DEVELOPING A BLOCKCHAIN-BASED SMART CONTRACT DATA PROVENANCE TRACKING APPLICATION WITHIN A SMART MANUFACTURING ENVIRONMENT

by
OMPHEMETSI LUCKY MOKALUSI

Dissertation submitted in fulfilment of the requirements for the Degree

MASTER of ENGINEERING in ELECTRICAL ENGINEERING (M_ENGE)

in the

Department of Electrical, Electronic and Computer Engineering

of the

Faculty of Engineering, Built Environment and Information Technology

at the

Central University of Technology, Free State

Supervisor: Prof. RB Kuriakose
Co-Supervisor: Prof. HJ Vermaak

BLOEMFONTEIN
2024

Declaration of Independent Work

I, OMPHEMETSI LUCKY MOKALUSI, identity number _____ and student number _____, do hereby declare that this research project submitted to the Central University of Technology, Free State for the Degree MASTER OF ENGINEERNG: ELECTRICAL ENGENIEERING, is my own independent work; and complies with the Code of Academic Integrity, as well as other relevant policies, procedures, rules and regulations of the Central University of Technology, Free State; and has not been submitted before to any institution by myself of any other person in fulfilment (or partial fulfilment) of the requirements for the attainment of any qualification.

11 January 2024

SIGNATURE OF STUDENT **DATE**

Dedication

To the Mokalusi Foundation.

Acknowledgements

I would like to express my heartfelt acknowledgments to those listed below for their valuable contributions to the completion of this research study.

First and foremost, I express my gratitude to the universe for granting me the knowledge to successfully complete this research study.

To Professor Rangith Baby Kuriakose, my supervisor, and Professor Herman Jacobus Vermaak, my co-supervisor, I am thankful for the guidance, unwavering support, and constant motivation you both offered during this research study. Your assistance was indispensable, and this research study would not have been possible without your input.

I am grateful to the Central University of Technology (CUT), Free State, for extending financial support that allowed me to participate in short courses and conferences. This support significantly enhanced my understanding of the research I undertook.

I would like to thank FoodBev SETA for awarding me a Research and Innovation bursary.

Finally, in a special way, I would like to thank my beloved wife Selloane, my loving daughter Lesego, and my son Katlego. Thank you.

Abstract

Data provenance across an end-to-end supply chain process refers to tracking the origin of every raw material, its processing, packaging, and distribution. Prior to reaching the end-user, a product usually goes through various suppliers and stages including processing, payment, manufacturing, and logistics. However, the origin and history of a product often remains centralised and unseen, making it difficult for the end-users to trace the product's journey. Centralisation has therefore been identified as a major challenge in achieving data provenance traceability across an end-to-end supply chain. This is because the traditional client-server architecture utilised in centralised systems stores data in a single location, making it vulnerable to single points of failure, data tampering, and unauthorised access. As a result, a lack of data provenance and standardisation for products in a manufacturing supply chain can occur, leading to a lack of traceability and transparency. Several authors have already proposed various solutions to address the challenge of data provenance traceability problems created by centralisation. However, there is still limited research on combating the centralisation in smart manufacturing and none when it comes to a cost-effective public blockchain platform. The current challenges that necessitated this study, stem from the limited research on how end-users can access critical data pertaining to the contents, composition, quantity and timestamp at time of packaging a product. The challenge of ensuring data provenance in a smart manufacturing environment can therefore be overcome by incorporating product data pertaining to the end-user order into a blockchain-based smart contract, making the data secure, immutable and traceable. This research study involves the development of a blockchain-based smart contract data provenance tracking application. To validate and demonstrate the developed solution, an experimental setup is utilised based on the case study of a water bottling plant. This is accomplished by firstly investigating the broader problem of data provenance traceability, and to identify a challenge described and establish the limitations that this research study intends to fill. This research study then goes on to determine the key factors that influence the choice of a blockchain platform for storing data provenance into smart contracts. It then compares the different blockchain platforms and identifies a cost-effective and suitable platform for this research study. The research study goes on to describe the proposed experimental setup that was developed to overcome this challenge and finally illustrates the results of the experiments. The results of this research study illustrate that centralisation, identified as a major challenge in achieving data provenance traceability, can be overcome by incorporating product data pertaining to the end-user order into a public blockchain-based smart contract, making the data

secure, immutable and traceable. This allows the end-user to “read” data provenance through a tag on the bottled water for traceability and transparency. The findings of this research study can also contribute to the existing body of knowledge regarding data provenance traceability. This contribution has the potential to significantly reduce the scale of product recalls, from millions of units to only a few hundred, further highlighting the importance of this research study.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction to Study Environment

1.1 Introduction

Traditional manufacturing plants have been focused on optimising operational and communication processes to increase profits in minimal time [1] across an end-to-end supply chain. Supply chains are characterised by upstream and downstream activities that involve the movement of products, raw materials, and services [2][3]. In a multi-stakeholder environment, supply chain applications require trust, traceability, data sharing, payment, and transparency [4] where data semantic information [5] is shared among different role players.

The intricacies of the traditional manufacturing environment are accentuated when moving into the fourth industrial revolution, commonly referred to as Industry 4.0. Data fragmentation [6], disintegration, and regulatory policies in supply chain management have made it challenging for traditional manufacturing environments to maintain product data traceability and transparency. Studies show that this has been hindered by factors such as the varying interests of and complex business processes [7].

Prior to reaching the end-user, a product usually goes through various suppliers and stages including processing, payment, manufacturing, and logistics. However, the origin and history of a product often remain centralised and unseen, making it difficult for the end-users to trace the product's journey.

Provenance [8] refers to the origin or earliest known history of an item and ensures the authenticity and integrity of items as they move across an end-to-end supply chain. Provenance must be tamper-proof [9] and there is an increasing demand for data provenance traceability across an end-to-end supply chain as it allows for tracking the history, application, or location of a product [10].

In a smart manufacturing environment, data provenance refers to the traceability of every raw material, its processing, packaging, and distribution across an end-to-end supply chain. Smart manufacturing refers to fully integrated and collaborative systems that can quickly respond to changing demands and conditions in the plant, suppliers, and end-user needs in real-time [11].

The lack of data provenance in verifying product or raw material data parameters stems from the fact that the data is often encrypted in a serialised numeric code format, posing difficulty for the end-user to track the product's origin and history. In most cases, data provenance is limited to a simple printing on the product's barcode indicating only the date of manufacturing and expiry.

The problem arises when end-users do not have access to data provenance storage platforms, making it difficult for them to verify the logged product's history and origin. This was evident with a listeriosis outbreak in South Africa, where several people lost their lives [12] due to the inability to track and query the contaminated meat back to its origin in time to effect a product recall.

The establishment of an end-to-end traceability framework, especially in food-sensitive sectors, necessitates a standardised approach handled by the Electronic Product Code Information Service (EPCIS) [2] developed by GS1 standard [13]. The GS1 global traceability standard sets a minimum set of requirements for traceability in business processes, ensuring an end-to-end traceability irrespective of the underlying technology. In this dissertation, a solution is proposed that leverages the GS1 global traceability standard on a cost-effective public decentralised distributed ledger termed blockchain [14]. This solution intends to address the challenge posed by the lack of data provenance in the existing systems.

This research study utilises the case study of a Make-to-Order (MTO) [15] smart manufacturing environment which produces bottled water. End-user orders are collected by the water bottling plant and executed using three smart manufacturing units operating in a decentralised manner using a proposed blockchain-based smart contract manufacturing protocol. Currently, end-users have no means of knowing the source and contents of the water, the plant where it was bottled, or tracking the delivery process.

The aim of this research study is thus to identify and implement a cost-effective and suitable blockchain platform which will be utilised to develop a blockchain-based smart contract data provenance tracking application within a smart manufacturing environment to log product or raw material information pertaining to the composition and production process. This is to grant the end-users access to a transparent decentralised storage platform to verify logged product or raw material data provenance related to the contents, composition, quantity and timestamp at the time of packaging across an end-to-end supply chain.

1.2 Problem statement

Data provenance in a smart manufacturing environment refers to the origin of every raw material, its processing, packaging, and distribution. The current challenges that necessitated this research study, stem from the limited research on how end-users can access critical data pertaining to the contents, composition, quantity and timestamp at the time of packaging a product.

1.3 Research hypothesis and objectives

1.3.1 Research hypothesis

The challenge of ensuring data provenance in a smart manufacturing environment can be overcome by incorporating product data pertaining to the end-user order into a blockchain-based smart contract, making the data secure, immutable and traceable.

1.3.2 Research aim and questions

The aim of this research study is to identify and implement a cost-effective and suitable blockchain platform which will be utilised to develop a blockchain-based smart contract data provenance tracking application within a smart manufacturing environment to log product or raw material information pertaining to the composition and production process. Therefore, the research questions that eminate from this research study are as follows:

- What are the means and benefits of including blockchain-based smart contracts to a smart manufacturing environment?
- What are the factors influencing the selection of a blockchain platform for incorporating data provenance into smart contracts?
- How to identify a cost-effective and suitable decetralised platforms to reduce expensive on-chain data storage?
- How can standard principles be applied to supply chains across various sectors leading to traceability and transparency?

1.3.3 Research objectives

The specific objectives of this research study are as follows:

- To investigate the broader problem of data provenance traceability, and to identify the challenge described and establish the limitations that this research study aims to fill.
- To determine the key factors that influence the selection of a blockchain platform for storing data provenance into smart contracts.
- To compare the different blockchain platforms and identify a cost-effective and suitable platform for this research study.
- To develop a blockchain-based smart contract data provenance tracking application within a smart manufacturing environment that can log critical product data pertaining to the contents, composition, quantity and timestamp at the time of packaging a product.

1.4 Research methodology

The aim of this research study is to identify and implement a cost-effective and suitable blockchain platform which will be utilised to develop a blockchain-based smart contract data provenance tracking application within a smart manufacturing environment to log product or raw material information pertaining to the composition and production process. The problem arises when the end-user has limited access to the tracking of such critical data provenance which may be utilised to accelerate the process of a product recall if the need arises.

The methodology of this research study will be divided into four stages as follows:

- The first stage will be defining the research problem by conducting a comprehensive literature review on data provenance traceability across an end-to-end supply chain to establish the limitations that this research study aims to fill.
- The second stage will be to discuss related work in blockchain-based smart contract data provenance traceability in a manufacturing supply chain, including an overview of different types of blockchain platforms and factors influencing their selection for storing data provenance into smart contracts.
- The third stage will focus on developing the blockchain-based smart contract data provenance tracking application within a smart manufacturing environment. A case study of a water bottling plant will be selected to validate and demonstrate the practical application of the experimental setup.

- Finally, the fourth stage will be the comparison of the developed solution with existing solutions to evaluate its effectiveness in addressing the challenge described in the problem statement of this research.

As described in Section 1.1, the case study for this research study will focus on a water bottling plant to validate and demonstrate the practical application of blockchain-based smart contract data provenance tracking in a smart manufacturing environment. The outcome of this design process will illustrate the ability to develop a data provenance tracking application that logs critical data parameters related to product composition and production flow. This application's design leverages blockchain-based smart contracts to overcome existing techniques' limitations. Therefore, this enables end-users to access and verify data provenance through a tag attached to the bottled water, providing increased transparency and traceability.

1.5 Dissertation layout

Chapter 1: The aim of Chapter 1 is to provide a comprehensive overview of the research study. This chapter introduces various elements, including the problem statement, hypothesis, research objectives, and research methodology.

Chapter 2: The aim of Chapter 2 is to explore the literature review conducted prior to the research study. The purpose of the literature review is to assist in establishing the limitations and provide a rationale for the research study. The chapter is organised such that it first focuses on introducing provenance. It then discusses the challenges associated with utilising centralised databases for data provenance traceability. The chapter concludes by examining the limitations of prior research which necessitated this study.

Chapter 3: The aim of Chapter 3 is to explain the research methodology utilised to develop a blockchain-based smart contract data provenance tracking application within a smart manufacturing environment to log product or raw material information pertaining to the composition and production process. In this research study, a blockchain-based smart contract data provenance tracking application will be developed utilising a cost-effective and suitable public blockchain platform. To validate the developed blockchain-based smart contract data provenance tracking application, a case study is selected.

Chapter 4: The aim of Chapter 4 is to describe and illustrate the results of the methodology discussed in Chapter 3. The results and the analysis thereof for data provenance collection, storage and tracking will be discussed in the same sequence as in Chapter 3. Firstly, the type of collected data provenance is described and the corresponding results are shared. Next, the analysis focuses on the results of data provenance stored. Finally, the tests and results of the data provenance tracking are shared.

Chapter 5: The aim of Chapter 5 is to discuss the results obtained from the tests carried out in Chapter 4. As discussed in the introduction, the aim of this research study is to identify and implement a cost-effective and suitable blockchain platform utilised to develop a blockchain-based smart contract data provenance tracking application within a smart manufacturing environment to log product or raw material information pertaining to the composition and production process.

Chapter 6: The aim of this chapter is to discuss the reflection on the previous chapters, research goals, recommendations, and future work.

Chapter 2: Literature Review

2.1 Introduction

The aim of this chapter is to explore the literature review conducted prior to the research study. The purpose of the literature review is to assist in establishing the limitations and provide a rationale for the research study. The chapter is organised such that it first focuses on introducing provenance. It then discusses the challenges associated with utilising centralised databases for data provenance traceability. The chapter concludes by examining the limitations of prior research which necessitated this study.

2.2 Provenance

Provenance, also known as lineage [8] or pedigree, refers to the historical record of data and its origins, showing how the data is utilised, processed, represented, stored, and disseminated, by whom and for what purpose [16]. It involves tracking the origins of data and the modifications made to it. It involves tracking the source of information and changes performed on it.

Subsequently, traceability ensures the ability to track the history and origins of product processes [17]–[21]. The management of provenance received extensive attention in database systems, as explored by Simmhan et al. [22], Pérez et al. [23] and Abiodun et al. [5]. These authors presented a comprehensive description of the taxonomy of provenance, which is divided into five divisions as follows:

- Application of provenance
- Provenance subject
- Provenance representation
- Provenance storing
- Provenance dissemination

Provenance is crucial in applications such as forensic analysis, as presented by Isaac Abiodun et al. [5], as it provides a digital proof for investigation. However, due to the enormous development in data, finding the origin can be a difficult task, leading to emerging studies in various applications such as Scientific Workflow Management Systems (SWfMS) [24]–[26], Database Management Systems (DBMS) [27]–[32], Hospital Information Systems [33], and supply chain.

Provenance has different meanings and representations in various applications, and its definition has been refined over time. For instance, Hasan et al. [34] emphasised the importance of protecting the contents of provenance, while Buneman et al. [35] distinguished between "Where and Why" provenance. Similarly, Ram and Liu [36] proposed a generic model called the seven "W's" model, which represent provenance as a set of seven points of data semantic information made up of What, When, Where, How, Who, Which and Why.

Furthermore, Woodruff et al. [37] emphasised the importance of tracking provenance in databases. Therefore, the authors utilised functions to transform data and subsequently reversed those functions in order to track and query provenance within a dataflow graph, such as a Directed Acyclic Graph (DAG). Meanwhile, Groth et al. [38] focused on capturing end-user actions and disseminating them visually as a DAG. This approach allows end-users the ability to traverse the graph to track and query semantic information to extract data provenance traceability. Both approaches demonstrate effective methods for tracking and querying data provenance.

This section is intended to provide an introduction to the terminology and technologies utilised for provenance. This introduction will assist in understanding the several problems encountered in centralised storage platforms as a problem in data provenance traceability within the manufacturing supply chain.

2.2.1 Provenance terminology

The terminology utilised in provenance is key to grasping the issues related to data security, integrity and trust. This section defines some of the key terms utilised in provenance as follows:

- Semantic information – referred to as the meaning [5] of data provenance.
- Provenance – defined as a process to identify and document the source [5] of metadata.
- Metadata – referred to as provenance [5], metadata is also known as the history of records, pedigree, or lineage, ensuring the tracking of the steps for data origin.

- Traceability – defined as the ability to track and query the history, application or location of an object [39][10].
- Provenance chain – it is made up of the time-order of data provenance records of P1|P2 . . . |Pi| . . . |Pn| [34][40].
- Auditor – an Auditor [34] is an end-user that is elected by the provenance management layer as an integrity verifier to provenance chain.

2.2.2 Classification of provenance problems

This section intends to classify provenance. As depicted in Figure 2.1. Ametepe et al. [40] classified provenance into four elements which are provenance security, integrity, availability, and confidentiality.

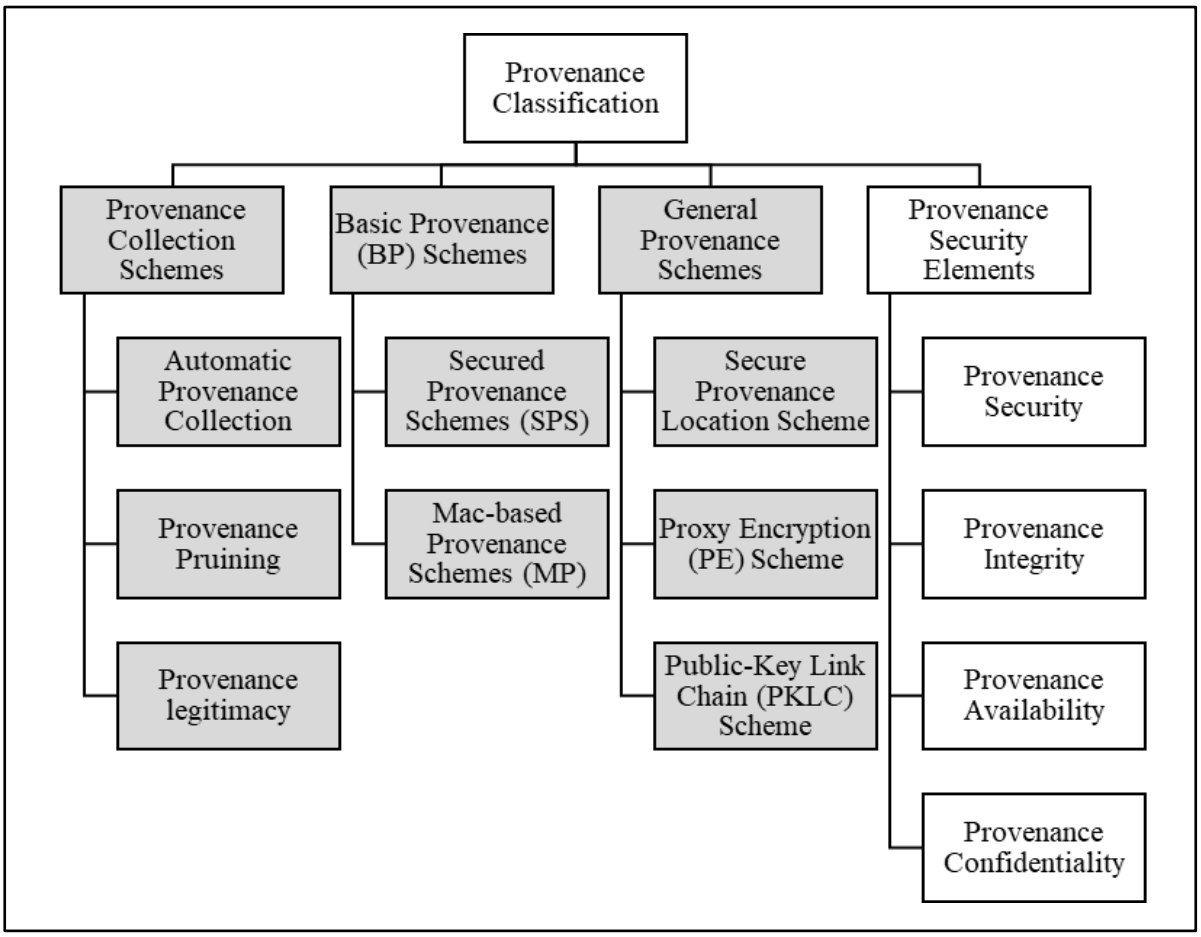


Figure 2.1 Provenance classification by Ametepe et al. 2021 [40]

The key parameters for the classification are described below as follows:

- Single point of failure – refers to the risk of data security loss due to accidental deletion or system failure in a centralised environment [6].
- Data tampering and loss – refers to the risk of intentional data manipulation, leading to data loss and integrity in the provenance chain [40].
- Traceability – refers to the ability to track and query a product's origin or history for increased transparency [39][10].

2.3 Cloud-based data provenance traceability problems in a supply chain

The problem of ensuring data provenance traceability across an end-to-end supply chain for traceability and transparency is a critical issue. It becomes even more challenging when end-users do not have access to centralised storage, making it difficult for them to verify critical data provenance. This is commonly known as a data provenance traceability problem and it can arise due to various reasons, such as lack of data standardisation.

This section thus intends to first discuss the data provenance traceability problems in more detail, and then focus on the system architecture domains that are relevant to this research study. Finally, this section sheds more light on centralisation which is a major challenge for provenance traceability across an end-to-end supply chain.

2.3.1 Data provenance traceability problems

Industrial automation is experiencing a significant transformation with the emergence of Industry 4.0, which requires smart manufacturing machines to be self-managing, self-configuring, self-organising, self-optimising, and self-enforcing, as described by Prause [41].

This is driven by the need for technology organisations to promote collaboration and improve the proliferation of automated production processes, from producing a product from raw materials to delivering it to the end-user, as highlighted by Jamwal et al. [42]. Mentzer et al. [43] define the supply chain as a complex system that involves many components and stakeholders who are directly involved in the process of moving products, finances, services, and information from upstream to downstream. In 2014, Kassahun et al. [44] classified traceability into three types, which are centralised, linear, and distributed. Data provenance traceability problems can be further classified based on the system architecture as centralised or decentralised, as depicted in Figure 2.2 by Kamble et al. [45].

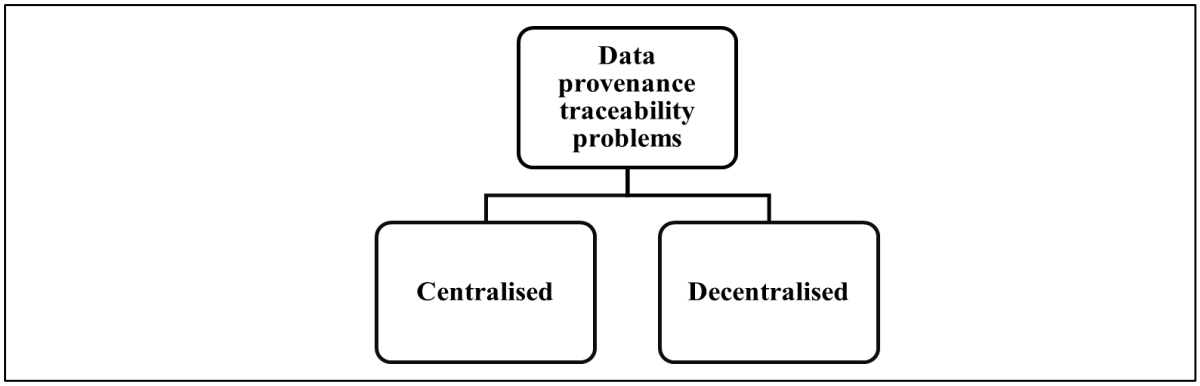


Figure 2.2 Data provenance traceability in a supply chain by Kamble et al. 2020 [45]

Zafar et al. [46] proposes a general overview of provenance and expanded the technologies of traditional supply chains into sub-domains. These are technologies utilised to log product data provenance associated with product composition and production process data parameters in the supply chain for traceability, as depicted in Figure 2.3. The technologies are classified as centralised, including Wireless Sensor Network (WSN), Internet of Things (IoT), cloud-based storage, and decentralised based storage, such as blockchain.

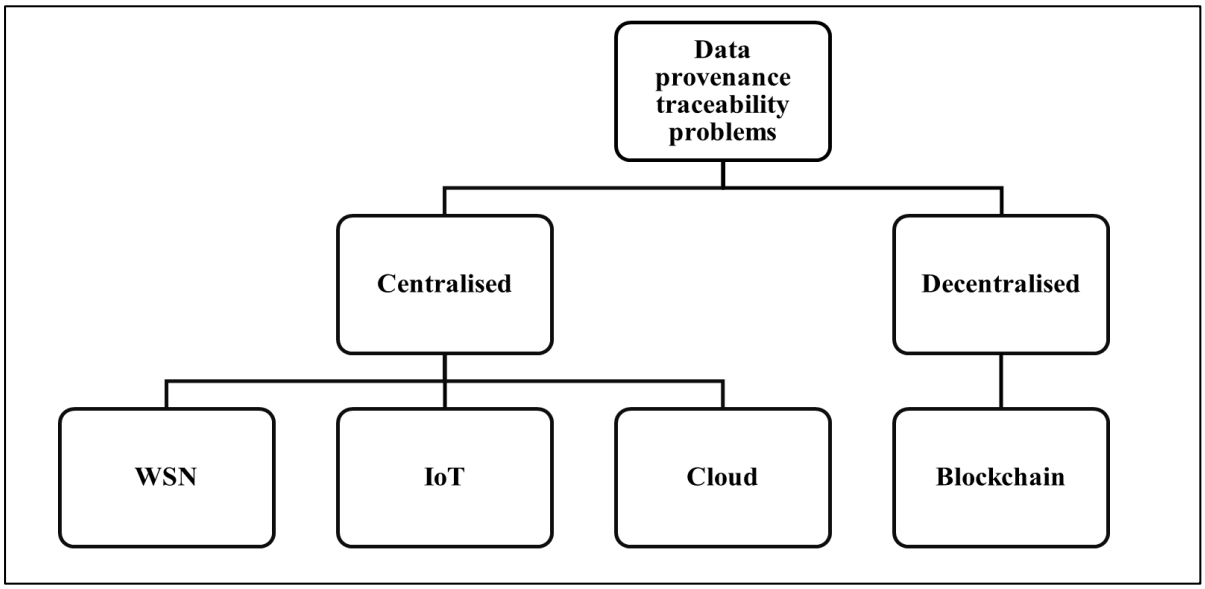


Figure 2.3 Data provenance domains by Zafar et al. 2017 [46]

The establishment of an end-to-end traceability framework, especially in food-sensitive sectors, necessitates a standardised approach handled by the Electronic Product Code Information Service (EPCIS) developed by GS1 standards [13]. The GS1 global traceability standard sets a minimum set of requirements for traceability in business processes, ensuring an end-to-end traceability framework irrespective of the underlying technology. The services of the GS1 global traceability standard are split into three categories as identify, capture and share.

GS1's identification standards are further sub-divided into five categories as follows:

1. The manufacturer and location are identified as Global Location Number (GLN).
2. The traceable item is identified as Global Trade Item Number (GTIN).
3. The logistics and shipping are identified as Serial Shipping Container Code (SSCC) and Global Shipment Identification Number (GSIN).
4. An asset is identified as Global Individual Asset Identifier (GIAI) and Global Returnable Asset Identifier (GARI).
5. The services are identified as Global Service Relation Number (GSLN) and Global Document Type Identifier (GDTI).

