* (pp. 29–40). Springer.

Hudic, A., Smith, P., & Weippl, E. R. (2017). Security assurance assessment methodology for hybrid clouds. *Computers & Security*, *70*, 723–743.

Hughes, L., Dwivedi, Y. K., Misra, S. K., Rana, N. P., Raghavan, V., & Akella, V. (2019). Blockchain research, practice and policy: Applications, benefits, limitations, emerging research themes and research agenda. *International Journal of Information Management*, *49*, 114–129.

Ibrahim, A. A. Z. A., Kliazovich, D., & Bouvry, P. (2016). Service level agreement assurance between cloud services providers and cloud customers. In *2016 16th ieee/acm international symposium on cluster, cloud and grid computing (ccgrid)* (pp. 588–591).

Jansen, W. A. (2011). Cloud hooks: Security and privacy issues in cloud computing. In *2011 44th hawaii international conference on system sciences* (pp. 1–10).

Janssen, M., Weerakkody, V., Ismagilova, E., Sivarajah, U., & Irani, Z. (2020). A framework for analysing blockchain technology adoption: Integrating institutional, market and technical factors. *International Journal of Information Management*, *50*, 302–309.

Jasmine, C. A. (2019). Impacts of covid-19 on company and efforts to support organization adaptable. *Dr. David F. Rico, PMP, ACP, CSM*, 67–70.

Jasti, A., Shah, P., Nagaraj, R., & Pendse, R. (2010). Security in multi-tenancy cloud. In *44th annual 2010 ieee international carnahan conference on security technology* (pp. 35–41).

Johng, H., Kim, D., Hill, T., & Chung, L. (2018). Using blockchain to enhance the trustworthiness of business processes: a goal-oriented approach. In *2018 ieee international conference on services computing (scc)* (pp. 249–252).

Kaewkasi, C. (2018). *Docker for serverless applications: Containerize and orchestrate functions using openfaas, openwhisk, and fn.* Packt Publishing Ltd.

Kalaiprasath, R., Elankavi, R., & Udayakumar, D. R. (2017). Cloud information accountability (cia) framework ensuring accountability of data in cloud and security in end to end process in cloud terminology. *International Journal Of Civil Engineering And Technology (Ijciet) Volume*, *8*, 376–385.

Katilu, V. M., Franqueira, V. N., & Angelopoulou, O. (2015). Challenges of data provenance for cloud forensic investigations. In *2015 10th international conference on availability, reliability and security* (pp. 312–317).

Kaufman, L. M. (2009). Data security in the world of cloud computing. *IEEE Security & Privacy*, *7*(4), 61–64.

Kavis, M. J. (2014). *Architecting the cloud: design decisions for cloud computing service models (saas, paas, and iaas).* John Wiley & Sons.

Khan, S., Gani, A., Wahab, A. W. A., Bagiwa, M. A., Shiraz, M., Khan, S. U., ... Zomaya, A. Y. (2016). Cloud log forensics: Foundations, state of the art, and future directions. *ACM Computing Surveys (CSUR)*, *49*(1), 1–42.

Ko, R. K., Lee, B. S., & Pearson, S. (2011). Towards achieving accountability, auditability and trust in cloud computing. In *International conference on advances in computing and communications* (pp. 432–444).

Krittanawong, C., Rogers, A. J., Aydar, M., Choi, E., Johnson, K. W., Wang, Z., &

Narayan, S. M. (2020). Integrating blockchain technology with artificial intelligence for cardiovascular medicine. *Nature Reviews Cardiology*, *17*(1), 1–3.

Kroes, N. (2010). The critical role of cities in making the digital agenda a reality. *Closing speech to Global Cities Dialogue Spring Summit of Mayors, European Commission-SPEECH/10/272, Brussels*, *28*.

Krystlik, J. (2017). With gdpr, preparation is everything. *Computer Fraud & Security*, *2017*(6), 5–8.

Kulkarni, S., Agrawal, P., et al. (2014). *Analysis of tcp performance in data center networks*. Springer.

Kumar, P. R., Raj, P. H., & Jelciana, P. (2018). Exploring data security issues and solutions in cloud computing. *Procedia Computer Science*, *125*, 691–697.

Kuo, T.-T., Kim, H.-E., & Ohno-Machado, L. (2017). Blockchain distributed ledger technologies for biomedical and health care applications. *Journal of the American Medical Informatics Association*, *24*(6), 1211–1220.

Kuzlu, M., Pipattanasomporn, M., Gurses, L., & Rahman, S. (2019). Performance analysis of a hyperledger fabric blockchain framework: throughput, latency and scalability. In *2019 ieee international conference on blockchain (blockchain)* (pp. 536–540).

Laili, Y., Tao, F., Zhang, L., Cheng, Y., Luo, Y., & Sarker, B. R. (2013). A ranking chaos algorithm for dual scheduling of cloud service and computing resource in private cloud. *Computers in Industry*, *64*(4), 448–463.

Laurier, W., Kiehn, J., & Polovina, S. (2018). Rea 2: A unified formalisation of the resource-event-agent ontology. *Applied Ontology*, *13*(3), 201–224.

Leavitt, N. (2009). Is cloud computing really ready for prime time? *Computer*(1), 15–20.

Lemoudden, M., Bouazza, N., & Ouahidi, B. (2014). Towards achieving discernment

and correlation in cloud logging. *Proceedings of the Applications of Information Systems in Engineering and Bioscience*, 202–207.

Li, H., Yu, L., & He, W. (2019). *The impact of gdpr on global technology development.* Taylor & Francis.

Li, J., Wu, J., & Chen, L. (2018). Block-secure: Blockchain based scheme for secure p2p cloud storage. *Information Sciences*, *465*, 219–231.

Li, J., Wu, J., Jiang, G., & Srikanthan, T. (2020). Blockchain-based public auditing for big data in cloud storage. *Information Processing & Management*, *57*(6), 102382.

Li, M., Lal, C., Conti, M., & Hu, D. (2020). Lechain: A blockchain-based lawful evidence management scheme for digital forensics. *Future Generation Computer Systems*.

Li, S., Qin, T., & Min, G. (2019). Blockchain-based digital forensics investigation framework in the internet of things and social systems. *IEEE Transactions on Computational Social Systems*, *6*(6), 1433–1441.

Liang, X., Shetty, S., Tosh, D., Kamhoua, C., Kwiat, K., & Njilla, L. (2017). Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. In *Proceedings of the 17th ieee/acm international symposium on cluster, cloud and grid computing* (pp. 468–477).

Lillard, T. V. (2010). *Digital forensics for network, internet, and cloud computing: a forensic evidence guide for moving targets and data.* Syngress Publishing.

Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L., & Leaf, D. (2011). Nist cloud computing reference architecture. *NIST special publication*, *500*(2011), 1–28.

Lone, A. H., & Mir, R. N. (2019). Forensic-chain: Blockchain based digital forensics chain of custody with poc in hyperledger composer. *Digital Investigation*, *28*,

44–55.

Loureiro, S. (2021). Security misconfigurations and how to prevent them. *Network Security*, *2021*(5), 13–16.

Lu, Y. (2018). Blockchain and the related issues: a review of current research topics. *Journal of Management Analytics*, *5*(4), 231–255.

Mainelli, M., & Smith, M. (2015). Sharing ledgers for sharing economies: an exploration of mutual distributed ledgers (aka blockchain technology). *Journal of Financial Perspectives*, *3*(3).

Makhlouf, R. (2020). Cloudy transaction costs: a dive into cloud computing economics. *Journal of Cloud Computing*, *9*(1), 1–11.

Manoj, S. K. A., Bhaskari, D. L., et al. (2016). Cloud forensics-a framework for investigating cyber attacks in cloud environment. *Procedia Computer Science*, *85*, 149–154.

Manral, B., Somani, G., Choo, K.-K. R., Conti, M., & Gaur, M. S. (2019). A systematic survey on cloud forensics challenges, solutions, and future directions. *ACM Computing Surveys (CSUR)*, *52*(6), 1–38.

Manvi, S. S., & Shyam, G. K. (2014). Resource management for infrastructure as a service (iaas) in cloud computing: A survey. *Journal of network and computer applications*, *41*, 424–440.

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision support systems*, *15*(4), 251–266.

Marinos, A., & Briscoe, G. (2009). Community cloud computing. In *Ieee international conference on cloud computing* (pp. 472–484).

Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J., & Ghalsasi, A. (2011). Cloud computing—the business perspective. *Decision support systems*, *51*(1), 176–189.

Martini, B., & Choo, K.-K. R. (2012). An integrated conceptual digital forensic

framework for cloud computing. *Digital Investigation*, *9*(2), 71–80.

Martini, B., & Choo, K.-K. R. (2014). Cloud forensic technical challenges and solutions: A snapshot. *IEEE Cloud Computing*, *1*(4), 20–25.

Marturana, F., Me, G., & Tacconi, S. (2012). A case study on digital forensics in the cloud. In *2012 international conference on cyber-enabled distributed computing and knowledge discovery* (pp. 111–116).

