UNIVERSITY OF BIRMINGHAM

THE APPLICATION OF BLOCKCHAINS TO RAILWAY CONDITION MONITORING

by

RAHMA AHMED G ALZAHRANI

A thesis submitted to the University of Birmingham for the degree of

DOCTOR OF PHILOSOPHY

Birmingham Centre for Railway Research and Education

School of Engineering

College of Engineering and Physical Sciences

University of Birmingham

May 2025

Abstract

Ageing infrastructure and fragmented data ownership present major challenges to remote condition monitoring technologies in the European railway sector. Despite the potential of these technologies to improve efficiency and safety, their deployment is often limited by issues related to data silos, stakeholder mistrust, and the lack of transparent, enforceable cost attribution models. This thesis investigates how blockchain and smart contract technologies can be leveraged to address these challenges. The research focuses on key questions: how blockchain can reduce centralisation and mistrust; how it can improve transparency and compliance in data cost attribution; how smart contracts can automate and streamline the attribution process; how blockchain can ensure data integrity without storing large volumes of data; and what practical applications blockchain may have in railway operations. A blockchain-based framework was designed and implemented to enable fair, transparent, and legally compliant attribution of data costs across stakeholders. The system incorporates smart contracts to enforce agreement clauses without third-party involvement. The performance of the developed framework was tested under various scenarios to assess scalability, execution efficiency, and compliance with railway sector requirements.

The primary contributions of this research are: the development of a cross-border data accounting framework; the establishment of operational links between the framework and real-world business and commercial processes; and a working proof-of-concept tailored to the European rail industry. These contributions demonstrate that blockchain can serve as a practical and scalable foundation for trusted, decentralised data management in multi-stakeholder transport environments.

In Memory of My Father, Ahmed...

Acknowledgements

First and foremost, I would like to express my profound gratitude to Allah, whose unwavering guidance and grace have been my foundation throughout this journey. His blessings have given me the strength and perseverance to overcome the many challenges I faced along the way.

I am deeply indebted to my supervisor, Dr. John M. Easton, whose support went far beyond academic guidance. His understanding and compassion during my social and personal difficulties were truly invaluable. His mentorship has profoundly shaped both my research and my personal growth, and for that, I am extremely grateful.

I also wish to express my sincere thanks to my internal examiner, Dr. Stuart Hillmansen, and my external examiner, Dr. Jay Daniel, for their valuable time, insightful feedback, and constructive critique during the viva. Their perspectives helped improve the quality of this work and have greatly contributed to my academic development.

I wish to acknowledge Simon Herko, whose contributions during the early stages of this project played a significant role in shaping the foundational ideas on which this work is built. His involvement helped set the direction that led to this thesis. I would also like to thank Dr.Mani Entezami, whose guidance in explaining and refining the use cases employed in this work has been of great support.

This research was made possible through the funding provided by the Shift2Rail Joint Undertaking (JU) under grant agreement No. 826156. The JU receives support from the European Union's Horizon 2020 research and innovation programme, as well as from Shift2Rail JU members other than the Union. I am sincerely thankful for this financial support, which has been instrumental in the successful completion of this project.

To my family, I owe an immeasurable debt of gratitude. My deepest thanks go to my mother, whose strength, love, and encouragement have been a constant source of support throughout this challenging journey. Her sacrifices and wisdom have carried me through some of the most difficult moments, and I could not have achieved this without her steadfast belief in me.

To my beloved sons, Khaled, Faisal, and Yousef, your understanding and patience have touched me in ways words cannot fully express. You have borne the weight of my absence with remarkable resilience, facing the challenges that came with it, despite your tender age. I am acutely aware of the sacrifices you have made, moments missed, time spent apart, and the emotional toll that comes with having a parent deeply engrossed in such an endeavor. Your love and perseverance have been a source of profound motivation for me. I am deeply humbled by your ability to endure, and it is for you that I have strived to complete this journey. Every step of this achievement belongs as much to you as it does to me.

Lastly, I would like to thank all those who supported me, both directly and indirectly, through this transformative period of my life. This achievement is as much yours as mine.

Contents

List of Figures

List of Tables

List of Acronyms

AI Artificial Intelligence. 17, 19, 20, 32, 81

API Application Programming Interface. 52, 85, 102, 215

B4CM Bockchain for Condition Monitoring. 3, 4

BPMN Business Process Model and Notation. 85

CA Certificate Authority. 52, 68, 220

CBTC Communication Based Train Control. 83

CVE Common Vulnerability Enumerations. 52

DaaP Data as a Product. 216

DAO Distributed Autonomous Organization. 54

DfT Department for Transport. 24, 65, 219

DLT Distributed Ledger Technology. xix, 6, 7, 39, 44, 57, 75, 77–80, 85

DSS Decision Support System. 28

ESA European Space Agency. 75

ESCC Endorsement System Chaincode. 68

FOC Freight train Operating Cmpanies. 24, 65

GAP Good Agricultural Practices. 76

GMP Good Manufacturing Practices. 76

GNSS Global navigation satellite system. 92

GUI Graphical User Interfaces. 102

HLF Hyperledger Fabric. xiii, xiv, 52, 63, 64, 66–68, 71, 74, 89, 100, 113, 115, 116, 136, 172, 176, 179, 182, 194, 196, 197, 199, 201, 212, 217–223

HS2 High Speed 2. 21

IAMS Intelligent Asset Management Systems. 85

ICT Information and Communication Technology. 19

IM Infrastructure Manager. 84

IMU Inertial Measurement Unit. 92, 93

IoT Internet of Things. 2, 10, 13, 17–20, 25–27, 32, 35–37, 75, 81, 87–89, 93–95, 99–104, 106, 111, 114, 212, 216, 218

IP Innovation Programme. xiii, 83, 84

IPFS InterPlanetary File System. 213

IPX Innovation Programme X. 3, 4, 84

IT Information Technology. 26

IV Initialization Vector. 42

MAC Message Authentication Code. 42

MD5 Message Digest 5. 42, 43

MVCC Multi-Version Concurrency Control. 68, 70, 217

NIST National Institute of Standards and Technology. 43

NR National Rail. 24

NSA National Security Agency. 43

OM Operation and Maintenance. 19, 20, 29

ORR The Office of Rail and Road. 65, 185, 220

OT Operational Technology. 26

PBFT Practical Byzantine Fault Tolerance. 50

PII Personally Identifiable Information. 53

PLC Programmable Logic Controller. 17

PoS Proof of Stake. 61

PoW Proof of Work. 61

RCF Rail Contact Fatigue. 25

RCM Remote Condition Monitoring. 1–3, 6, 24, 214, 215

RSSB Rail Safety and Standards Board. 2, 24

SDSS Smart Decision Support Systems. 17, 19

SHA-1 Secure Hash Algorithm 1. 42, 43

SLA Service Level Agreement. 84, 98, 210, 218

SPV Simplified Payment Verification. 59

SUT System Under Test. 177, 182, 186

TOC Train Operating Companies. 24, 65

TPS Transactions Per Second. 177, 184–191, 193, 194, 196, 199, 201

TTP Trusted Third Party. 7, 33–36

VR Virtual Reality. 17, 20

VSCC Validation System Chaincode. 68, 70

XIRCM Cross-Industry RCM. 2

ZKP zero-knowledge proofs. 35

Chapter 1

Introduction

This chapter provides a comprehensive overview of the major obstacles impacting the effectiveness of Remote Condition Monitoring (RCM) in the railway sector and examines how Blockchain technology can help overcome these issues. Section 1.1 outlines the current technological landscape and the emergence of data sharing issues. Section 1.2 introduces the core problem and highlights how Blockchain could serve as a solution. Research gaps and questions are identified in Section 1.3, followed by the research aims and contributions in Section 1.4. Section 1.5 provides an outline of the thesis structure, and Section 1.6 lists the related publications.

1.1 Overview

The continuous improvement of service quality within the railway industry is an ongoing endeavor. The industry has increasingly integrated various technologies to transition from a primarily mechanical and electronic infrastructure to one that is more information-driven. One such technology is Remote Condition Monitoring (RCM), which plays a crucial role in enhancing the overall robustness, ease of access, safety, and dependability of the railway network. By leveraging RCM, it becomes possible to detect and diagnose both existing and potential faults, thereby enabling preventive maintenance. This approach helps to prevent system breakdowns, high-cost malfunctions, and delays. As a result, cutting-edge computing and sensing technologies are increasingly essential to meet the growing need for real-time monitoring

of railway assets, ensuring that maintenance can be carried out promptly. Consequently, there is a continuous integration of sensors and smart devices, leading to the generation of vast amounts of data.

Railway RCM activities can generally be categorised into four primary categories based on the location of the monitoring sensors and the assets being monitored: train monitoring train, infrastructure monitoring infrastructure, train monitoring infrastructure, and infrastructure monitoring train (Ward et al., 2011). In railway systems with a single infrastructure manager, such as in Great Britain, sensors installed on assets owned by one stakeholder but used to monitor assets owned by another typically fall into the train monitoring infrastructure or infrastructure monitoring train quadrants. For instance, sensors mounted on fixed infrastructure that monitor wheel flats on rolling stock illustrate this scenario (Alemi, Corman, and Lodewijks, 2016). In such cases, the stakeholder benefiting from the system might not be the party responsible for the costs associated with setting up and maintaining the monitoring equipment. This misalignment of costs and benefits can hinder the adoption of technologies that offer net business benefits to the entire rail system, a challenge that is likely to be amplified with the rise of the Internet of Things (IoT).

Addressing this challenge requires fostering stronger collaboration among railway industry stakeholders, particularly in cases where equipment is installed on trains to monitor infrastructure or on infrastructure to monitor trains. Such cooperation would facilitate the full utilisation of RCM across the rail industry through the sharing of RCM data. To address this issue Rail Safety and Standards Board (RSSB), on behalf of the Cross-Industry Remote Condition Monitoring Strategy Group, initiated the Cross-Industry RCM (XIRCM) research program, known as the T1010 project (Sparkrail, 2014). The initial findings of this research were presented by the RSSB and Network Rail at the IET RCM conference in 2014 (Tucker and Hall, 2014). To effectively develop business cases for new monitoring and sensing hardware, it is crucial to assign value to data that is generated by one party but utilised by another. In addressing the associated cost challenges, Project T1010 proposed that commercial agreements should be established between involved parties prior to the installation of any new monitoring systems (Sparkrail,

2016). Examples of these proposed agreements are detailed in Appendix A. However, these agreements have certain limitations that impede comprehensive data management and precise cost allocation among stakeholders. Specifically, they do not resolve the requirement for a reliable intermediary to enforce compliance with the agreement, nor do they provide assurances against the potential for misconduct by the parties involved.

Given these limitations, it is posited that employing technologies such as Blockchain could significantly enhance cost attribution and data sharing among parties, effectively addressing the identified challenges. The secure and auditable characteristics of Blockchain technology can incentivise participants to share RCM data and equitably distribute the associated costs. As Blockchain operates on three foundational protocols: decentralisation, cryptography, and consensus, its adoption can promote interoperability among business processes and stakeholders (Crosby et al., 2016). Blockchain-driven solutions have the potential to improve the efficiency and security of transactions, leveraging the immune censorship and immutable nature of distributed trust platforms that this innovative technology provides.