A comprehensive description of the identification keys for a traceable item is provided by GS1's framework for the design of interoperable traceability systems across an end-to-end supply chain [39]. The illustration in Figure 2.4 depicts traceability identification implemented across an end-to-end supply chain.

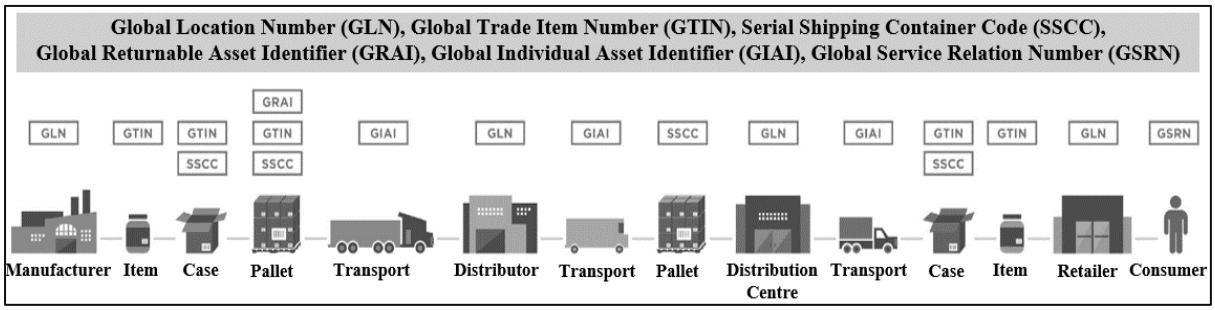


Figure 2.4 Supply chain end-to-end traceability incorporating GS1 standards [47]

GS1's capturing standards are sub-divided into two categories as follows:

- Barcodes
- Electronic Product Code (EPC)/Radio Frequency Identification (RFID)

As depicted in Figure 2.4, barcodes and RFID tags are commonly used for capturing data provenance of traceable items as they move through different stages across an end-to-end supply chain. Barcodes are widely utilised to track items at the case level [6] with information being stored in a local database [48]. Barcodes are printed on almost all products in stores and can be electronically scanned utilising laser or vision-based systems, as long as they are within line-of-sight.

There are different types of barcodes such as European Article Number (EAN)/Universal Product Code (UPC), QR code, Data Matrix, Databar, 1D, and 2D barcodes. However, barcodes can be prone to challenges such as poor print quality, smudged ink, and damaged labels, which can affect their accuracy and reliability, as depicted in Figure 2.5.

Figure 2.5 Challenges with barcode systems [49]

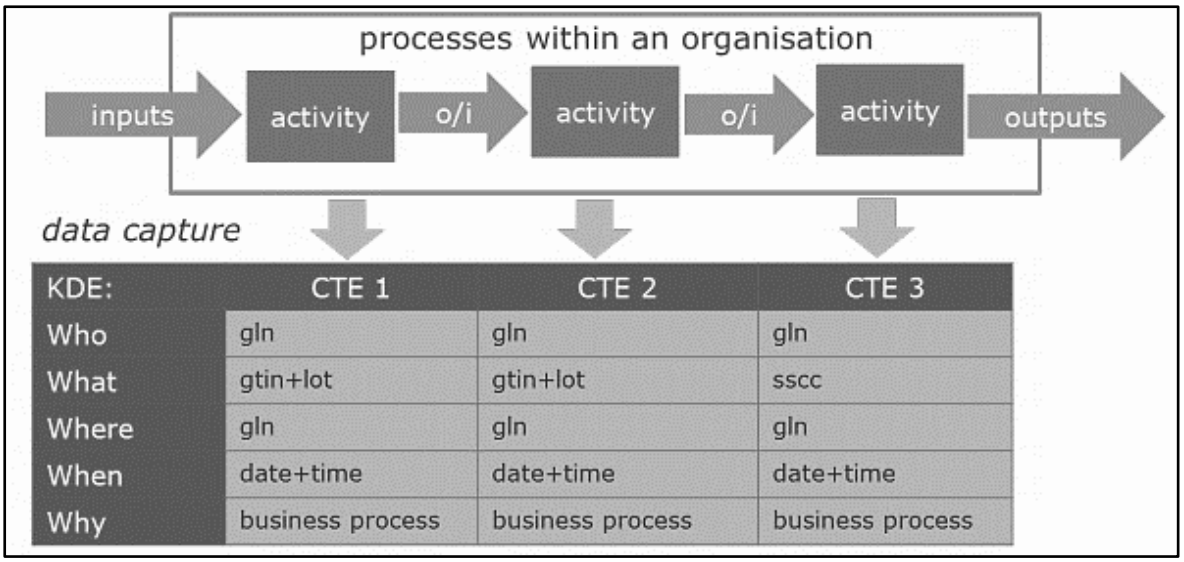
RFID tags are widely utilised for tracking items, cases, pallets, and packages [6]. Similar to barcodes, both RFID and barcodes capture data with a reader and store it in a local database that can be either read-only or read/write [48]. However, RFID tags do not require line-of-sight and can be read from a distance, unlike barcodes. RFID tags and readers operate in two modes of communication, Far Field Communication for distances up to a few meters and Near Field Communication (NFC) for a maximum distance of 10 cm [48].

RFID operates at different frequencies, including Low Frequency (LF) 125-135 KHz, High Frequency (HF) 13.56 MHz, and Ultra High Frequency (UHF). It must be noted that NFC operates at a frequency of 13.56 MHz [48]. One of the major drawbacks of barcodes and RFIDs is that they are linked to a local database [48]. This means that end-users without access to the local database are unable to verify product provenance.

GS1's sharing standards are sub-divided into three data exchanges as follows:

- Electronic Product Code Information Service (EPCIS) events.
- Global Data Synchronisation Network (GDSN).
- Electronic data exchange (EDI) standards.

In order to minimise the risks associated with local storage, GS1 standards play a crucial role in establishing an end-to-end traceability system that captures data provenance as Key Data Elements (KDEs) which emit Critical Tracking Events (CTEs). This is depicted in Figure 2.6.



KDE:	CTE 1	CTE 2	CTE 3
Who	gln	gln	gln
What	gtin+lot	gtin+lot	sscc
Where	gln	gln	gln
When	date+time	date+time	date+time
Why	business process	business process	business process

Figure 2.6 Critical Tracking Events (CTE) and Key Data Elements (KDE) example [39]

As depicted in Figure 2.6, the CTEs are events that occur as the traceable item is traded among different role players. The KDEs capture the semantic information of data provenance with the five "W's", which are Who, What, Where, When and Why, as follows:

- Who – the Global Location Number (GLN) of the manufacturer or the Global Service Relation Number (GSRN) of the end-user.
- What – the unique identifier (UID) of the trackable item, such as the Global Trade Item Number (GTIN) or Serial Shipping Container Code (SSCC), as well as the batch/lot.
- Where – the GLN of the physical location identifier (ID), such as an address or global position service coordinates.
- When – the event date and time as Coordinated Universal Time (UTC) offset in a human-readable standardised ISO 8601 format.
- Why – the business process step or reason for tracking the data provenance, such as a product recall.

This enables scalability due to events of sharing business process data provenance across an end-to-end supply chain facilitated by the Global Data Synchronisation Network (GDSN). The GDSN is a decentralised storage system that allows for the sharing of interactions events among role players globally.

Electronic data exchange (EDI) is utilised to enable this sharing, with GS1 standards Extensible Markup Language (XML) being a common format. However, JavaScript Object Notation (JSON) is a lightweight alternative to XML that requires less coding and is faster to transmit data, making it suitable for tracking and querying data [50].

2.3.2 Centralisation as a problem in data provenance traceability in a supply chain

As mentioned in the previous section, there are several problems with centralised data provenance traceability systems utilising traditional client server architecture [51] with risks such as:

- Single point of failure
- Data tampering and loss
- Traceability and transparency

Centralisation has thus been identified as a major challenge in achieving data provenance traceability across an end-to-end supply chain. The traditional client-server architecture utilised in centralised systems stores data in a single location, making it vulnerable to single points of failure, data tampering, and unauthorised access. As a result, a lack of data provenance and standardisation for products in a manufacturing supply chain can occur, leading to a lack of traceability and transparency.

According to Bagdai et al. [52], corruption prevails where there is a lack of traceability and transparency. This was evident during the 2017 listeriosis outbreak in South Africa [12] which led to 216 fatalities because contaminated meat products could not be traced back to their origin in time to effect a product recall.

To overcome these issues, a decentralised distributed storage is needed, such as a blockchain-based smart contract which can offer solutions to centralisation by minimising risks of ensuring provenance, traceability, immutability and trust [53].

A paper [15] exploring the means and benefits of including blockchain-based smart contracts to a smart manufacturing environment was published for this research study. It highlighted the challenges of centralized data storage and how these challenges can be overcome by incorporating data provenance into blockchain-based smart contracts for traceability and transparency.

2.4 Blockchain-based data provenance traceability in a manufacturing supply chain

This section, firstly, discusses the different types of blockchain platforms and the factors that influence their selection for incorporating data provenance into smart contracts, based on a research study done [54]. Secondly, it discusses a comparison to identify a cost-effective and suitable blockchain platform, as per the research study done [55]. Thirdly, it analyses the various implementations of blockchain-based smart contract data provenance traceability, based on the literature review done by Dasaklis et al. [13].

2.4.1 Types of blockchain platforms

This section discusses the different types of blockchain platforms. The concept of integrating blockchain technology was initially presented in the *Bitcoin* whitepaper authored by Satoshi Nakamoto in 2008 [56][13]. The aim of blockchain platforms is to establish a decentralised distributed ledger for time-stamped transactions among various computers in a peer-to-peer network [57][58], that eliminates the necessity for cloud-based storage that is vulnerable to single point of failure.

As a result, the role players can utilise a Uniform Resource Locator (URL) link to browse and check the network platform's status or validate transactions through a blockchain explorer without relying on third parties. Blockchain technology is a write-only decentralised digital ledger, meaning that transactions or data are trackable and irreversible and can only be added but not edited or removed [41][59]. Every transaction is stored on the blockchain and grouped together in blocks. From a database perspective, the blockchain can be viewed as a blockchain-structured database, where data is packaged into blocks and connected utilising a chain structure.

Blockchain platform block transactions are initiated from a wallet like *MetaMask* [60][61]. The first block in the chain is known as the genesis block, which has no parent block. These blocks are connected utilising cryptographic principles, providing a secure and tamper-proof ledger for storing data provenance. The newly generated block has the SHA256 algorithm applied to it. This is depicted in Figure 2.7.

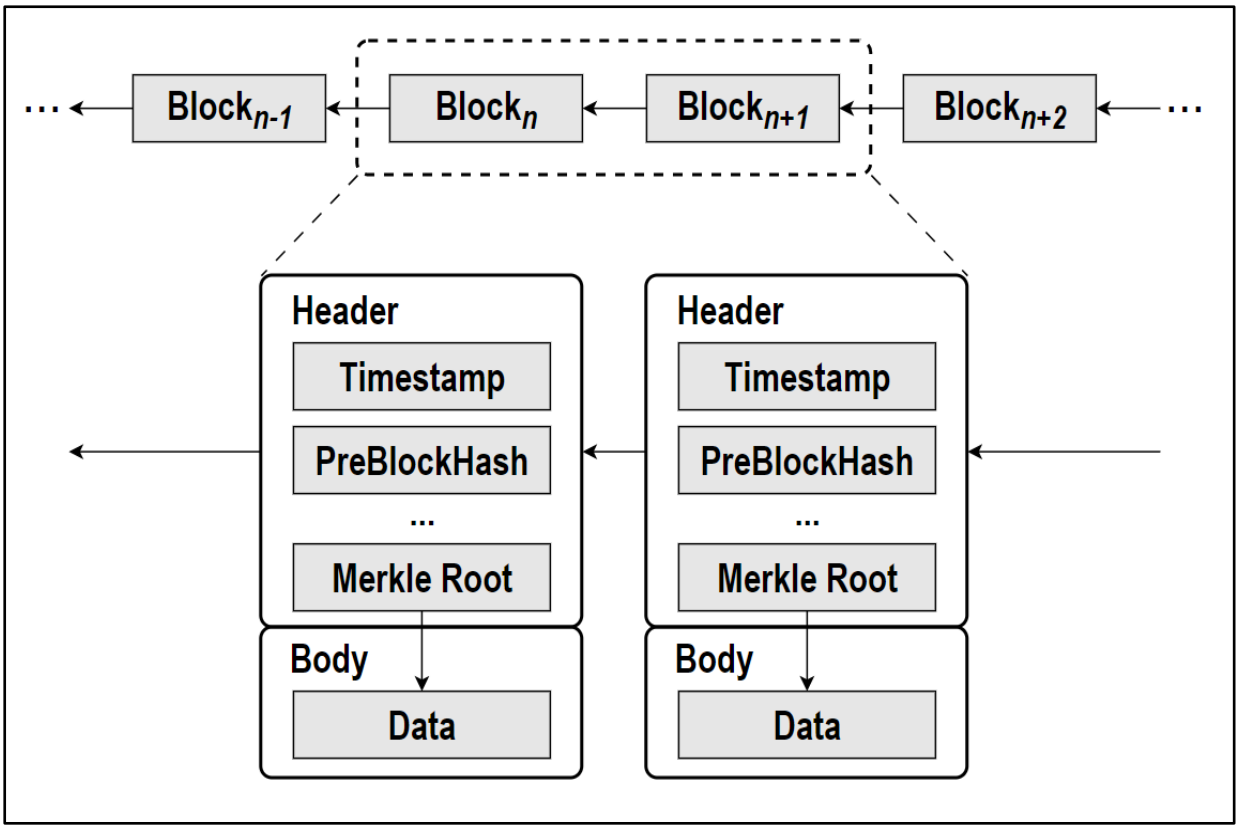


Figure 2.7 Blockchain structure [56]

As depicted in Figure 2.7, the block of the blockchain platform comprises a header and body. The header contains important fields including the timestamp, the hash code of the previous blockchain platform block [62], transaction hash, and the root of a Merkle tree [63]. The concept of a Merkle hash root was initially introduced by Ralph Merkle in 1987 and is represented by a binary or multi-tree structure based on hash value. This is depicted in Figure 2.8.

A Merkle hash root utilises layers of hashing to ensure the integrity of transaction order within a block. Any modification to a Merkle tree can result in changes to the root, thereby breaking the chain connecting each node. In order to construct a Merkle tree [64][65], the dataset $Data_1$, $Data_2$, $Data_3$ and $Data_4$ are subjected to cryptographic hashing utilising a function, such as SHA256.

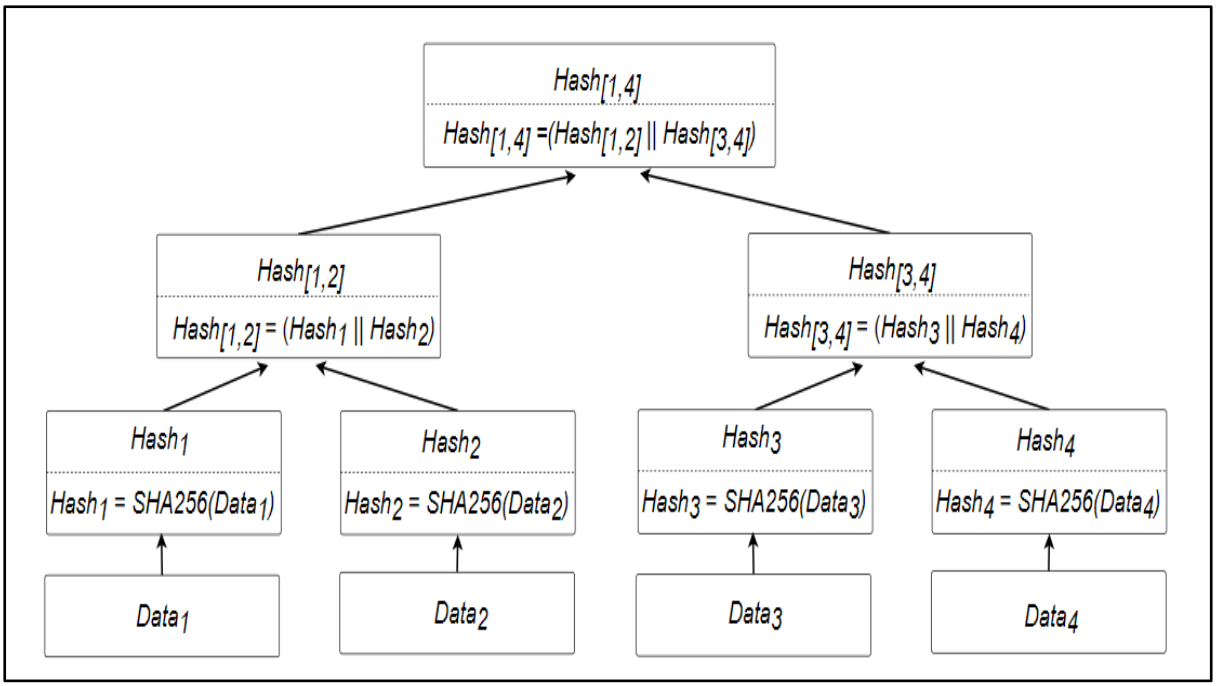


Figure 2.8 An illustration of a "balanced" Merkle tree [64]

As depicted in Figure 2.8 each parent node in a Merkle tree derives its hash value from its child's nodes $Data_n$. The resulting hashes of node $Hash_1$ and $Hash_2$ are concatenated to create a new parent node $Hash_{[1,2]}$. Similarly, the resulting hashes of nodes $Hash_3$ and $Hash_4$ are concatenated to create a parent node $Hash_{[3,4]}$. Finally, the resulting hashes of nodes $Hash_{[1,2]}$ and $Hash_{[3,4]}$ are concatenated to create the root hash value node $Hash_{[1,4]}$, which is linked to every value in the tree.

The concept of implementing a Merkle tree in blockchain technology was first introduced by *Bitcoin* [66]. A Merkle tree is then utilised for ensuring integrity and validation, with the root being incorporated into the blockchain platform to enable traceability and immutability. The body section of the block stores comprehensive data regarding the recorded transactions. This facilitates secure and transparent record-keeping of automated transactions in blockchain technology, which are enforced by smart contracts [67].

The blockchain-based smart contracts first originated from Nick Szabo in 1997 [68]. These blockchain-based smart contract contracts are self-executing, self-enforcing, self-verifiable, and self-constraining pieces of source code that reside on a network platform [15]. Ethereum [69] introduced blockchain-based smart contracts, and it continues to be widely utilised for developing decentralised applications.

The Figure 2.9 depicts the blockchain-based smart contract development workflow, which is divided into four sections as follows:

- Developing the blockchain-based smart contract.
- Compiling the blockchain-based smart contract and extracting the Application Binary Interface (ABI) and bytecode.
- Connecting and deploying the blockchain-based smart contract to blockchain platforms.
- Call functions and transactions.

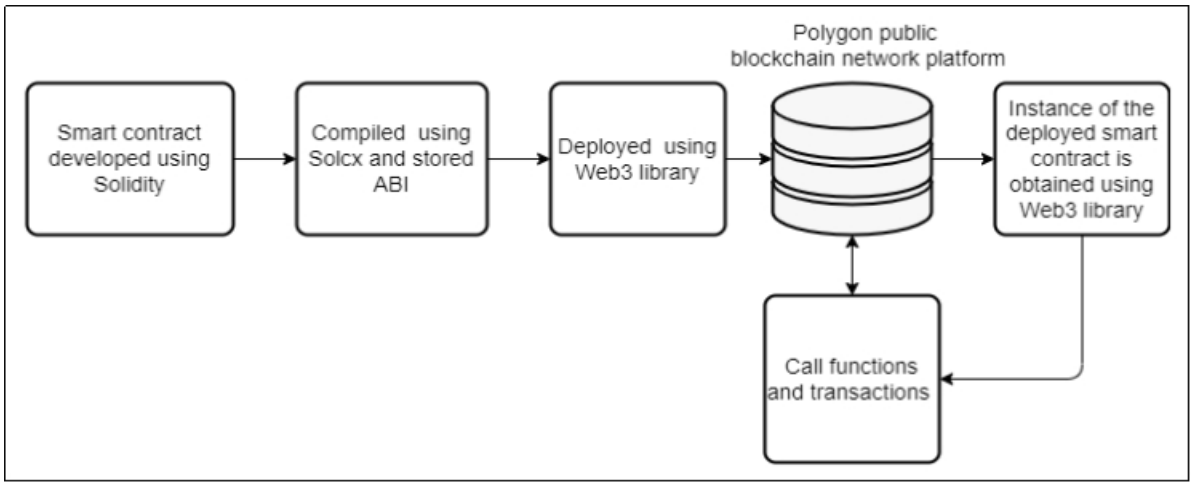


Figure 2.9 The blockchain-based smart contract development workflow

The blockchain-based smart contract is developed in the *Solidity* programming language [70] and compiled utilising compilers such as py-solc-x [71] or *Truffle* framework [72]. It can be seen in Figure 2.9 that the compiled blockchain-based smart contract generates the Application Binary Interface (ABI) and bytecode in JavaScript Object Notation (JSON) markup format which are required for deployment and interaction with end-users.

This interaction is facilitated by the web3.py [73] set of libraries with Ethereum Virtual Machine (EVM) through an Application Program Interface (API) such as *Infura*, *Moralis*, Web3, and the JSON Remote Procedure Call (JSON-RPC) [72] protocol. Once the blockchain-based smart contract is deployed, it cannot be modified, but its function methods can be invoked utilising their address and ABI.

The function methods of the blockchain-based smart contract enable calling and querying transaction records. This is achieved by utilising keywords, meaning the block containing the queried results can be located through a double-layer query index [74]. This index, acting as a mapping hash table, consists of two parts:

1. Inter-block lookup table index locates blocks with query results corresponding to the queried keyword. This utilises sequential linked lists based on block numbers to enhance query efficiency.
2. Intra-block recursive model index locates the blocks with query results by utilising keywords obtained from the inter-block lookup table index.

Zhang et al. [74] published a comprehensive survey on the efficiency, reliability, and security of data query in blockchain systems, comparing various articles in terms of indices. The various types of existing blockchain platforms' principles [75] include traceability, immutability, and transparency utilising indices based on a Merkle tree. This makes blockchain technology a potential solution for ensuring data provenance traceability in products.

This section discusses the different types of blockchain platforms. The decentralised blockchain platforms can be split into three [76], as depicted in Figure 2.10.

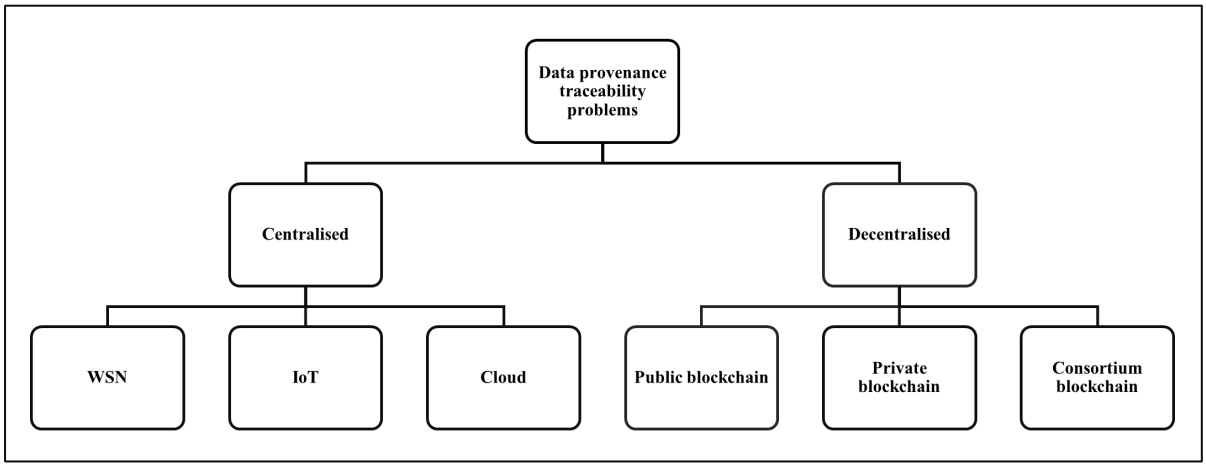


Figure 2.10 The three types of blockchain platforms

The three types of blockchain platforms are as follows:

- Public blockchain – these are permissionless [77] network platforms that are accessible to all internet users to interact and view the history of transactions. There is no central authority [6] and all peers can take part in the consensus mechanism to improve data security, integrity, and trust. Public blockchain platforms include and are not limited to *Bitcoin* [56], *Ethereum*[78], *Tron* [79] and *Polygon* [80][81].

However, the efficiency of permissionless public blockchain platforms is low [76], which affects adoption cost and is their main drawback [82].

- Private blockchain – these are permissioned [77] network platforms that are centrally managed by a single organisation [83]. Private blockchain platforms include and are not limited to *Multichain* [84].

However, since permissioned [77] private blockchain platforms are centralised [85], the decentralised concept is therefore compromised [82] and this is their main drawback.

- Consortium blockchain – these network platforms have permissioned or permissionless access authority for end-users, and can be public or private. They are considered partially decentralised [86] since only organisations that form part of the network platform have access permission authority. Consortium blockchain platforms are implemented with a pre-selected group of peer organisations controlling the consensus mechanism. Consortium blockchain network platforms include and are not limited to *Hyperledger* [87], *Corda* [87][88] and *Quorum* [89].

However, because permissioned [76][77] consortium blockchain platforms are centralised [85], the decentralised concept is compromised [82], and this is their main drawback.

Torky and Hassanein [76] compare public, private, and consortium blockchain platform types, as summarised in Table 2.1. Therefore, selecting a suitable blockchain platform is crucial for solving the challenge of data provenance traceability.

Table 2.1 The comparison between blockchain types by Torky and Hassanein in 2020 [76]

Blockchain type	Public	Private	Consortium
Centralisation	No	Yes	Partial
Access authority	Permissionless	Permissioned	Permissioned
Efficiency	Low	High	High
Trust	All peers	Single organisation	Selected peers

2.4.2 Factors influencing the selection of a blockchain platform

This section discusses factors influencing the selection of a blockchain platform for incorporating data provenance into smart contracts. The main aim of this research is to develop a blockchain-based smart contract data provenance tracking application within a smart manufacturing environment. The application will give the end-user access to a decentralised, permissionless and transparent public blockchain platform as discussed in Section 2.4.1.

Ramachandran et al.[4] depict a decision-tree for selecting the suitable blockchain platform, as depicted in Figure 2.11. This research proposes the selection of a public blockchain platform as the suitable solution for incorporating data provenance that is secure, immutable, traceable, and transparent.