Marty, R. (2011). Cloud application logging for forensics. In *proceedings of the 2011 acm symposium on applied computing* (pp. 178–184).

McKemmish, R. (1999). *What is forensic computing?* Australian Institute of Criminology Canberra.

McKemmish, R. (2008). When is digital evidence forensically sound? In *Ifip international conference on digital forensics* (pp. 3–15).

Mingxiao, D., Xiaofeng, M., Zhe, Z., Xiangwei, W., & Qijun, C. (2017). A review on consensus algorithm of blockchain. In *2017 ieee international conference on systems, man, and cybernetics (smc)* (pp. 2567–2572).

Mohamed, H., & Ali, H. (2018). *Blockchain, fintech, and islamic finance: Building the future in the new islamic digital economy*. Walter de Gruyter GmbH & Co KG.

Montasari, R. (2017). A standardised data acquisition process model for digital forensic investigations. *International Journal of Information and Computer Security*, *9*(3), 229–249.

Montasari, R., & Hill, R. (2019). Next-generation digital forensics: Challenges and future paradigms. In *2019 ieee 12th international conference on global security, safety and sustainability (icgs3)* (pp. 205–212).

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

Mullarkey, M. T., & Hevner, A. R. (2015). Entering action design research. In

International conference on design science research in information systems (pp. 121–134).

Muthurajkumar, S., Ganapathy, S., Vijayalakshmi, M., & Kannan, A. (2015). Secured temporal log management techniques for cloud. *Procedia Computer Science*, *46*, 589–595.

Nakamoto, S. (2019). *Bitcoin: A peer-to-peer electronic cash system* (Tech. Rep.). Manubot.

Nakamoto, S., & Bitcoin, A. (2008). A peer-to-peer electronic cash system. *Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf*.

Nakamoto, S., et al. (2008). *Bitcoin: A peer-to-peer electronic cash system.(2008).*

Nawari, N. O., & Ravindran, S. (2019). Blockchain and the built environment: Potentials and limitations. *Journal of Building Engineering*, *25*, 100832.

Nidumolu, S. R., Menon, N. M., & Zeigler, B. P. (1998). Object-oriented business process modeling and simulation:: A discrete event system specification framework. *Simulation Practice and Theory*, *6*(6), 533–571.

Niranjanamurthy, M., Nithya, B., & Jagannatha, S. (2019). Analysis of blockchain technology: pros, cons and swot. *Cluster Computing*, *22*(6), 14743–14757.

NIST, S. (2012). *800-145: The nist definition of cloud computing.*

Noura, H. N., Salman, O., Chehab, A., & Couturier, R. (2020). Distlog: A distributed logging scheme for iot forensics. *Ad Hoc Networks*, *98*, 102061.

Nyaletey, E., Parizi, R. M., Zhang, Q., & Choo, K.-K. R. (2019). Blockipfs-blockchain-enabled interplanetary file system for forensic and trusted data traceability. In *2019 ieee international conference on blockchain (blockchain)* (pp. 18–25).

Ølnes, S., Ubacht, J., & Janssen, M. (2017). *Blockchain in government: Benefits and implications of distributed ledger technology for information sharing.* Elsevier.

Olsen, P., Borit, M., & Syed, S. (2019). Applications, limitations, costs, and ben-

efits related to the use of blockchain technology in the food industry. *Nofima rapportserie*.

Pahl, C. (2015). Containerization and the paas cloud. *IEEE Cloud Computing*, *2*(3), 24–31.

Pahl, C., & Lee, B. (2015). Containers and clusters for edge cloud architectures–a technology review. In *2015 3rd international conference on future internet of things and cloud* (pp. 379–386).

Palmer, J. (2001). Tracing bodies: gender, genre, and forensic detective fiction. *South Central Review*, *18*(3/4), 54–71.

Pappa, G. L., & Freitas, A. (2009). *Automating the design of data mining algorithms: an evolutionary computation approach*. Springer Science & Business Media.

Patrascu, A., & Patriciu, V.-V. (2014). Logging system for cloud computing forensic environments. *Journal of Control Engineering and Applied Informatics*, *16*(1), 80–88.

Peffers, K., Tuunanen, T., & Niehaves, B. (2018). *Design science research genres: introduction to the special issue on exemplars and criteria for applicable design science research*. Taylor & Francis.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems*, *24*(3), 45–77.

Peng, L., Feng, W., Yan, Z., Li, Y., Zhou, X., & Shimizu, S. (2020). Privacy preservation in permissionless blockchain: A survey. *Digital Communications and Networks*.

Pichan, A., Lazarescu, M., & Soh, S. T. (2015). Cloud forensics: Technical challenges, solutions and comparative analysis. *Digital Investigation*, *13*, 38–57.

Pichan, A., Lazarescu, M., & Soh, S. T. (2018). Towards a practical cloud forensics

logging framework. *Journal of information security and applications*, *42*, 18–28.

Poisel, R., Malzer, E., & Tjoa, S. (2013). Evidence and cloud computing: The virtual machine introspection approach. *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, *4*(1), 135–152.

Popović, K., & Hocenski, Ž. (2010). Cloud computing security issues and challenges. In *The 33rd international convention mipro* (pp. 344–349).

Potey, M. M., & Nikumbh, D. D. (2013). Achieving accountability and secure logging to increase trust in cloud environment. *International Journal of Computer Applications*, *73*(17).

Pourmajidi, W., & Miranskyy, A. (2018). Logchain: Blockchain-assisted log storage. In *2018 ieee 11th international conference on cloud computing (cloud)* (pp. 978–982).

Praehofer, H. (1991). System theoretic formalisms for combined discrete-continuous system simulation. *International Journal of General System*, *19*(3), 226–240.

Prakash, T., Kakkar, M., & Patel, K. (2016). Geo-identification of web users through logs using elk stack. In *2016 6th international conference-cloud system and big data engineering (confluence)* (pp. 606–610).

Putz, B., Menges, F., & Pernul, G. (2019). A secure and auditable logging infrastructure based on a permissioned blockchain. *Computers & Security*, *87*, 101602.

Quang-Hung, N., Nien, P. D., Nam, N. H., Tuong, N. H., & Thoai, N. (2013). A genetic algorithm for power-aware virtual machine allocation in private cloud. In *Information and communication technology-eurasia conference* (pp. 183–191).

Quick, D., Martini, B., & Choo, R. (2013). *Cloud storage forensics*. Syngress.

Raghuram, Y., & Bangalore, S. S. (2016, February 9). *Remote trust attestation and geolocation of servers and clients in cloud computing environments.* Google Patents. (US Patent 9,256,742)

Rahman, A. F. A., Daud, M., & Mohamad, M. Z. (2016). Securing sensor to cloud ecosystem using internet of things (iot) security framework. In *Proceedings of the international conference on internet of things and cloud computing* (pp. 1–5).

Ramgovind, S., Eloff, M. M., & Smith, E. (2010). The management of security in cloud computing. In *2010 information security for south africa* (pp. 1–7).

Rong, C., Nguyen, S. T., & Jaatun, M. G. (2013). Beyond lightning: A survey on security challenges in cloud computing. *Computers & Electrical Engineering*, *39*(1), 47–54.

Rosado, T., & Bernardino, J. (2014). An overview of openstack architecture. In *Proceedings of the 18th international database engineering & applications symposium* (pp. 366–367).

Roschke, S., Cheng, F., & Meinel, C. (2009). Intrusion detection in the cloud. In *2009 eighth ieee international conference on dependable, autonomic and secure computing* (pp. 729–734).

Rousseau, D. M., Sitkin, S. B., Burt, R. S., & Camerer, C. (1998). Not so different after all: A cross-discipline view of trust. *Academy of management review*, *23*(3), 393–404.

Ruan, K., & Carthy, J. (2012). Cloud computing reference architecture and its forensic implications: a preliminary analysis. In *International conference on digital forensics and cyber crime* (pp. 1–21).

Ruan, K., Carthy, J., Kechadi, T., & Baggili, I. (2013). Cloud forensics definitions and critical criteria for cloud forensic capability: An overview of survey results. *Digital Investigation*, *10*(1), 34–43.

Ruan, K., Carthy, J., Kechadi, T., & Crosbie, M. (2011). Cloud forensics. In *Ifip international conference on digital forensics* (pp. 35–46).

Ruparelia, N. B. (2016). *Cloud computing*. Mit Press.

Ryoo, J., Rizvi, S., Aiken, W., & Kissell, J. (2013). Cloud security auditing: challenges and emerging approaches. *IEEE Security & Privacy*, *12*(6), 68–74.

Saberi, S., Kouhizadeh, M., Sarkis, J., & Shen, L. (2019). Blockchain technology and its relationships to sustainable supply chain management. *International Journal of Production Research*, *57*(7), 2117–2135.

Salah, K., Rehman, M. H. U., Nizamuddin, N., & Al-Fuqaha, A. (2019). Blockchain for ai: Review and open research challenges. *IEEE Access*, *7*, 10127–10149.

Samani, R., Reavis, J., & Honan, B. (2014). *Csa guide to cloud computing: Implementing cloud privacy and security*. Syngress.