While Blockchain remains in its early stages of development, ongoing efforts are being made to explore its applicability and potential adoption across various sectors, including the industrial sector (Friedlmaier, Tumasjan, and Welpe, 2018; Risius and Spohrer, 2017). The key obstacles to the adoption of this disruptive technology in industry have been thoroughly investigated and analysed (Biswas and Gupta, 2019). In the railway industry, Blockchain-based systems have already found applications in areas such as ticketing, invoicing, and freight consignment (Mcmahon, T. Zhang, and Dwight, 2020). Beginning in 2009, Shift2Rail, a Joint Undertaking funded by the European Union, has endorsed numerous research and innovation projects aimed at enhancing the productivity, efficiency, and sustainability of the railway sector (Haltuf, 2016). Blockchain technology has piqued Shift2Rail's interest and has emerged as a focus area. Consequently, several research and innovation projects have been funded to investigate its potential applications in rail transport. Among these initiatives, Bockchain for Condition Monitoring (B4CM) launched in January 2020, and a considerable part of the research presented in this thesis has been carried out as part of the project B4CM, which falls in Innovation

Programme X (IPX) [1].

1.2 Problem Statement

The B4CM initiative is managed by Shift2Rail as part of IPX, aiming to implement a Blockchain-based solution to allocate data costs across different rail organisations' boundaries. As detailed in Chapter 2, monitoring the condition of railway assets is an essential requirement for their operations, as it increases the longevity of the physical asset and reduces the risk of incidents leading to delays, service disruption, or even catastrophes. Although there is continued development using new technologies, as detailed in 2.3.2, there is still an inability to provide accurate and timely information on the condition of the asset based on the "Network Rail Challenge Statements" that were reviewed in 2.4.1. Therefore, the availability of abundant data is crucial for achieving quick and successful data analytics to establish a good and comprehensive understanding of asset condition.

According to the literature, "big" data refers to an information asset marked by high volume, velocity, and variety, requiring specialised technology and analysis techniques to convert it into value. The value derived from big data is not inherent in the data itself but in the methods used to uncover value, patterns, and insights that might otherwise be obscured or undiscovered.

The railway network comprises multiple diverse groups of stakeholders who produce or utilise data for their business activities. The main challenges encountered are the mistrust among network participants, problems with data quality and its evaluation, and the lack of a sustainable model for long-term revenue generation. These challenges are compounded by the fragmented nature of the railway ecosystem, where stakeholders such as infrastructure managers, train operators, and third-party service providers often operate under differing standards and priorities. This leads to data silos, where valuable condition monitoring data remains underutilised due to poor accessibility, lack of interoperability, or concerns over data misuse. Additionally, there is currently no effective incentive mechanism to encourage stakeholders to share high-quality

[1] https://www.b4cm.co.uk/

data. Without assurance that their data contributions will be acknowledged, compensated, or protected, many organisations are reluctant to participate in collaborative data sharing initiatives. Trust is another persistent issue: stakeholders often hesitate to engage in direct data exchange due to concerns about data manipulation, ownership disputes, or unauthorised exploitation of sensitive information. As a result, despite the abundance of potential data, its true value remains untapped, undermining the goal of proactive and predictive asset management.

The rationale for introducing Blockchain technology lies in its potential to systematically address these trust and incentive-related barriers. By providing an immutable, transparent ledger and programmable logic via smart contracts, Blockchain offers a framework for secure, auditable, and fair data transactions—something traditional systems lack. Through smart contracts, it becomes possible to define and enforce data usage agreements, implement automatic reward mechanisms, and manage data rights without needing a centralised authority. This aligns with the unique needs of the railway sector, where decentralisation and trustless collaboration are key to effective digital transformation. This scenario facilitates the use of Blockchain technology (serving as software infrastructure) and smart contracts (acting as an application layer) to ensure the secure exchange of data sources within a trusted environment. Additionally, these technologies can be employed to establish a direct reward system for the exchange of data sources. Organisations within the railway can gain financial advantages from their data while also encouraging sustainable data monetisation and building trust among different stakeholders.

The proposed platform is expected to eventually offer the following:

- The participants in the process can manage and control their own data autonomously, without the need for intermediaries or fully centralised storage solutions.

- The data exchange is governed by implemented open standards and can be enhanced through the use of smart contracts, which will facilitate the monetisation of data transactions.

- Hence, the potential to generate revenue through data exchange within the digital ecosystem, especially in the B2B (business-to-business) sector, along with the automation of

governance procedures within the digital ecosystem, is enabled.

- All individuals involved in the transaction have access to the same information, leading to quicker data exchange and improved data quality and analysis.
- The use of smart contracts could solve the problem of managing the marketplace with flexibility. The information will be distributed, and its value will vary based on its application or predefined attributes. Unlike current data exchange markets, this approach does not depend on any reliable intermediaries.
- Data producers and consumers can collaborate to establish a network for data-driven value transfer. The fundamental component of this network consists of agreements adhered to by all members. The network will autonomously log transactions, enabling data owners to review how their data is utilised. This approach can facilitate data circulation and enhance data-heavy applications.

Table 1.1 offers an overview of the expected benefits of the proposed platform, as previously mentioned, in the context of utilising DLT.

1.3 Research Gap and Research Questions

Current RCM systems operate in data silos, with very little industry-level oversight of what data is being collected, who is using it and what benefit. The lack of integration between stakeholders across the industry in remote condition monitoring hinders the full exploitation of these isolated data silos. As this problem has been recognised previously in project T1010, the suggested solution, which was provided through template commercial agreements, does not cope with current technical development as it suffers from the following limitations:

- They do not provide a mechanism for recording the evidence needed to enforce the agreements.
- They don't address the need for robust communication support so that data is made consistent and reliable.

Opportunity	Advantages of applying DLT
Develop an innovative method for leveraging data sources within the Railway ecosystem to generate income.	• The immediate exchange of value among participants. • Efficient management and regulation of data sources.
Create a viable framework for sharing data sources in a sustainable manner by implementing motivating factors.	• Participants can exchange goods or services with each other smoothly and transparently, even when they do not know one another. • An incentive-based reward system is in place for data sources.
Tracking the use of data sources.	Having a ledger that can be audited and is unable to be changed guarantees the credibility of past data and how it is used.

Table 1.1: Advantages and potential gains of implementing DLT.

- A Trusted Third Party (TTP) to enforce compliance to the agreement is still required.

Motivated by these challenges, our ultimate aim in this research is to develop solutions by leveraging state-of-the-art Blockchain and smart contract technologies to overcome all these limitations. The eminent advantages of Blockchain technology, namely immutability, traceability, and transparency, can be exploited effectively to manage fair data cost attribution fully complying with the railway sector legislation.

1.3.1 Research Questions

The following research questions are proposed in this research:

Research Question 1: Considering that data silos are typical in the rail sector, can Blockchain technology address the fundamental issues of data centralisation and stakeholder mistrust?

Research Question 2: Can Blockchains improve the transparency of the data costs attribution process and compliance with agreement clauses and therefore reduce the need for a third

party to enforce the terms of the agreement?

Research Question 3: How can the efficiency of the data cost attribution process be improved via smart contracts?

Research Question 4: In light of the fact that Blockchains aren't meant to be used for storing huge amounts of data, how can the features of Blockchain technology be used to ensure the integrity of the data that is exchanged?

Research Question 5: What are the potential applications of Blockchains in the context of the railway sector?

1.4 Research Aim, Contributions and Objectives

1.4.1 Research Aim

The main objective is to build and develop a Blockchain-based testbed to automate the attribution of data costs across organisational boundaries and subsequently evaluate how the framework functions within the rail industry setting.

1.4.2 Research Contributions

1. **Development of a Framework for Cross-Border Data Accounting.**

   The first contribution of this thesis includes the development of use cases for applying Blockchain in the rail sector and a framework for attributing data costs through the use of Blockchains.

2. **Building Connections Between the Framework and Business and Commercial Operations.**

   This thesis aims to create smart contracts to facilitate data transfer and examine payment models, thereby elucidating the connection between the framework and the UK's commercial environment.

3. **Implemented Proof-of-Concept for the European Rail Industry.**

This thesis will create a functional testbed that will be used to showcase the framework and its integration into the financial systems of the European rail industry. The testbed will utilise actual railway data or simulated sensor installations. The formal results of this task will consist of various deliverables.

1.4.3 Research Objectives

1. Creation of use cases to support the application of Blockchain in the railway industry.
2. Development of a framework based on the Blockchain technology for the cost attribution in systems across industry boundaries.
3. Incorporate the developed Blockchain-based framework into the financial operations of the railway sector.
4. Enable extendable work for the future developer to build upon the developed test-bed based on a known and complete working configuration.
5. Influence the best practice in innovation development and technology uptake in leading and evolving domains within the railway industry by disseminating all findings and lessons learned in this research.

As shown in Table 1.2, the research objectives and questions are mapped to the corresponding chapters where each objective is addressed and each question is answered. This mapping illustrates how the study systematically fulfills its aims and ensures that all research questions are comprehensively tackled throughout the thesis.

Objective	Research Question	Answered In
1	1 & 4 & 5	Chapter 3, Chapter 4 , and Chapter 5
2	2 & 3	Chapter 5
3	2 & 3 & 4	Chapter 4 and Chapter 5
4	1 & 5	Chapter 3 , Chapter 4 , Chapter 5 and Chapter 6
5	1 & 5	Chapter 3 and Chapter 7

Table 1.2: Mapping of research objectives and research questions to the corresponding chapters in this thesis.

1.5 Thesis Structure

This thesis is structured into seven chapters. The first chapter, which is the present, provides a general overview of the research domain and introduces the key challenges that motivated this study. It also outlines the problem statement, defines the research aim, and presents the core research questions and objectives. Additionally, it highlights the main contributions of the study and includes a list of related publications.

Chapter 2: Condition Monitoring in Railway Industry

Chapter 2 covers condition monitoring within the railway sector. It includes a thorough review of the technologies currently used in this domain, examining their role in enhancing the reliability and safety of railway operations. The chapter also explores the critical role of the IoT in advancing condition monitoring practices, with particular attention to challenges identified by Network Rail. Additionally, the chapter addresses the concept of big data, highlighting the complexities it introduces to condition monitoring, particularly in relation to data processing, storage, and interpretation.

Chapter 3: Blockchain Technology Background and Applications

Chapter 3 delves into essential background information crucial for the research. It introduces all the core concepts that make up the blockchain network. The chapter covers Hyperledger Fabric components and why it was chosen as the primary infrastructure for the proposed system. The chapter also reviews Blockchain applications in various industries and closely explores its implementation in the railway industry.

Chapter 4: IoT Integration and Use Cases

Chapter 4 presents a simulator system designed to emulate sensors that produce data for hashing and transmission to the Blockchain. In addition, it outlines and describes the two use cases that will be used to evaluate the developed blockchain application.