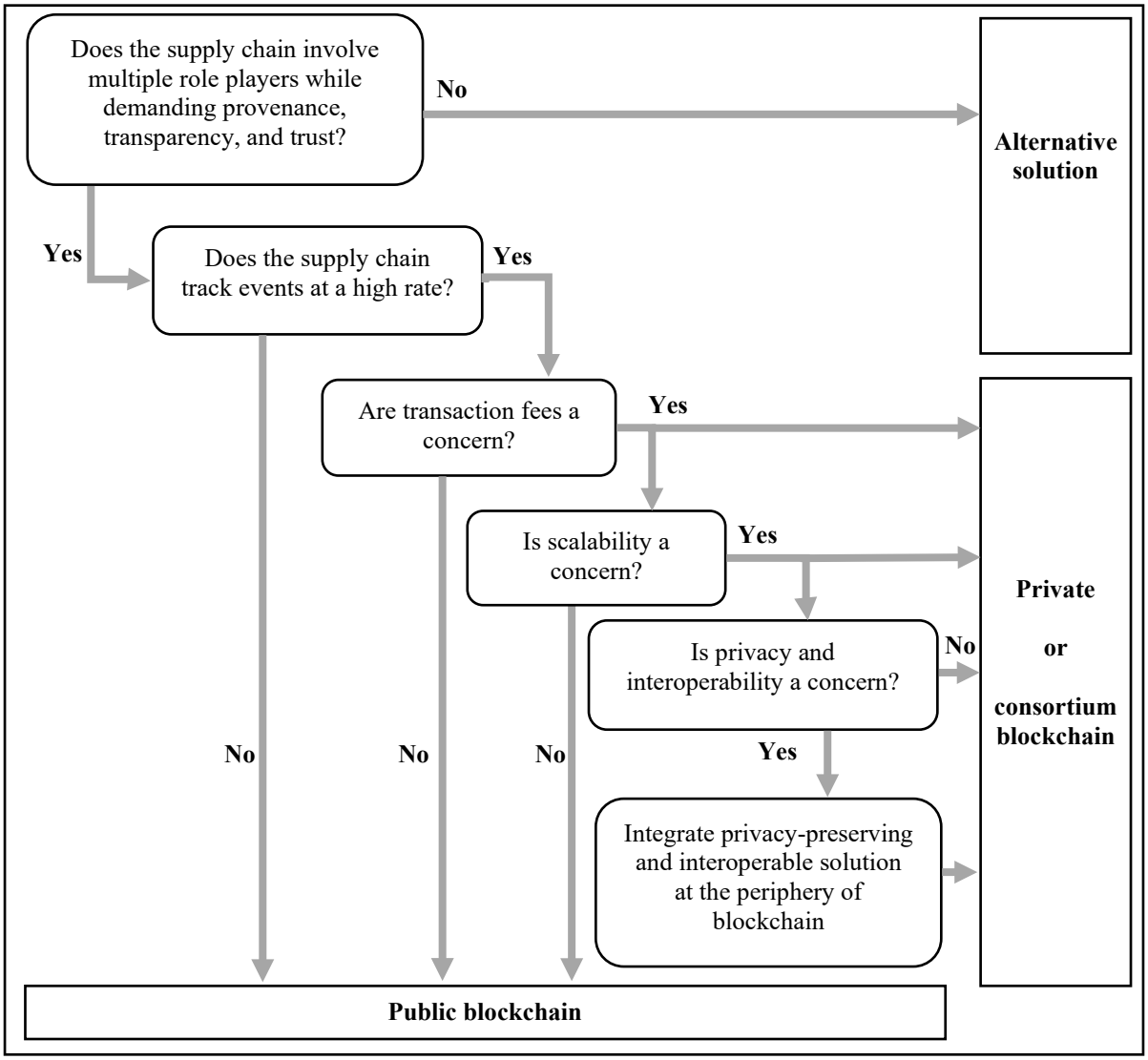


Figure 2.11 A decision-tree for selecting the suitable blockchain platform [90]

A paper detailing the comparison [54] was published for this research study. In the paper, different types of public blockchain platforms were compared, such as *Ethereum* (legacy), *Ethereum* 2, *Polygon*, and *Tron*. The comparison was based on several factors, such as performance, maturity, centralised or decentralised operation, and cost, among others. Table 2.2 presents a summary of the comparison of these public blockchain platforms.

Table 2.2 A comparison of blockchain platforms

	Ethereum (legacy)	**Ethereum 2.0**	**Polygon**	**Tron**
Currency and symbol	ETH	ETH	MATIC	TRON
Cost	High [91]	Low [92]	Low [93]	Low [94]
Performance	< 5 minutes [91]	High [95]	30 seconds [93]	3 seconds [94]
TPS	Low [91]	High [96]	High [97]	High [94]
Throughput	Low [95]	High [96]	High [97]	High [93]
Consensus algorithm	PoW [78]	PoS [96]	PoS [97]	DPoS [94]
Cheapest transfer fee	$3.2606 [91]	$0 [95]	$0 [93]	$0 [94]

Traditional manufacturing plants have been focused on optimising operational and communication processes to increase profits in a shorter amount of time across an end-to-end supply chain. However, the efficiency of permissionless public blockchain platforms is low, which affects the cost of adoption, as depicted in Table 2.2. This is due to the consensus mechanism, such as the Proof of Work (PoW) mining algorithm, which is a significant factor in selecting a blockchain platform.

As a result, organisations are switching from the PoW algorithm to new consensus mechanisms, such as Proof of Stake (PoS), which promise lower transaction fees and energy costs [98]. Such factors have negative financial implications for deploying or interacting with blockchain-based smart contracts that incorporate data provenance in modern manufacturing supply chain.

2.4.3 A comparison of transaction fees of blockchain smart contracts

This section focuses on comparing and identifying a cost-effective and suitable blockchain platform for incorporating data provenance into smart contracts. A paper [99] comparing transaction fees for various data types and data sizes of blockchain-based smart contracts on a selection of blockchain platforms was published for this research study.

The main aim of data provenance is to provide traceability assurance at a low cost for its storage, which supports data integrity and trust [100]. There have been significant contributions from public blockchain platforms such as *Tron*, *Polkadot* [101], *Cardano* [102] and *Solana* [103], as well as layer 2 solutions such as *Polygon*, which may fulfil the requirements for sustainable blockchain-based smart contract platform solutions in terms of cost and latency.

At the time of publishing the paper [99], *Ethereum* 2.0 was still not fully implemented. Based on the analysis in Table 2.3, the results point to *Tron* and *Polygon* as the public blockchain platforms of choice, with factors such as low cost, high performance, high throughput, and the utilisation of efficient Proof of Stake (PoS) consensus algorithms for incorporating data provenance into blockchain-based smart contracts.

Table 2.3 A comparison of transaction fees for various data types and data sizes

Blockchain network	Byte32 (256 bits)	String (256 bits)	Byte32 (768 bits)	String (768 bits)	Byte32 (2304 bits)	String (2304 bits)	Trans action fees
Ethereum (Legacy) (ETH)	64,50	139,18	111,16	486,83	227,72	1555,04	High [91]
Ethereum 2.0 (ETH)	0,83	1,74	1,40	3,89	3,10	10,86	Low [92]
Tron (TRX)	0,00	0,00	0,00	0,00	0,00	0,00	Low [94]
Polygon (MATIC)	0,00	0,00	0,00	0,01	0,00	0,00	Low [93]

In the paper [99], it was found that both *Polygon* and *Tron* had low transaction fees based on the literature review presented in Table 2.2. However, it was noted that *Tron's* consensus mechanism Delegated Proof of Stake (DPoS), is still in its early stages and not yet considered secure enough due to governance issues [79][102]. As a result, for this research study, *Polygon* [99] is considered as the suitable public blockchain platform.

2.4.4 Blockchain-based smart contract data provenance traceability applications

The previous section compared and identified the most cost-effective and suitable blockchain platform. This section analyses various blockchain-based smart contract data provenance traceability applications, with respect to the literature review by Dasaklis et al. [13]. The available blockchain-enabled smart contract traceability implementations encompass products from various supply chain application domains, such as agriculture or agri-food, electronics, food, generic supply chain, manufacturing, pharmaceuticals, electronics, apparel, aviation, automobile, industry, and construction.

As discussed in the previous section, the efficiency of permissionless public blockchain platforms is low, which can increase adoption costs. To address this, the InterPlanetary File System (IPFS) [104] is a distributed peer-to-peer distributed file system that is independent and does not require trust in others, resulting in no single point of failure as in traditional servers. This lowers the risk of attacks. Therefore, IPFS is utilised to store large-sized data provenance by generating content-addressed hash with a fixed length. The generated content-addressed hash or Content Identifier (CID) can be stored on the blockchain-based smart contract through indexing as discussed in Section 2.4.1 to retrieve a large-sized data provenance from the IPFS [105]. This improves the blockchain storage size problem and data provenance traceability [106][107].

Similar to blockchain technology as discussed in Section 2.4.1, the concept of Merkle tree Directed Acyclic Graph (DAG) is utilised in IPFS. IPFS Merkle tree DAGs are an immutable form of distributed ledger [108] technology which does not suffer from the scalability issues that are common in blockchain technology. Merkle DAG nodes are immutable. Any change in a node would alter its identifier and thus affect all the ascendants in the DAG, essentially creating a different DAG.

Ruan et al. [109] presented LineageChain, a fine-grained data provenance traceability solution for blockchains platforms to achieve efficient tracking and querying of semantic information. The authors captured verifiable semantic information during the execution of blockchain-based smart contracts and stored it in a Merkle tree, which was converted into a DAG [110]. The solution also incorporated a skip list index utilised to navigate and validate previous states for optimising data provenance tracking and queries.

IPFS Merkle hash tree DAG node is created by utilising Python IPFS Application Programming Interface (API) [111] which accepts data provenance as a JSON markup format that is stored with the function *IPFS DAG PUT* to obtain a unique node hash. There is no requirement to "balance" IPFS Merkle hash tree DAG, as depicted in Figure 2.12. Therefore, to validate whether a node is in the IPFS Merkle hash tree DAG, only a few hashes of nodes are required rather than the entire tree.

The command *IPFS DAG GET* then accepts a node hash that is utilised to inspect the content history and origin of a stored data provenance JSON file. This enables the role players to traverse the IPFS Merkle hash tree DAG to read the required version that is secure, immutable and traceable. The ability to track and query IPFS Merkle hash tree DAGs, facilitates an end-to-end supply chain tracking, enabling verification of product history and origin to ensure the integrity of data provenance [37].

This data provenance tracking approach was done by mapping hash table indexing to query the stored content-addressed hash or Content Identifier (CID) on the public blockchain-based smart contract. This CID enables the role players to traverse the IPFS Merkle hash tree DAG to read data provenance that is secure, immutable and traceable.

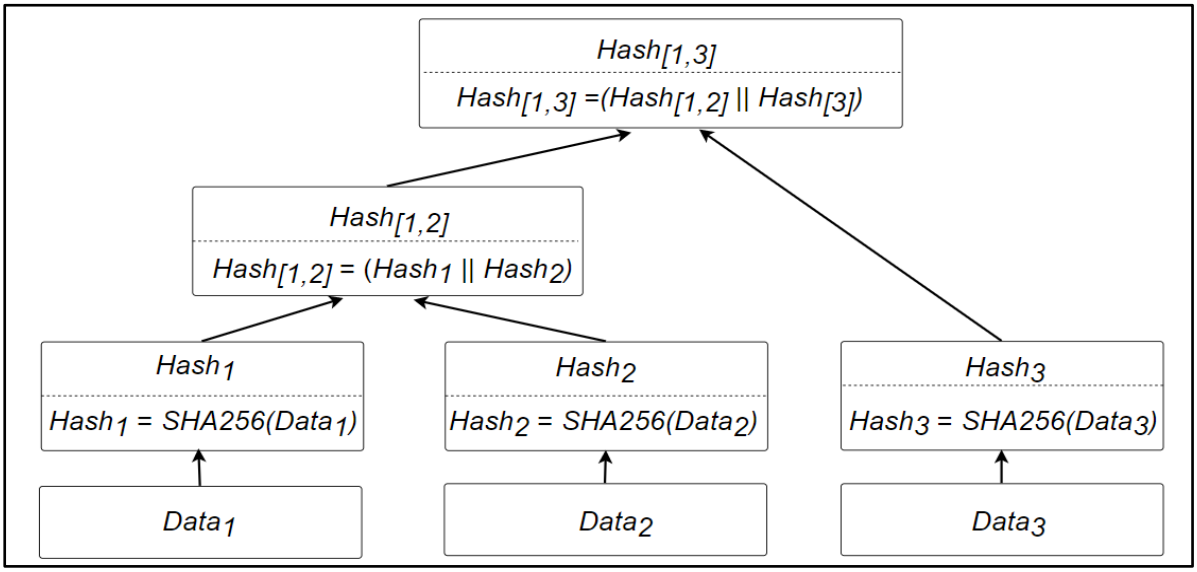


Figure 2.12 Illustration of a "trickle" Merkle tree [112]

By combining blockchain technology, EPCIS [113][2] and IPFS [106][107], a secure, immutable and traceable approach is achieved for storing data provenance. In the following section, a summary of the research done on the various data provenance traceability problems and the corresponding solutions implemented using blockchain-based smart contracts is presented.

2.4.5 Summary of data provenance traceability problems and solution methods

This section tabulates the number of recorded studies done on the various data provenance traceability problems discussed in Section 2.3.1 with respect to the blockchain based smart contract traceability and transparency solutions discussed in Section 2.4.4. This is presented in Table 2.4. An analysis of the table is then performed to determine the most common type of data provenance traceability problem and the preferred solution.

Table 2.4 Number of studies done on the various data provenance traceability problems

	Agnostic	Agricultural product	Agri-food products	Agrochemicals/Pesticides	Airbags	Aircraft spare parts	Aquatic products	Automotive, electrical, and	Beef	Bread	Buckwheat	Cotton	Crops	Dairy products	Drugs	Eggs	Electronics	E-products	Fish and seafood products	Furniture	Goods	Grains	Meat	Medical supplies	Mobile phones	Pepper	Perishable food products	Perishable medical diagnostic kits	Precast concrete products	Soyabean	Steel	Vaccines	Warehouse management	Wheat, flour	Wind turbine blade	Wine	Wood
Agriculture/agri-food	1	3	2	1					1		1		1													1				1						1	
Electronics																	2								2												
Food	3	1					1		2	1				1		1			1			1	1				2							1			
Generic Supply Chain	10																	2		1	2			1									1		1		1
Manufacturing	1							1																1							1						
Other domains					1	1						1																	1								
Pharmaceuticals	1														8												1					2					
Total	16	4	2	1	1	1	1	1	3	1	1	1	1	1	8	1	2	2	1	1	2	1	1	2	2	1	2	1	1	1	1	2	1	1	1	1	1

A graphical representation of Table 2.4 depicts that the generic supply chain problems are the most researched among various application domains of data provenance traceability problems. This is depicted in Figure 2.13.

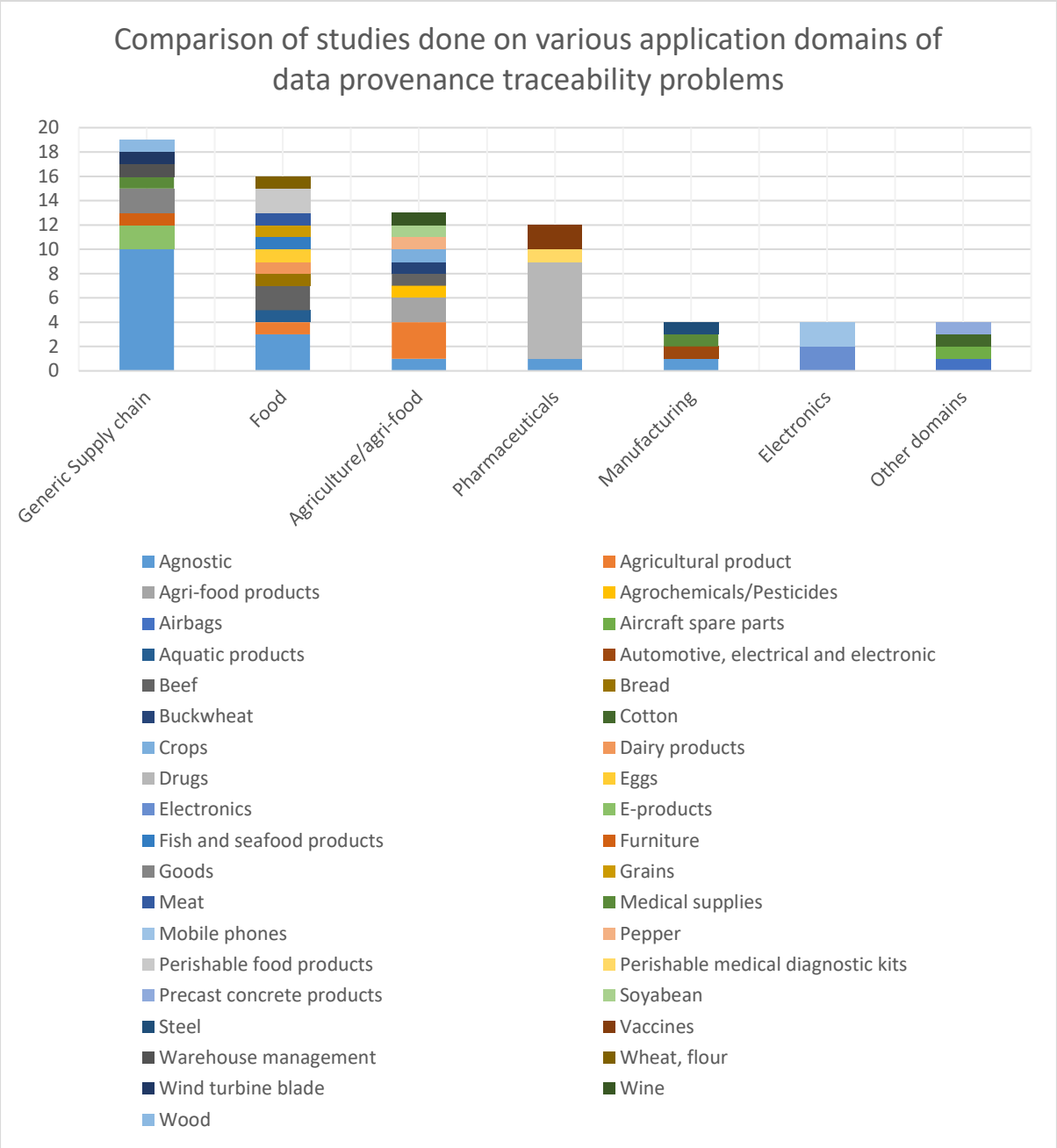


Figure 2.13 Comparison of studies conducted on several application domains

Figure 2.14 depicts that manufacturing is among the least researched with limited or no research on bottled water as a product.

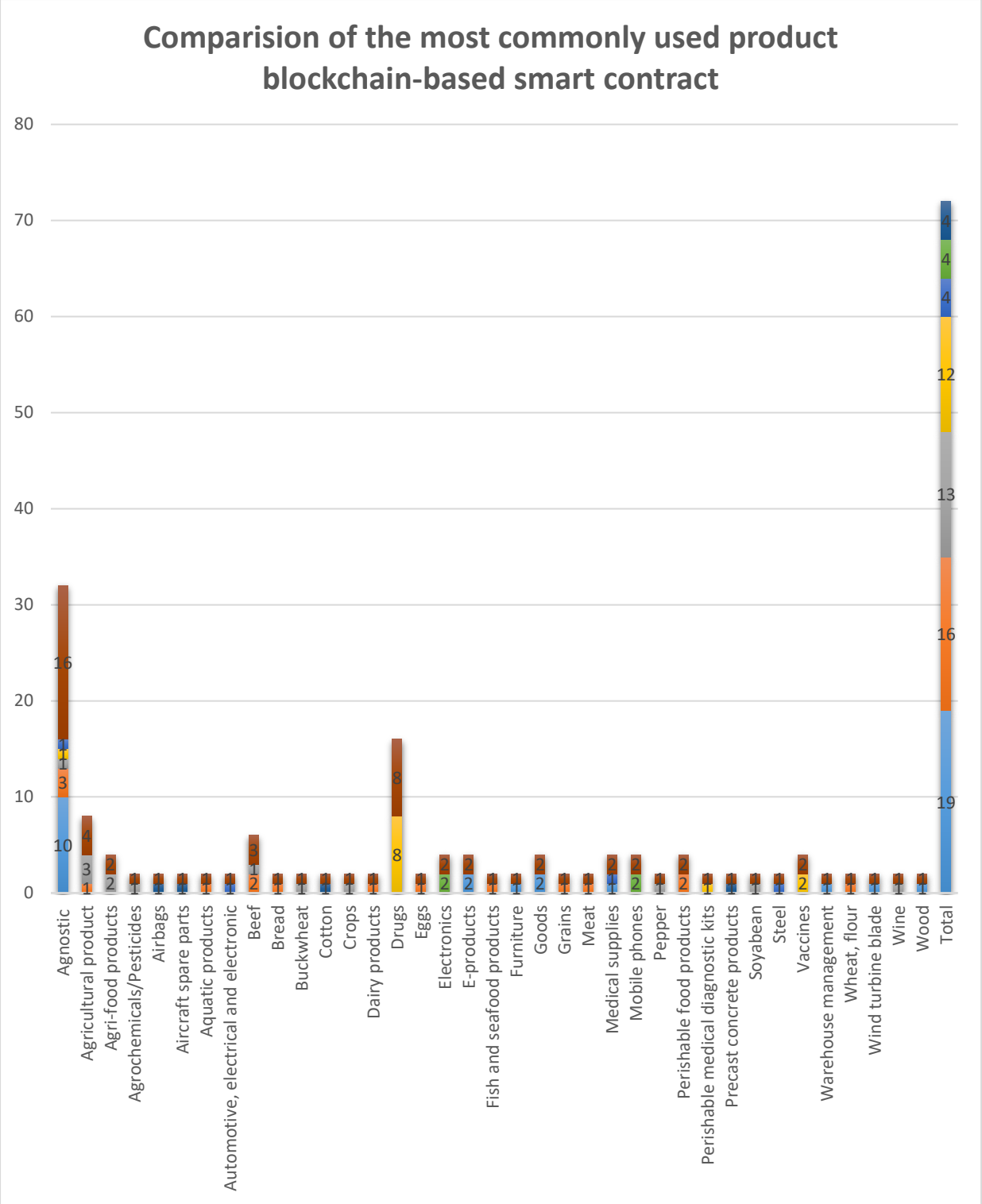


Figure 2.14 Most commonly utilised product blockchain-based smart contract

2.5 Limitations of existing research

This specific research study focuses on developing a blockchain-based smart contract data provenance tracking application for a smart manufacturing environment. Data provenance traceability and transparency are crucial in such an environment. The challenges related to data provenance traceability are split into two categories, being centralised and decentralised architecture problems, as discussed in Section 2.3.1. The centralised category looks at problems concerned with cloud-based data provenance traceability in a supply chain, while the decentralised category looks at problems concerned with blockchain-based data provenance traceability in the manufacturing supply chain.

Centralisation has been identified as a major problem in achieving data provenance traceability across an end-to-end supply chain, discussed in Section 2.3.2. Therefore, the hypothesis of the research study is that these challenges can be overcome by incorporating data provenance into blockchain-based smart contracts for traceability and transparency. As depicted in the summary in Section 2.4.5, the amount of research done [13] on blockchain-enabled smart contract data provenance traceability in products in a manufacturing supply chain was seemingly limited. This is based on the analysis of 72 peer-reviewed papers from 2018 to 2021.

These limitations are classified as follows:

- Performance – to maximise the performance across an end-to-end supply chain, traceability solutions involve leveraging factors that influence the selection of a blockchain platform for incorporating data provenance into smart contracts.

Ethereum (legacy) blockchain platform is the most utilised due to its well-established ecosystem of developer tools and best practices. However, the efficiency of permissionless public blockchain platforms like *Ethereum* (legacy) is low, which affects the cost of adoption, as discussed in Section 2.4.1.

- Layer 2 blockchain platform solutions – to improve the cost of adoption, novel approaches that utilise layer 2 blockchain network remain a challenge.

The challenge in improving the cost of adoption is through novel approaches that utilise layer 2 blockchain platform solutions. Despite the 72 peer-reviewed papers, there is a limited number of solutions developed utilising layer 2 blockchain platforms, like *Polygon*, which leverage the foundation of *Ethereum* (legacy) to reduce transaction fees, as discussed in Section 2.4.2 and Section 2.4.3.

- Investment and operational costs – based on the large volume of transactions across an end-to-end supply chain, high costs are incurred in terms of data storage.

Hybrid approaches that enable decentralised storage to minimise on-chain data storage, such as InterPlanetary File System (IPFS), are on the increase, as discussed in Section 2.4.4. However, of the 72 peer-reviewed papers, there is limited development and utilisation of IPFS, in conjunction with Merkle tree Directed Acyclic Graph (DAG) [37].

- Standardisation and certification of the supply chain traceability process – establishing an end-to-end traceability framework, particularly in health and safety-sensitive sectors [114] necessitates a standardised approach handled by the Electronic Product Code Information Service (EPCIS) developed by GS1 standards [13].

GS1 standard principles can be applied to supply chains across various sectors, including food and beverage, pharmaceuticals, medical devices and agri-food. However, among the 72 peer-reviewed papers, there is limited implementation of EPCIS by GS1 standards.

The major drawback of these existing solutions is that there is very little research on combating the centralisation of manufacturing of bottled water and none when it comes to cost-effective public blockchain platforms.

To conclude Chapter 2 this research aims to utilise the identified cost-effective and suitable blockchain platform in Section 2.4.3 to develop a blockchain-based smart contract data provenance tracking application within a smart manufacturing environment. The application will log product or raw material data provenance pertaining to the composition and production flow. The research localisation of this study is depicted in Figure 2.15 and will be used for advancing the research methodology in Chapter 3.

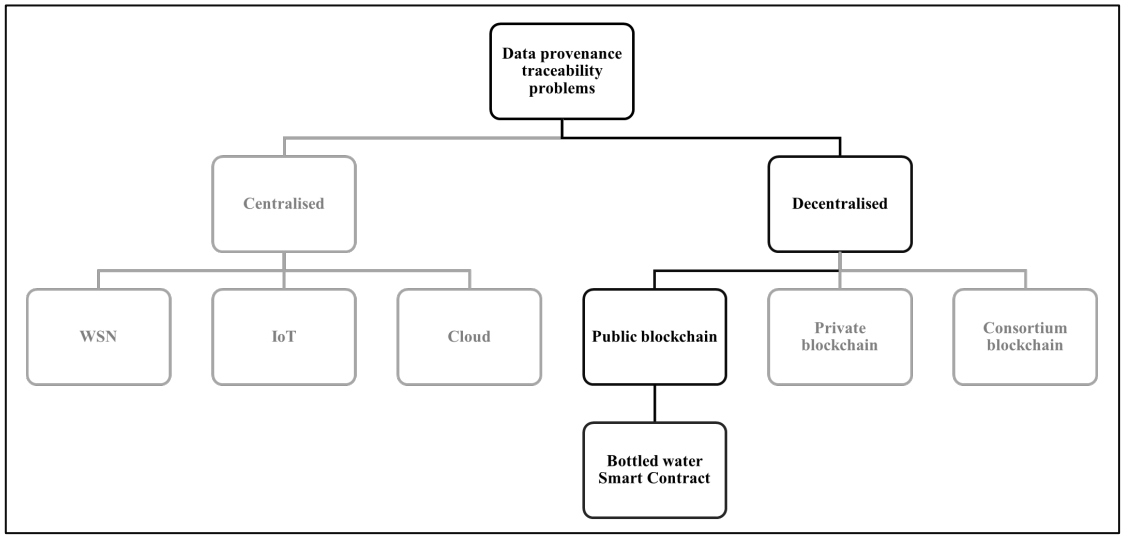


Figure 2.15 Research problem localisation

Chapter 3: Research methodology

3.1 Introduction

The aim of this chapter is to explain the methodology utilised to develop a blockchain-based smart contract data provenance tracking application within a smart manufacturing environment to log product or raw material information pertaining to the composition and production process. In this research study, a blockchain-based smart contract data provenance tracking application will be developed utilising *Polygon* public blockchain platform described in Section 2.4.3 as it is deemed secure, immutable, traceable, and transparent.