Samarati, P., di Vimercati, S. D. C., Murugesan, S., & Bojanova, I. (2016). Cloud security: Issues and concerns. *Encyclopedia on cloud computing*, 1–14.

Sandre, A. (2015). *Digital diplomacy: Conversations on innovation in foreign policy*. Rowman & Littlefield.

Sankar, L. S., Sindhu, M., & Sethumadhavan, M. (2017). Survey of consensus protocols on blockchain applications. In *2017 4th international conference on advanced computing and communication systems (icaccs)* (pp. 1–5).

Santos, N., Gummadi, K. P., & Rodrigues, R. (2009). Towards trusted cloud computing. *HotCloud*, *9*(9), 3.

Schenker, D. G. N. (2018). *Deploy, scale, orchestrate, and manage containers with docker and kubernetes*. PACKT.

Schulz, S., Rozenblit, J. W., Mrva, M., & Buchenriede, K. (1998). Model-based codesign. *Computer*, *31*(8), 60–67.

Sharma, P. K., Chen, M.-Y., & Park, J. H. (2017). A software defined fog node based distributed blockchain cloud architecture for iot. *IEEE Access*, *6*, 115–124.

Shirazi, F., Seddighi, A., & Iqbal, A. (2017). Cloud computing security and privacy: an empirical study. In *International conference on human-computer interaction*

(pp. 534–549).

Simmon, E. (2018). Evaluation of cloud computing services based on nist sp 800-145. *NIST Special Publication*, *500*, 322.

Simon, H. A. (2019). *The sciences of the artificial*. MIT press.

Simou, S., Kalloniatis, C., Kavakli, E., & Gritzalis, S. (2014). Cloud forensics: identifying the major issues and challenges. In *International conference on advanced information systems engineering* (pp. 271–284).

Singh, S., Jeong, Y.-S., & Park, J. H. (2016). A survey on cloud computing security: Issues, threats, and solutions. *Journal of Network and Computer Applications*, *75*, 200–222.

Singla, K., Bose, J., & Katariya, S. (2018). Machine learning for secure device personalization using blockchain. In *2018 international conference on advances in computing, communications and informatics (icacci)* (pp. 67–73).

Smith, C., & Kumar, A. (2018). Crypto-currencies–an introduction to not-so-funny moneys. *Journal of Economic Surveys*, *32*(5), 1531–1559.

Sompolinsky, Y., & Zohar, A. (2015). Secure high-rate transaction processing in bitcoin. In *International conference on financial cryptography and data security* (pp. 507–527).

Son, S. J., & Kwon, Y. (2017). Performance of elk stack and commercial system in security log analysis. In *2017 ieee 13th malaysia international conference on communications (micc)* (pp. 187–190).

Sousa, J., Bessani, A., & Vukolic, M. (2018). A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform. In *2018 48th annual ieee/ifip international conference on dependable systems and networks (dsn)* (pp. 51–58).

Stanciu, A. (2017). Blockchain based distributed control system for edge computing. In *2017 21st international conference on control systems and computer science*

(cscs) (pp. 667–671).

Stelly, C., & Roussev, V. (2017). Scarf: A container-based approach to cloud-scale digital forensic processing. *Digital Investigation*, *22*, S39–S47.

Subashini, S., & Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. *Journal of network and computer applications*, *34*(1), 1–11.

Subramanian, N., & Jeyaraj, A. (2018). Recent security challenges in cloud computing. *Computers & Electrical Engineering*, *71*, 28–42.

Szabo, N. (1996). Smart contracts: building blocks for digital markets. *EXTROPY: The Journal of Transhumanist Thought,(16)*, *18*(2).

Takeda, H., Veerkamp, P., & Yoshikawa, H. (1990). Modeling design process. *AI magazine*, *11*(4), 37–37.

Tanwar, S., Parekh, K., & Evans, R. (2020). Blockchain-based electronic healthcare record system for healthcare 4.0 applications. *Journal of Information Security and Applications*, *50*, 102407.

Taylor, M., Haggerty, J., Gresty, D., & Lamb, D. (2011). Forensic investigation of cloud computing systems. *Network Security*, *2011*(3), 4–10.

Team, I. G. P. (2020). *Eu general data protection regulation (gdpr)–an implementation and compliance guide*. IT Governance Ltd.

Thakkar, P., Nathan, S., & Viswanathan, B. (2018). Performance benchmarking and optimizing hyperledger fabric blockchain platform. In *2018 ieee 26th international symposium on modeling, analysis, and simulation of computer and telecommunication systems (mascots)* (pp. 264–276).

Tian, H., Chen, Z., Chang, C.-C., Huang, Y., Wang, T., Huang, Z.-a., ... Chen, Y. (2019). Public audit for operation behavior logs with error locating in cloud storage. *Soft Computing*, *23*(11), 3779–3792.

Tian, Z., Li, M., Qiu, M., Sun, Y., & Su, S. (2019). Block-def: A secure digital evidence framework using blockchain. *Information Sciences*, *491*, 151–165.

Tosh, D. K., Shetty, S., Liang, X., Kamhoua, C. A., Kwiat, K. A., & Njilla, L. (2017). Security implications of blockchain cloud with analysis of block withholding attack. In *Proceedings of the 17th ieee/acm international symposium on cluster, cloud and grid computing* (pp. 458–467).

Truong, N. B., Sun, K., Lee, G. M., & Guo, Y. (2019). Gdpr-compliant personal data management: A blockchain-based solution. *IEEE Transactions on Information Forensics and Security*, *15*, 1746–1761.

Tsai, W.-T., Shao, Q., Sun, X., & Elston, J. (2010). Real-time service-oriented cloud computing. In *2010 6th world congress on services* (pp. 473–478).

Turnbull, J. (2014). *The docker book: Containerization is the new virtualization*. James Turnbull.

Ullah, S., Xuefeng, Z., & Feng, Z. (2013). Tcloud: a multi–factor access control framework for cloud computing. *International Journal of Security and Its Applications*, *7*(2), 15–26.

Vacca, J. R. (2016). *Cloud computing security: foundations and challenges*. CRC Press.

Vaishnavi, V., Kuechler, W., & Petter, S. (2004). Design science research in information systems. *January*, *20*, 2004.

Varshney, T., Sharma, N., Kaushik, I., & Bhushan, B. (2019). Authentication & encryption based security services in blockchain technology. In *2019 international conference on computing, communication, and intelligent systems (icccis)* (pp. 63–68).

Verginadis, Y., Michalas, A., Gouvas, P., Schiefer, G., Hübsch, G., & Paraskakis, I. (2017). Paasword: A holistic data privacy and security by design framework for

cloud services. *Journal of Grid Computing*, *15*(2), 219–234.

Vincze, E. A. (2016). Challenges in digital forensics. *Police Practice and Research*, *17*(2), 183–194.

Viriyasitavat, W., Da Xu, L., Bi, Z., & Sapsomboon, A. (2018). Blockchain-based business process management (bpm) framework for service composition in industry 4.0. *Journal of Intelligent Manufacturing*, 1–12.

Voigt, P., & Von dem Bussche, A. (2017). The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*.

vom Brocke, J., Hevner, A., & Maedche, A. (2020). Introduction to design science research. In *Design science research. cases* (pp. 1–13). Springer.

Wahyudi, E., Riadi, I., & Prayudi, Y. (2018). Virtual machine forensic analysis and recovery method for recovery and analysis digital evidence. *International Journal of Computer Science and Information Security*, *16*.

Wainer, G. A., & Mosterman, P. J. (2016). *Discrete-event modeling and simulation: theory and applications*. CRC press.

Wang, C., Chow, S. S., Wang, Q., Ren, K., & Lou, W. (2011). Privacy-preserving public auditing for secure cloud storage. *IEEE transactions on computers*, *62*(2), 362–375.

Wani, M. A., AlZahrani, A., & Bhat, W. A. (2020). File system anti-forensics–types, techniques and tools. *Computer Fraud & Security*, *2020*(3), 14–19.

Warren, J. R., MacArthur, P. J., & Crosslin, R. L. (1994). A dynamic modeling toolkit to add rigor to business process re-engineering. In *Hicss (4)* (pp. 683–692).

Wattenhofer, R. (2017). *Distributed ledger technology: The science of the blockchain*. CreateSpace Independent Publishing Platform.

Wei, Y.-C., Wu, W.-C., & Chu, Y.-C. (2018). Performance evaluation of the recommendation mechanism of information security risk identification. *Neurocomputing*,

279, 48–53.

Weissman, C. D., & Bobrowski, S. (2009). The design of the force. com multitenant internet application development platform. In *Proceedings of the 2009 acm sigmod international conference on management of data* (pp. 889–896).

Wieringa, R. J. (2014). *Design science methodology for information systems and software engineering*. Springer.

Xia, Q., Sifah, E. B., Smahi, A., Amofa, S., & Zhang, X. (2017). Bbds: Blockchain-based data sharing for electronic medical records in cloud environments. *Information*, *8*(2), 44.