Chapter 5: Proof of Concept and Implementation

Chapter 5 introduces a proof of concept that details the design and implementation of the developed system.

Chapter 6: Benchmarking The Developed Application Performance

Evaluating the performance of the implemented smart contracts within the system is presented in Chapter 6. Performance benchmarking is achieved by presenting the throughput and the average latency as the metrics utilised.

Chapter 7: Discussion and Conclusion

Chapter 7 offers a discussion and a combined summary of the results to demonstrate whether the project addresses the research questions. It concludes the thesis and sheds light on some topics for further work.

1.6 Publications

The publications completed prior to submission are:

- Blockchain application in remote condition monitoring.
  - 2020 IEEE International Conference on Big Data (Big Data).
  - Published in 2020.
  - **Cite as:** R. A. Alzahrani, S. J. Herko and J. M. Easton, "Blockchain Application in Remote Condition Monitoring," 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 2020, pp. 2385-2394, doi: 10.1109/BigData50022.2020.9377895.
- Blockchain-hosted data access agreements for remote condition monitoring in rail.
  - The Journal of The British Blockchain Association.
  - Published in 2021.
  - **Cite as:** R.A. Alzahrani, S.J. Herko and J.M. Easton, "Blockchain-Hosted Data Access Agreements for Remote Condition Monitoring in Rail", The Journal of The British Blockchain Association, 4 (2) (2021), pp. 1-9, https://doi.org/10.31585/jbba-4-2-(3)2021.

Chapter 2

Condition Monitoring in Railway Industry

2.1 Introduction

This chapter provides an overview of condition monitoring in the rail industry, its benefits and challenges, and the current state-of-the-art techniques and technologies employed to enhance asset condition monitoring in the industry. This chapter demonstrates the potential applications of the big data concept in the rail transport sector, as well as its advantageous impact on operations and technology from a systemic point of view. Section 2.2 highlights the deregulation process of the national railway system. Section 2.3 provides an overview of the key technologies gaining traction in Industry 4.0, particularly those applied to condition monitoring in the railway sector. Section 2.4 sheds light on the applications of IoT technology in condition monitoring. The role of big data, techniques, and challenges in condition monitoring is covered in section 2.5. Section 2.6 explores the monetisation of data and the use of micropayments in data exchanges. Finally, the conclusion will be provided in Section 2.7.

2.2 Background

In terms of technology and operations, the railway industry as a whole is highly complex. A railway consists of both fixed and moving components that work together technologically. Consumption, operational conditions, and weather are some factors that affect the performance of

technical systems. The performance of technical systems also impacts operations. The primary service, transportation, is generated through the interaction between technical elements and humans, including both passengers and crew members who fulfill specific roles. All operations must achieve the objectives of reliability, on-time performance, safety, and environmental protection. At the organisational level, where various actors and organisations interact, planning, coordination, and regulation are integrated. It follows that the railway is a great subject for system research.

Since national railway organisations that included rolling stock and infrastructure were privatised and divided into two groups, operators and infrastructure managers, the railway industry has seen significant changes (C. Nash, 2008). The operators, including the freight and passenger service operators, took control of the traffic flow, while the infrastructure manager took control of the fixed assets. Traffic was completely deregulated, and these businesses were privatised throughout Europe (Smith and C. A. Nash, 2023). The number of train-kilometers has been gradually rising in this situation, primarily for passenger trains. The number of participants in the rail transportation sector has grown in the meantime. The railway transportation sector has transformed as a result of improved infrastructure, faster trains, and more advanced train technology. With regard to organisation, operations, and technology, rail transportation is a very complicated business.

2.2.1 Maintenance Strategies in Railway

Common maintenance strategies include reactive, preventive, and predictive maintenance as depicted in Figure 2.1 (Velmurugan and Dhingra, 2015;Bhebhe and Zincume, 2020). These approaches are designed to ensure operational continuity, safeguard the environment, and maintain product quality within acceptable standards.

1. Reactive maintenance is aimed at minimising response time, equipment downtime, and labor requirements. While this strategy can reduce immediate maintenance costs, it does not account for potential failures in advance, sometimes resulting in unnecessary replacements or overlooked repairs (Swanson, 2001). A subset of this approach, corrective

maintenance, involves unscheduled repairs following equipment failure. Its effectiveness is contingent upon the system's ability to function at acceptable performance levels post-repair. However, due to time constraints and limited planning, this technique often results in incomplete maintenance focused only on immediate failure symptoms, leading to elevated costs and suboptimal engineering outcomes.

Another reactive technique is Run-To-Failure, where equipment is intentionally operated until it fails. Although this technique and corrective maintenance share reactive characteristics, Run-To-Failure is more structured, with prearranged plans for spare parts, staffing, and procedures, thereby reducing disruptions to production. Despite this, Run-To-Failure remains unpredictable and resource-intensive, requiring a substantial inventory of spare components (Bhebhe and Zincume, 2020).

2. Predictive maintenance emerged to proactively address failures by monitoring equipment condition in real-time. It aims to lower failure rates and repair frequency, detect potential issues before breakdowns occur, and minimise both downtime and response time (Mitchell and Murry, 1995). Techniques such as ultrasonic testing, lubrication analysis, and vibration monitoring are commonly used, often in conjunction with advanced sensor technologies (Selcuk, 2017). This strategy contributes significantly to improving system reliability, availability, safety, efficiency, and product or service quality. Its implementation relies on two key assumptions: that equipment degradation can be detected before failure occurs and that unanticipated failure imposes unacceptable risks in terms of cost, safety, and environmental impact. Effective predictive maintenance involves three key stages: data acquisition, data processing, and maintenance decision-making (Selcuk, 2017).

3. Preventive maintenance, illustrated in Figure 2.1, includes time-based or calendar-driven and routine maintenance practices. This strategy is typically scheduled based on the elapsed operational time or expected life span of components, drawing from probabilistic models and historical failure data (Alaswad and Xiang, 2017). Its primary objectives are

to enhance safety, reliability, and availability while also controlling labor and inventory costs.

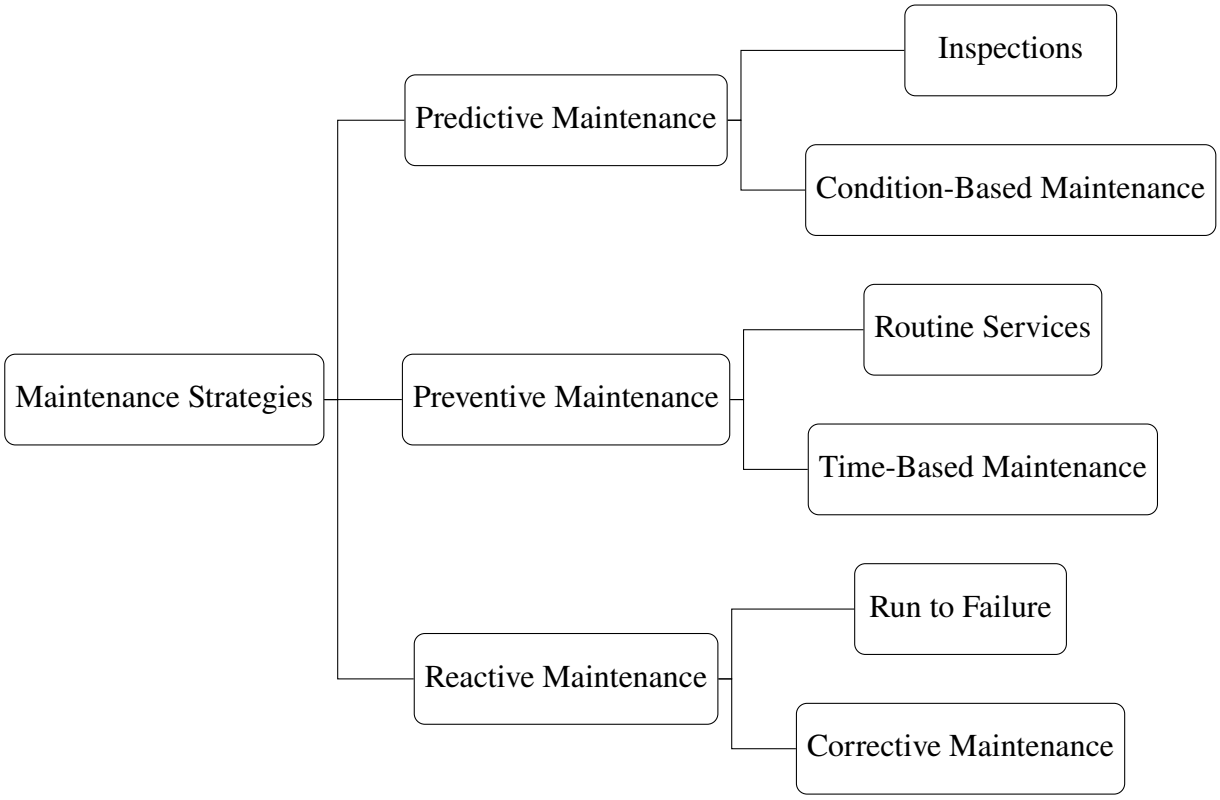

Figure 2.1: Maintenance strategies and their classifications (Bhebhe and Zincume, 2020;A. Al-abdulkarim, D. Ball, and Tiwari, 2014).

This comprehensive categorisation provides a foundational understanding of maintenance strategies in the railway industry.

2.3 Industry 4.0 Technologies Applied to Condition Monitoring in Railway

Authors in (Landscheidt and Kans, 2016) claim that Industry 4.0 represents a fourth generation of industrial activity, characterised by smart systems and Internet-enabled solutions. When production became mechanised in the nineteenth century, the first revolution occurred. The second revolution took place in the previous century when parts and procedures were standardised and production became electrified. Production digitisation is commonly referred to as the third

industrial revolution, which began with the advent of Programmable Logic Controllers (PLCs) in the late 1960s.

The transport industry, specifically the railway sector, has embraced many benefits brought by Industry 4.0. Indeed, Industry 4.0 is founded on the technologies that facilitate communication, identify climatic factors, gather detailed data, and make customised instrumentation decisions (Gerhátová, Zitrickỳ, and Klapita, 2021), all of which are vital for optimising and maintaining processes and infrastructure in the railway industry. The leading technologies utilised in the rail transport sector comprise: Artificial Intelligence (AI),IoT, Cloud Computing, Big Data, Cybersecurity, Modeling and Simulation, Smart Decision Support Systems (SDSS), Computer Vision and Virtual Reality (VR). These technologies improve resource and information utilisation, streamline industry processes, and open new avenues for product innovation (Niebel, Rasel, and Viete, 2019). As a result, services are getting better quality, there are new savings, and resources are being used more effectively.

Within the monitoring domain, the objective is to concentrate on the oversight of railway infrastructure related to technological assets, identify anomalies in the rails and train machinery, and monitor environmental pollution, among other aspects. It's important to highlight that condition monitoring aims at the prompt identification of problems within railway or train infrastructure to guarantee safe train operations and reduce maintenance costs (Jamshidi et al., 2018). Therefore, implementing real-time monitoring in railway infrastructure can enhance the reliability, availability, maintainability, and safety of the rail network (Hoelzl et al., 2022).