In order to validate and demonstrate the experimental setup of the developed blockchain-based smart contract data provenance tracking application, a case study is selected. In this research study, the water bottling plant located at the Bloemfontein campus of the Central University of Technology (CUT) [115] was chosen as the case study.

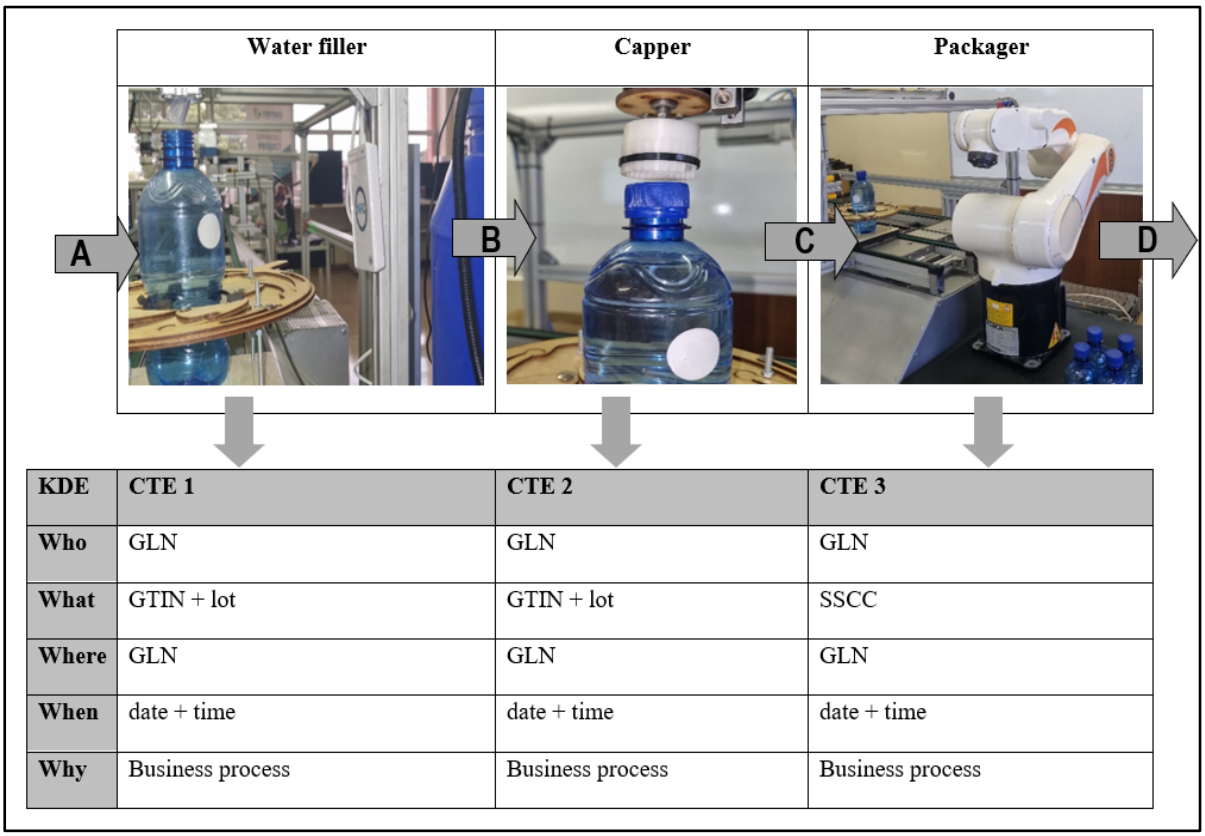
3.2 The Water bottling plant: A case study

The water bottling plant in this research study is a smart manufacturing environment utilising the Make-to-Order (MTO) approach [116]. Kuriakose and Vermaak [117] discuss contents of a business plan with the aim of the water bottling plant to produce bottled water for internal use during meetings, conferences, and other functions at various faculties of CUT. According to Kuriakose and Vermaak [118], the water bottling plant's production requirements include bottling 5000 litres of water per month, sourcing and storing water containing all minerals, and producing filled and capped 500ml and 750ml bottles.

The Critical Tracking Events (CTEs) are events that occur as the traceable item is traded among different role players. The Key Data Elements (KDEs) capture the semantic information of data provenance with the five “W’s”, which are Who, What, Where, When and Why.

This research study proposes a solution to ensure data provenance in smart manufacturing environments by incorporating bottled water data into a blockchain-based smart contract, ensuring the security, immutability, and traceability of the data. The solution utilises GS1 standard’s five W’s to collate semantic information on the KDEs related to end-user’s new orders for bottled water, including Who, What, Where, When and Why, as discussed in Section 2.3.1.

The proposed solution for data provenance collection involves the utilisation of Near Field Communication (NFC) tags. NFC antennas are mounted at designated points A, B, C, and D in Figure 3.1. The NFC antennas are mounted at the start and end of each Smart Manufacturing Unit (SMU), designated for water filling, capping, and packaging activities. Each SMU activity triggers a CTE, which is then incorporated into a blockchain-based smart contract to ensure data security, immutability, and traceability.



KDE	CTE 1	CTE 2	CTE 3
Who	GLN	GLN	GLN
What	GTIN + lot	GTIN + lot	SSCC
Where	GLN	GLN	GLN
When	date + time	date + time	date + time
Why	Business process	Business process	Business process

Figure 3.1 The schematic outline of a water bottling plant at CUT

The results of this research study design process will illustrate how data provenance tracking applications within a smart manufacturing environment can be utilised to log bottled water composition and production flow for traceability purposes. By incorporating bottled water data into a blockchain-based smart contract, the research study intends to overcome the limitations of existing techniques and allow the end-user to access data provenance through a tag attached to the bottled water. This section has discussed the primary considerations of the water bottling plant and why it was selected for this study.

3.3 Experimental setup

This section illustrates the proposed experimental setup for blockchain-based smart contract data provenance tracking application within a water bottling plant. To facilitate the design process, the schematic outline of the water bottling plant depicted in Figure 3.1 is split into three subsystems. This is depicted in Figure 3.2.

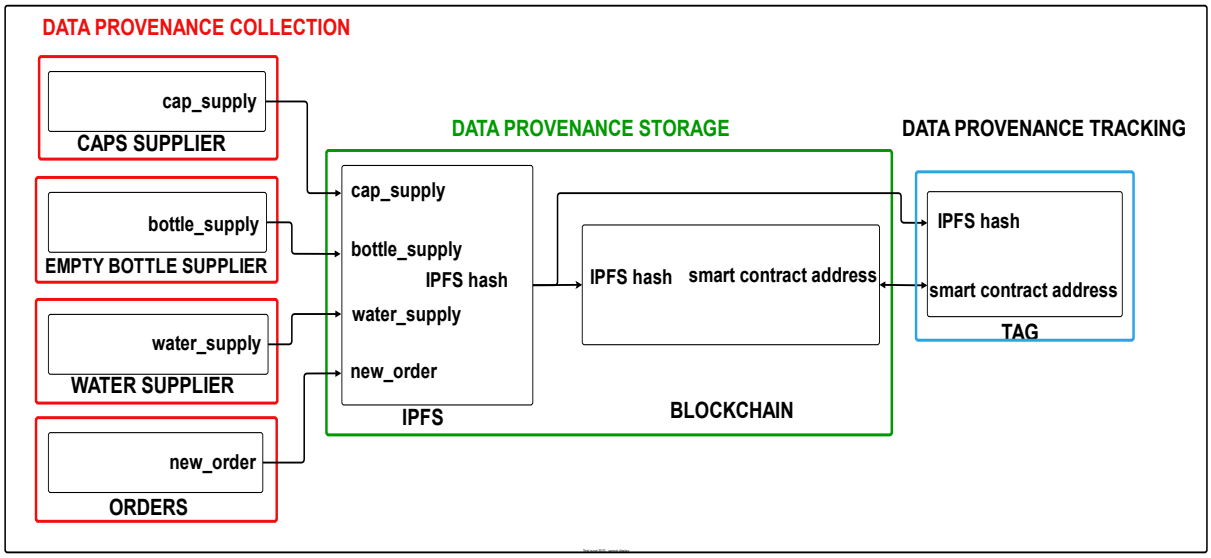


Figure 3.2 The flow diagram of the experimental setup split into three

The three subsystems with their specific tasks are defined as follows:

- Data provenance collection
- Data provenance storage
- Data provenance tracking

The experimental setup with the different subsystems is depicted in Figure 3.3. The block diagram of the experimental setup depicts the three subsystems described in Figure 3.2.

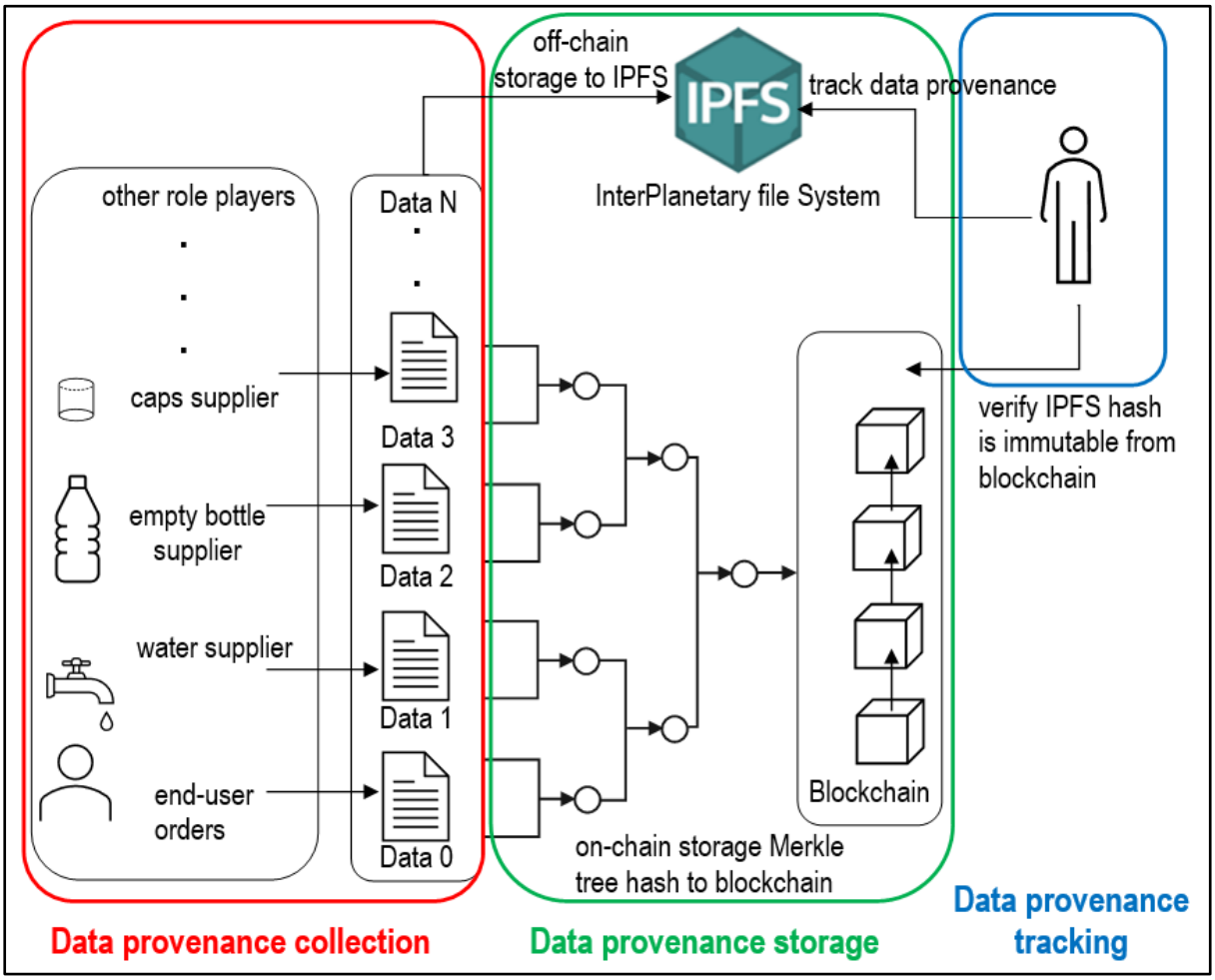


Figure 3.3 The block diagram of the experimental setup

The following section provides a detailed explanation of each component of the experimental setup, starting with data provenance collection, followed by storage and tracking. It is important to note that certain functions of the water bottling plant, such as capping, labelling, and packaging have been omitted from the experimental setup for convenience.

3.3.1 Data provenance collection

This section discusses how data provenance can be collected in the experimental setup. In this specific experimental setup, there are three sources for data, as depicted in Figure 3.3. These input sources include the end-user's new order, water supplier and empty bottle supplier. The end-user provides order requirements, followed by the water sourced from a specific supplier, and the empty bottles and caps from other suppliers. If additional role players need to be added, provisions can be made.

The GS1 standards which are fundamental in identifying, capturing, and sharing data provenance, are implemented in this experimental setup, as discussed in Section 2.3.1. The data is then written in the Near Field Communication (NFC) tag in a JavaScript Object Notation (JSON) [50] markup format and shared as object of $Data_n$ (Data N), where N is an index set starting from zero. The NTAG21x [119][120] series are utilised, which are compatible with most NFC-enabled devices. This is depicted in Figure 3.4. Each role player data is collated through an NFC tag with its Unique Identifier (UID) as Global Trade Item Number (GTIN).

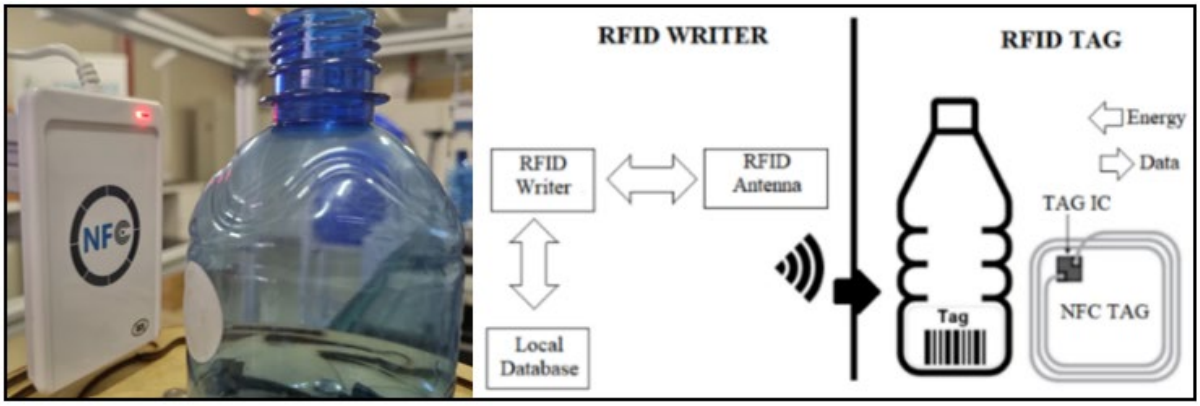


Figure 3.4 NFC antenna and tag for data provenance collection

The end-user's new order includes various attributes pertaining to the specific end-user requirements. These attributes include the UID, represented by a GTIN of the trackable end-user order. The end-user requirements include additional attributes such as the number of 500ml bottles, and the number of 750ml bottles. The date and time required attribute specifies the end-user's *new_order* required delivery date, in Coordinated Universal Time (UTC) offset format. This data is collected utilising GS1 standards Key Data Elements (KDEs) in a JSON markup format. The Figure 3.5 describes a JSON markup format array *new_order*, the first entry object of $Data_0$.

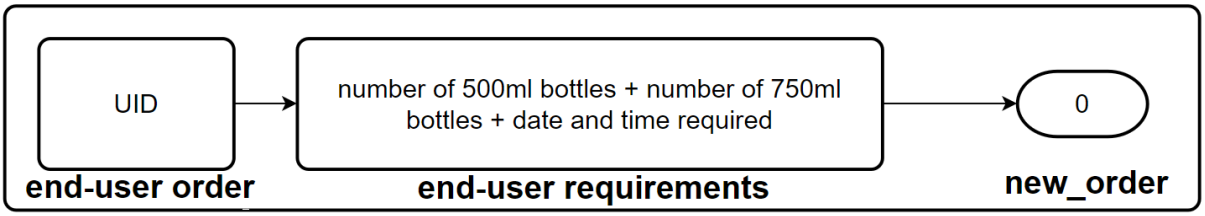


Figure 3.5 End-user's new order data provenance collection

In the water bottling plant, the purified water is sourced from a specific supplier, and the NFC tag UID is utilised as GTIN to identify the water source. The collected data includes attributes such as the Batch/Lot number, Volume and pH level of the water. The collected data attributes are written in the NFC tag in a JSON markup format, which includes the GTIN and KDEs shared as *water_supply*, the first entry object of $Data_0$. This is depicted in Figure 3.6.

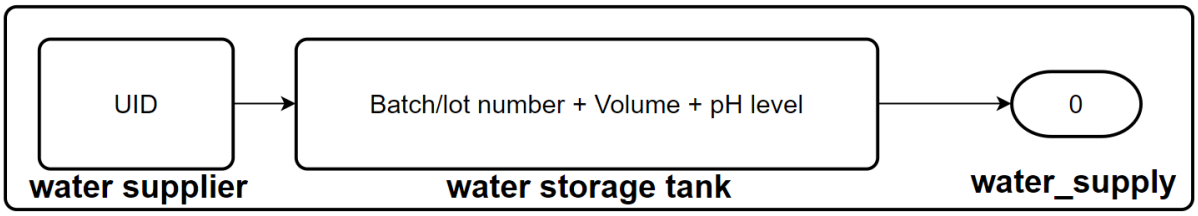


Figure 3.6 Water supply data provenance collection

The empty bottles are sourced from another supplier and the NFC tag UID is utilised as the GTIN. The collected data attributes include the Batch/Lot number, the number of 500ml bottles, and the number of 750ml bottles from the supplier. The collected data is written in the NFC tag in a JSON markup format, which includes the GTIN and KDEs shared as *bottle_supply*, the first entry object of Data0. This is depicted in Figure 3.7.

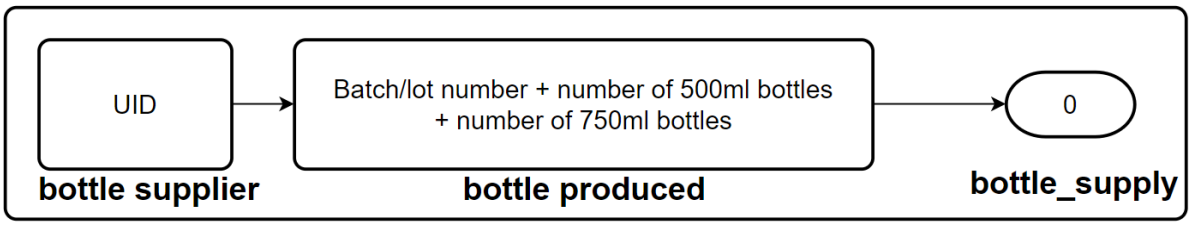


Figure 3.7 Bottle supply data provenance collection

This section has discussed how data provenance was collected in the experimental setup for the blockchain-based smart contract data provenance tracking application within a water bottling plant.

3.3.2 Data provenance storage

A study that was done as part of this research into the factors that influencing the selection of a blockchain platform resulted in this input. The efficiency of permissionless public blockchain platforms is low, which can increase adoption costs as discussed in Section 2.4.2. Therefore, the InterPlanetary File System (IPFS) [104] is utilised to store large-sized data provenance by generating content-addressed hash with a fixed length.

The research study done [55] compared transaction fees for various data types and data sizes of blockchain-based smart contracts on different blockchain platforms as discussed in Section 2.4.3. Based on the findings, a layer 2 solution called *Polygon* [99] public blockchain platform was identified. The blockchain-based smart contract granularity will be limited to Critical Tracking Events (CTE) to reduce transaction fees associated with storing large amounts of data provenance.

To validate and demonstrate the experimental setup, a scenario is illustrated where an end-user places an order through a software application. A data provenance tracking application is developed utilising React Native framework which interprets JavaScript. The Graphical User Interface (GUI) is provided in Appendix A. A data provenance tracking application contains blocks where the end-user can enter various attributes which include their name, number of 500ml bottles, and the number of 750ml bottles and required delivery date. On completion, the end-user can place the order to the water bottling plant.

The water bottling plant collates end-user order and attributes are written in the Near Field Communication (NFC) tag in a JavaScript Object Notation (JSON) markup format. All sources collected are concatenated into a single JSON markup format and stored in the IPFS for off-chain storage. The role player sources are stored in the IPFS as node $Hash_0$, as depicted in Figure 3.8. The contents of node $Hash_0$ can be utilised to optimise the water bottling plant before production starts [118] by informing the Smart Manufacturing Unit (SMU) of the appropriate bottle volume and quantity to fill the end-user's new order.

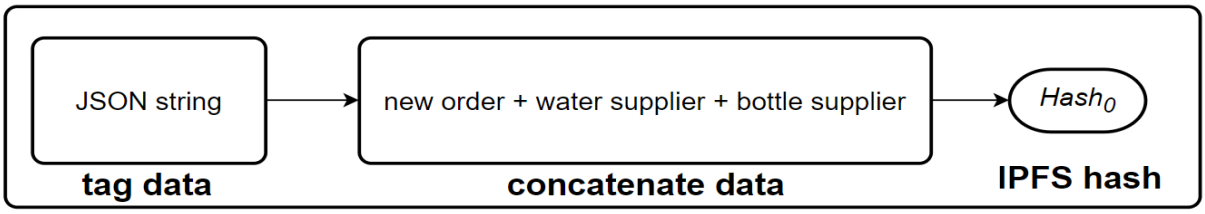


Figure 3.8 The role player's sources storing in the IPFS

The end-user's new order, represented as node $Hash_1$, is stored in the IPFS as *new_order*, the first entry object of $Data_0$. The end-user's *new_order* will be filled with water sourced from *water_supply* and empty bottles from *bottle_supply*, both of which are stored in the IPFS as entry $Data_0$. This is depicted in Figure 3.9.

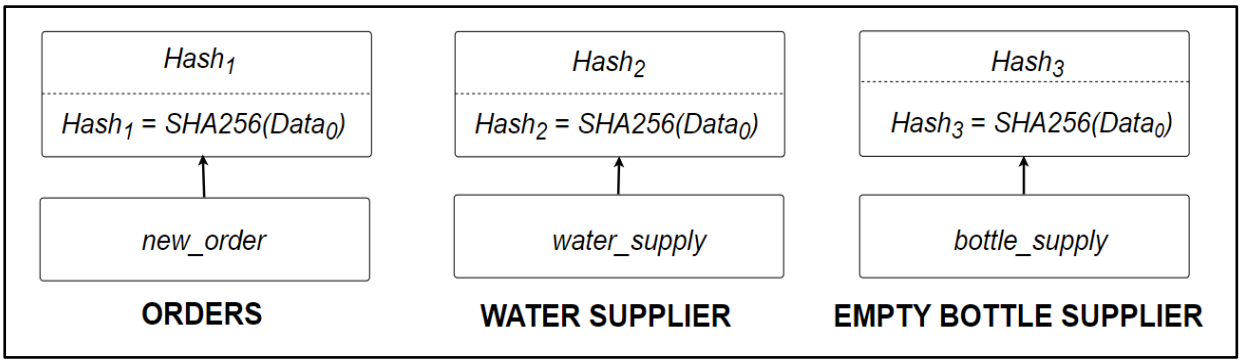


Figure 3.9 The IPFS utilised to shorten data provenance collected into hashes

The concept of Merkle tree Directed Acyclic Graph (DAG) [110] is utilised in IPFS, as discussed in Section 2.4.4. The ability to track and query the IPFS Merkle tree DAG, facilitates an end-to-end supply chain tracking, enabling verification of product history and origin to ensure the integrity of data provenance [37]. Therefore, the principles of the IPFS Merkle tree DAG are implemented with Smart Manufacturing Unit (SMU), tasked with filling the water bottles, as depicted in Figure 3.10.

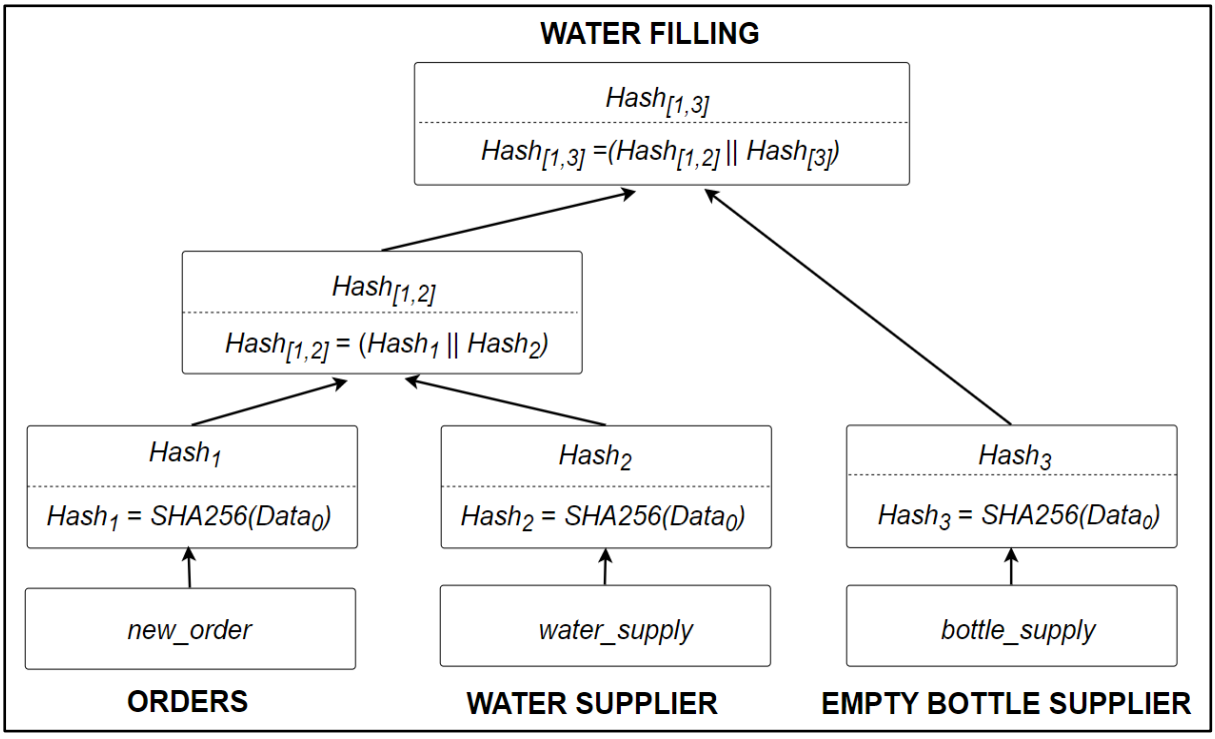


Figure 3.10 Merkle tree for data provenance storage for water filling

As depicted in Figure 3.10, IPFS Merkle tree DAG node is created by utilising Python IPFS Application Programming Interface (API) [111]. The function *IPFS DAG PUT* accepts data provenance as a JSON markup format that is stored to obtain a shortened unique content-addressed hash or Content Identifier (CID). The *water_supply* entry $Data_0$ generates node $Hash_1$, while the *bottle_supply* entry $Data_0$ generates another node $Hash_2$.