Xia, Z., Wang, X., Sun, X., & Wang, Q. (2015). A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. *IEEE transactions on parallel and distributed systems*, *27*(2), 340–352.

Xu, X., Weber, I., Staples, M., Zhu, L., Bosch, J., Bass, L., ... Rimba, P. (2017). A taxonomy of blockchain-based systems for architecture design. In *2017 ieee international conference on software architecture (icsa)* (pp. 243–252).

Xun (Brian) Wu, W. S. (2018). *A beginner's guide to developing enterprise-grade decentralized applications*. PACKT.

Yan, Q., & Yu, F. R. (2015). Distributed denial of service attacks in software-defined networking with cloud computing. *IEEE Communications Magazine*, *53*(4), 52–59.

Yang, C., Huang, Q., Li, Z., Liu, K., & Hu, F. (2017). Big data and cloud computing: innovation opportunities and challenges. *International Journal of Digital Earth*, *10*(1), 13–53.

Yeluri, R., & Castro-Leon, E. (2014). *Building the infrastructure for cloud security: A solutions view*. Springer Nature.

Yin, S., Zhang, N., & Dong, H. (2020). Preventing covid-19 from the perspective of

industrial information integration: Evaluation and continuous improvement of information networks for sustainable epidemic prevention. *Journal of Industrial Information Integration*, *19*, 100157.

Zawoad, S., Dutta, A. K., & Hasan, R. (2013). Seclaas: secure logging-as-a-service for cloud forensics. In *Proceedings of the 8th acm sigsac symposium on information, computer and communications security* (pp. 219–230).

Zawoad, S., Dutta, A. K., & Hasan, R. (2015). Towards building forensics enabled cloud through secure logging-as-a-service. *IEEE Transactions on Dependable and Secure Computing*, *13*(2), 148–162.

Zawoad, S., & Hasan, R. (2013). Cloud forensics: a meta-study of challenges, approaches, and open problems. *arXiv preprint arXiv:1302.6312*.

Zeigler, B. P. (1976). Theory of modeling and simulation. jhon wiley & sons. *Inc., New York, NY*.

Zeigler, B. P. (2014). *Object-oriented simulation with hierarchical, modular models: intelligent agents and endomorphic systems.* Academic press.

Zelkowitz, M. V., & Wallace, D. R. (1998). Experimental models for validating technology. *Computer*, *31*(5), 23–31.

Zhang, F., Huang, Y., Wang, H., Chen, H., & Zang, B. (2008). Palm: security preserving vm live migration for systems with vmm-enforced protection. In *2008 third asia-pacific trusted infrastructure technologies conference* (pp. 9–18).

Zhang, G., Li, T., Li, Y., Hui, P., & Jin, D. (2018). Blockchain-based data sharing system for ai-powered network operations. *Journal of Communications and Information Networks*, *3*(3), 1–8.

Zhang, Y. (2015, June 23). *Network bandwidth allocation in multi-tenancy cloud computing networks.* Google Patents. (US Patent 9,065,734)

Zhang, Y., Juels, A., Reiter, M. K., & Ristenpart, T. (2014). Cross-tenant side-

channel attacks in paas clouds. In *Proceedings of the 2014 acm sigsac conference on computer and communications security* (pp. 990–1003).

Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017). An overview of blockchain technology: Architecture, consensus, and future trends. In *2017 ieee international congress on big data (bigdata congress)* (pp. 557–564).

Zimmerman, S., & Glavach, D. (2011). Cyber forensics in the cloud. *IA Newsletter*, *14*(1), 4–7.

Zissis, D., & Lekkas, D. (2012). Addressing cloud computing security issues. *Future Generation computer systems*, *28*(3), 583–592.