The technological solutions within the monitoring domain have diverse objectives. For instance, some solutions aim to assist the train driver in making improved decisions using gathered travel data, while others concentrate on supervising the train equipment to avert mechanical failures. Outlined below are various condition monitoring solutions in the railway sector that leverage Industry 4.0 technologies:

2.3.1 Rail Condition Monitoring

In monitoring rail conditions, the suggested solutions focused on using Industry 4.0 technologies to inspect the condition of the rail and its assets to identify potential defects. This approach aims to reduce maintenance costs and prevent accidents on the rails, among other benefits. The infrastructure of railway systems consists of two main parts: the superstructure and the substructure (Qian et al., 2019). Typically, the superstructure includes components such as rails, sleepers, and fastening mechanisms, whereas the substructure comprises elements like subgrades, ballast, and sub-ballast layers. Monitoring mechanisms can be employed during two distinct phases in the lifecycle of a railroad. The first phase pertains to the production of railway equipment, while the second phase involves monitoring once the equipment is operational on the railway. During the manufacturing phase, monitoring systems primarily aim to ensure defect-free production. Conversely, once the equipment is in place on the railways, these systems continuously monitor the performance of mechanical components and feed data to intelligent systems to avert potential failures and hazards during train operations. Researchers in (Jwo et al., 2021) introduced a deep-learning model aimed at automating the inspection of railway wheelsets to enhance the precision and effectiveness of conventional manual inspection methods for wheelset assembly quality. The neural network model they proposed, built on ResNet-50 (He et al., 2016), processes images sized at 400 × 602 pixels. It achieved a match accuracy of 100% with the actual values in a data set of 386 test images and has been implemented in an actual manufacturing environment.

2.3.2 Train Condition Monitoring

The proposed solutions in this field focus on collecting train data to analyse and extract insights on train efficiency, environmental impact, or equipment malfunctions. Typically, research in this category involves real-time data collection and processing to enhance day-to-day operations and profitability. Researchers in (Brezulianu et al., 2020) introduced a freight-train monitoring system that utilises wireless sensors, the IoT, and web applications for representation purposes.

This system, named FEDORATA, is capable of tracking various freight train metrics such as location, vibration, temperature, speed, and energy usage, helping with administrative and technical decisions aimed at lowering maintenance expenses.

2.3.3 Technologies in Condition Monitoring

The advancement of Information and Communication Technology (ICT) and decision-making powered by big data is crucial for the fourth industrial revolution. Industry 4.0 is marked by the combination of computerisation supported by cyber-physical systems and smart factories utilising the IoT (Amadi-Echendu et al., 2010). The term "Internet of Things" refers to a network of cyber-physical systems that communicate via embedded systems, another name for them (Le and Jeong, 2016). Combining these two ideas results in a distributed ecosystem of embedded systems that interact with one another. Therefore, in today's competitive environment, people, organisations,cities, and,systems are becoming more networked, instrumented, and intelligent. In the railway transport industry, which is increasingly adopting Industry 4.0 principles, effective monitoring tools remain a significant challenge. Based on several studies related to rail and train monitoring, as shown in Table 2.1, the main obstacle is creating these tools. Various proposed solutions often involve a blend of different technologies and methods. In most cases, using IoT technology to capture real-time data is the first phase. Next, the gathered data is transmitted to processing algorithms (sometimes utilising cloud technology) for cleaning purposes. Subsequently, the proposed solution is implemented using procedures based on AI, SDSS, Computer Vision, and simulation modeling. Ultimately, alerts or condition analysis reports are sent to personnel to address any arising needs.

In addition to IoT, big data analytics plays a crucial role in enabling "smart railways". The integration of cyber-physical systems, IoT, and cloud computing forms the foundation for these advanced systems, which allow for more intelligent and predictive operations in the railway sector. In reality, enhanced Operation and Maintenance (OM) through self-learning and intelligent systems that predict failures, diagnose issues, and trigger maintenance actions could be considered one application area of big data that has elevated expectations. These

Domain	Studies	Applied Technology
Rail Monitoring	(Mujica, Henche, and Portilla, 2021)	IoT, AI
	(Cui et al., 2019)	IoT, AI, Big Data
	(Karakose and Yaman, 2020)	IoT, Computer Vision, Big Data, AI
	(J. Xu et al., 2018)	VR, Modeling and Simulation
	(Ran et al., 2021)	IoT , Computer Vision
Train Monitoring	(Biao Wang et al., 2020)	AI , Modeling and Simulation
	(L. Jin et al., 2017)	IoT
	(Brezulianu et al., 2020)	IoT
	(Bernal, Spiryagin, and Cole, 2018)	IoT
	(Malakar and Roy, 2018)	IoT, Modeling and Simulation
	(Fayyaz and Johnson, 2020)	IoT, AI, Computer Vision

Table 2.1: Monitoring studies and applied technologies.

systems depend significantly on accessing and ensuring the quality of data and merging data from various sources to derive relevant information for subsequent analysis . To date, these services have primarily been applied within the process and manufacturing industries. However, they evidently hold significant potential in other domains, such as the railway sector, given the complexity and vast amounts of high-quality data produced and recorded.

In railway OM, big data analytics will utilise cutting-edge technologies for predictive analytics. As part of OM services, big data is collected, analysed, visualised, and used for decision-making. In asset management, big data is used to address another typical weakness, that is, to predict the status of assets. In order to evaluate whether an asset is likely to fulfill its purpose, OM are based on predicting the remaining useful life (Galar, Seneviratne, and Kumar, 2018).

2.4 Internet of Things in Railway Condition Monitoring

The railway industry continues to face challenges with the monitoring and maintenance of its assets, which can lead to costly maintenance operations, safety risks to workers, and delays in train services (Pintelon, Nagarur, and Van Puyvelde, 1999, Carretero et al., 2003). One of the main problems is not having enough information about the condition of the asset to plan efficient maintenance (Network Rail, 2018). Additionally, data acquisition methods are risky because

rail operators have to access hazardous areas to check and repair assets (British Transport Police, 2018). New approaches are needed to guarantee safe working conditions for railway operators and reliable service delivery. The UK rail network had 1.718 billion journeys in 2016 and is expanding with projects like High Speed 2 (HS2). Trains must be punctual and reliable because there is a growing need for more transportation capacity (Office of Rail Regulation, 2013). Generally, the quality of the train services depends on the efficiency of the tracks, stations, power supply, communication systems, and signals (Stenström, Parida, and Galar, 2012; Faiz and Edirisinghe, 2009). To make sure trains run smoothly, it is important to know how well the train parts are working. This means regularly collecting information on their current condition and predicting what might happen in the future so that maintenance can be performed before something goes wrong (Lohman, Fortuin, and Wouters, 2004). Thus, there is a need for better ways to collect and manage data. This includes acquiring the data, putting them together in one place, analysing them and effectively managing them (Bilal et al., 2016).

2.4.1 Challenges in Rail Assets Maintenance

Network Rail is working on a rail technical strategy for 2019-2024, and has released challenge statements for innovation and improvement priorities[1]. Network Rail owns and operates Britain's main rail network, which includes numerous rail infrastructure elements, including tracks, tunnels, signals, bridges, and stations across England, Scotland, and Wales. Network Rail's Strategic Business Plan, known as CP6, seeks to guarantee trains operate safely, punctually, and effectively (Network Rail, 2018).

To understand the research requirements related to Network Rail's issues, 'Network Rail Challenge Statements' were examined as part of this study. As illustrated in Table 2.2, nine distinct problem areas were selected from four challenge statements to ensure broad coverage and avoid redundancy. These areas reveal a common shortcoming with current rail asset monitoring methods: the inability to provide precise and timely information on the condition of the asset.

[1] https://www.networkrail.co.uk/industry-and-commercial/research-development-and-technology/research-and-development-programme/challenge-statements/

This hampers prompt intervention when an asset is deteriorating and puts workers at risk. The use of digital solutions within IoT-based systems for the maintenance of rail assets shows great promise for revolutionising the maintenance and management of rail assets, and many researchers have explored this potential further.

Table 2.2: Limitations of current assets monitoring techniques (Reference: Network Rail).

Challenge Statement Domain	**Problem Area**	**Failure Causes**	**Specific Research Needs**	**Benefit**
Building and civils	Earthworks – Detection of asset failure by means other than train drivers.	• Subjective and sometimes unreliable datasets. • Late identification of failing assets. • Lack of holistic understanding.	• Novel techniques and cost-effective technologies to consistently acquire and store ground investigation data to better understand soil characteristics across the asset base.	• Reduction in schedule and costs by fixing before disruption and failure. • Better knowledge of assets with quantitative data across the entire geotechnical asset portfolio.
Electrical Power	Intelligent Assets and Condition Monitoring	• High maintenance cost. • Late mitigation of asset failure.	Development of new tools, techniques, equipment, and understanding to improve the reliability of Electrical and Power assets as part of the Contact System and Distribution Infrastructure.	• Improved timetable reliability and asset management. • Reduced maintenance cost and consumption of electrical and power assets.

Table 2.2: Limitations of current assets monitoring techniques (Reference: Network Rail).

Challenge Statement Domain	**Problem Area**	**Failure Causes**	**Specific Research Needs**	**Benefit**
Maintenance	Enabling Transition to Predict and Prevent Maintenance Regimes	• Condition monitoring data are not yet used to generate predictive and preventive maintenance regimes. • The present data lack the quality and detail necessary to fully implement predictive and preventive maintenance.	Provision of additional insight through the integration of condition monitoring and train-borne monitoring data.	Optimised maintenance costs, which could be increased if this provides a greater reduction in renewal costs.
Maintenance	Automating Inspection and Maintenance Activities to Remove Workforce from High-Risk Areas and Improved Data Capture.	Failure to intervene due to a lack of knowledge of the failure mode, data collection, and data analysis.	The development of a distributed sensor network enables data to be provided to modeling, analytical, and decision tools to support systems.	• Using decision support tools, schedule effective inspections with minimal disruption. • Increase in network capacity.
Lineside and track	Safe and Effective Lineside Inspections	• The current approach is paper-based. • Lack of full condition assessments.	• Future inspections should make use of remote techniques. • Evaluate different data sources and captures so the inspection regime is enhanced.	• Significant reduction in asset failure and reactive maintenance. • Decrease in financial costs to the business due to performance fines. • Reduction in safety risk.

Table 2.2: Limitations of current assets monitoring techniques (Reference: Network Rail).

Challenge Statement Domain	**Problem Area**	**Failure Causes**	**Specific Research Needs**	**Benefit**
Lineside and track	Lineside Asset Management	• Weakness has been identified in using Ellipse for work bank management and assets database. • Asset information is currently captured manually.	• Explore technologies to provide timely information. • Develop proactive measures.	• Access to an up-to-date asset record that can offer condition status and risk evaluation. • A unified source of truth where all information about the asset and the work performed on it is documented in a single location.
Lineside and track	Improved Application of Friction Management to Prevent Defects, Derailments & Extend Rail Life	Poor data management and lack of RCM.	Analyse the costs and benefits of improved friction management for each of the key stakeholders, including National Rail (NR), Department for Transport (DfT), RSSB, vehicle manufacturers and Freight train Operating Cmpaniess (FOCs)/Train Operating Companiess (TOCs).	Minimise the total lifecycle cost of assets, monitor the risk profile of track assets, and evaluate the risk profile of track workers.