The parent IPFS Merkle tree DAG hash value, node $Hash_{[1,2]}$ is derived from child nodes, $Hash_1$ and $Hash_2$. At this IPFS Merkle tree DAG node $Hash_{[1,2]}$ the pH level of water in the tank is measured to update the data provenance stored as node $Hash_2$. This is depicted in Figure 3.11.

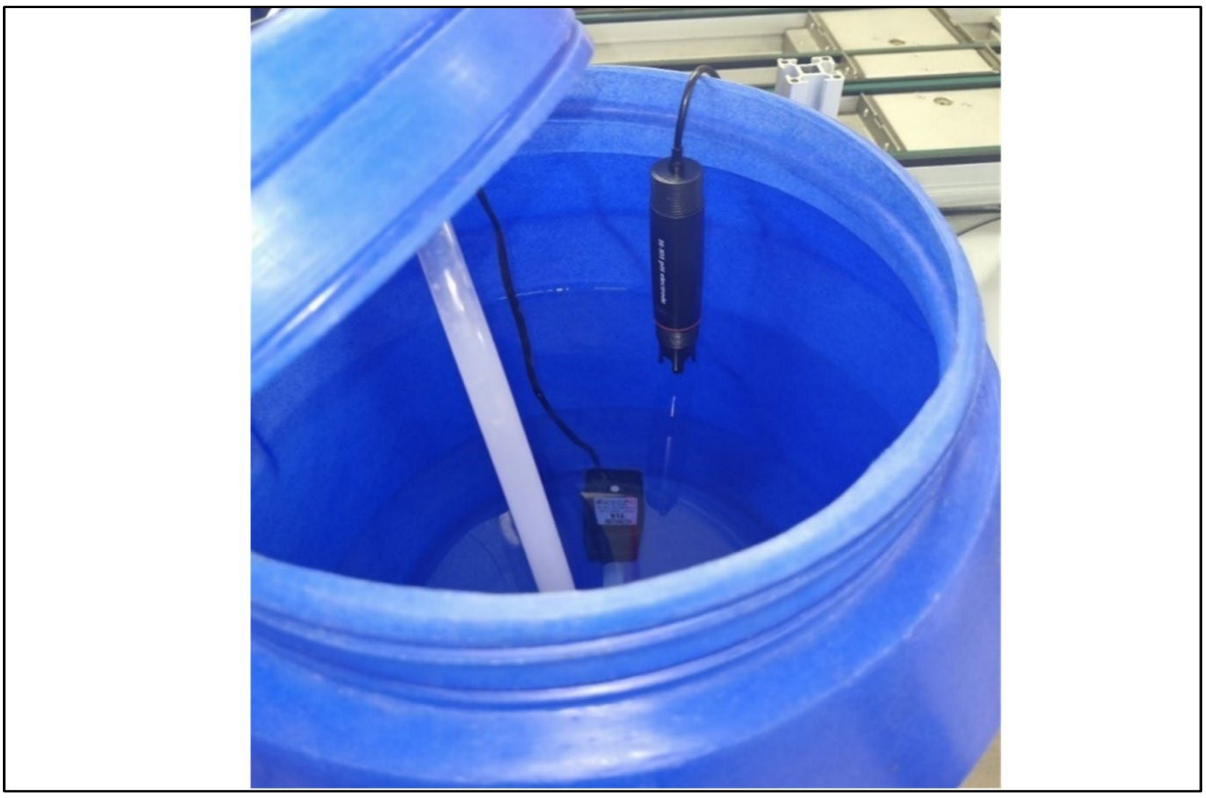


Figure 3.11 Measuring pH level of water in the tank

Smart Manufacturing Unit 1 (SMU1) also utilises proximity sensors to count the water bottles. This is depicted in Figure 3.12. This is utilised to update data provenance stored as IPFS Merkle tree DAG node $Hash_3$. As depicted in Figure 3.10, the concatenation of node $Hash_{[1,2]}$ and $Hash_3$ result in a root node $Hash_{[1,3]}$ which represents Critical Tracking Event CTE_1. This data is then incorporated into the *Polygon* to incorporate the root node $Hash_{[1,3]}$ for CTE_1 blockchain-based smart contract. Antenna B appends the NFC tag with IPFS Merkle tree DAG node $Hash_{[1,3]}$ to create a data provenance chain for an end-to-end traceability.

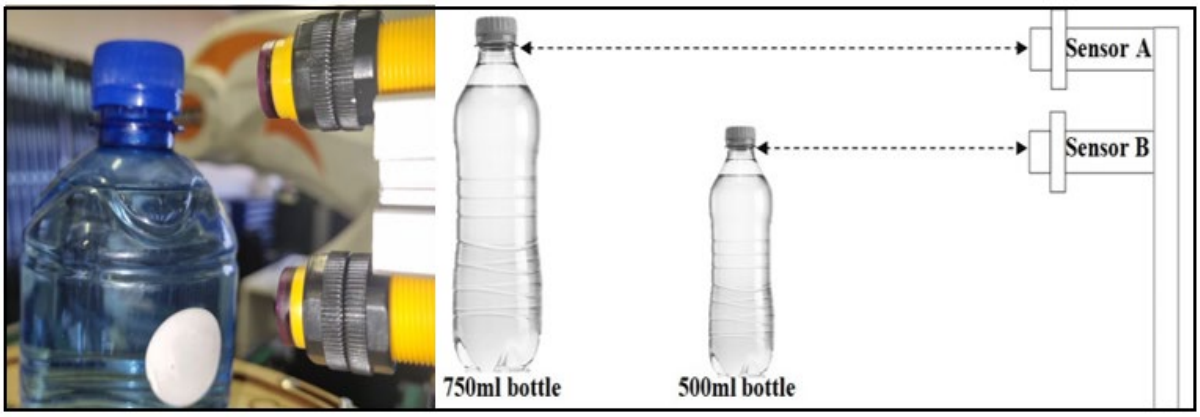


Figure 3.12 Counting the water bottles

As mentioned in Section 2.4.1 the blockchain-based smart contract is developed in the Solidity programming language [70] compiled utilising a Python script. The blockchain-based smart contract flow diagram is depicted in Figure 3.13 assigned with arbitrary name *BottledWaterProvenance.sol* to track the data provenance of bottled water.

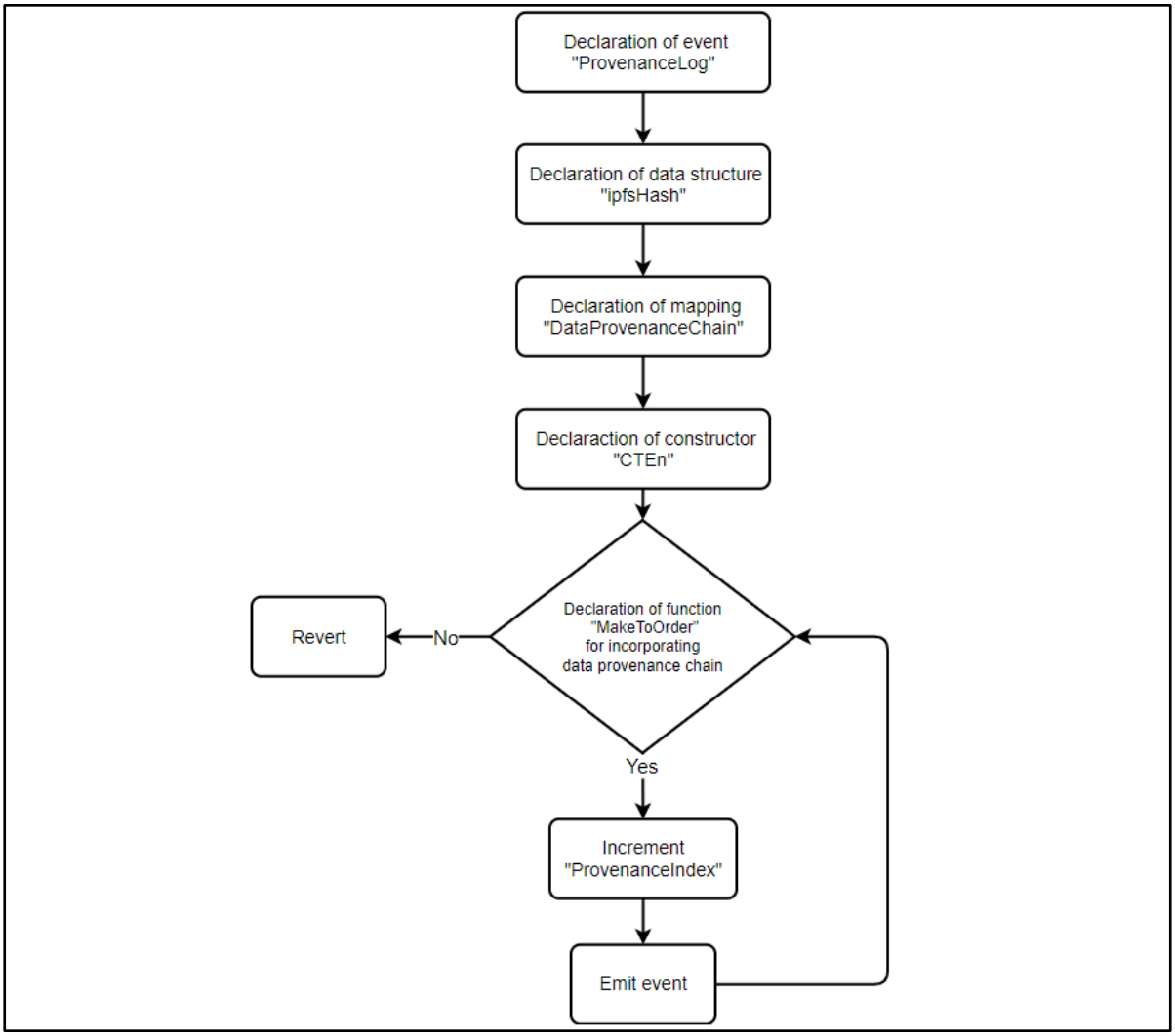


Figure 3.13 The flow diagram of the blockchain-based smart contract

As depicted in Figure 3.13**,** the blockchain-based smart contract is developed as follows:

- An event *ProvenanceLog* to log the data provenance chain is declared. This is done utilising the event keyword that accepts two arguments of an indexed unassigned integer 256 and a string named CTE.
- The data structure for the provenance chain is declared. This is done utilising a structure named *ipfsHash* that contains a string named CTE.
- An index mapping to incorporate data provenance chain is declared. This is done utilising a mapping keyword named *DataProvenanceChain* that accepts a public uint256 value. This mapping is equivalent to a hash table *ProvenanceIndex* index uint256 set to 1.
- The constructor is declared. This is done utilising a function that is executed when the blockchain-based smart contract is deployed. The constructor accepts a string argument named hash CTE_n. The *DataProvenanceChain* mapping hash table is then set with a uint256 value of 0 and initial IPFS Merkle tree DAG node $Hash_0$ as hash CTE_0. The *ProvenanceLog* event log is emitted with an index of 0 and hash CTE_0 value.
- The function named *MakeToOrderStage* is declared and accepts a string argument named hash CTE_n. The function is for incorporating the data provenance chain hash CTE_n and utilising mapping hash table *ProvenanceIndex* index that is incremented for every entry.
- Finally, an event is emitted with *ProvenanceIndex* index variable and the hash CTE_n value that is accessible by all the role players.

As discussed in Section 2.4.1, the blockchain-based smart contract is compiled and deployed to *Polygon*. Once the blockchain-based smart contract is deployed, its generated content-addressed *Transaction Hash* transaction hash is written in the NFC tag as a Uniform Resource Locator (URL) link. This section has discussed how data provenance was stored off-chain in the IPFS Merkle tree DAG and its root hash incorporated into the blockchain-based smart contract.

3.3.3 Data provenance tracking

As discussed in the introduction, the aim of this research study is to develop a blockchain-based smart contract data provenance tracking application in order to ensure traceability. Therefore, the final part of the experimental setup illustrates how the end-user can access and track critical data pertaining to the history and origin of contents, composition, quantity and timestamp of water bottle filling.

This section introduces an anomaly with the end-user's order placed. A scenario of one of the bottles not being filled with water is illustrated. To validate and demonstrate this anomaly, data provenance tracking is conducted in relation with the version stored in Section 3.3.2. A flow diagram of the data provenance tracking approach is depicted in Figure 3.14.

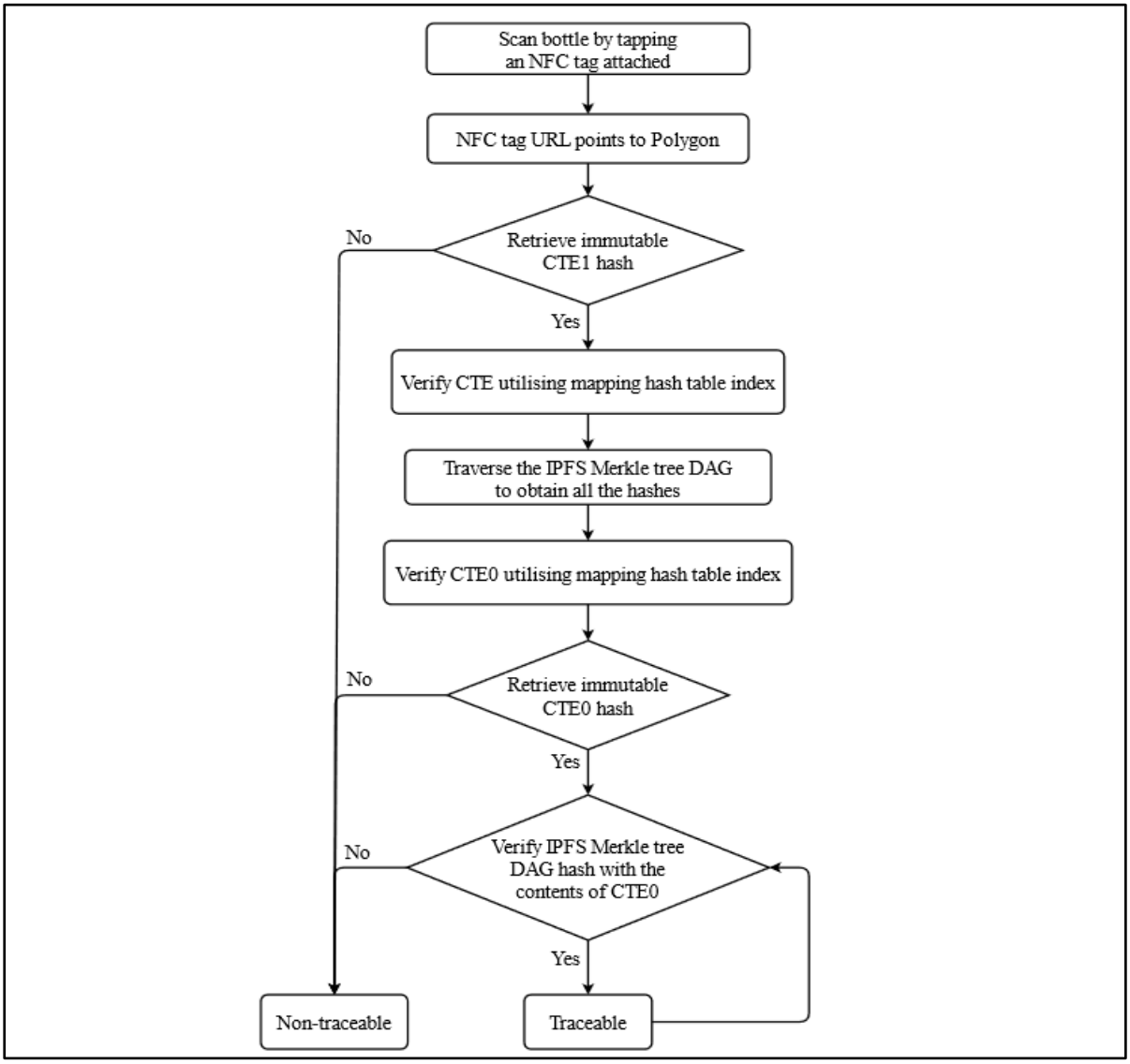


Figure 3.14 A flow diagram of the data provenance tracking approach

The end-user can read data provenance by tapping a Near Field Communication (NFC) tag attached to the bottled water, which contains a content-addressed transaction hash Uniform Resource Locator (URL) link. A URL link directs the end-user to the deployed blockchain-based smart contract transaction hash on the *Polygon* public blockchain platform. This is to retrieve the root InterPlanetary File System (IPFS) Merkle tree Directed Acyclic Graph (DAG) node $Hash_{[1,3]}$, which represents a Critical Tracking Event CTE_1 for water filling which is immutable.

The *CTE1* associated with node $Hash_{[1,3]}$ from *Polygon* is verified utilising a *DataProvenanceChain* mapping hash table *ProvenanceIndex* index uint256 value set to 1. The verified *ProvenanceIndex* index will correspond to the water filler described in Figure 3.1 of the experimental setup. A top-down data provenance tracking approach [38] is achieved by the ability to traverse the IPFS Merkle tree DAG [37] with node $Hash_{[1,3]}$. This is depicted in Figure 3.15.

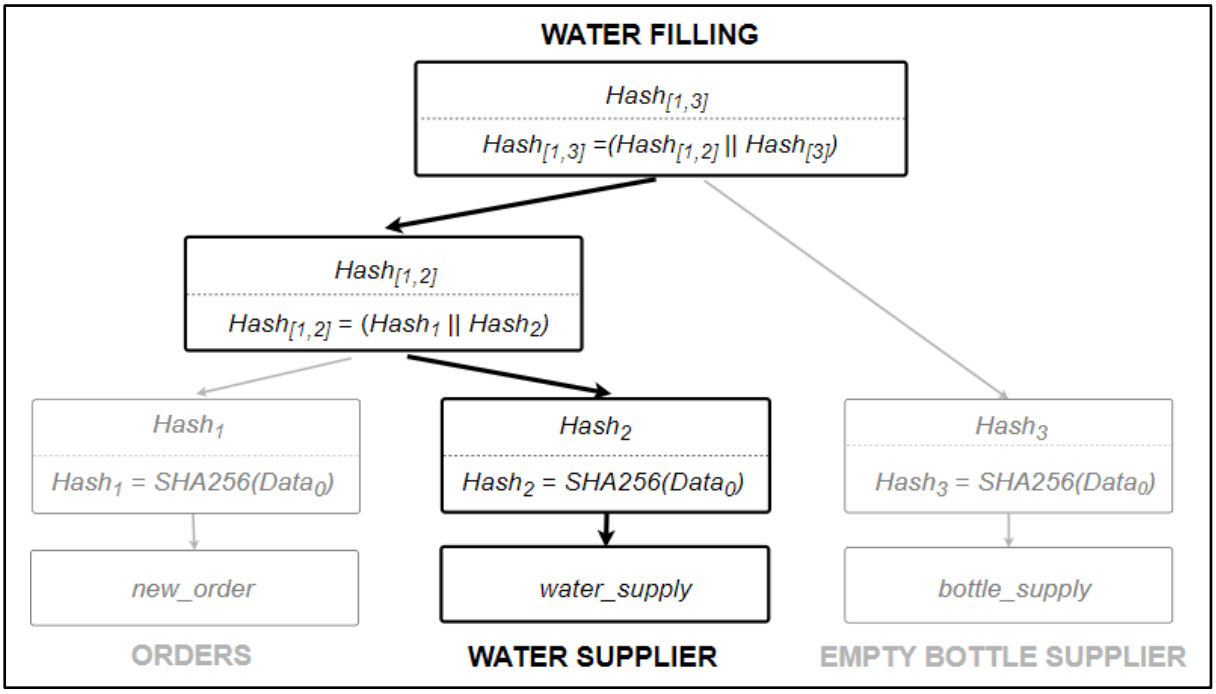


Figure 3.15 IPFS Merkle tree DAG inversion method for data provenance tracking

As depicted in Figure 3.15, modifying a node would result in a change to its root hash, subsequently impacting all the ancestor nodes within the Merkle tree, leading to the creation of a different DAG. A top-down data provenance tracking approach is as follows:

- Firstly, the links from the parent node $Hash_{[1,3]}$ are then verified, with two child links. The one pointing to node $Hash_{[1,2]}$ of the measured water pH level in the tank. The other as node $Hash_3$ of the counted water bottles.
- Follow the link to node $Hash_{[1,2]}$ and retrieve its content utilising its IPFS Merkle tree DAG hash, which has two child links pointing to node $Hash_1$ and $Hash_2$.
- The content corresponding to the *new_order* is retrieved by following the link to node $Hash_1$.
- Similarly, the content corresponding to the *water_supply* is retrieved by following the link to node $Hash_2$.
- Once the contents of node $Hash_1$ and node $Hash_2$ have been retrieved, navigate back to node $Hash_{[1,3]}$ and follow its links to node $Hash_3$, which includes the content corresponding to the empty *bottle_supply*.

After retrieving the contents of *new_order*, *water_supply* and *bottle_supply*, the hash CTE_0 is obtained from *Polygon.* The block of the blockchain platform comprises a header and body as discussed in Section 2.4.1. This is depicted in Figure 3.16. The hash CTE_0 is verified utilising a mapping IPFS Merkle tree DAG hash table of index *ProvenanceIndex* with a public uint256 value set to 0. The blockchain-based smart contract includes hashes CTEs for verifying the mapped hash table.

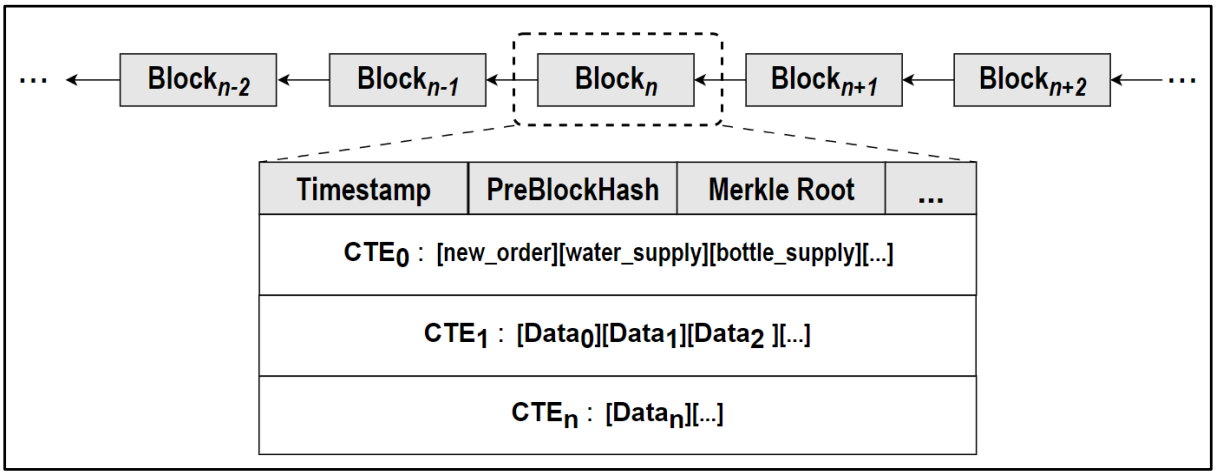


Figure 3.16 A mapping hash table of provenance-driven indexing

This mapping hash table indexing [98] enhances the ability to traverse the IPFS Merkle tree DAG to track and query semantic information to extract data provenance traceability, as discussed in Section 2.4.4. Each *ProvenanceIndex* includes a verifiable IPFS Merkle tree DAG hash with comprehensive data provenance associated with the end-user's *new_order* history and origin that is secure, immutable and traceable. It can be seen from Figure 3.16 that the hash CTE_0 corresponding to node $Hash_0$ is utilised as follows:

- The entry $Data_0$ of the *new_order* array is verified by examining the contents of node $Hash_0$ corresponding to node $Hash_1$.
- Similarly, the entry $Data_0$ of the *water_supply* array is verified by examining the contents of node $Hash_0$ corresponding to node $Hash_2$.
- Lastly, the entry $Data_0$ of the *bottle_supply* array is then verified by examining the contents of node $Hash_0$ corresponding to node $Hash_3$.

In conclusion Chapter 3 discussed how the end-user can access critical data pertaining to the history and origin of contents, composition, quantity and timestamp of water bottle filling, ensuring data provenance that is secure, immutable, traceable, and transparent. Table 3.1 provides a summary of data provenance hashes generated to clarify the results which will be presented in Chapter 4.

Table 3.1 Summary of data provenance hashes generated

Hashn	$Hash_n$	Critical Tracking Event
hash0	$Hash_0$	All the supplier's raw materials as CTE_0
hash1	$Hash_1$	New order
hash2	$Hash_2$	Water supplied
hash3	$Hash_3$	Empty bottle supplied
hash12	$Hash_{[1,2]}$	Order associated with the water supplied
hash13	$Hash_{[1,3]}$	Water filled root node as CTE_1

Chapter 4: Results and Analysis

4.1 Introduction

The aim of this chapter is to describe and illustrate the results of the methodology discussed in Chapter 3. The results and the analysis thereof for data provenance collection, storage and tracking will be discussed in the same sequence as in Chapter 3. Firstly, the type of collected data provenance is described and the corresponding results are shared. Next, the analysis focuses on the results of data provenance stored. Finally, the tests and results of the data provenance tracking are shared.