Appendices

Appendix A

BCFL Code

The Angular App is integrated into BCFL designed architecture to enable an App interface face that easy administration and facilitate a secure environment for BCFL supplychain system administrators:

```
/*
org.Supplychain Network Angular App Code
*/

import { AngularTestPage } from './app.po';
import { ExpectedConditions, browser, element, by } from 'protractor';
import {} from 'jasmine';


describe('Starting tests for supplychain-app', function() {
let page: AngularTestPage;
```

```
beforeEach(() => {
page = new AngularTestPage();
});

it('website title should be supplychain-app', () => {
page.navigateTo('/');
return browser.getTitle().then((result)=>{
expect(result).toBe('supplychain-app');
})
});

it('network-name should be supplychain-network@0.0.1',() => {
element(by.css('.network-name')).getWebElement()
.then((webElement) => {
return webElement.getText();
})
.then((txt) => {
expect(txt).toBe('supplychain-network@0.0.1.bna');
});
});

it('navbar-brand should be supplychain-app',() => {
element(by.css('.navbar-brand')).getWebElement()
.then((webElement) => {
return webElement.getText();
})
```

```
.then((txt) => {
expect(txt).toBe('supplychain-app');
});
});


it('Product component should be loadable',() => {
page.navigateTo('/Product');
browser.findElement(by.id('assetName'))
.then((assetName) => {
return assetName.getText();
})
.then((txt) => {
expect(txt).toBe('Product');
});
});

it('Product table should have 10 columns',() => {
page.navigateTo('/Product');
element.all(by.css('.thead-cols th')).then(function(arr) {
expect(arr.length).toEqual(10); // Addition of 1 for 'Action' column
});
});

it('Customer component should be loadable',() => {
page.navigateTo('/Customer');
```

```
browser.findElement(by.id('participantName'))
.then((participantName) => {
return participantName.getText();
})
.then((txt) => {
expect(txt).toBe('Customer');
});
});

it('Customer table should have 5 columns',() => {
page.navigateTo('/Customer');
element.all(by.css('.thead-cols th')).then(function(arr) {
expect(arr.length).toEqual(5); // Addition of 1 for 'Action' column
});
});

it('Manufacturer component should be loadable',() => {
page.navigateTo('/Manufacturer');
browser.findElement(by.id('participantName'))
.then((participantName) => {
return participantName.getText();
})
.then((txt) => {
expect(txt).toBe('Manufacturer');
});
});
```

```
it('Manufacturer table should have 5 columns',() => {
page.navigateTo('/Manufacturer');
element.all(by.css('.thead-cols th')).then(function(arr) {
expect(arr.length).toEqual(5); // Addition of 1 for 'Action' column
});
});

it('Distributor component should be loadable',() => {
page.navigateTo('/Distributor');
browser.findElement(by.id('participantName'))
.then((participantName) => {
return participantName.getText();
})
.then((txt) => {
expect(txt).toBe('Distributor');
});
});

it('Distributor table should have 5 columns',() => {
page.navigateTo('/Distributor');
element.all(by.css('.thead-cols th')).then(function(arr) {
expect(arr.length).toEqual(5); // Addition of 1 for 'Action' column
});
});
```

```
it('Retailer component should be loadable',() => {
page.navigateTo('/Retailer');
browser.findElement(by.id('participantName'))
.then((participantName) => {
return participantName.getText();
})
.then((txt) => {
expect(txt).toBe('Retailer');
});
});

it('Retailer table should have 5 columns',() => {
page.navigateTo('/Retailer');
element.all(by.css('.thead-cols th')).then(function(arr) {
expect(arr.length).toEqual(5); // Addition of 1 for 'Action' column
});
});



it('MoveProduct component should be loadable',() => {
page.navigateTo('/MoveProduct');
browser.findElement(by.id('transactionName'))
.then((transactionName) => {
return transactionName.getText();
})
```

```
.then((txt) => {
expect(txt).toBe('MoveProduct');
});
});



});


\textbf{Here is the codes for the BCFL Angular APP user interface}


import { Component, OnInit, Input } from '@angular/core';
import { FormGroup, FormControl, Validators, FormBuilder }
from '@angular/forms';
import { RetailerService } from './Retailer.service';
import 'rxjs/add/operator/toPromise';

@Component({
  selector: 'app-retailer',
  templateUrl: './Retailer.component.html',
  styleUrls: ['./Retailer.component.css'],
  providers: [RetailerService]
})
export class RetailerComponent implements OnInit {

  myForm: FormGroup;
```

```
private allParticipants;
private participant;
private currentId;
private errorMessage;

email = new FormControl('', Validators.required);
firstName = new FormControl('', Validators.required);
lastName = new FormControl('', Validators.required);
type = new FormControl('', Validators.required);


constructor(public serviceRetailer: RetailerService,
fb: FormBuilder) {
  this.myForm = fb.group({
    email: this.email,
    firstName: this.firstName,
    lastName: this.lastName,
    type: this.type
  });
};

ngOnInit(): void {
  this.loadAll();
}
```

```
loadAll(): Promise<any> {
  const tempList = [];
  return this.serviceRetailer.getAll()
  .toPromise()
  .then((result) => {
    this.errorMessage = null;
    result.forEach(participant => {
      tempList.push(participant);
    });
    this.allParticipants = tempList;
  })
  .catch((error) => {
    if (error === 'Server error') {
      this.errorMessage = 'Could not connect to REST server. Please
      check your configuration details';
    } else if (error === '404 - Not Found') {
      this.errorMessage = '404 - Could not find API route. Please
      check your available APIs.';
      this.errorMessage = error;
    }
  });
}


      /**
 * Event handler for changing the checked state of a
 checkbox (handles array enumeration values)
```

```
   * @param {String} name - the name of the participant field to update
   * @param {any} value - the enumeration value for which
   to toggle the checked state
   */
  changeArrayValue(name: string, value: any): void {
    const index = this[name].value.indexOf(value);
    if (index === -1) {
      this[name].value.push(value);
    } else {
      this[name].value.splice(index, 1);
    }
  }


        /*
        * Checkbox helper, determining whether an enumeration value should
        be selected or not (for array enumeration values
   * only). This is used for checkboxes in the participant updateDialog.
   * @param {String} name - the name of the participant field to check
   * @param {any} value - the enumeration value to check for
   * @return {Boolean} whether the
   specified participant field contains the provided value
   */
  hasArrayValue(name: string, value: any): boolean {
    return this[name].value.indexOf(value) !== -1;
  }
```

```
  addParticipant(form: any): Promise<any> {
    this.participant = {
      $class: 'org.supplychain.network.Retailer',
      'email': this.email.value,
      'firstName': this.firstName.value,
      'lastName': this.lastName.value,
      'type': this.type.value
    };

    this.myForm.setValue({
      'email': null,
      'firstName': null,
      'lastName': null,
      'type': null
    });

    return this.serviceRetailer.addParticipant(this.participant)
    .toPromise()
    .then(() => {
      this.errorMessage = null;
      this.myForm.setValue({
        'email': null,
        'firstName': null,
        'lastName': null,
        'type': null
      });
```

```
      this.loadAll();
    })
    .catch((error) => {
      if (error === 'Server error') {
        this.errorMessage = 'Could not connect to REST server. Please
        check your configuration details';
      } else {
        this.errorMessage = error;
      }
    });
  }


  updateParticipant(form: any): Promise<any> {
    this.participant = {
      $class: 'org.supplychain.network.Retailer',
      'firstName': this.firstName.value,
      'lastName': this.lastName.value,
      'type': this.type.value
    };

    return this.serviceRetailer.updateParticipant(form.get('email').value,
    .toPromise()
    .then(() => {
      this.errorMessage = null;
      this.loadAll();
```

```
    })
    .catch((error) => {
      if (error === 'Server error') {
        this.errorMessage = 'Could not connect to REST server. Please
        check your configuration details';
      } else if (error === '404 - Not Found') {
        this.errorMessage = '404 - Could not find API route. Please
        check your available APIs.';
      } else {
        this.errorMessage = error;
      }
    });
  }


  deleteParticipant(): Promise<any> {

    return this.serviceRetailer.deleteParticipant(this.currentId)
    .toPromise()
    .then(() => {
      this.errorMessage = null;
      this.loadAll();
    })
    .catch((error) => {
      if (error === 'Server error') {
        this.errorMessage = 'Could not connect to REST server. Please