Table 2.2: Limitations of current assets monitoring techniques (Reference: Network Rail).

Challenge Statement Domain	**Problem Area**	**Failure Causes**	**Specific Research Needs**	**Benefit**
Lineside and track	Rail Head Squats	• Apply corrective maintenance. • Poor analysis and understanding of the root causes of squat defects.	• Utilise track condition data to enhance the comprehension of rail squat development. • Implementing ultrasonic detectors on trains in service to capture a more comprehensive view of rail head squat progression.	• Assist in creating maintenance plans to reduce the expansion of squat. • Create proactive maintenance strategies that target the fundamental sources of squat defect formation in areas with high risk. • Support implementation of efficient and effective corrective maintenance.
Lineside and track	Re-Profiling Rail to Remove Defects & Extend Rail Life	Manual methods of Rail Contact Fatigue (RCF)	• Develop methods or systems to regulate conicity using data from diverse inspection methods.	• Increased rail life. • Condition-based management of rail profiles, resulting in more effective utilisation of rail profiling machinery and tools.

2.4.2 IoT-based Applications for Rail Condition Monitoring

IoT-based applications are made to automate data collection, interpretation, and physical asset control (Gubbi et al., 2013). Four levels of data and control flow between physical devices and IoT applications are commonly seen in an Internet of Things system, as shown in Figure 2.2 (J. Jin et al., 2014).

According to (X. Zheng, Z. Cai, and Y. Li, 2018; Gbadamosi et al., 2019), the hierarchical structure includes:

1. Edge/fog Information Technology (IT) and Operational Technology (OT) for processing, analysing and controlling data.

2. Devices such as sensors and actuators are used for collecting data and controlling assets.

3. IoT gateways for aggregating or segregating data.

4. Cloud-based IoT applications for managing data and control.

Hardware, software, networking, and communication components are combined and synchronised in IoT systems to deliver immediate insights and management of physical assets (Kochovski and Stankovski, 2018). In an industrial manufacturing sector, IoT has been used to give consumers context-aware information (Alexopoulos, Makris, et al., 2016). Aleopoulos et al. came to the conclusion that layered structures, event-driven systems, and context-aware methodologies are some of the commonalities among industrial IoT system architectures (Alexopoulos, Sipsas, et al., 2018).

The capability to gather prompt data on the state of assets and support or automate the control, repair, or replacement of those assets' components is a key objective to achieve optimal maintenance of rail assets. This objective perfectly aligns with the fundamental capabilities of IoT, which can provide connectivity to a large number of device components, including sensors, actuators, and user tools for the administration, control, and monitoring of rail assets (I. Lee and K. Lee, 2015).

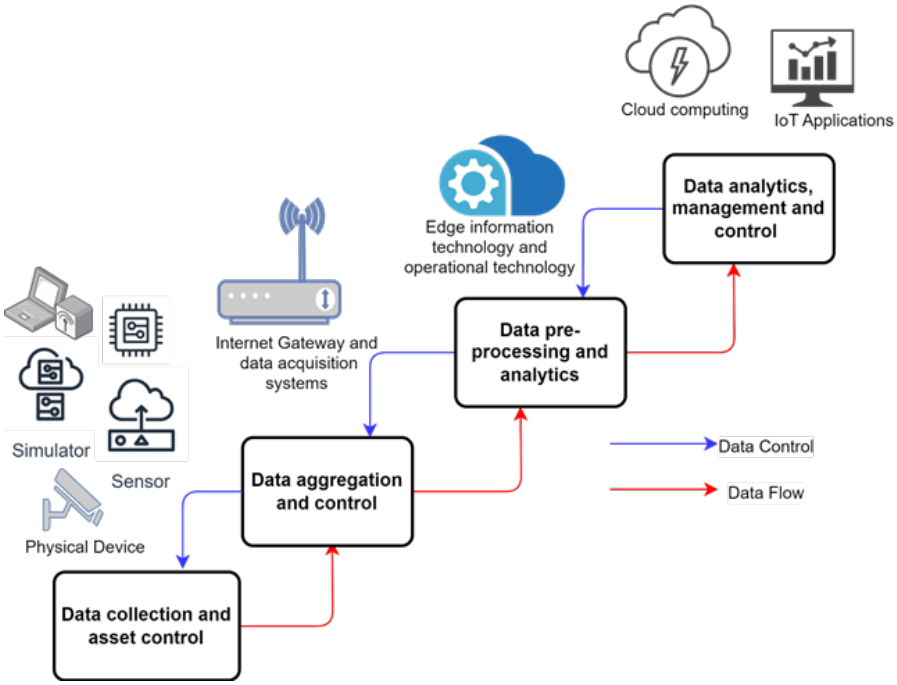

Figure 2.2: IoT Application data and control flow

There are numerous present and potential applications for IoT in different areas of maintenance of rail assets. However, more research and development is required to address previously identified issues in the maintenance of rail assets, and the implementation of IoT remains the most suitable solution.

2.5 Big Data in Railway Condition Monitoring

Previously, Zikopoulos and Eaton introduced a big data model based on three key aspects: volume, variety, and velocity (Zikopoulos and Eaton, 2011). Subsequently, Lomotey and Deters expanded the 3Vs model into the 5Vs model by including value and veracity (Lomotey and Deters, 2013). The 5Vs are applicable for characterising the data produced in asset management. A high rate of samples per second collected from each measurement point using sensors like accelerometers or audio sensors exemplifies velocity. A substantial amount of data is produced because of the many measurement points and this indicates the volume of data. Some maintenance-related data, including free text comments for completed maintenance actions or failure reports, are structured, while others are not. Moreover, the forms of the data come from

many systems. This is where a variety of data for asset management is sourced.

When effectively utilised in asset management, this data can be highly valued, provided that its accuracy and uncertainty are thoroughly assessed and managed. Furthermore, it's critical to comprehend the value of data, particularly how it can increase efficiency and effectiveness in maintenance management, such as through better decision-making. Choosing the most cost-effective approach for data processing is equally important.

Large-scale asset data can be employed for data mining to uncover new patterns and connections that are not immediately apparent. With the advent of the big data approach, maintenance Decision Support Systems (DSSs) are now able to integrate contextual data (Galar, Thaduri, et al., 2015). Underlying causes of failures are an example of valuable insights that can be discovered. These insights can inform better design and better maintenance planning.

Successful maintenance decision-making requires a robust DSS grounding in knowledge discovery. As depicted in Figure 2.3, the primary stages of the knowledge discovery process include data acquisition, which includes the collection of pertinent data and its content management; data transition; data fusion; data analysis and mining; and information extraction and visualisation, which involves the merging of data and information from multiple sources to aid maintenance decisions. Figure 2.3 demonstrates how data fusion, big data analytics, and context sensing can be employed to derive real-time insights and solutions for maintenance challenges.

Figure 2.3: Overview of maintenance information workflow.

Data fusion is the process of combining data from various sensors with knowledge from other sources to draw conclusions (Nichols, 2017). Data fusion becomes essential when handling information from a variety of sensors or different sources. When used for maintenance decision support, knowledge discovery integrates data mining and knowledge discovery using the eMaintenance paradigm (J. Lee, 2003). As shown in Figure 2.3, getting the appropriate judgement for the context sensing is essential. Nevertheless, the implementation of eMaintenance in industrial applications is confronted with challenges categorised as follows: Organisational, architectural,

infrastructural, content and contextual, and integrative factors make up the first four categories (Karim et al., 2016). In fact, big data in the OM sector uses cutting-edge systems for prognostic analytics and offers decisions that are feasible.

2.5.1 Big Data Techniques Applied to Railway Condition Monitoring

The goals of big data in railways are primarily anticipated to facilitate the adoption of Big Data technologies in the areas of predictive algorithms from diverse data sources, scalable data architectures, real-time communications, and visualisation methods. The software tools that are already on hand to help with asset monitoring within the railway system are described in this section. Currently, there are two groups of software tools available to help maintain the railway system. On the one hand, there are software tools provided by rolling stock manufacturers, who are more often in charge of the management and maintenance of the vehicles, and on the other, there are software tools for infrastructure managers.

The first family may include tools from prominent European firms, specifically ALSTOM and SIEMENS, the primary providers in the European rail sector (Galar, Seneviratne, and Kumar, 2018). Below are some of the areas they supported:

- Information management (such as scheduling tasks for corrective and proactive maintenance, staff, spare parts, and paperwork).
- Condition-based monitoring, which uses vehicle parameters and radio transmission to a server on the ground. Unfortunately, there is relatively little information to track and just a small number of parameters with little predictive power. Typically, the metrics are more performance indicators than alarms.
- Predictive data analysis for maintenance (HealthHub), which uses developed algorithms for forecasting future component performance, employs sophisticated data analytics to track asset health, estimates how long they will last, and replaces assets only when necessary. The algorithms and formulas are severely constrained by the volume of available data, and only straightforward statistical analyses are offered.

- Algorithms for optimising and improving processes to lower the price of component lifecycles. Due to their dependence on inadequately accurate predictive algorithms, optimisation is severely constrained.

For the second family, several IT solutions companies offer general-purpose solutions that are often tailored for the infrastructure operator and include a variety of supported areas, as detailed in (Galar, Seneviratne, and Kumar, 2018), like:

- Information management, including task scheduling for corrective and preventive maintenance, staff, spare parts, and paperwork.

- Online and offline condition monitoring. Unfortunately, there is relatively little information to track and just a small number of factors with little predictive power. The parameters typically function more as alarms than performance indicators.

- Visualisation of the environment for mobile devices and online technologies. The quality of the visualisation primarily depends on the accuracy and availability of the data.

Given that the evaluated information sources are relatively small (a few hundred megabytes) and pertain only to fleet or infrastructure, existing software tools for maintaining railway systems do not qualify as 'Big Data technology' (Galar, Seneviratne, and Kumar, 2018). Consequently, prediction algorithms lack precision and scalability, leaving room for improvement in optimisation capabilities.

2.5.2 Expectations and Challenges of Big Data for Railway Condition Monitoring

Referring to Figure 2.3 in Section 2.5, at data acquistion stage, the data to be collected may include unstructured data from heterogeneous sources managed through big data analytics engines such as HADOOP, combined with a NoSQL database. During the data analysis phase, methods such as regression analysis, machine learning, deep learning, Bayesian inference, and

data mining may be employed. The choice of techniques is highly dependent on the attributes of the data as well as the systems used for data acquisition, management, and processing.