4.2 Results of data provenance collection

The main aim of the first subsystem of the experimental setup is to describe the type of data provenance collected and written in the Near Field Communication (NFC) tag. The first set of results are the end-user's new order data provenance collection, derived from Figure 3.5 and described in Section 3.3.1. The result of data provenance collection is depicted in Figure 4.1 and are from an order placed by the end-user (Customer A). The customer order is detailed in Appendix A.

```
{
  "Who": "(GSRN)9004000219900",
  "What": {
    "GTIN": "(GTIN)040A16AD36480",
    "Requirements": {
      "num500ml": "12",
      "num750ml": "6",
      "RequiredDeliveryDate": "2023-06-15T10:00:00+01:00"
    }
  },
  "Where": "(GLN)9001101530910",
  "When": "2023-05-30T15:00:00+01:00",
  "Why": "new_order"
}
```

Figure 4.1 The collected *new_order* object of $Data_0$

It can be seen from Figure 4.1 that the collected data provenance represented as an array of the end-user's *new_order* object of $Data_0$ is in JavaScript Object Notation (JSON) markup format. This JSON markup format follows the Key Data Elements (KDEs) approach developed by the GS1 standards. The KDEs capture the semantic information of data provenance with the five W's, which are Who, What, Where, When and Why as follows:

- *Who* – as a Global Service Relation Number (GSRN) identifies a specific end-user ((GSRN)9004000219900).
- *What* – encompasses various attributes pertaining to the specific end-user requirements. These attributes include the NFC tag's unique identifier (UID), represented by a Global Trade Item Number (GTIN) of the trackable end-user's *new_order* ((GTIN)040A16AD36480). The *Requirements* field includes additional attributes such as *num500ml*, which specifies the required number of 500ml bottles (12), and *num750ml* which indicates the number of 750ml bottles (6). The *RequiredDeliveryDate* attribute specifies the date and time (2023-06-15T10:00:00+01:00) when the end-user's *new_order* is required in Coordinated Universal Time (UTC) offset format.
- *Where* – is represented by a Global Location Number (GLN) indicates the physical location ID ((GLN)9001101530910) of the end-user.
- *When* – specifies the date and time (2023-05-30T15:00:00+01:00) the end-user's *new_order* was placed in UTC offset format.
- *Why* – provides information about the business process (*new_order*).

The next set of results are derived from Figure 3.6, and is represented as an array of the *water_supply* object of $Data_0$. The result of collected data provenance is shown in Figure 4.2 .

```
{
  "Who": "(GLN)9104000219900",
  "What": {
    "GTIN": "(GTIN)04131F6AD3680",
    "Batch_lot": "(Batch/lot)WS-200",
    "Volume": "(ml)10500",
    "pH": "8"
  },
  "Where": "(GLN)9101101530910",
  "When": "2023-05-30T14:00:00+01:00",
  "Why": "water_supply"
}
```

Figure 4.2 The collected *water_supply* object of $Data_0$

It can be seen from Figure 4.2 that the collected *water_supply* KDEs are described as follows:

- *Who* – is represented by a GLN that identifies a specific water supplier ((GLN)9104000219900).
- *What* – encompasses various attributes pertaining to the water sourced. These attributes include the NFC tag's UID, represented by a GTIN of the water sourced ((GTIN)04131F6AD3680). The *What* includes additional attributes such as the *Batch_lot* number ((Batch/lot)WS-200), the *Volume* in millilitres ((ml)10500) and the *pH* level (8) of the water sourced.
- *Where* – is represented by a GLN indicates the physical location ID ((GLN)9101101530910) of the water supplier.
- *When* – specifies the date and time (2023-05-30T14:00:00+01:00) the purified water was sourced.
- *Why* – provides information about the business process (*water_supply*).

The last set of results are derived from Figure 3.7, represented as an array of the *bottle_supply* object of $Data_0$. This is depicted in Figure 4.3.

```
{
  "Who": "(GLN)9204000219900",
  "What": {
    "GTIN": "(GTIN)042F1F6AD36480",
    "Batch_lot": "(Batch/lot)BS-300",
    "num500ml": "24",
    "num750ml": "12"
  },
  "Where": "(GLN)9201101530910",
  "When": "2023-05-30T14:00:00+01:00",
  "Why": "bottle_supply"
}
```

Figure 4.3 The collected empty *bottle_supply* object of $Data_0$

It can be seen from Figure 4.3 that the collected *bottle_supply* KDEs are described as follows:

- *Who* – is represented by a GLN that identifies a specific supplier of empty bottles ((GLN)9204000219900).
- *What* – encompasses various attributes pertaining to the production of the empty bottle, including the NFC tag's UID, represented by a GTIN of the empty bottles ((GTIN)042F1F6AD36480).
- *What* – includes additional attributes such as the *Batch_lot* number ((Batch/lot)BS-300), *num500ml* as the number of 500ml bottles (24), and *num750ml* as the number of 750ml bottles (12) from the empty bottle supplier.
- *Where* – is represented by a GLN indicates the physical location ID ((GLN)9201101530910) of the empty bottle supplier.
- *When* – specifies the date and time (2023-05-30T14:00:00+01:00) when the empty bottle was produced.
- *Why* – provides information about the business process (*bottle_supply*). The first subsystem section has described the type of collected data provenance and the corresponding results shared.

4.3 Results of data provenance storage

The main aim of the second subsystem of the experimental setup is to describe how data provenance was stored off-chain in the InterPlanetary File System (IPFS) Merkle tree Directed Acyclic Graph (DAG) and its root hash incorporated into the blockchain-based smart contract. The first set of results is from all role player sources in Section 3.3.2. The results are derived from Figure 3.8 and concatenated into a single JavaScript Object Notation (JSON) markup format, stored in the IPFS for off-chain storage.

The results of a generated IPFS Merkle tree DAG node *Hash₀* (hash0) are depicted in Figure 4.4. These results are the terminal output from a Python script as *Hashn* (Datan['Cid']['/']) which is semantic information Content Identifier (CID). The contents of node *Hash₀* (hash0) are JSON markup format arrays of an end-user's *new_order*, *water_supply* and *bottle_supply* as Datan objects derived from Figure 3.9.

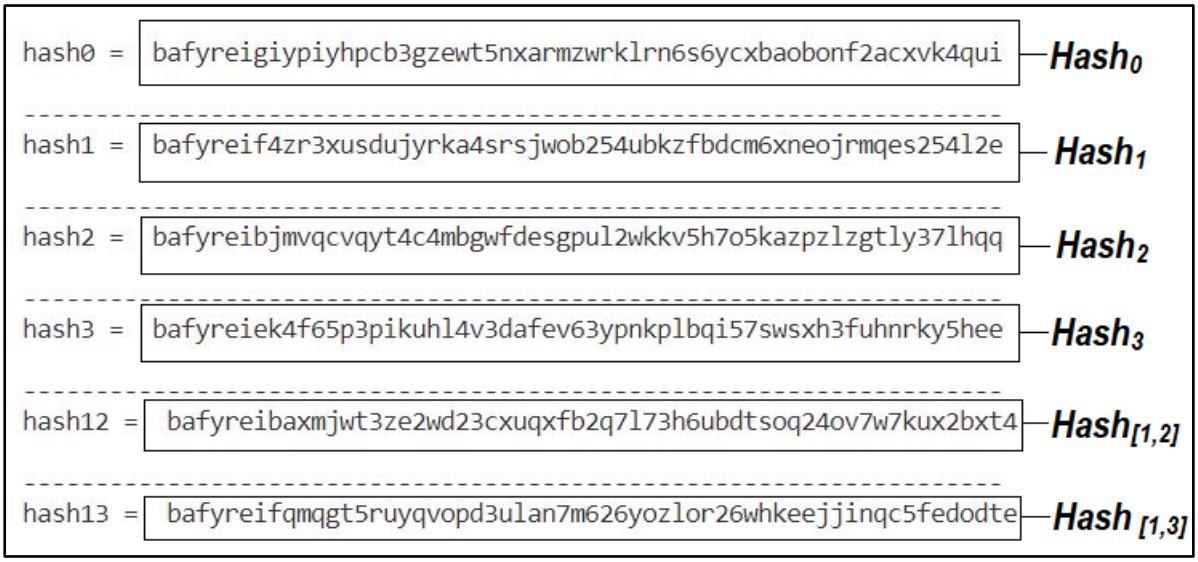


Figure 4.4 The results of the data provenance of stored IPFS Merkle tree DAG *Hashn*

This is illustrated with a scenario of an end-user (Customer A) placing an order, as provided in Appendix A, and written in the Near Field Communication (NFC) tag. The end-user's *new_order* as node *Hash₁* (hash1) is filled with water sourced from *water_supply* and empty bottles from *bottle_supply*, both of which are stored as JSON object *Data₀*. As seen in Figure 4.4, the *water_supply* entry *Data₀* generated node *Hash₂* (hash2), while the *bottle_supply* entry *Data₀* as node *Hash₃* (hash3). These hashes are the child nodes of IPFS Merkle tree DAG for data provenance storage implemented with Smart Manufacturing Unit (SMU) tasked with filling the water bottles.

The next set of results are links from the hashes of child nodes derived from Figure 3.10. This is depicted in Figure 4.5.


```
{
  "Who": "(GLN)9604000219900",
  "What": {
    "GTIN": "(GTIN)040A16AD36480",
    "hash12": {
      "hash1": "bafyreif4zr3xusdujyrka4srsjwob254ubkzfbdcm6xneojrmqes254l2e",
      "hash2": "bafyreibjmvqcvqyt4c4mbgwfdesgpul2wkkv5h7o5kazpzlzgtly37lhqq"
    },
    "Volume": "(ml)10500",
    "pH": "8"
  },
  "Where": "(GLN)9601101530910",
  "When": "2023-06-14T10:25:00+01:00",
  "Why": "water_supply"
}
```


Figure 4.5 Data provenance stored as hash12

It can be seen from Figure 4.5 that the collected *bottle_supply* object of $Data_0$ Key Data Elements (KDEs) are described as follows:

- *Who* – as a Global Location Number (GLN) identifies the water bottling plant ((GLN)9604000219900).
- *What* – encompasses various attributes pertaining to the *pH* level of the measured water in the tank derived from Figure 3.11 stored in the *ipfsHash* node $Hash_{[1,2]}$ (hash12). These attributes include the NFC tag's unique identifier (UID), represented by a Global Trade Item Number (GTIN) corresponding to the end-user's *new_order* ((GTIN)040A16AD36480) for traceability.
- *What* – encompasses additional attributes, such as the links from the parent node $Hash_{[1,2]}$ (hash2), verified, with two child links pointing to node $Hash_1$ (hash1) and another to $Hash_2$ (hash2). The *Volume* in millilitres ((ml)10500) and the measured water *pH* level (8). The water *Volume* ((ml)10500) validates the combined filled number of 500ml bottles (12) and the number of 750ml bottles (6) for traceability.
- *Where* – is represented by a Global Location Number (GLN) validates the physical location ID ((GLN)9601101530910) of the water bottling plant.
- *When* – validates the date and time (2023-06-14T10:25:00+01:00) when the *pH* level (8) was measured during the water filling process.
- *Why* – provides information about the business process (*water_supply*).

The next set of results are derived from Figure 3.12 and presented as IPFS Merkle tree DAG node $Hash_{[1,3]}$ (hash13). This is depicted in Figure 4.6.


```
{
  "Who": "(GLN)9604000219900",
  "What": {
    "GTIN": "(GTIN)040A16AD36480",
    "hash13": {
      "hash12": "bafyreibaxmjwt3ze2wd23cxuqxfb2q7l73h6ubdtsoq24ov7w7kux2bxt4",
      "hash3": "bafyreiek4f65p3pikuhl4v3dafev63ypnkplbqi57swsxh3fuhnrky5hee"
    },
    "num500ml": "12",
    "num750ml": "6"
  },
  "Where": "(GLN)9601101530910",
  "When": "2023-06-14T10:30:00+01:00",
  "Why": "water_filling"
}
```


Figure 4.6 Data provenance stored as hash13

It can be seen from Figure 4.6 that the collected *bottle_supply* object of $Data_0$ KDEs are described as follows:

- *Who* – identifies the water bottling plant ((GLN)9604000219900). The KDE *What* includes the NFC tag's UID, represented by a GTIN corresponding to the end-user's *new_order* ((GTIN)040A16AD36480) for traceability.
- *What* – encompasses additional attributes, such as the links from the parent $Hash_{[1,3]}$ (hash13) verified, with two child links pointing to node $Hash_{[1,2]}$ (hash12) and another to node $Hash_3$ (hash3). The *num500ml* validates the counted number of 500ml bottles (12) and *num750ml* the number of 750ml bottles (6) by the proximity sensor in Figure 3.12 for traceability.
- *Where* – validates physical location identification ((GLN)9601101530910) of the water bottling plant.
- *When* – validates the date and time (2023-06-14T10:30:00+01:00) of the bottles counted during the water filling process.
- *Why* – provides information about the business process (*water_filling*).

The next set of results are derived from Figure 3.13 and presented as emitted event log of blockchain-based smart contract deployed to *Polygon*. This is depicted in Figure 4.7.

Figure 4.7 The result of an immutable emitted event of stored CTE_1

It can be seen from Figure 4.7 that the blockchain-based smart contract is assigned with arbitrary name *BottledWaterProvenance.sol* (A). The function *MakeToOrderStage* (B) incorporates the *DataProvenanceChain* data provenance chain mapping hash table with the two arguments as *ProvenanceIndex* index and a string IPFS Merkle tree DAG node Hashn (hashn) as *CTEn*. The *ProvenanceLog* (C) event log is then emitted with *ProvenanceIndex* (D) and *ipfsHash* node $Hash_{[1,3]}$ (hash13) as CTE1 (E). Finally, the event log *ProvenanceIndex* (D) is incremented (1).

Once the blockchain-based smart contract is deployed, its generated content-addressed transaction hash is written in the Near Field Communication (NFC) tag as a Uniform Resource Locator (URL) link. A URL link directs the end-user to the deployed blockchain-based smart contract content-addressed transaction hash on the *Polygon* public blockchain platform. This is depicted in Figure 4.8.

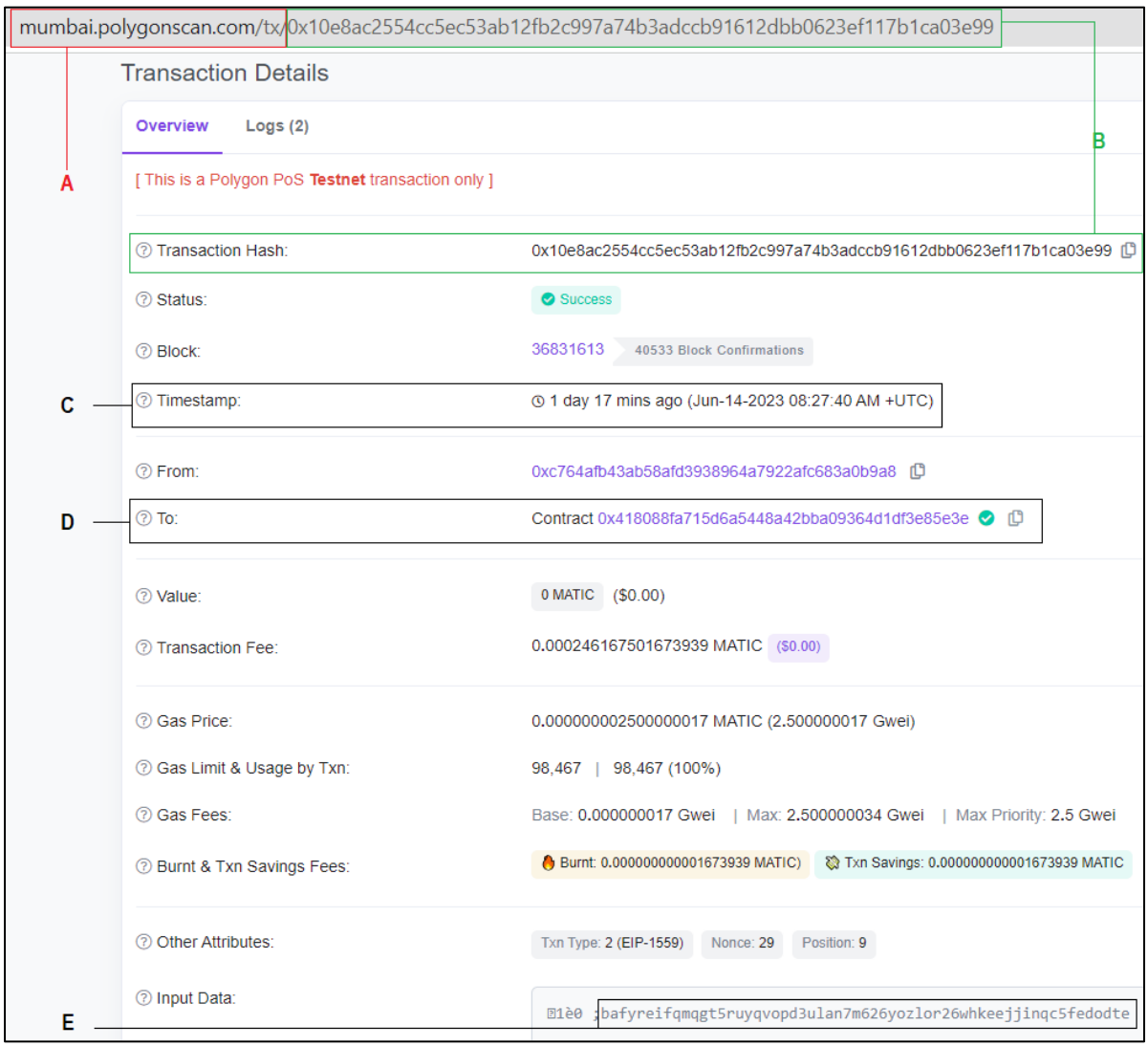


Figure 4.8 The result of an immutable, traceable and transparent node CTE_1

As depicted in Figure 4.8 the end-user can access critical data pertaining to the history and origin of the contents, composition, quantity and *TimeStamp* (C) of water bottle filling. Therefore, the challenge of ensuring data provenance in a water bottling plant is overcome by incorporating bottled water data pertaining to the end-user's *new_order* (E) hash CTE_1 into a blockchain-based smart contract (D), which is secure, immutable, traceable, and transparent. The second subsystem section described the results of data provenance stored.

4.4 Results of data provenance tracking

The main aim of the third subsystem of the experimental setup is to describe tests and results of a data provenance tracking application. This is described in Section 3.3.3. To track data provenance, the end-user can read data provenance by tapping a Near Field Communication (NFC) tag attached to the bottled water. The first set of results will correspond to the water filler Critical Tracking Event CTE_1 from *Polygon* described in Figure 4.8. However, an anomaly was introduced in the end-user's *new_order* with a scenario of one of the bottles not being filled with water. The results of this anomaly are depicted in Figure 4.9.

Figure 4.9 The results of the data provenance IPFS hash of stored anomaly

It can be seen in Figure 4.9 that the anomaly affected *ipfsHash* InterPlanetary File System (IPFS) Merkle tree Directed Acyclic Graph (DAG) node $Hash_2$ (hash2), $Hash_{[1,2]}$ (hash12) and $Hash_{[1,3]}$ (hash13). This is in relation to the initially stored version in Figure 4.4. The veracity and robustness testing of the developed blockchain-based smart contract was done on Remix Integrated Development Environment (IDE).

A top-down data provenance tracking approach was used to validate and demonstrate the anomaly based on the immutability of the hash CTE_1 associated with node $Hash_{[1,3]}$ (hash13) from *Polygon*. The mapping hash table indexing enhances the ability to traverse the IPFS Merkle hash tree DAG to query semantic information to extract data provenance traceability that is secure, immutable and traceable. The result is depicted in Figure 4.10.

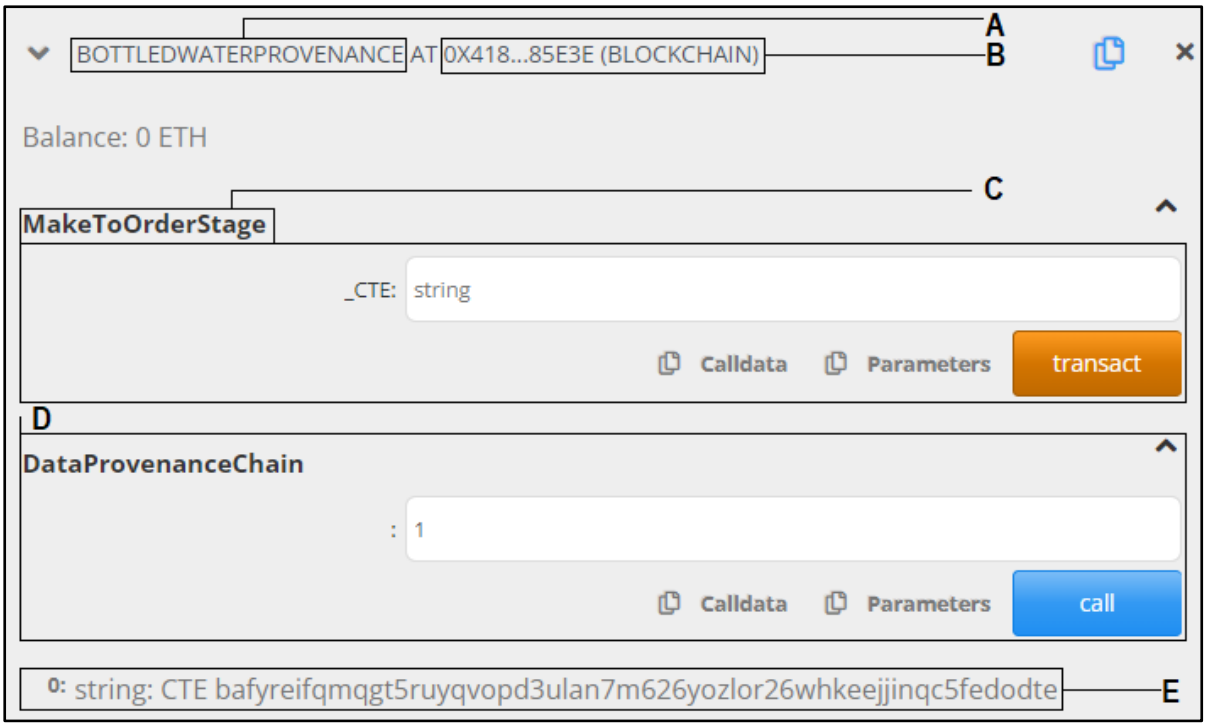


Figure 4.10 The result of *ProvenanceIndex* with public uint256 value set to 1 and hash13

As depicted in Figure 4.10, the blockchain-based smart contract with arbitrary name *BottledWaterProvenance.sol* (A) is developed and assigned an address (B). The function *MakeToOrderStage* (C) incorporates the Critical Tracking Event *ipfsHash*. This is in addition to *DataProvenanceChain* mapping hash table with the two arguments. The first argument as *ProvenanceIndex* index (D) uint256 value set to 1 and the other as a string IPFS Merkle tree DAG node $Hash_{[1,3]}$ (hash13) as CTE_1 (E).

The parent node $Hash_{[1,3]}$ (hash13) is utilised to trace and retrieve the contents history and origin of a stored data provenance JSON markup format from IPFS Merkle tree DAG as a Concise Binary Object Representation (CBOR). This is depicted in Figure 4.11. The results will correspond to a traceable data provenance in Figure 4.6 as node $Hash_{[1,3]}$ (hash13). However, the anomaly affected node $Hash_{[1,3]}$ (hash13). This is indicated as Content Identifier (CID).

Figure 4.11 The result of a traceable and transparent data provenance for hash13

It can be seen in Figure 4.11 that the anomaly also affected node $Hash_{[1,2]}$ (hash12). Therefore, the next set of results represent node $Hash_{[1,2]}$ (hash12). This is depicted in Figure 4.12. The result confirms the anomaly as it is different from the data provenance stored as $Hash_{[1,2]}$ (hash12) depicted in Figure 4.5.

Figure 4.12 The result of a traceable and transparent data provenance for hash12

It can be seen from Figure 4.12 that the collected *bottle_supply* object of $Data_0$ KDEs are described as follows:

- *Who* – a GLN identifies the water bottling plant ((GLN)9604000219900).
- *What* – encompasses various attributes pertaining to the NFC tag's UID, represented by a GTIN corresponding to the end-user's *new_order* ((GTIN)040A16AD36480) for traceability. The *What* encompasses additional attributes, such as the links from the parent node $Hash_{[1,2]}$ (hash2), verified, with two child links pointing to node $Hash_1$ (hash1) and another to $Hash_2$ (hash2). The *Volume* in millilitres ((ml)10000) and the measured water *pH* level (7).
- *Where* – is represented by a GLN validates the physical location ID ((GLN)9601101530910) of the water bottling plant.
- *When* – validates the date and time (2023-06-14T10:25:00+01:00) the *pH* level (7) was measured during the water filling process.
- *Why* – provides information about the business process (*water_supply*).

The next set of results for IPFS Merkle tree DAG node $Hash_1$ (hash1) is represented as an array *new_order* object of $Data_0$. This is depicted in Figure 4.13 and the result corresponds to a traceable collected *new_order* object of $Data_0$ depicted in Figure 4.1.

Figure 4.13 The result of a traceable and transparent data provenance for hash1

The next set of results for IPFS Merkle tree DAG *node Hash*$_2$ (hash2) is represented as an array *water_supply* object of *Data*$_0$. This is depicted in Figure 4.14. The result confirms the anomaly as it is different from the data provenance stored as hash12 depicted in Figure 4.5.

Figure 4.14 The result of a traceable and transparent data provenance for hash2

It can be seen from Figure 4.14 that the collected *water_supply* objects of *Data*$_1$ KDE are described as follows:

- *Who* – is represented by a GLN that identifies a specific water supplier ((GLN)9104000219901).
- *What* – encompasses various attributes pertaining to the water sourced. These attributes include the NFC tag's UID, represented by a GTIN of the water sourced ((GTIN)04131F6AD3681). The *What* includes additional attributes such as the *Batch_lot* number ((Batch/lot)WS-200), the *Volume* in millilitres ((ml)10000) and the *pH* level (7) of the water sourced. The water *Volume* ((ml)10000) validates the anomaly as it is not enough to complete the combined filled number of 500ml bottles (12) and the number of 750ml bottles (6). This results with one 500ml bottle not filled.
- *Where* – is represented by a GLN indicates the physical location ID ((GLN)9101101530910) of the water supplier.
- *When* – specifies the date and time (2023-05-30T14:00:00+01:00) the purified water was sourced.
- *Why* – provides information about the business process (*water_supply*).

The next set of results for IPFS Merkle tree DAG *node Hash₃* (hash3) is represented as an array *bottle_supply* object of *Data₀*. This is depicted in Figure 4.15 and the result corresponds to a traceable Figure 4.3 data provenance.

Figure 4.15 The result of a traceable and transparent data provenance for hash3

After retrieving the contents of *new_order*, *water_supply* and *bottle_supply*, the hash *CTE₀* associated with IPFS Merkle tree DAG node *Hash₀* (hash0) from *Polygon* public blockchain platform can be verified. This is depicted in Figure 4.16, and verified utilising a *DataProvenanceChain* mapping hash table *ProvenanceIndex* index uint256 value set to 0 (A) and corresponds to a traceable hash *CTE₀* (B).