```

```
        check your configuration details';
      } else if (error === '404 - Not Found') {
        this.errorMessage = '404 - Could not find API route. Please
        check your available APIs.';
      } else {
        this.errorMessage = error;
      }
    });
  }

  setId(id: any): void {
    this.currentId = id;
  }

  getForm(id: any): Promise<any> {

    return this.serviceRetailer.getparticipant(id)
    .toPromise()
    .then((result) => {
      this.errorMessage = null;
      const formObject = {
        'email': null,
        'firstName': null,
        'lastName': null,
        'type': null
      };
```

```
if (result.email) {
  formObject.email = result.email;
} else {
  formObject.email = null;
}

if (result.firstName) {
  formObject.firstName = result.firstName;
} else {
  formObject.firstName = null;
}

if (result.lastName) {
  formObject.lastName = result.lastName;
} else {
  formObject.lastName = null;
}

if (result.type) {
  formObject.type = result.type;
} else {
  formObject.type = null;
}

this.myForm.setValue(formObject);
```

```
    })
    .catch((error) => {
      if (error === 'Server error') {
        this.errorMessage = 'Could not connect to REST server. Please
        check your configuration details';
      } else if (error === '404 - Not Found') {
        this.errorMessage = '404 - Could not find API route. Please
        check your available APIs.';
      } else {
        this.errorMessage = error;
      }
    });

  }

  resetForm(): void {
    this.myForm.setValue({
      'email': null,
      'firstName': null,
      'lastName': null,
      'type': null
    });
  }
}
```

```
*/

import { Injectable } from '@angular/core';
import { DataService } from '../data.service';
import { Observable } from 'rxjs/Observable';
import { Retailer } from '../org.supplychain.network';
import 'rxjs/Rx';

// Can be injected into a constructor
@Injectable()
export class RetailerService {

  private NAMESPACE = 'Retailer';

  constructor(private dataService: DataService<Retailer>) {
  };

  public getAll(): Observable<Retailer[]> {
    return this.dataService.getAll(this.NAMESPACE);
  }

  public getparticipant(id: any): Observable<Retailer> {
    return this.dataService.getSingle(this.NAMESPACE, id);
  }
```

```
public addParticipant(itemToAdd: any)
: Observable<Retailer> {
    return this.dataService.add(this.NAMESPACE, itemToAdd);
  }

  public updateParticipant(id: any, itemToUpdate: any): Observable<Retailer
    return this.dataService.update(this.NAMESPACE, id, itemToUpdate);
  }

  public deleteParticipant(id: any): Observable<Retailer> {
    return this.dataService.delete(this.NAMESPACE, id);
  }

}
```

is the codes for the BCFL Angular APP use case product interface algorithm

```
        /**
   BCFL Code
   */
  changeArrayValue(name: string, value: any): void {
    const index = this[name].value.indexOf(value);
    if (index === -1) {
      this[name].value.push(value);
    } else {
      this[name].value.splice(index, 1);
```

```
    }
  }


        /**
         * Checkbox helper, determining whether an enumeration
         value should be selected or not (for array enumeration values
   * only). This is used for checkboxes in the asset updateDialog.
   * @param {String} name - the name of the asset field to check
   * @param {any} value - the enumeration value to check for
   * @return {Boolean} whether the specified asset field contains the
   provided value
   */
  hasArrayValue(name: string, value: any): boolean {
    return this[name].value.indexOf(value) !== -1;
  }

  addAsset(form: any): Promise<any> {
    this.asset = {
      $class: 'org.supplychain.network.Product',
      'productId': this.productId.value,
      'producttype': this.producttype.value,
      'size': this.size.value,
      'description': this.description.value,
      'quantity': this.quantity.value,
      'unitPrice': this.unitPrice.value,
      'totalPrice': this.totalPrice.value,
```

```
    'owner': this.owner.value,
    'issuer': this.issuer.value
  };

  this.myForm.setValue({
    'productId': null,
    'producttype': null,
    'size': null,
    'description': null,
    'quantity': null,
    'unitPrice': null,
    'totalPrice': null,
    'owner': null,
    'issuer': null
  });

  return this.serviceProduct.addAsset(this.asset)
  .toPromise()
  .then(() => {
    this.errorMessage = null;
    this.myForm.setValue({
      'productId': null,
      'producttype': null,
      'size': null,
      'description': null,
      'quantity': null,
```

```
        'unitPrice': null,
        'totalPrice': null,
        'owner': null,
        'issuer': null
      });
      this.loadAll();
    })
    .catch((error) => {
      if (error === 'Server error') {
          this.errorMessage = 'Could not connect to REST server. Please
          check your configuration details';
      } else {
          this.errorMessage = error;
      }
    });
  }



  updateAsset(form: any): Promise<any> {
    this.asset = {
      $class: 'org.supplychain.network.Product',
      'producttype': this.producttype.value,
      'size': this.size.value,
      'description': this.description.value,
      'quantity': this.quantity.value,
      'unitPrice': this.unitPrice.value,
```

```
      'totalPrice': this.totalPrice.value,
      'owner': this.owner.value,
      'issuer': this.issuer.value
    };

    return this.serviceProduct.updateAsset(form.get('productId').
    value, this.asset)
    .toPromise()
    .then(() => {
      this.errorMessage = null;
      this.loadAll();
    })
    .catch((error) => {
      if (error === 'Server error') {
        this.errorMessage = 'Could not connect to REST server. Please
        check your configuration details';
      } else if (error === '404 - Not Found') {
        this.errorMessage = '404 - Could not find API route. Please
        check your available APIs.';
      } else {
        this.errorMessage = error;
      }
    });
  }
```

```
deleteAsset(): Promise<any> {

  return this.serviceProduct.deleteAsset(this.currentId)
  .toPromise()
  .then(() => {
    this.errorMessage = null;
    this.loadAll();
  })
  .catch((error) => {
    if (error === 'Server error') {
      this.errorMessage = 'Could not connect to REST server. Please
      check your configuration details';
    } else if (error === '404 - Not Found') {
      this.errorMessage = '404 - Could not find API route. Please
      check your available APIs.';
    } else {
      this.errorMessage = error;
    }
  });
}

setId(id: any): void {
  this.currentId = id;
}

getForm(id: any): Promise<any> {
```

```
return this.serviceProduct.getAsset(id)
.toPromise()
.then((result) => {
  this.errorMessage = null;
  const formObject = {
    'productId': null,
    'producttype': null,
    'size': null,
    'description': null,
    'quantity': null,
    'unitPrice': null,
    'totalPrice': null,
    'owner': null,
    'issuer': null
  };

  if (result.productId) {
    formObject.productId = result.productId;
  } else {
    formObject.productId = null;
  }

  if (result.producttype) {
    formObject.producttype = result.producttype;
  } else {
```

```
    formObject.producttype = null;
  }

  if (result.size) {
    formObject.size = result.size;
  } else {
    formObject.size = null;
  }

  if (result.description) {
    formObject.description = result.description;
  } else {
    formObject.description = null;
  }

  if (result.quantity) {
    formObject.quantity = result.quantity;
  } else {
    formObject.quantity = null;
  }

  if (result.unitPrice) {
    formObject.unitPrice = result.unitPrice;
  } else {
    formObject.unitPrice = null;
  }
```

```
    if (result.totalPrice) {
      formObject.totalPrice = result.totalPrice;
    } else {
      formObject.totalPrice = null;
    }

    if (result.owner) {
      formObject.owner = result.owner;
    } else {
      formObject.owner = null;
    }

    if (result.issuer) {
      formObject.issuer = result.issuer;
    } else {
      formObject.issuer = null;
    }

    this.myForm.setValue(formObject);

  })
  .catch((error) => {
    if (error === 'Server error') {
      this.errorMessage = 'Could not connect to REST server. Please
      check your configuration details';
```

```
      } else if (error === '404 - Not Found') {
        this.errorMessage = '404 - Could not find API route. Please
        check your available APIs.';
      } else {
        this.errorMessage = error;
      }
    });
  }

  resetForm(): void {
    this.myForm.setValue({
      'productId': null,
      'producttype': null,
      'size': null,
      'description': null,
      'quantity': null,
      'unitPrice': null,
      'totalPrice': null,
      'owner': null,
      'issuer': null
      });
  }

}
```