In the railway sector, these data acquisition and analysis processes are essential for enabling the broader integration of big data technologies across various domains. The main objectives of integrating big data in the railway sector are to enhance predictive algorithms, develop scalable data frameworks, improve real-time communication, and refine visual representation techniques. The achievement of these goals poses significant challenges in the maintenance of infrastructure assets, particularly within the complex railway environment. In this context, there are three main focus areas: predicting the wear and tear of railway system components, estimating the maintenance expenses for railway infrastructure and vehicles, and tracking the condition of infrastructure and vehicles (Binder, Mezhuyev, and Tschandl, 2023). To achieve these objectives, several advanced techniques and systems must be implemented in the railway sector. According to (Galar, Seneviratne, and Kumar, 2018), these objectives involve the following:

- Real-time prediction algorithms capable of managing multiple data sources while ensuring privacy throughout processing. These algorithms entail feature and instance selection, transforming continuous data into discrete forms, data compression, employing ensemble classifiers and regression techniques, and synchronising data geographically and chronologically.

- Flexible data structures that utilise a virtualisation layer to connect data acquisition and data analytics. Additionally, it involves the incorporation of advanced database features in order to integrate various types of data sources within a high-performing cloud-based system.

- The implementation of open interface gateways equipped with monitoring systems providing timestamp and position synchronisation, versatile communication support (such as mobility and aggregation), and priority protocols is crucial for enabling real-time data transmission, which is essential for Big Data Communications.

- The employment and development of applications for visualisation techniques including infographics and virtual/augmented reality.

Together, these advancements aim to enhance the efficiency and effectiveness of railway infrastructure management, enabling smarter and more predictive maintenance practices.

2.6 Data Monetisation and Micropayment

Big data is generally distributed across various locations and is held by multiple stakeholders. To efficiently manage this data and maximise its value in different sectors, data owners should be encouraged to share their data with others for analysis and integration with other datasets. Data from different resources, if used together, could be more useful. For many enterprises involved in data processing and analytics, particularly new ventures and researchers in AI and big data fields, possessing valuable datasets is crucial for conducting experiments and performing statistical analysis. Data exchange allows for the effective utilisation of data assets and improves the importance of data. This can be done through data owners selling their data to generate profit. Additionally, data consumers can overcome data collection issues and fully exploit data resources. However, ownership of these data is the main issue, as disclosing the data to others may not be desirable for many data owners for several reasons. This hesitation in providing data between stakeholders leads to an isolated data island problem.

In particular, the railway sector exemplifies how the exchange of IoT data or historical data can be implemented at low costs. The production costs of these data are often not affected by the number of consumers, so producers are often willing to charge very small amounts for their offerings if the payment system allows for it. However, credit card payments, which are commonly used for online purchases, involve a significant minimum fee per transaction, such as 20 cents, and are not suitable for charging smaller amounts. This highlights the need for micropayments, which are defined as financial transactions that are smaller or close to the minimum credit card transaction fees (Herzberg, 2003) or any financial transaction under USD 10 (W. Wang et al., 2023).

Despite the potential for low-cost exchanges, current data exchange methods are primarily conducted privately among data providers and consumers, presenting numerous limitations. This is evident from the agreement templates suggested in project T1010 (Sparkrail, 2016); see Appendix A. Firstly, the absence of an open trading platform necessitates multiple steps and negotiations to finalise a transaction. Initially, the data user (buyer) must make numerous inquiries to locate a reliable data owner (provider) and establish contact. Furthermore, the provider often lacks sales opportunities, requiring investment in manpower and resources to promote data sales. Consequently, both parties incur extra costs to complete a transaction, while an open trading platform could minimise pre-transaction costs and reduce resource waste. Secondly, the current transactions do not guarantee a high level of security and reliability. Private trading is inherently insecure, and in the event of malice from either party, a trusted third party must intervene to handle subsequent complaints. Moreover, the post-transaction time, resource, and economic costs pose hidden threats to this trade form. Thirdly, the consumer often desires to exchange the entire dataset in one transaction, forcing the provider to opt for a single transaction due to storage difficulties. This means that the consumer gains full rights to use and process the data, making it challenging to prevent the re-trading of pre-owned data. As a result, private data trading is highly risky for all data providers and consumers.

Due to technological advancements and societal demands, various data trading platforms have emerged online[2]. Regardless of their openness level, these platforms are all centralised, raising concerns about the protection of the data being sold and the justice of the transactions, as they rely entirely on potentially untrustworthy platforms. Consequently, there remains a need for an open and reliable trading platform.

One of the key contributions of this research is the development of a fair cost attribution model among parties engaged in exchanging condition monitoring data. To achieve this, Blockchain technology, which will be discussed in Chapter 3, is utilised to remove the need for TTP. Most studies that have created data marketplaces and eliminated TTP have primarily focused on weather, data privacy, and integrity, as discussed in 2.6.1, or payment models, as discussed in

[2]https://www.bdex.com/

2.6.2.

2.6.1 Fair Data Exchange

In 1997, Asokan et al. put forward fair exchange protocols that did not require a third party under any circumstances (Asokan, Schunter, and Waidner, 1997). These protocols enhanced efficiency by eliminating the constant involvement of a third party. In 1998, Asokan et al. introduced a novel protocol known as "optimistic fair exchange," where digital signatures or encrypted data are exchanged among two individuals (Asokan, Shoup, and Waidner, 1998). This approach relied exclusively on the TTP when one party failed to adhere to the rules.

Further advancing the field, Avoine and Vaudenay in (Avoine and Vaudenay, 2004) developed a pioneering fair exchange protocol that also did not involve a third party, using the verifiable secret sharing mechanism within decentralised environments, which laid the groundwork for fair exchange in the Bitcoin network (Nakamoto, 2008).

Using Bitcoin, Bentov and Kumaresan in (Bentov and Kumaresan, 2014) designed additional decentralised fair exchange protocols. Zyskind et al. introduced a decentralised system for managing personal data using Blockchain technology (Zyskind, Nathan, et al., 2015). The purpose is to safeguard extensive amounts of personal information from being gathered and controlled by external entities. In this system, Blockchain serves as an access-control manager that operates independently without the intervention of a third party or its protocols.

In a related vein, Vu et al. utilised Blockchain technology to enhance users' overall experience and decrease the expenses associated with delivering content in content delivery networks (Vu, Chatzinotas, and Ottersten, 2019). This application further illustrates the versatility of blockchain solutions across various domains.

Additionally, a different data-sharing mechanism is proposed in (Desai et al., 2018), where data hash values are encrypted with a symmetrical key and stored in a secure location off-chain by the data provider before the transaction is processed. This innovative approach allows all providers in the cloud to promote their data services and public keys. To enable consumers to gain one-time access to the appropriate records, smart contracts were generated on the fly, and

activity was logged on the chain to be used in the resolution of any potential disputes.

Further emphasising the importance of privacy in data transactions, Agora is a data marketplace introduced in (Koutsos et al., 2022). This marketplace prioritises data privacy, result verification, and secure payments by incorporating a framework that includes data generators, brokers, and clients. In this model, brokers remunerate data generators for their contributions, process the data, and then trade the results to data clients. By employing encryption, the marketplace guarantees that data generators cannot access any information about the transmitted data. Furthermore, by using zero-knowledge proofs (ZKP), the marketplace guarantees that data generators cannot tamper with the data and allows consumers to validate the results they receive. Although this solution covers key elements like data privacy, it falls short in adapting to dynamic changes among data generators and mainly emphasises privacy aspects in data marketplaces.

2.6.2 Payments Models Towards Fair Costs Distribution

There are two common trading models on any trading site: post-paid and pre-paid. The former suggests a trust in the consumer (buyer) that payment will be made as agreed after the data is obtained correctly. Conversely, the pre-paid model indicates a trust in the provider that the data will be submitted afterward if the payment is made as agreed. However, neither model guarantees total satisfaction for both parties, and both bear some risk in cases where one party may misbehave. Consequently, there is a requirement for a TTP to provide both the provider and the consumer with an escrow service.

An example of pre-paid model is a subscription-based model for trading data on cloud platforms, introduced by Al-Zahrani (F. A. Alzahrani, 2020). In this proposed model, the ledger tracks all subscriptions and orders, including those requests that have not been concluded and finalised. Additionally, the authors of IDMoB in (Özyilmaz, Doğan, and Yurdakul, 2018) suggest a Blockchain-based IoT data marketplace. In this system, devices transfer their datasets to Swarm, a decentralised system for file exchange. Consumers can search for specific sensor data by querying a smart contract and make payments through payment channels. Uploaded IoT data is encoded using a symmetric key to block unapproved access. Once a dataset is purchased,

the buyer and seller share the symmetric key, and the buyer can access the data using the given file identifier. Users can also rate the data sources through a voting mechanism. However, this proposal fails to address the sale of real-time data, highlighting a gap in current models.

To address such gaps, the literature presents various examples that utilise Blockchains as an escrow to eliminate the need for a TTP. This approach ensures that the consumer's payment is held until the data is submitted, and the consumer agrees that it has been properly processed. For example, a study by Meijers et al. introduced a Blockchain-based IoT data trading system that proposed an enhanced data trading strategy to minimise transactional costs by reducing the number of executed transactions and thus minimising smart contract execution fees (Meijers et al., 2021). Through off-chain agreements, consumers and producers establish trade conditions. To acknowledge data receipt, consumers must provide a receipt to the provider, who can then submit a receipt to a smart contract acting as an escrow that holds the payment for the exchange. Although the authors primarily aimed to present a Blockchain-based IoT data exchange system, they acknowledged the need for additional functionalities to transform it into a fully functional data marketplace.

Further expanding on this theme, Sober et al. in (Sober et al., 2023) created and operated a data marketplace that utilises the Blockchain and IoT. The marketplace follows a three-layer structure and utilises smart contracts to enforce marketplace rules. By using a proxy, sellers and buyers can easily integrate IoT devices, even with limited resources. Moreover, a broker assists in facilitating data trading, managing tasks that require significant resources, and handling conflict resolution if sellers and buyers cannot reach an agreement during the negotiation phase. They also examined the costs associated with using smart contracts, discussing challenges that arose during implementation. Specifically, the costs related to data trading cover both the execution of smart contracts and storage fees, given that they implemented their application on Ethereum, which demonstrates considerably high costs, particularly for the data provider. Notably, their design lacks a mechanism for trustless dispute resolution, indicating an area for future improvement. Notably, their design lacks a mechanism for trustless dispute resolution, indicating an area for future improvement.

2.7 Conclusion

In conclusion, the railway industry, characterised by its intricate interplay of technology, operations, and human involvement, faces significant challenges in enhancing operational efficiency and safety. Industry 4.0 technologies, particularly IoT and Big Data, present promising avenues to revolutionise rail asset monitoring and maintenance practices. By integrating real-time data collection, predictive analytics, and advanced monitoring solutions, the sector can achieve increased reliability, reduced maintenance costs, and improved safety measures.

However, the effective implementation of these technologies hinges upon overcoming existing challenges, such as data management, predictive maintenance capabilities, and fair data exchange mechanisms. The proposed frameworks for data monetisation and micropayment can facilitate the seamless sharing of information, fostering collaborative innovation throughout the industry.

Ultimately, as the railway sector continues to evolve, embracing these advances will not only enhance operational efficiency but also secure a sustainable future in the transportation domain. Continuous investment in research and development, coupled with regulatory support, will be crucial in realising the full potential of smart railways, paving the way for a transformative impact on both service delivery and customer satisfaction.