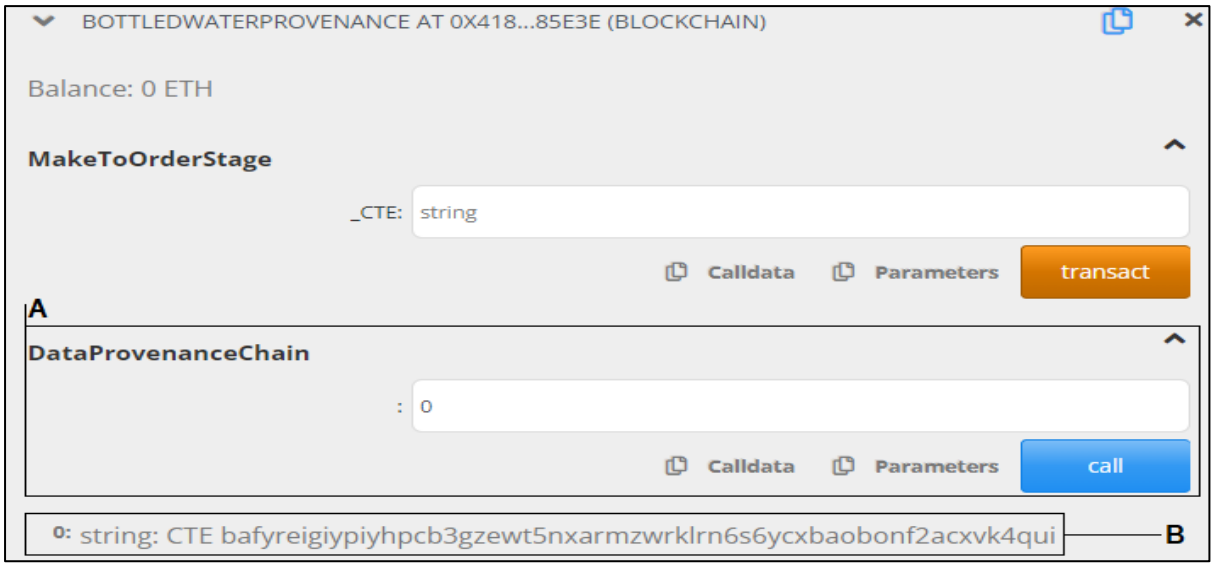


Figure 4.16 The result of *ProvenanceIndex* with public uint256 value set to 0 and hash0

The result of hash CTE_0 associated with node $Hash_0$ (hash0) were utilised for processing an end-user's *new_order*. This is depicted in Figure 4.17. The provenance data from all sources collected are concatenated into a single JSON markup format. The results of collected data provenance include the arrays, the end-user's *new_order* (10), *water_supply* (5), and *bottle_supply* (3) objects of $Data_n$.

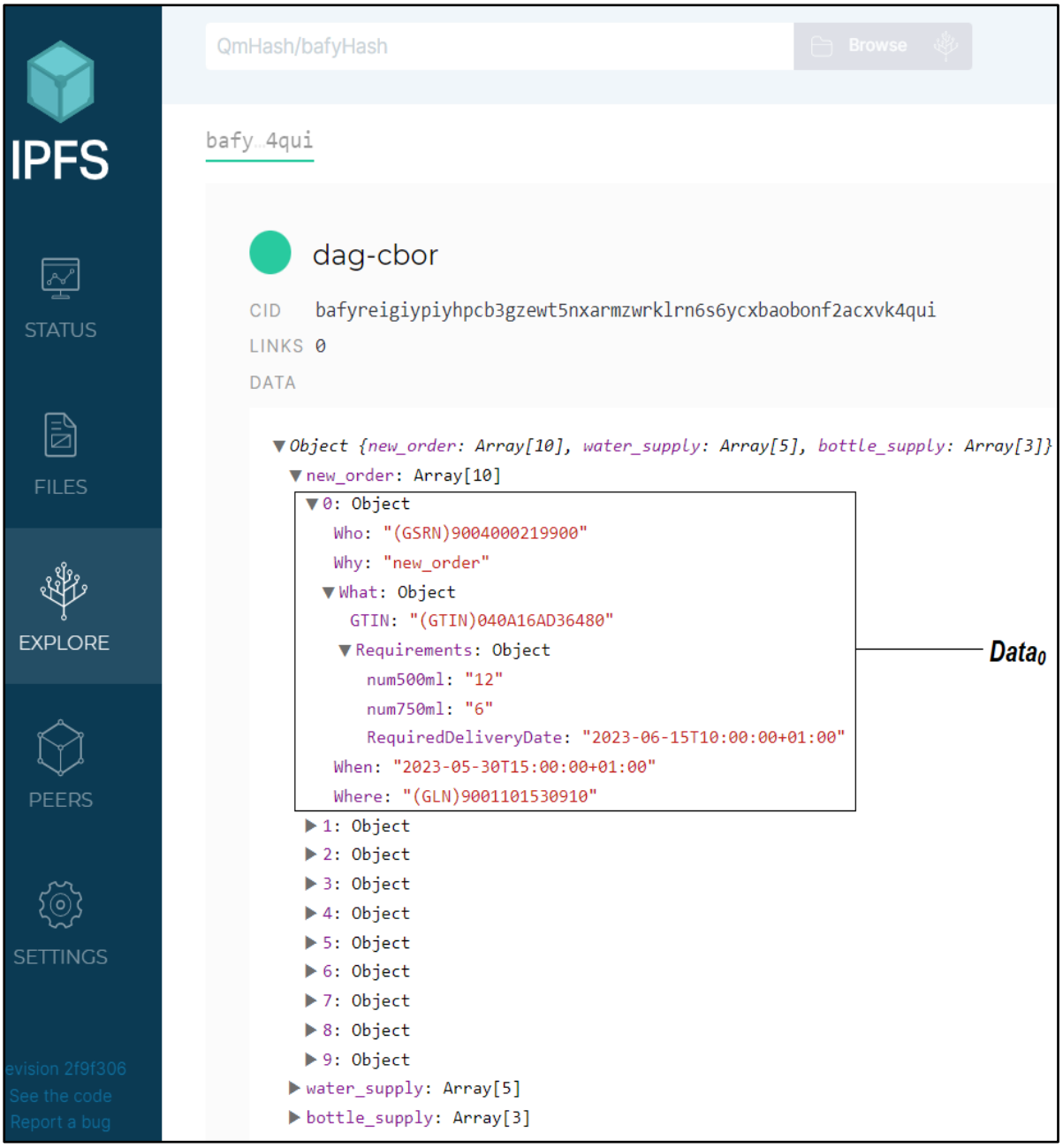


Figure 4.17 The result of an immutable, traceable and transparent node CTE_0

As depicted in Figure 4.17, IPFS Merkle tree DAG $Hash_0$ (hash0) is utilised to further enable the end-user to access critical data pertaining to the history and origin of the contents, composition, quantity and timestamp of water bottle filling as follows:

- The entry $Data_0$ of the *new_order* array is traceable and corresponds to Figure 4.1.
- Similarly, the entry $Data_0$ of the *water_supply* array is traceable and corresponds to Figure 4.2 .
- Lastly, the entry $Data_0$ of the *bottle_supply* array is traceable and corresponds to Figure 4.3.

The final subsystem section has described the tests and results of incorporating product data pertaining to the end-user's new order into a public blockchain-based smart contract, making the data secure, immutable and traceable. This allows the end-user to "read" data provenance through a tag attached on the bottled water for traceability and transparency. The findings of this research study can also contribute to the existing body of knowledge regarding data provenance traceability. This contribution has the potential to significantly reduce the scale of product recalls, from millions of units to only a few hundred, further highlighting the importance of this research study.

Chapter 5: Research Contributions

5.1 Introduction

The aim of this chapter is to discuss the results obtained from the tests carried out in the previous chapter. The contribution of this research study involves the development of an experimental setup for incorporating data provenance into blockchain smart contracts in a smart manufacturing environment process. Firstly, this experimental setup ensures that data provenance pertaining to the end-user order, raw materials used, and timestamp at each stage of production is incorporated into a blockchain smart contract to make it immutable and traceable. Secondly, the end-user can read data provenance by tapping a Near Field Communication (NFC) enabled device on an NFC tag attached to the bottled water. This NFC tag contains a content-addressed transaction hash Uniform Resource Locator (URL) link to a blockchain-based smart contract with critical data provenance. Lastly, this experimental setup eliminates the need for end-users to possess knowledge of cryptography wallet setup and configurations.

5.2 Comparison of the developed solution with existing solutions

The current challenges that necessitated this study, stem from the limited research on how end-users can access critical data pertaining to the bottled water contents, composition, quantity and timestamp at time of packaging a product within a smart manufacturing environment. As discussed in Section 2.5, centralisation has been identified as a major challenge in achieving data provenance traceability across an end-to-end supply chain.

As seen in Table 5.1**,** the developed solution is decentralised compared with most other existing solutions, and even against its main competitor by Cao et al. [122] and Kuhn et al. [123] in the manufacturing supply chain which are permissioned, based on the literature review done by Dasaklis et al. [13].

The developed solution was implemented on a public blockchain platform. As previously mentioned in terms of performance, the efficiency of public blockchain platforms like *Ethereum* (legacy) developed by Alkhader et al. [62] is low, which affects the cost of adoption.

Table 5.1 Comparison of the developed solution with existing solutions

	Alkhader et al. [62] (Medical supplies)	Alkhader et al. [124] (Agnostic)	Cao et al. [122] (Steel)	Kuhn et al. [123] (Automotive)	Subashini et al. [105] (Agri-food)	Developed solution (Bottled water)
Decentralisation	Yes	Yes	No	No	No	Yes
Performance	No	No	N/A	N/A	Yes	Yes
Layer 2 network	No	No	No	No	Yes	Yes
IPFS	Yes	Yes	No	No	Yes	Yes
GS1 standardisation	No	No	No	No	No	Yes

The aim of this research study is to identify and implement a cost-effective and suitable blockchain platform which will be utilised to develop a blockchain-based smart contract data provenance tracking application. To fulfil the aim of this research, firstly, the developed solution selection of blockchain platform for storing data provenance into smart contracts was influenced based on the key factors determined [54].

Secondly, the developed solution utilised the identified [99] cost-effective and suitable novel layer 2 *Polygon* public blockchain platform solution as compared with most other existing solutions by Alkhader et al. [62][124] based on *Ethereum* (legacy). Lastly, a search strategy was conducted to find current relevant papers for the period of 2022 to 2023.

A predefined set of terms for searching within the paper titles utilised, included the following keywords, “provenance” and “Polygon”. The search resulted with the main competitor solution by Subashini et al. [105], with tests conducted on *Ethereum* (legacy), and *Polygon*. However, the solution was developed as permissioned.

The test results reinforced the developed solution as cost-effective public blockchain network platform. The developed solution and the searched main competitor solution by Subashini et al. [105] further utilised hybrid approaches such as InterPlanetary File System (IPFS) that enable decentralised storage to reduce expensive on-chain data storage.

The developed solution and the searched main competitor solution can be considered as health and safety-sensitive sectors that necessitate a standardised approach handled by the Electronic Product Code Information Service (EPCIS) implemented by GS1 standards [13]. The developed solution implemented standardisation combating the centralisation of manufacturing of bottled compared with searched main competitor by Subashini et al. [105], which is permissioned for agri-food.

Table 5.1 summarises the comparison of the developed solution with existing solutions alongside the functionality of each blockchain-based data provenance tracking implementation. The developed solution allows end-users to access secure, immutable, transparent, and traceable data provenance. Therefore, the developed solution reduce the scale of product recalls, from millions of units to only a few hundred, further highlighting the importance of this research study.

5.3 Research contributions

This research study developed a blockchain-based smart contract data provenance tracking application within a smart manufacturing environment. The following contributions from the study are considered to be novel:

5.3.1 Contributions to existing knowledge

This research studied several journal articles written in the field of data provenance traceability and, specifically, from Dasaklis et al. [13]. This article was able to capture the advances made in the field of study from 2018 to 2021 and highlighted the importance of selecting the proper blockchain platform architecture. As part of contributing to new knowledge, a journal article [54] titled "*Factors influencing the selection of a blockchain platform for incorporating data provenance into smart contracts*" was published in 2023. This article was able to highlight the importance of selecting a suitable blockchain platform for storing data provenance into smart contracts based on the key factors determined.

As stated in a comparison study in Section 2.4.5, there is limited or lack of research on combating the centralisation of manufacturing of bottled water and none when it comes to cost-effective public blockchain platforms.

As part of contributing to new knowledge, a journal article [99] titled “A comparison of transaction fees for various data types and data sizes of blockchain smart contracts on a selection of blockchain platforms” was published in 2022. This article was able to identify [99] a cost-effective and suitable public blockchain platform solution. A total of four journal articles in engineering were published or were in publication at the time of submitting this dissertation.

5.3.2 Contribution to Industry 4.0

The contribution of this research study is in the context of the Fourth Industrial Revolution, commonly referred to as Industry 4.0, specifically focusing on the smart manufacturing environment. This research study utilises a water bottling plant as a case study, which consists of three smart manufacturing units.

The first unit is designated to fill the water bottles, the second unit caps the bottles, and the third unit packs the bottles according to the end-user order. Radio Frequency Identification (RFID) tags are utilised to store data pertaining to the provenance of a specific end-user’s new order and the measured data collected by incorporating various sensors from the respective smart manufacturing units.

Subsequently, this research study contributes to the existing knowledge in the domain of Industry 4.0 by addressing data provenance traceability in the context of bottled water production within a smart manufacturing environment. The developed solution enhances the semantic information exchange in the manufacturing communication process by considering the five W’s, known as Who, What, Where, When, and Why. It enables collaborations, responsiveness, resource-driven operations, includes a physical layer, and incorporates Big Data analysis capabilities at each stage of the entire smart manufacturing environment process. This generic protocol has the potential for standardisation and provenance traceability across various fields within the Industry 4.0 domain.

Chapter 6: Conclusions

6.1 Introduction

The aim of this chapter is to discuss the reflection on the previous chapters, research goals, recommendations, and future work.

6.2 Reflection on the previous chapters

The initial chapter introduced the research study and specified the problem the research intended to solve. Thereafter, a plausible hypothesis was identified, and a list of research study objectives were specified. These objectives served as the foundation to formulate the research methodology of the study. The first chapter was concluded with an overview of the dissertation's structure and layout.

In the second chapter, a comprehensive literature review was performed. This was done by initially focusing on the broader problem of data provenance traceability, identifying the challenge described and establishing the limitations that this research study aimed to fill. The chapter also determined the key factors that influenced the selection of a blockchain platform for storing data provenance into smart contracts. The chapter finally, made a comparison of various blockchain platforms to determine the most cost-effective and suitable platform for this research study.

The third chapter delved into the research methodology utilised to develop a blockchain-based smart contract data provenance tracking application within a smart manufacturing environment to log product or raw material information pertaining to the composition and production process. In this research study, a blockchain-based smart contract data provenance tracking application was developed utilising *Polygon* public blockchain platform. This chapter illustrated how the water bottling plant, the case study in this research, was utilised to validate and demonstrate the practical application of the experimental setup.

The fourth chapter described and illustrated the results of the methodology utilised to develop a blockchain-based smart contract data provenance tracking application. This was done by firstly describing the type of collected data, followed by the analysis of the results of data provenance stored and, finally, the tests and results of the data provenance tracking.

Finally, this chapter discusses the results obtained from the tests conducted. This is done by comparing the developed solution with existing ones to evaluate its effectiveness in addressing the challenge described in the problem statement of this study. This comparison highlighted the limitation of end-users being unable to access critical data pertaining to the contents, composition, quantity, and timestamp at the time of packaging a product. To overcome this limitation, the developed solution incorporated product data pertaining to the end-user's new order into a blockchain-based smart contract, making the data secure, immutable and traceable.

6.3 Research goals

The main goal of the research study was to develop a blockchain-based smart contract data provenance tracking application within a smart manufacturing environment. However, this presented several challenges as present centralised systems store data in a single location, making it vulnerable to single points of failure, data tampering, and unauthorised access. This research study thus aimed to address the challenge posed by the problem of data provenance in the existing centralised systems.

Therefore, to address this unique issue, an initial step was to conduct a comprehensive review of the literature, with a specific focus on data provenance traceability. Secondly, the key factors that influenced the selection of a blockchain platform for storing data provenance into smart contracts needed to be determined. These different types of blockchain platforms needed to be compared to identify a cost-effective and suitable platform.

The literature review was conducted in Chapter 2. It focused initially on provenance and its classification. The chapter then went on to discuss the problems associated with cloud-based data provenance traceability. In Section 2.3.2, the problem of centralisation, which is the focus of this study in achieving data provenance traceability across an end-to-end supply chain, was detailed. This chapter also gave a comprehensive discussion into blockchain-based data provenance traceability in a manufacturing supply chain and the limitations of existing research. Finally, the limitations of the existing research reinforced the need for the study as stated in the problem statement.

As mentioned previously, a case study was selected to validate and demonstrate the experimental setup of the blockchain-based smart contract data provenance tracking application. The data provenance needed to be collected, stored and tracked. This was accomplished in Chapter 3, where a case study of a water bottling plant was chosen.

The bottling plant produces more than one variant (500ml and 750ml) of the same product. Section 3.3 details how the proposed experimental setup for blockchain-based smart contract data provenance tracking application was written in Python development platform and tested on Remix Integrated Development Environment (IDE).

Firstly, the inputted data provenance to the experimental setup needed to be collected from three input sources which included the end-user's new order, water supplier, and empty bottle supplier. The collected data provenance was provided by utilisation of Near Field Communication (NFC) tags which are read by NFC antennas mounted at designated points of the water bottling plant. This was discussed in Section 3.3.1.

Secondly, the data provenance needed to be stored off-chain in the InterPlanetary File System (IPFS) Merkle tree Directed Acyclic Graph (DAG) with its root hash incorporated into the blockchain-based smart contract. The blockchain-based smart contract was developed in the *Solidity* programming language compiled utilising a Python script. This was further discussed in Section 3.3.2. This chapter was rounded off by ensuring that the end-user can access and track critical data pertaining to the history and origin of contents, composition, quantity and timestamp of water bottle fillings. This was discussed in Section 3.3.3.

The results of the research study then needed to be collated in a structured manner. This was done in Chapter 4. Firstly, the type of collected data provenance was described and the corresponding results shared. This was discussed in Section 4.2. Next, the analysis of the results of data provenance stored in the IPFS Merkle tree DAG and blockchain-based smart contract was done. This was discussed in Section 4.3.

The veracity and robustness testing of the developed blockchain-based smart contract was done on Remix IDE. The results were able to bring to light the obstacles described in the problem statement and reinforce the importance of ensuring data provenance traceability in a water bottling plant.

The chapter was rounded off by describing the tests and results of the data provenance tracking. This was done by validating and demonstrating the experimental setup utilising the root IPFS Merkle tree DAG which represents hash of a Critical Tracking Event (CTE). Here, the three input sources, including arrays of ten end-users' new orders, five water sources, and three empty bottle suppliers were provided to the experimental setup.

The inputted arrays utilised GS1 standard's five W's to collate semantic information on the Key Data Elements (KDEs) related to end-users' new orders for bottled water, including Who, What, Where, When, and Why. This was done to facilitate the tracking and querying of data provenance associated with the end-user's new order.

This data provenance tracking approach was done by mapping hash table indexing to query a public blockchain-based smart contract to extract Content Identifier (CID) hash. This CID hash enables the role players to traverse the IPFS Merkle tree DAG to read the required version that is secure, immutable and traceable. Merkle DAG nodes are immutable. Any change in a node would alter its identifier and thus affect all the ascendants in the DAG, essentially creating a different DAG.

This ensured that the end-users could access critical data pertaining to the contents, composition, quantity, and timestamp of water bottle fillings. The developed blockchain-based smart contract data provenance tracking application can be scaled to packaging a product and end-to-end supply chain traceability. This was done to reduce the scale of product recalls, from millions of units to only a few hundred, further highlighting the importance of this research study. This was done in Chapter 5.

6.4 Recommendations and future work

This dissertation is part of an ongoing research project conducted at the Central University of Technology (CUT), Free State, focusing on the analysis of different data provenance traceability issues and the utilisation of blockchain-based smart contracts to address centralisation, ensuring data is secure, immutable, traceable, and transparent.

The conclusions derived from this specific research study lay the groundwork for various future research opportunities. This research specifically targeted the centralisation of manufacturing processes within the bottled water industry and explored cost-effective public blockchain platforms – an area that had received limited research attention in the past. However, there are several aspects that require further investigation.

As alluded earlier in the research methodology, certain functions of the water bottling plant, such as capping, labelling, and packaging, have been omitted from the experimental setup for convenience. As of the completion of this dissertation, the experimental setup validated and demonstrated the results pertaining to the water filling process to reduce transaction fees associated with storing large amounts of data provenance. Subsequently, the granularity of the blockchain-based smart contract data provenance was limited to Critical Tracking Events (CTEs).

In addition, a journal article [15] titled "Exploring the means and benefits of including blockchain smart contracts to a smart manufacturing environment: Water bottling plant case study" was published in 2022 as a contribution to new knowledge. The article presented a proposed scaled-up experimental setup for further exploration.

Furthermore, in a Make-to-Order (MTO) smart manufacturing plant at CUT, the semantic information can be utilised to autonomously procure raw materials, machine service parts, and generate blockchain-based smart contract events for machine maintenance, thereby eliminating the need for third-party deliveries. However, the scalability of storing large amounts of data provenance may present challenges in this expansion.

Hence, future research studies may need to focus on examining a fine-grained data provenance traceability solution utilising a public blockchain-based smart contract to enable efficient querying of semantic information. The findings of this research study can also contribute to the existing body of knowledge regarding data provenance traceability. This contribution has the potential to significantly reduce the scale of product recalls, from millions of units to only a few hundred, further highlighting the importance of this research study.

6.5 Scientific outcomes

- Mokalusi, O.L., Kuriakose, R.B., Vermaak, H.J. (2022). Exploring the Means and Benefits of Including Blockchain Smart Contracts to a Smart manufacturing environment: Water Bottling Plant Case Study. In: Nagar, A.K., Jat, D.S., Marín-Raventós, G., Mishra, D.K. (eds) Intelligent Sustainable Systems. Lecture Notes in Networks and Systems, vol 334. Springer, Singapore. https://doi.org/10.1007/978-981-16-6369-7_27
- Mokalusi, O.L., Kuriakose, R.B., Vermaak, H.J. (2023). Factors Influencing the Selection of a Blockchain Platform for Incorporating Data Provenance into Smart Contracts. In: Yang, XS., Sherratt, S., Dey, N., Joshi, A. (eds) Proceedings of Seventh International Congress on Information and Communication Technology. Lecture Notes in Networks and Systems, vol 464. Springer, Singapore. https://doi.org/10.1007/978-981-19-2394-4_47
- Mokalusi, O.L., Kuriakose, R.B., Vermaak, H.J. A comparison of transaction fees for various data types and data sizes of blockchain smart contracts on a selection of blockchain platforms. 7th Edition of ICT4SD International ICT summit & Awards 2022. Springer, Goa, India LNNS. ISSN: 2367-3370, Series: https://www.springer.com/series/15179
- Mokalusi, O.L., Kuriakose, R.B., Vermaak, H.J. Designing an experimental setup for incorporating data provenance into blockchain smart contracts in a Smart Manufacturing Environment: Water bottling plant case study. 5th Edition of the “Future Smart City” International Conference. Springer, Malaysia (Accepted and under publication).

References

[1] O. Prokopenko, G. Prause, V. Otenko, M. Cherkashyna, I. Kara, and I. Imnadze, "Adaptation of logistics companies to operation under the Covid-19 pandemic restrictions," *Acta Logistica (AL)*, vol. 10, no. 1, 2023, doi: 10.22306/al.v10i1.349.

[2] F. Dietrich, L. Louw, and D. Palm, *A Systematic Literature Review Of Blockchain-Based Traceability Solutions*. Gottfried Wilhelm Leibniz Universität Hannover, 2023.

[3] P. Azevedo, J. Gomes, and M. Romão, "Supply chain traceability using blockchain," *Operations Management Research*, pp. 1–23, 2023.

[4] G. S. Ramachandran *et al.*, "Blockchain in Supply Chain: Opportunities and Design Considerations," in *Handbook on Blockchain*, Cham: Springer International Publishing, 2022, pp. 541–576.

[5] O. Isaac Abiodun, M. Alawida, A. Esther Omolara, and A. Alabdulatif, "Data provenance for cloud forensic investigations, security, challenges, solutions and future perspectives: A survey," *Journal of King Saud University - Computer and Information Sciences*, Oct. 2022, doi: 10.1016/J.JKSUCI.2022.10.018.

[6] S. Jabbar, H. Lloyd, M. Hammoudeh, B. Adebisi, and U. Raza, "Blockchain-enabled supply chain: analysis, challenges, and future directions," *Multimed Syst*, vol. 27, no. 4, pp. 787–806, Aug. 2021, doi: 10.1007/S00530-020-00687-0/FIGURES/9.

[7] S. Chowdhury, O. Rodriguez-Espindola, P. Dey, and P. Budhwar, "Blockchain technology adoption for managing risks in operations and supply chain management: evidence from the UK," *Ann Oper Res*, vol. 327, no. 1, pp. 539–574, 2023, doi: 10.1007/s10479-021-04487-1.

[8] N. G. Kuftinova, A. V Ostroukh, O. I. Maksimychev, C. B. Pronin, and A. K. Yadav, "Implementation of the Data Fabric Architecture as a Sustainable Development of Industrial Platform Technologies in Road Transport Systems," in *2023 Systems of Signals Generating and Processing in the Field of on Board Communications*, 2023, pp. 1–5. doi: 10.1109/IEEECONF56737.2023.10092048.

[9] D. Batista *et al.*, "Exploring Blockchain Technology for Chain of Custody Control in Physical Evidence: A Systematic Literature Review," *Journal of Risk and Financial Management*, vol. 16, no. 8, 2023, doi: 10.3390/jrfm16080360.

[10] W. A. H. Ahmed and B. L. MacCarthy, "Blockchain-enabled supply chain traceability – How wide? How deep?," *Int J Prod Econ*, vol. 263, p. 108963, 2023, doi: https://doi.org/10.1016/j.ijpe.2023.108963.

[11] A. Shadravan and H. R. Parsaei, "Impacts of industry 4.0 on smart manufacturing," in *Proceedings of the 13th International Conference on Industrial Engineering and Operations Management (IEOM Society)*, pp. 7–9.

[12] C. D. K. Tchatchouang *et al.*, "Listeriosis outbreak in south africa: A comparative analysis with previously reported cases worldwide," *Microorganisms*, vol. 8, no. 1. MDPI AG, Jan. 01, 2020. doi: 10.3390/microorganisms8010135.

[13] T. K. Dasaklis, T. G. Voutsinas, G. T. Tsoulfas, and F. Casino, "A Systematic Literature Review of Blockchain-Enabled Supply Chain Traceability Implementations," *Sustainability 2022, Vol. 14, Page 2439*, vol. 14, no. 4, p. 2439, Feb. 2022, doi: 10.3390/SU14042439.

[14] R. Saxena, E. Gayathri, and L. Surya Kumari, "Semantic analysis of blockchain intelligence with proposed agenda for future issues," *International Journal of System Assurance Engineering and Management*, vol. 14, no. 1, pp. 34–54, 2023, doi: 10.1007/s13198-023-01862-y.