Appendix B

BCFL Simulation Results

B.1 Initial Results

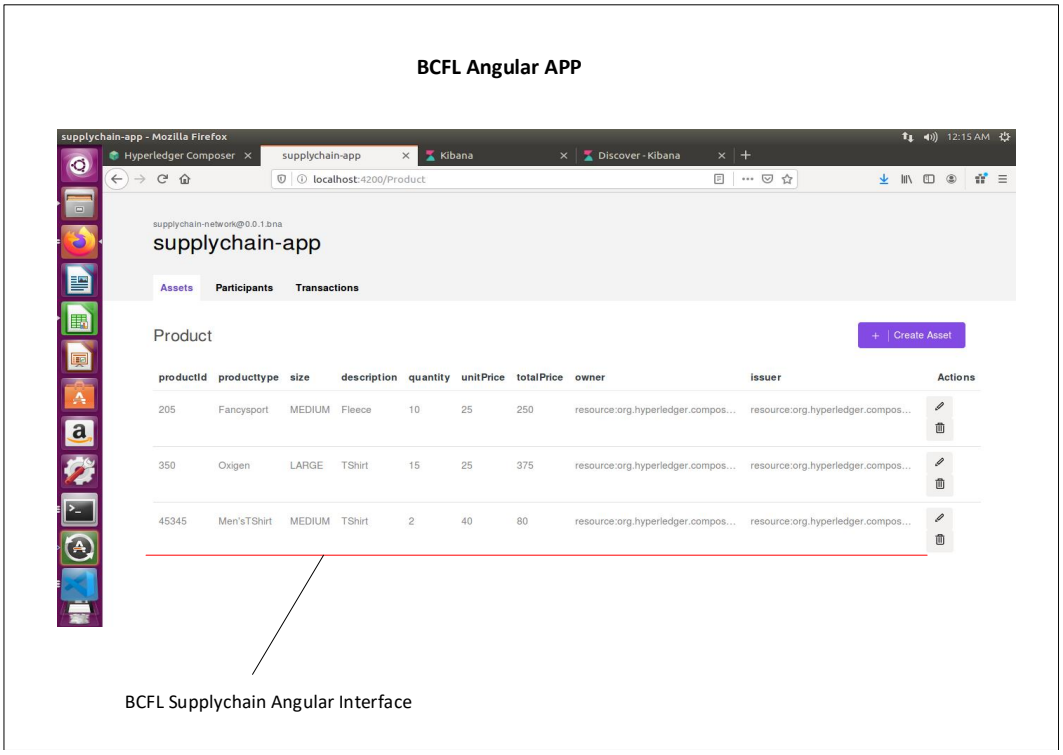


Figure B.1: *BCFL Supply chain Angular App Interface*

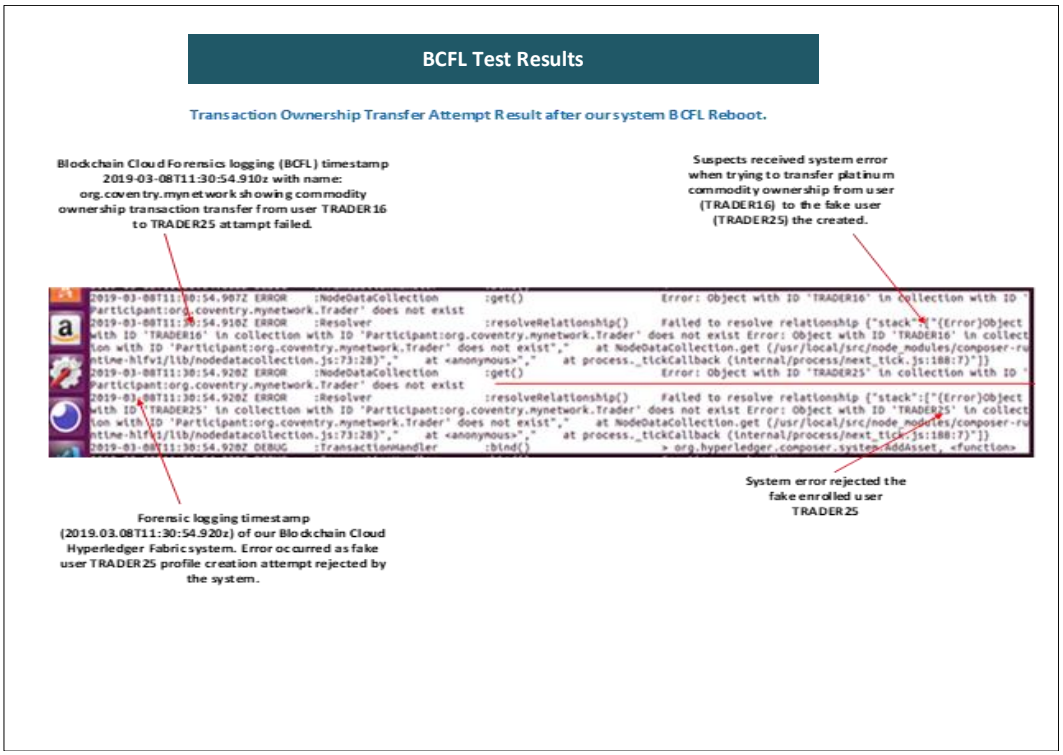


Figure B.2: *Identification of Log evidence after System Reboot*

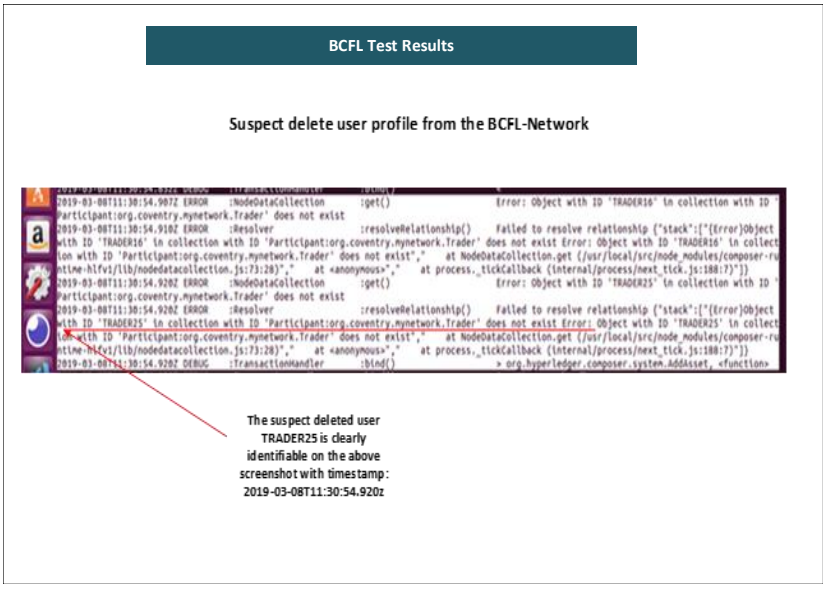


Figure B.3: *User Profile Deleted*

Figure B.4: *Log Forensic Acquisition*

Figure B.5: *User Identification by Chaincode*

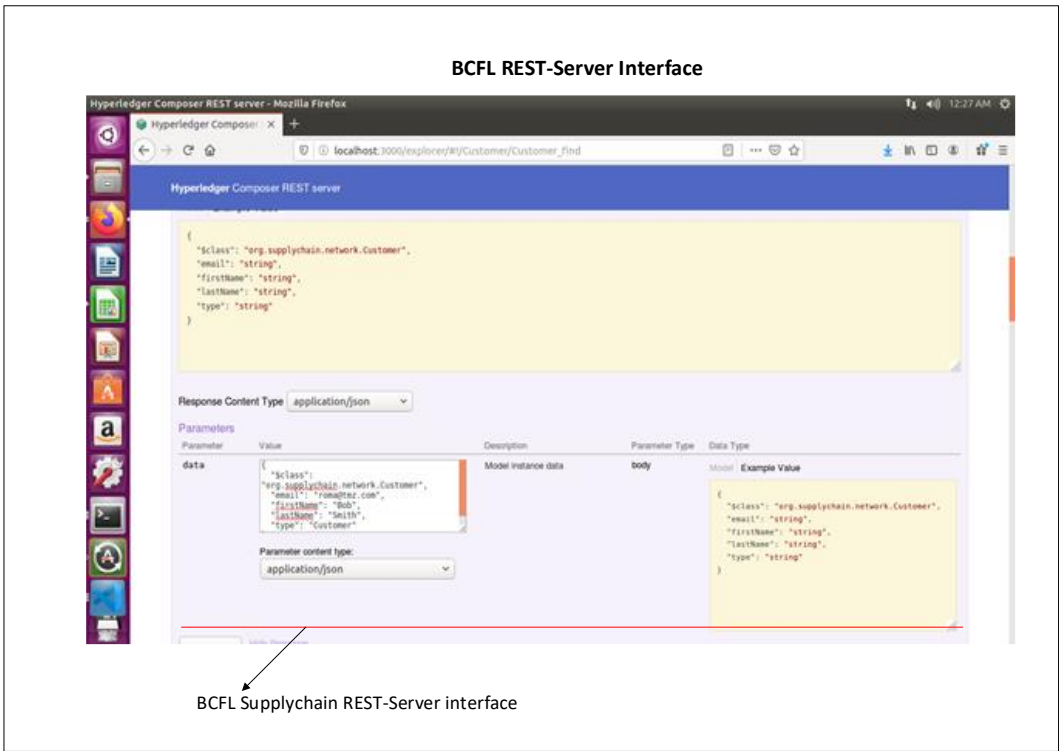


Figure B.6: *BCFL REST-Server Supplychain Interface*

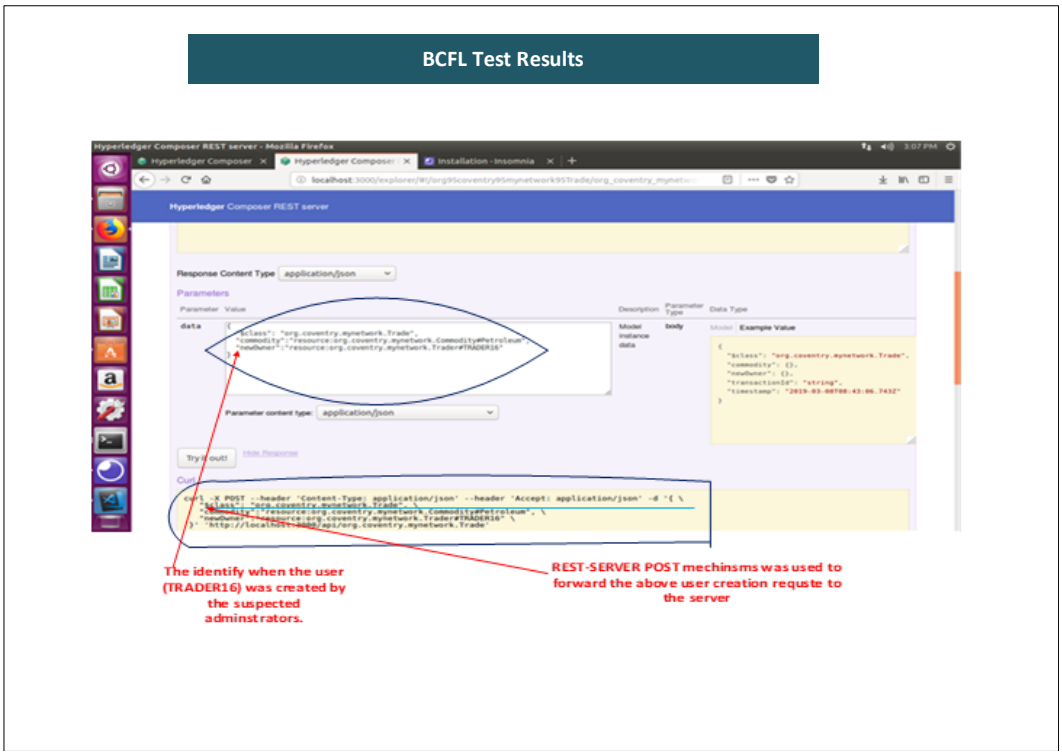


Figure B.7: *REST Server and user activity timestamp*

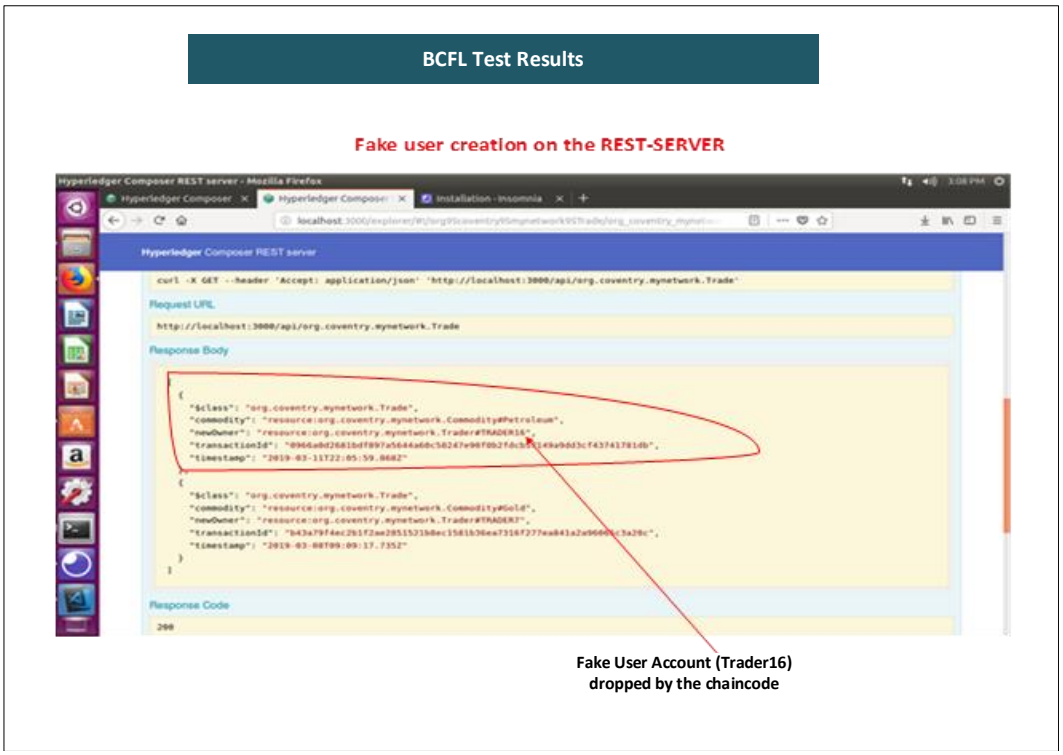


Figure B.8: *REST Server and user activity timestamp*

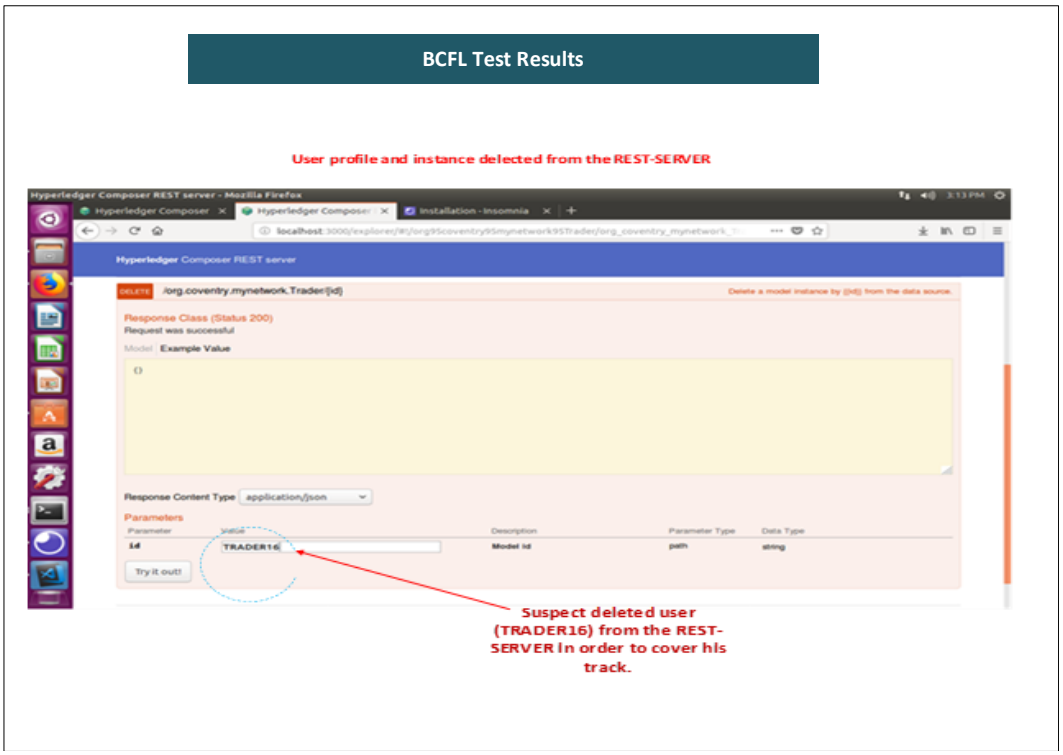