Chapter 3

Blockchain Technology Background and Applications

3.1 Introduction

From a technical standpoint, this chapter introduces Blockchain as one of the DLTs. Section 3.2 outlines the initiatives that came before the advent of Blockchain, and then explores the key characteristics of DLTs in Section 3.3 to delve into the discussions surrounding them. Section 3.4 discusses decentralisation, ledger, consensus algorithms, and smart contracts as core concepts in Blockchain. In Section 3.5, Hyperledger Fabric components are introduced to make it clear why this platform is chosen to build the test-bed framework as detailed in Section 3.6. Grasping the essential features of Blockchain will enable us to understand the present and forthcoming potential of this technology within the industrial sector. Consequently, Section 3.7 gives a summary of Blockchain applications across different industries. Section 3.8 introduces certain initiatives within Shift2Rail that have employed Blockchain technology to develop novel solutions in the railway sector. Finally, the chapter will be concluded in Section 3.9.

3.2 Historical Timeline of Blockchain

The necessity to minimise dependence on trust models that rely on centralised authorities or external mechanisms spurred the creation of the Blockchain framework (Fernández-Caramés and Fraga-Lamas, 2018). The fundamental ideas of Blockchain can be linked to an initial proposal by Haber and Stornetta (Haber and Stornetta, 1991) that tackles the dependability of centralised time-stamping services. Their proposal was driven by common trust issues such as collusion, integrity, and forgery. Their research laid the foundation for a tamper-resistant data structure that connects the hashes of records in a sequence that only allows for appending, even though it was not referred to as "Blockchain" at the time. Despite Bitcoin being the most famous crypto today, some cryptos appeared early and represent important steps in the evolution of digital currency but embody different approaches and philosophies. An example of such cryptos are:

- **Digicash:** Known as the inventor of electronic money, David Chaum introduced the essential concepts that underlie encrypted messaging tools. This idea was presented in his 1983 article, Blind Signatures for Untraceable Payments (Chaum, 1983; Chaum, 1985). He launched a business named DigiCash in 1990, maybe the first time digital money was developed for use primarily electronically. The business introduced the first-ever electronic cash transaction over the Internet in 1994 (*DigiCash - company brochure* 1997).

  Digicash differs from Bitcoin in several ways :

  1. **Anonymity:** Bitcoin is not anonymous; it is pseudonymous. Although the owner of a Bitcoin wallet isn't publicly known, a Bitcoin address's transaction history is visible on the public Blockchain and may be tracked. On the other side, DigiCash provided its users with high levels of anonymity. It made use of a cryptographic mechanism to enable money transactions without disclosing the identity of the payer.
  2. **Creation and Distribution:** Bitcoin is created or "mined" by a decentralised process in which miners solve challenging mathematical puzzles to validate transactions and

add them to the Blockchain. They are awarded with new bitcoins in return. There was no comparable method involved in the development of DigiCash. DigiCash was produced and distributed by DigiCash Inc. as a centralised system.

3. **Open Source:** Anyone may see, check, or alter the source code for Bitcoin. As a result, there have been several Bitcoin "forks" in which programmers have produced new cryptocurrencies based on the original code of Bitcoin. The DigiCash system was not open source; instead, it was a proprietary system managed by DigiCash Inc.

- **B-Money:** B-money was more a theoretical proposal by Wei Dai in 1998 and was never implemented as a functioning system (Dai, 1998). The original B-Money proposal included a mechanism for creating money, where each participant would determine the amount of money in their own account, but it lacked the specific mechanism for consensus that Bitcoin's proof-of-work provided. The use of a "ledger" was also mentioned in the B-money proposal, but did not specify the Blockchain data structure that is now synonymous with Bitcoin and many other cryptocurrencies (Wirdum, 2018).

Bitcoin (Nakamoto, 2008) was the first to successfully implement the concept of a Blockchain. The following year, this concept evolved into Bitcoin, a decentralised digital currency designed to remove the necessity for reliable financial intermediaries. Subsequently, Ethereum was developed as a decentralised platform for applications, extending Blockchain's applications beyond cryptocurrencies and introducing the idea of smart contracts (Wood, 2014; Buterin, 2014). This type of Blockchain infrastructure allows various sectors and applications to harness the benefits of Blockchain including decentralisation, permanence, transparency, data integrity, removal of central authorities, and a reliable consensus process (Seebacher and Schüritz, 2017; Christidis and Devetsikiotis, 2016a; Dinh et al., 2018). Since that point, numerous Blockchain-based applications have emerged in various fields including healthcare, insurance, energy, and transportation among others (Fernández-Caramés and Fraga-Lamas, 2018; W. Cai et al., 2018). Following this trend, Hyperledger Fabric introduced a modular, permissioned Blockchain platform tailored to meet business requirements (Androulaki et al., 2018).

3.2.1 Cryptographic Hash Functions

A cryptographic hash function is a mathematical algorithm designed to map an input of arbitrary length into a fixed-size bit string, often referred to as a hash or message digest. For such a function to be considered cryptographically secure, it must exhibit a set of essential properties. First, it must be *deterministic*, ensuring that the same input always produces the same output. The function should also support *efficient computation*, allowing rapid evaluation even for large data sets. A crucial aspect is the *one-way property*, which makes it computationally infeasible to reconstruct the original input from the hash value. Moreover, a cryptographic hash function must exhibit the *avalanche effect*, whereby even a minimal change in the input leads to a significantly different output. Lastly, *collision resistance* is fundamental, meaning that it should be computationally difficult to find two distinct inputs that produce the same hash output.

Due to these properties, cryptographic hash functions are foundational tools in various security protocols and systems. They are extensively used in the generation and verification of digital signatures, the construction of Message Authentication Codes (MACs), and the implementation of authentication mechanisms. Furthermore, they play a vital role in ensuring file integrity and are utilised in checksums and data fingerprinting techniques to detect and prevent tampering or data corruption (M. R. Anwar, Apriani, and Adianita, 2021; Madhuravani and Murthy, 2013).

3.2.2 The Merkle-Damgård Construction

The Merkle-Damgård construction, introduced by Merkle and Damgård in 1989, serves as the underlying structural design for several well-known hash functions, including Message Digest 5 (MD5), Secure Hash Algorithm 1 (SHA-1), and SHA-2. This construction utilises an iterative compression function to process input data. Specifically, the input message is first divided into fixed-length blocks. Each block is then processed sequentially using the compression function, beginning with an Initialization Vector (IV) that serves as the input for the first block. The output of each iteration is used as input for the next, allowing the construction to preserve the chaining of the data through all message blocks. Once the final block is processed, the output

of the last iteration yields the final hash value. The strength and flexibility of this construction contributed to its widespread adoption in early cryptographic hash function designs (Zellagui, Hadj-Said, and Ali-Pacha, 2019; AlOdat and S. Khan, 2019).

3.2.3 Evolution and Characteristics of Prominent Hash Algorithms

The MD5 algorithm was developed by Ronald Rivest in 1992 and produces a 128-bit hash value. At the time of its creation, MD5 was widely adopted for verifying data integrity due to its speed and simplicity. However, its security has since been compromised. The algorithm is now considered cryptographically broken, primarily due to its vulnerability to collision attacks and birthday attacks. As a result, the National Institute of Standards and Technology (NIST) has formally deprecated MD5 for use in security-sensitive applications, especially for password storage and digital authentication (Debnath, Chattopadhyay, and Dutta, 2017).

In response to the shortcomings of MD5, the SHA-1 was introduced in 1995 as part of the U.S. Government's Capstone project. SHA-1 generates a 160-bit hash value and incorporates structural improvements over MD5, offering enhanced collision resistance at the time. It was widely deployed in cryptographic systems such as TLS, SSL, SSH, and IPsec, serving as a core component in securing communications. Despite these enhancements, advances in cryptanalysis eventually demonstrated the feasibility of successful collision attacks against SHA-1 as well. Consequently, NIST officially retired SHA-1 from active use in secure applications as of December 31, 2013, recommending migration to more secure alternatives such as SHA-2 (AlOdat, Abbas, and S. U. Khan, 2019).

The SHA-2 family, designed by the National Security Agency (NSA) in 2001, was developed to address the vulnerabilities identified in earlier algorithms. This family encompasses multiple variants, including SHA-224, SHA-256, SHA-384, SHA-512, and their truncated forms. Each variant differs in output length and internal structure, offering a broad range of cryptographic strength tailored to different security needs. SHA-2 algorithms provide significantly greater collision resistance and remain robust against current cryptanalytic techniques. They are integral to widely used protocols such as SSL, TLS, S/MIME, and SSH and have also been adopted in

blockchain technologies, including Bitcoin, where they support transaction validation. Due to their ongoing security and adaptability, SHA-2 functions continue to be the preferred choice in modern cryptographic applications (Martino and Cilardo, 2019).

To further strengthen the cryptographic landscape and provide an alternative to the Merkle-Damgård construction, the SHA-3 algorithm was introduced. Developed by Guido Bertoni and colleagues, SHA-3 is based on the Keccak algorithm and was standardised as FIPS 202 in 2015. Unlike its predecessors, SHA-3 adopts a sponge construction, which absorbs input data and squeezes out the final hash value through an internal permutation mechanism. This architecture provides enhanced resistance to a variety of attacks, including length extension and differential cryptanalysis. SHA-3 is also well-suited for hardware-based implementations, offering flexibility and efficiency in constrained environments. While not intended to replace SHA-2 immediately, SHA-3 serves as a future-proof solution, ensuring long-term cryptographic resilience (Sharma and Mittal, 2019).

3.3 Distributed Ledger Technologies Characteristics

DLT such as Blockchain enables parties to exchange digital data on a peer-to-peer basis without relying on third parties or intermediaries, regardless of their geographical distance or the degree to which they trust each other. Data represents anything that can be transformed into a digital form, including money, insurance agreements, contracts, land ownership, healthcare information, and personal identification certificates. For clarity, Blockchain is referred to as a member belonging to the larger community of DLTs (Natarajan, Krause, and Gradstein, 2017). Through the use of certain types of databases called DLTs, data can be stored, shared, and synchronised across a distributed network of devices. A specific subset of DLT known as Blockchain technologies uses cryptographic methods to maintain storage and synchronisation of data within "chains of blocks." The distinction relates to how network participants disseminate, validate, and record data. In other words, all Blockchains are DLTs, but not all DLTs are Blockchains. Using Blockchain technology, nodes are connected to create a distributed database that records

data transactions chronologically (Wright and De Filippi, 2015). Because of the unique method transactions are stored and validated, it is known as a "Blockchain".The 'Blocks' are units of information that are organised and encrypted to represent a set number of transactions; further details on this will be provided in 3.4.2. In order for a new block of data to be validated across the network, a majority of nodes (depending on the consensus protocol) must agree. As illustrated in 3.4.3, one of the most well-known cryptographic techniques for reaching a consensus is Proof-of-Work, which involves solving a challenging mathematical equation by a node or computer (a "miner") and confirming its result by other participants as part of the network. The entire procedure makes sure that every block is assembled so that it evidently connects to the block before it and to the block after it, producing a "chain of blocks". Throughout the network, each node contributes a unique record to the Blockchain, which is updated and synchronised continuously.