[15] O. L. Mokalusi, R. B. Kuriakose, and H. J. Vermaak, "Exploring the means and benefits of including blockchain smart contracts to a smart manufacturing environment: water bottling plant case study," in *Intelligent Sustainable Systems: Selected Papers of WorldS4 2021, Volume 2*, Springer, 2021, pp. 309–319.

[16] S. N. Khan, F. Loukil, C. Ghedira-Guegan, · Elhadj Benkhelifa, and · Anoud Bani-Hani, "Blockchain smart contracts: Applications, challenges, and future trends," 2021, doi: 10.1007/s12083-021-01127-0.

[17] L. A. Risso, G. M. D. Ganga, M. Godinho Filho, L. A. de Santa-Eulalia, T. Chikhi, and E. Mosconi, "Present and future perspectives of blockchain in supply chain management: a review of reviews and research agenda," *Comput Ind Eng*, vol. 179, p. 109195, 2023, doi: https://doi.org/10.1016/j.cie.2023.109195.

[18] A. Hassoun *et al.*, "Implementation of relevant fourth industrial revolution innovations across the supply chain of fruits and vegetables: A short update on Traceability 4.0," *Food Chem*, vol. 409, p. 135303, 2023, doi: https://doi.org/10.1016/j.foodchem.2022.135303.

[19] Y. Chen, Y. Lu, L. Bulysheva, and M. Yu. Kataev, "Applications of Blockchain in Industry 4.0: a Review," *Information Systems Frontiers*, 2022, doi: 10.1007/s10796-022-10248-7.

[20] F. F. Rad *et al.*, "Industry 4.0 and supply chain performance: A systematic literature review of the benefits, challenges, and critical success factors of 11 core technologies," *Industrial Marketing Management*, vol. 105, pp. 268–293, 2022, doi: https://doi.org/10.1016/j.indmarman.2022.06.009.

[21] H. Treiblmaier and M. Garaus, "Using blockchain to signal quality in the food supply chain: The impact on consumer purchase intentions and the moderating effect of brand familiarity," *Int J Inf Manage*, vol. 68, p. 102514, 2023, doi: https://doi.org/10.1016/j.ijinfomgt.2022.102514.

[22] Y. L. Simmhan, B. Plale, and D. Gannon, "A Survey of Data Provenance in e-Science", doi: 10.1145/1084805.1084812.

[23] B. Pérez, J. Rubio, and C. Sáenz-Adán, "A systematic review of provenance systems," *Knowl Inf Syst*, vol. 57, no. 3, pp. 495–543, Dec. 2018, doi: 10.1007/S10115-018-1164-3.

[24] J. Freire, D. Koop, E. Santos, and C. T. Silva, "Provenance for computational tasks: A survey," *Comput Sci Eng*, vol. 10, no. 3, pp. 11–21, May 2008, doi: 10.1109/MCSE.2008.79.

[25] S. B. Davidson and J. Freire, "Provenance and scientific workflows: Challenges and opportunities," *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 1345–1350, 2008, doi: 10.1145/1376616.1376772.

[26] S. M. S. da Cruz, M. L. M. Campos, and M. Mattoso, "Towards a taxonomy of provenance in Scientific Workflow Management Systems," *SERVICES 2009 - 5th 2009 World Congress on Services*, no. PART 1, pp. 259–266, 2009, doi: 10.1109/SERVICES-I.2009.18.

[27] B.; Glavic and K. R. Dittrich, “Data provenance: A Cctegorization of existing approaches,” pp. 227–241, 2007, doi: 10.5167/uzh-24450.

[28] P. Buneman and S. B. Davidson, “Data provenance-the foundation of data quality,” 2010.

[29] J. Cheney, L. Chiticariu, and W.-C. Tan, “Provenance in Databases: Why, How, and Where,” *Foundations and Trends R in Databases*, vol. 1, no. 4, 2009, doi: 10.1561/1900000006.

[30] P. Buneman, S. Khanna, and W. C. Tan, “Why and where: A characterization of data provenance?,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 1973, pp. 316–330, 2001, doi: 10.1007/3-540-44503-X_20/COVER.

[31] P. Buneman and W. C. Tan, “Provenance in databases,” *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 1171–1173, 2007, doi: 10.1145/1247480.1247646.

[32] W.-C. Tan and U. Santa Cruz, “Provenance in Databases: Past, Current, and Future,” 2007.

[33] J. Vázquez-Salceda *et al.*, “EU PROVENANCE Project: An Open Provenance Architecture for Distributed Applications,” *Agent Technology and e-Health*, pp. 45–63, Apr. 2008, doi: 10.1007/978-3-7643-8547-7_4.

[34] R. Hasan, R. Sion, and M. Winslett, “Introducing secure provenance: Problems and challenges,” *StorageSS’07 - Proceedings of the 2007 ACM Workshop on Storage Security and Survivability*, pp. 13–18, 2007, doi: 10.1145/1314313.1314318.

[35] P. Buneman, A. P. Chapman, and J. Cheney, “Provenance Management in Curated Databases,” 2006.

[36] S. Ram and J. Liu, “Understanding the semantics of data provenance to support active conceptual modeling,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4512 LNCS, pp. 17–29, 2008, doi: 10.1007/978-3-540-77503-4_3/COVER.

[37] A. Woodruff and M. Stonebraker, “Supporting fine-grained data lineage in a database visualization environment,” *Proc Int Conf Data Eng*, pp. 91–102, 1997, doi: 10.1109/ICDE.1997.581742.

[38] D. P. Groth, "Information provenance and the knowledge rediscovery problem," *Proceedings of the International Conference on Information Visualization*, vol. 8, pp. 345–351, 2004, doi: 10.1109/IV.2004.1320167.

[39] G. G. T. Standard, "GS1's framework for the design of interoperable traceability systems for supply chains." GS1, 2017.

[40] W. Ametepe, C. Wang, S. K. Ocansey, X. Li, and F. Hussain, "Data provenance collection and security in a distributed environment: a survey," *International Journal of Computers and Applications*, vol. 43, no. 1, pp. 11–25, 2021, doi: 10.1080/1206212X.2018.1501937.

[41] G. Prause, "Smart contracts for smart supply chains," *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 2501–2506, 2019, doi: 10.1016/j.ifacol.2019.11.582.

[42] A. Jamwal, R. Agrawal, V. K. Manupati, M. Sharma, L. Varela, and J. Machado, "Development of cyber physical system based manufacturing system design for process optimization," *IOP Conf Ser Mater Sci Eng*, vol. 997, no. 1, 2020, doi: 10.1088/1757-899X/997/1/012048.

[43] J. T. Mentzer *et al.*, "DEFINING SUPPLY CHAIN MANAGEMENT," *JOURNAL OF BUSINESS LOGISTICS*, vol. 22, no. 2, p. 1, 2001.

[44] A. Kassahun *et al.*, "Enabling chain-wide transparency in meat supply chains based on the EPCIS global standard and cloud-based services," *Comput Electron Agric*, vol. 109, pp. 179–190, Nov. 2014, doi: 10.1016/J.COMPAG.2014.10.002.

[45] S. S. Kamble, A. Gunasekaran, and R. Sharma, "Modeling the blockchain enabled traceability in agriculture supply chain," *Int J Inf Manage*, vol. 52, Jun. 2020, doi: 10.1016/J.IJINFOMGT.2019.05.023.

[46] F. Zafar *et al.*, "Trustworthy data: A survey, taxonomy and future trends of secure provenance schemes," *Journal of Network and Computer Applications*, vol. 94, pp. 50–68, Sep. 2017, doi: 10.1016/J.JNCA.2017.06.003.

[47] GS1 Global, "GS1's framework for the design of interoperable traceability systems for supply chains: Release 2.0, Ratified, Aug 2017," 2017.

[48] N. Jardine, G. A. Gericke, R. R. Kuriakose, and H. J. Vermaak, "Wireless SMART Product Tracking using Radio Frequency Identification," *2019 IEEE 2nd Wireless*

Africa Conference, WAC 2019 - Proceedings, no. August, 2019, doi: 10.1109/AFRICA.2019.8843418.

[49] "All industries Archives - GS1." https://www.gs1za.org/tech-docs/all-industries/ (accessed May 27, 2021).

[50] A. Craig, A. Marquerita, and J. Abernathy Michael, "Real-time Algorithmic Exchange and Processing of Pharmaceutical Quality Data and Information," *Int J Pharm*, p. 123342, 2023, doi: https://doi.org/10.1016/j.ijpharm.2023.123342.

[51] J.-T. Yang, W.-Y. Chen, C.-H. Li, S. C.-H. Huang, and H.-C. Wu, "APPFLChain: A Privacy Protection Distributed Artificial-Intelligence Architecture Based on Federated Learning and Consortium Blockchain," Jun. 2022, doi: 10.48550/arxiv.2206.12790.

[52] N. Bagdai, P. van der Molen, and A. Tuladhar, "Does uncertainty exist where transparency is missing? Land privatisation in Mongolia," *Land use policy*, vol. 29, no. 4, pp. 798–804, Oct. 2012, doi: 10.1016/J.LANDUSEPOL.2011.12.006.

[53] P. Jahanbin, S. C. Wingreen, R. Sharma, B. Ijadi, and M. M. Reis, "Enabling affordances of blockchain in agri-food supply chains: A value-driver framework using Q-methodology," *International Journal of Innovation Studies*, vol. 7, no. 4, pp. 307–325, 2023, doi: https://doi.org/10.1016/j.ijis.2023.08.001.

[54] O. L. Mokalusi, R. B. Kuriakose, and H. J. Vermaak, "Factors Influencing the Selection of a Blockchain Platform for Incorporating Data Provenance into Smart Contracts," pp. 517–525, 2023, doi: 10.1007/978-981-19-2394-4_47.

[55] O. L. Mokalusi, R. B. Kuriakose, and H. J. Vermaak, "A Comparison of Transaction Fees for Various Data Types and Data Sizes of Blockchain Smart Contracts on a Selection of Blockchain Platforms," in *ICT Systems and Sustainability: Proceedings of ICT4SD 2022*, Springer, 2022, pp. 709–718.

[56] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System." 2008.

[57] G. Zyskind, O. Nathan, and A. S. Pentland, "Decentralizing privacy: Using blockchain to protect personal data," *Proceedings - 2015 IEEE Security and Privacy Workshops, SPW 2015*, pp. 180–184, Jul. 2015, doi: 10.1109/SPW.2015.27.

[58] M. Srikanth, R. N. V. J. Mohan, and M. C. Naik, "Blockchain based Crop Farming Application Using Peer-to-Peer," *xidian journal*, vol. 16, pp. 168–175, 2022.

[59] P. Cui, J. Dixon, U. Guin, and D. Dimase, "A Blockchain-Based Framework for Supply Chain Provenance," *IEEE Access*, vol. 7, pp. 157113–157125, 2019, doi: 10.1109/ACCESS.2019.2949951.

[60] W.-M. Lee, "Using the MetaMask Chrome Extension," in *Beginning Ethereum Smart Contracts Programming: With Examples in Python, Solidity, and JavaScript*, Berkeley, CA: Apress, 2019, pp. 93–126. doi: 10.1007/978-1-4842-5086-0_5.

[61] R. S. Duran *et al.*, "EASY E-VOTE: An Ethereum-based Voting System using IPFS and MetaMask for Student Government Election," in *2023 International Conference on Information Technology (ICIT)*, 2023, pp. 815–820. doi: 10.1109/ICIT58056.2023.10225798.

[62] W. Alkhader, N. Alkaabi, K. Salah, R. Jayaraman, J. Arshad, and M. Omar, "Blockchain-based traceability and management for additive manufacturing," *IEEE Access*, vol. 8, no. October, pp. 188363–188377, 2020, doi: 10.1109/ACCESS.2020.3031536.

[63] R. C. Merkle, "A digital signature based on a conventional encryption function," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 293 LNCS, pp. 369–378, 1988, doi: 10.1007/3-540-48184-2_32/COVER.

[64] H. Huang, J. Lin, B. Zheng, Z. Zheng, and J. Bian, "When Blockchain Meets Distributed File Systems: An Overview, Challenges, and Open Issues," *IEEE Access*, vol. 8, pp. 50574–50586, 2020, doi: 10.1109/ACCESS.2020.2979881.

[65] K. Miyachi and T. K. Mackey, "hOCBS: A privacy-preserving blockchain framework for healthcare data leveraging an on-chain and off-chain system design," *Inf Process Manag*, vol. 58, no. 3, p. 102535, 2021, doi: https://doi.org/10.1016/j.ipm.2021.102535.

[66] Q. Zhang, Y. He, R. Lai, Z. Hou, and G. Zhao, "A survey on the efficiency, reliability, and security of data query in blockchain systems," *Future Generation Computer Systems*, vol. 145, pp. 303–320, 2023, doi: https://doi.org/10.1016/j.future.2023.03.044.

[67] G. Schroeder Meike and Prause, "Reduction of Supply Chain Risks by Using Blockchain Technology," in *Reliability and Statistics in Transportation and Communication*, I. and P. O. Kabashkin Igor and Yatskiv, Ed., Cham: Springer International Publishing, 2023, pp. 151–161.

[68] N. Szabo, “Formalizing and Securing Relationships on Public Networks,” *First Monday*, vol. 2, no. 9, Sep. 1997, doi: 10.5210/FM.V2I9.548.

[69] G. Wood and others, “Ethereum: A secure decentralised generalised transaction ledger,” *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[70] P. Martina, R. Dhanvardini, R. Vijay, R. Amirtharajan, and P. Pravinkumar, “Design development and execution of Smart Contract : An Overview,” in *2023 International Conference on Computer Communication and Informatics (ICCCI)*, 2023, pp. 1–5. doi: 10.1109/ICCCI56745.2023.10128536.

[71] “py-solc-x.” https://pypi.org/project/py-solc-x/ (accessed Sep. 04, 2023).

[72] Q. and L. X. Cai Liang and Li, “Fundamentals of Ethereum: Application Development,” in *Advanced Blockchain Technology: Frameworks and Enterprise-Level Practices*, Singapore: Springer Nature Singapore, 2022, pp. 101–148. doi: 10.1007/978-981-19-3596-1_3.

[73] “web3.py,” 2023. https://pypi.org/project/web3/ (accessed Sep. 04, 2023).

[74] Q. Zhang, Y. He, R. Lai, Z. Hou, and G. Zhao, “A survey on the efficiency, reliability, and security of data query in blockchain systems,” *Future Generation Computer Systems*, vol. 145, pp. 303–320, 2023, doi: 10.1016/j.future.2023.03.044.

[75] E. Yontar, “Critical success factor analysis of blockchain technology in agri-food supply chain management: A circular economy perspective,” *J Environ Manage*, vol. 330, p. 117173, 2023, doi: https://doi.org/10.1016/j.jenvman.2022.117173.

[76] M. Torky and A. E. Hassanein, “Integrating blockchain and the internet of things in precision agriculture: Analysis, opportunities, and challenges,” *Comput Electron Agric*, vol. 178, Nov. 2020, doi: 10.1016/J.COMPAG.2020.105476.

[77] L. A. Risso, G. M. D. Ganga, M. Godinho Filho, L. A. de Santa-Eulalia, T. Chikhi, and E. Mosconi, “Present and future perspectives of blockchain in supply chain management: a review of reviews and research agenda,” *Comput Ind Eng*, vol. 179, p. 109195, 2023, doi: https://doi.org/10.1016/j.cie.2023.109195.

[78] “G. Wood. 2016. Ethereum: a secure decentralised transaction ledger, EIP-150 revision. (2016).”, Accessed: Oct. 12, 2021. [Online]. Available: http://gavwood.com/paper.pdf

[79] M. Grida, M. Magdy, and G. Hussein, “Can Public Blockchains Mitigate Semiconductors Supply Chain Disruption?,” 2023.

[80] A. Musamih, K. Salah, R. Jayaraman, I. Yaqoob, Y. Al-Hammadi, and J. Antony, “Blockchain-based solution for COVID-19 vaccine waste reduction,” *J Clean Prod*, vol. 372, p. 133619, 2022, doi: https://doi.org/10.1016/j.jclepro.2022.133619.

[81] S. K. Rana *et al.*, “Decentralized Model to Protect Digital Evidence via Smart Contracts Using Layer 2 Polygon Blockchain,” *IEEE Access*, vol. 11, pp. 83289–83300, 2023, doi: 10.1109/ACCESS.2023.3302771.

[82] S. Rouhani and R. Deters, “Performance analysis of Ethereum transactions in private blockchain,” *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS*, vol. 2017-November, pp. 70–74, Apr. 2018, doi: 10.1109/ICSESS.2017.8342866.

[83] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends,” *Proceedings - 2017 IEEE 6th International Congress on Big Data, BigData Congress 2017*, pp. 557–564, Sep. 2017, doi: 10.1109/BIGDATACONGRESS.2017.85.

[84] P. Hegde and P. K. R. Maddikunta, “Amalgamation of Blockchain with resource-constrained IoT devices for healthcare applications – State of art, challenges and future directions,” *International Journal of Cognitive Computing in Engineering*, vol. 4, pp. 220–239, 2023, doi: https://doi.org/10.1016/j.ijcce.2023.06.002.

[85] P. D. Ameyaw and W. T. de Vries, “Transparency of Land Administration and the Role of Blockchain Technology, a Four-Dimensional Framework Analysis from the Ghanaian Land Perspective,” *Land 2020, Vol. 9, Page 491*, vol. 9, no. 12, p. 491, Dec. 2020, doi: 10.3390/LAND9120491.

[86] S. Dey, S. Saha, A. K. Singh, and K. McDonald-Maier, “FoodSQRBlock: Digitizing Food Production and the Supply Chain with Blockchain and QR Code in the Cloud,” *Sustainability 2021, Vol. 13, Page 3486*, vol. 13, no. 6, p. 3486, Mar. 2021, doi: 10.3390/SU13063486.

[87] S. Kumar and M. K. Barua, “Exploring the hyperledger blockchain technology disruption and barriers of blockchain adoption in petroleum supply chain,” *Resources Policy*, vol. 81, p. 103366, 2023, doi: https://doi.org/10.1016/j.resourpol.2023.103366.

[88] M. and P. P. W. and T. C. and R. M. J. Minango Juan and Zambrano, "Proof of Concepts of Corda Blockchain Technology Applied on the Supply Chain Area," in *Trends in Artificial Intelligence and Computer Engineering*, O. S. and R. M. R. and D. C. A. and L.-E. W. Botto-Tobar Miguel and Gómez, Ed., Cham: Springer Nature Switzerland, 2023, pp. 619–631.

[89] J. Ktari, T. Frikha, F. Chaabane, M. Hamdi, and H. Hamam, "Agricultural Lightweight Embedded Blockchain System: A Case Study in Olive Oil," *Electronics (Basel)*, vol. 11, no. 20, 2022, doi: 10.3390/electronics11203394.

[90] G. Sankar Ramachandran *et al.*, "Blockchain in Supply Chain: Opportunities and Design Considerations Springer Nature," 2021.

[91] "ETH Gas Station | Consumer oriented metrics for the Ethereum gas market." https://legacy.ethgasstation.info/ (accessed Oct. 18, 2021).

[92] "What Is Gas (Ethereum)?" https://www.thebalance.com/gas-ethereum-5189030 (accessed Oct. 12, 2021).

[93] "53.3-29.2 Gwei | Polygon Gas Tracker | PolygonScan." https://polygonscan.com/gastracker/ (accessed Oct. 18, 2021).

[94] "TRON Ecosystem Introduction." https://developers.tron.network/docs/getting-started (accessed Oct. 17, 2021).

[95] "The Unveiling of Ethereum 2.0 - ETH Gas Station." https://legacy.ethgasstation.info/blog/the-unveiling-of-ethereum-2-0/ (accessed Oct. 17, 2021).

[96] S. Leonardos, B. Monnot, D. Reijsbergen, E. Skoulakis, and G. Piliouras, "Dynamical Analysis of the EIP-1559 Ethereum Fee Market," in *Proceedings of the 3rd ACM Conference on Advances in Financial Technologies*, in AFT '21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 114–126. doi: 10.1145/3479722.3480993.

[97] "Polygon | Ethereum's Internet of Blockchains." https://polygon.technology/ (accessed Oct. 17, 2021).

[98] "10 Breakthrough Technologies 2022: Proof of stake | MIT Technology Review." https://www.technologyreview.com/2022/02/23/1044960/proof-of-stake-cryptocurrency/ (accessed Aug. 23, 2022).

[99] O. L. Mokalusi, R. B. Kuriakose, and H. J. Vermaak, "A comparison of transaction fees for various data types and data sizes of blockchain smart contracts on a selection of blockchain platforms," 2023, Accessed: Sep. 03, 2022. [Online]. Available: https://www.springer.com/series/15179

[100] O. Isaac Abiodun, M. Alawida, A. Esther Omolara, and A. Alabdulatif, "Data provenance for cloud forensic investigations, security, challenges, solutions and future perspectives: A survey," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, Part B, pp. 10217–10245, 2022, doi: https://doi.org/10.1016/j.jksuci.2022.10.018.

[101] A. Buldas *et al.*, "An Ultra-Scalable Blockchain Platform for Universal Asset Tokenization: Design and Implementation," *IEEE Access*, vol. 10, pp. 77284–77322, 2022, doi: 10.1109/ACCESS.2022.3192837.

[102] M. Magdy, M. Grida, and G. Hussein, "Disruption mitigation in the semiconductors supply chain by using public blockchains," *J Supercomput*, 2023, doi: 10.1007/s11227-023-05543-2.

[103] P. P. Ray, "Web3: A comprehensive review on background, technologies, applications, zero-trust architectures, challenges and future directions," *Internet of Things and Cyber-Physical Systems*, vol. 3, pp. 213–248, 2023, doi: https://doi.org/10.1016/j.iotcps.2023.05.003.

[104] "py-ipfs-http-client," 2023. https://pypi.org/project/ipfshttpclient/ (accessed Sep. 04, 2023).

[105] B. Subashini and D. Hemavathi, "Scalable Blockchain Technology for Tracking the Provenance of the Agri-Food," *Computers, Materials and Continua*, vol. 75, no. 2, pp. 3339–3358, 2023, doi: 10.32604/CMC.2023.035074.

[106] R. W. Ahmad, K. Salah, R. Jayaraman, I. Yaqoob, M. Omar, and S. Ellahham, "Blockchain-Based Forward Supply Chain and Waste Management for COVID-19 Medical Equipment and Supplies," *IEEE Access*, vol. 9, pp. 44905–44927, 2021, doi: 10.1109/ACCESS.2021.3066503.

[107] L. Wang *et al.*, "Smart Contract-Based Agricultural Food Supply Chain Traceability," *IEEE Access*, vol. 9, pp. 9296–9307, 2021, doi: 10.1109/ACCESS.2021.3050112.

[108] H. Y. Wu, X. Yang, C. Yue, H.-Y. Paik, and S. S. Kanhere, "Chain or DAG? Underlying data structures, architectures, topologies and consensus in distributed ledger technology: A review, taxonomy and research issues," *Journal of Systems Architecture*, vol. 131, p. 102720, 2022, doi: https://doi.org/10.1016/j.sysarc.2022.102720.

[109] P. Ruan, T. T. A. Dinh, Q. Lin, M. Zhang, G. Chen, and B. C. Ooi, "LineageChain: a fine-grained, secure and efficient data provenance system for blockchains," *The VLDB Journal 2021 30:1*, vol. 30, no. 1, pp. 3–24, Feb. 2021, doi: 10.1007/S00778-020-00646-1.

[110] J. Benet, "IPFS-Content Addressed, Versioned, P2P File System (DRAFT 3)," 2014.

[111] "Python IPFS API Documentation Release 0.4.1 py-ipfs-api team," 2017.

[112] S. and Z. J. and B. D. and L. D. and L. J. Liang Xueping and Shetty, "Towards Decentralized Accountability and Self-sovereignty in Healthcare Systems," in *Information and Communications Security*, C. and C. L. and L. D. Qing Sihan and Mitchell, Ed., Cham: Springer International Publishing, 2018, pp. 387–398.

[113] Q. Lin, H. Wang, X. Pei, and J. Wang, "Food Safety Traceability System Based on Blockchain and EPCIS," *IEEE Access*, vol. 7, pp. 20698–20707, 2019, doi: 10.1109/ACCESS.2019.2897792.

[114] K. Behnke and M. F. W. H. A. Janssen, "Boundary conditions for traceability in food supply chains using blockchain technology," *Int J Inf Manage*, vol. 52, no. May 2019, p. 101969, 2020, doi: 10.1016/j.ijinfomgt.2019.05.025.

[115] "Central University of Technology." http://www.cut.ac.za/ (accessed May 01, 2015).

[116] N. Kumar and D. Mahto, "Assembly Line Balancing: A Review of Developments and Trends in Approach to Industrial Application," *Global Journal of Researches in Engineering*, vol. 13, no. 2, p. 23, 2013.

[117] R. Kuriakose, "Optimization of a Real-Time Multi-Mixed Make-to-Order Assembly Line to Reduce Positive Drift," Central University of Technology, Free State, 2019.

[118] R. B. Kuriakose and H. Vermaak, "Customized Mixed Model Stochastic Assembly Line Modelling Using Simulink," *International journal of simulation: systems, science & technology*, Mar. 2019, doi: 10.5013/IJSSST.A.20.S1.06.

[119] N. C. K. Yiu, "Decentralizing Supply Chain Anti-Counterfeiting and Traceability Systems Using Blockchain Technology," *Future Internet 2021, Vol. 13, Page 84*, vol. 13, no. 4, p. 84, Mar. 2021, doi: 10.3390/FI13040084.

[120] Anthony, M. C. Lee, R. R. Pearl, I. S. Edbert, and D. Suhartono, "Developing an anti-counterfeit system using blockchain technology," *Procedia Comput Sci*, vol. 216, pp. 86–95, Jan. 2023, doi: 10.1016/J.PROCS.2022.12.114.

[121] T.-Y. Li *et al.*, "Learned-Index-Based Semantic Keyword Query on Blockchain," *Mathematics 2023, Vol. 11, Page 2055*, vol. 11, no. 9, p. 2055, Apr. 2023, doi: 10.3390/MATH11092055.

[122] Y. Cao, F. Jia, and G. Manogaran, "Efficient Traceability Systems of Steel Products Using Blockchain-Based Industrial Internet of Things," *IEEE Trans Industr Inform*, vol. 16, no. 9, pp. 6004–6012, Sep. 2020, doi: 10.1109/TII.2019.2942211.

[123] M. Kuhn, F. Funk, G. Zhang, and J. Franke, "Blockchain-based application for the traceability of complex assembly structures," *J Manuf Syst*, vol. 59, pp. 617–630, Apr. 2021, doi: 10.1016/J.JMSY.2021.04.013.

[124] W. Alkhader, K. Salah, A. Sleptchenko, R. Jayaraman, I. Yaqoob, and M. Omar, "Blockchain-Based Decentralized Digital Manufacturing and Supply for COVID-19 Medical Devices and Supplies," *IEEE Access*, vol. 9, pp. 137923–137940, 2021, doi: 10.1109/ACCESS.2021.3118085.

Appendix A