Figure B.9: *REST Server User Activity*

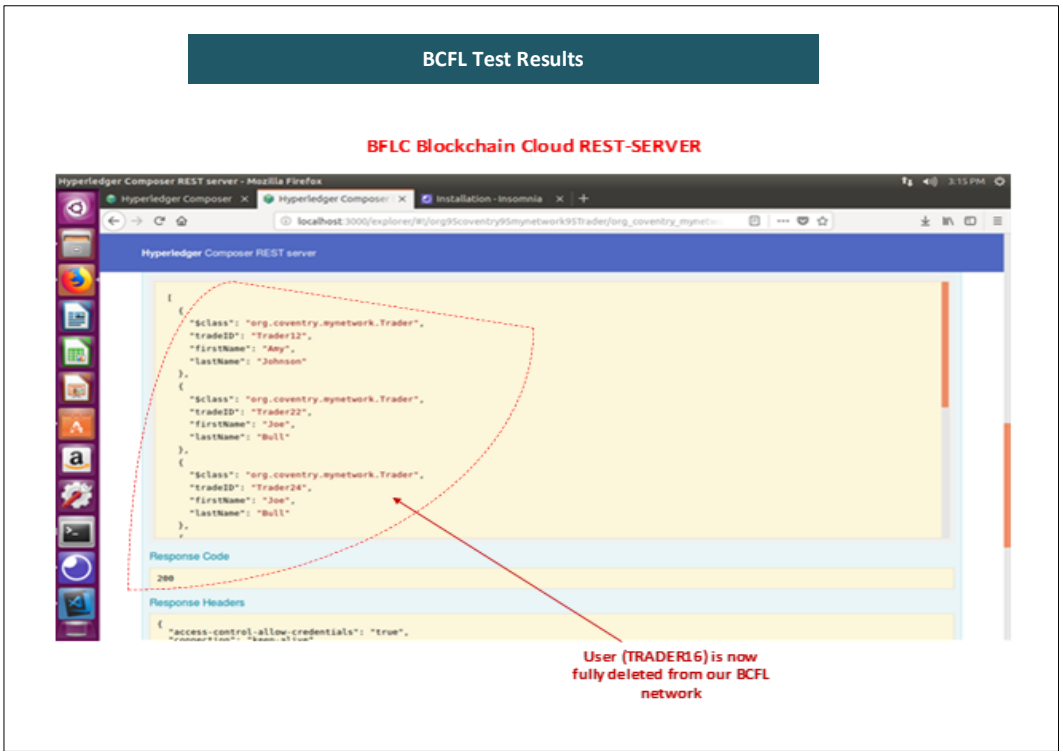


Figure B.10: *REST Server User Activity*

Certificate of Ethical Approval

Applicant:

Kenny Awuson - David

Project Title:

Blockchain Logging: An Approach to maintain Forensics Evidence Integrity in the Cloud Ecosystem

This is to certify that the above named applicant has completed the Coventry University Ethical Approval process and their project has been confirmed and approved as Low Risk

Date of approval:

26 March 2020

Project Reference Number:

P93768

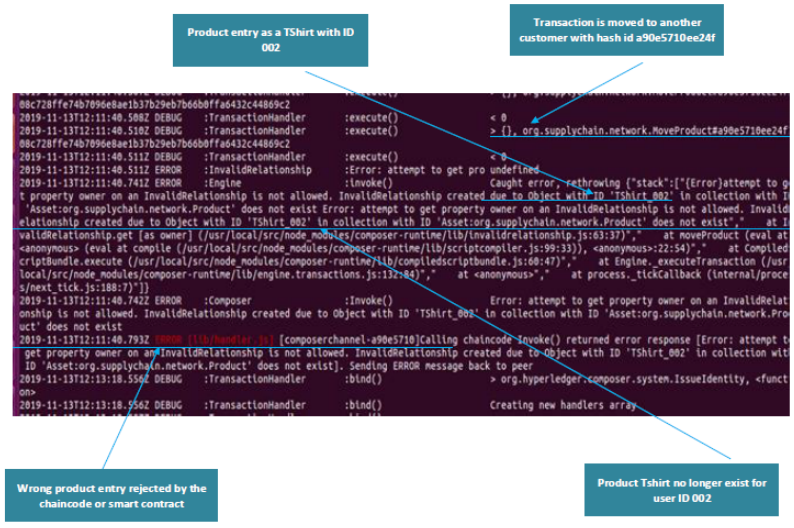


Figure B.11: *BCFL Results*

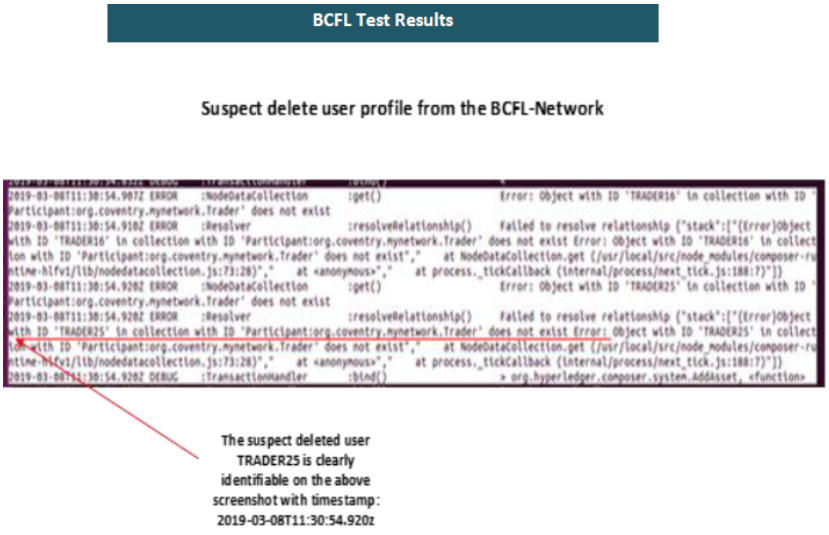


Figure B.12: *BCFL Results*

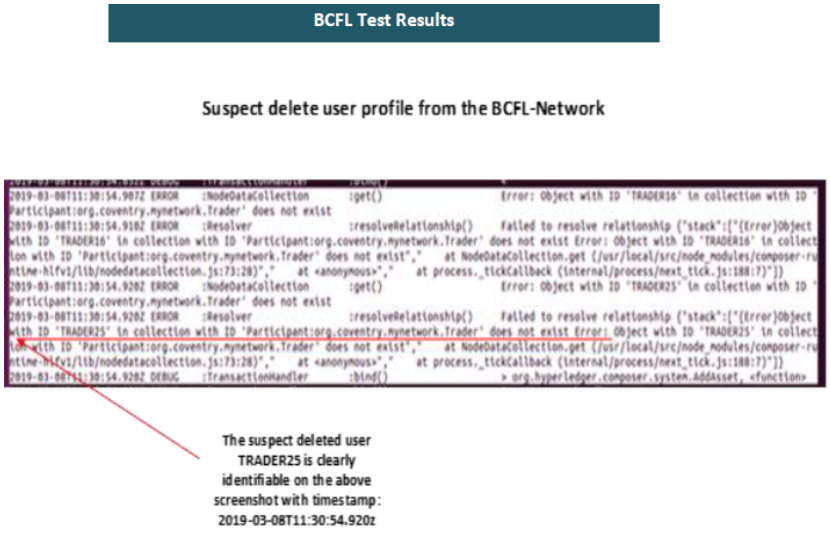


Figure B.13: *BCFL Results*