A Blockchain functions in the sense of a ledger or database that maintains and authenticates records of every transaction ever carried out through the network. In view of the fact that the system is constantly updated and validated, it would be very difficult for unauthorised changes to take place without anyone noticing these changes and tampering with it. Additionally, in public Blockchains, all transactions are available for validation and inspection at any time by anyone. While in private Blockchains, only authorised parties may access this information. Cryptographic signatures or public-private keys guarantee that access is secured and protected. Since there is no single point of failure, it is theoretically more resistant to outages or cyberattacks. Given the number and distribution of nodes, it is very challenging to attack the majority at once or bring down the entire network.

There is no 'new' technology behind Blockchain; rather, it is a hybrid of several previously developed technologies, including peer-to-peer networks, cryptographic methods, consensus protocols, and distributed storage of data (Narayanan and Clark, 2017). As mentioned above in 3.2, this combination was first created in the decentralised cryptocurrency known as Bitcoin, which Satoshi Nakamoto first introduced in 2008 (Nakamoto, 2008).

It is still difficult to escape the connection between Blockchain and Bitcoin and problems

such as fraud, tax evasion, money laundering, and other illegal activities. Despite the debates surrounding Bitcoin and other cryptocurrencies, Blockchain has recently gained attention for several important features or properties (Tasca and Tessone, 2017):

- **Decentralisation:** As detailed in 3.4.1, a distributed network lacks a centralised authority for transactions and operates with participants who are unfamiliar with one another. For transactions to be added to the Blockchain and to be verified and validated, a consensus mechanism is required. Among these is Proof-of-Work, which uses the computing power of nodes to solve challenging mathematical problems. Other mechanisms are also being developed, such as Proof-of-Stake. The algorithms for consensus are detailed in 3.4.3.

- **Redundancy:** The Blockchain is made to withstand failures, interruptions, and interferences by ensuring that multiple updated and verified copies are present throughout the network. Unlike centralised systems with a single point of failure, this feature prevents nodes from becoming disconnected, from experiencing hardware failures, or from experiencing power outages.

- **Transparency:** All participants in a Blockchain network can access a ledger, or certain participants may only have access to certain predefined sets of ledgers. Transparent transactions boost network auditability and trust, and consensus mechanisms ensure that all Internet connections are treated equally.

- **Timestamping:** As part of the Blockchain-based transactional system, the timestamp serves as a public and irrevocable link between a specific date and time and the data associated with it. It is helpful for tracking transactions and ensuring data existence because this secure tracking and verification feature enables parties to verify information and transactions.

- **Immutability:** Blockchain transactions are tamper-proof and irreversible because they are cryptographically recorded and confirmed through consensus. A unique, historical version that is shared by all participants is ensured by features like immutability, non-repudiation, and non-forgery.

- **Digital Signatures:** The integrity and authenticity of transactions and data transfers are ensured by the use of cryptography with public and private keys in a Blockchain. Based on public and private keys, each participant has a unique identity. Only recipients with access to the relevant public key can open encrypted messages or transactions, and only a specific recipient can decrypt them using their private key.

- **Automation and Smart Contracts:** Blockchain technology automates transactions without requiring human coordination, resolving any conflicts and guaranteeing that there is only one legitimate transaction. It also acts as the foundational layer for smart contracts, which automatically execute and enforce agreements in accordance with if-then logic, recording and validating agreements (Szabo, 1997); (Buterin, 2014). These agreements involve trade-offs with associated benefits and restrictions.

3.3.1 Permissionless and Permissioned Blockchain:

Blockchains that don't require permission, like those used by Bitcoin, Ethereum, and Litecoin, let anyone participate. These distributed ledgers use consensus mechanisms for sending transactions, accessing records, and validating them. In permissionless blockchains using Proof-of-Work consensus algorithm, cryptocurrency (such as Bitcoin) is often generated as a reward for miners who validate transactions and secure the network. (S. Anwar et al., 2020).

Blockchains with permissions, like Ripple, Chain, and Hyperledger, are distributed databases where participants can be preselected or given access by an organisation, consortium, or central administrator. So only a select group of participants have the ability to read, modify, and access the Blockchain, often resulting in a more centralised governance structure compared to permissionless blockchains. These private and trusted parties or intermediaries ensure network consensus or the upkeep of the Blockchain according to a predetermined set of rules. With this configuration, the ledger can be operated and protected without the use of native currencies or their associated proof-of-work system. It is important to understand that permissioned Blockchains can be broadly classified into two types: consortium Blockchains, such as R3 or Corda, in which consensus is carried out by a predetermined group of organisations; and private

Blockchains, such as MONAX or Multichain, for primarily internal use within an organisation.

The precise characteristics that set permissionless Blockchains apart from permissioned Blockchains are a topic of disagreement among technologists. Currently, they are both evolving as spectrums, where hybrids can combine the best features of both (Danezis and Meiklejohn, 2015). Businesses or organisations may be able to create permissioned networks on top of Blockchain platforms like Ethereum in some circumstances. Hybrid blockchains, such as Dragonchain, combine features of both permissioned and permissionless models, allowing businesses to operate within a controlled environment while still leveraging some of the transparency and immutability of permissionless blockchains. Figure 3.1 provides an overview of the four categories of Blockchain networks.

3.3.2 Scalability and Performance:

For the overall deployment of Blockchain technology, the controversy over permissioned versus permissionless Blockchains is crucial. Due to the fact that all nodes in a distributed network must verify and send transactions across a network, the number of transactions that can be performed on a permissionless or public Blockchain is limited. The Bitcoin network, for example, verifies about 300,000 transactions per day, with each transaction verified every 10 minutes. On average, Visa's electronic payment processing network processes 150 million transactions per day. However, scaling up these Blockchains is not straightforward. Due to the way their architecture was originally intended, public Blockchains have an inherent limit on the number of transactions and the amount of data they can contain in a given 'block'. In the past, there have been fierce debates over possible solutions regarding block size expansion (in Bitcoin, the maximum block size is 1MB) or introducing sidechain or off-chain protocols (Nabilou, 2019). The debate over Bitcoin's block size culminated in the 2017 hard fork that created Bitcoin Cash, which adopted a larger block size to increase transaction throughput. Several alternative mechanisms, such as sharding and proof-of-stake, are being developed to improve the scalability of public blockchains by reducing the burden on individual nodes for storing and validating transactions (Zamani, Movahedi, and Raykova, 2018; Durand, Anceaume,

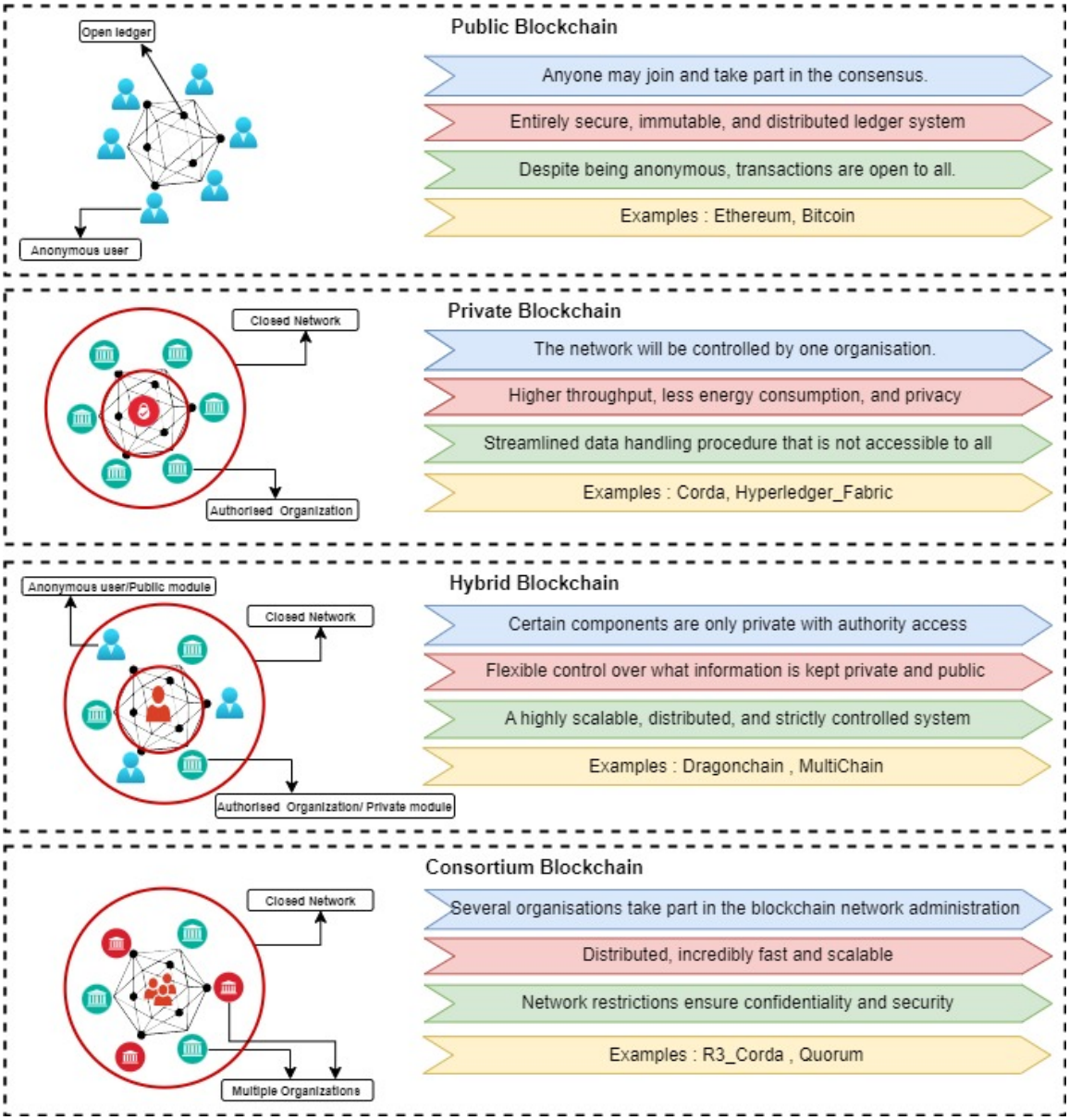

Figure 3.1: Blockchain network types.

and Ludinard, 2019). While permissioned Blockchains may face some scalability limitations, they are generally better suited to address these issues more efficiently. Their restricted and predefined set of participants allows specific nodes to allocate more computing power, resulting in faster transaction processing. Additionally, consensus mechanisms like Practical Byzantine Fault Tolerance (PBFT) or Raft, which limit the number of validating nodes, further enhance scalability by speeding up transaction validation.

3.3.3 Energy Consumption

Public Blockchains that rely on Proof-of-Work, as introduced by Nakamoto in 2008 (Nakamoto, 2008), have come under scrutiny due to their significant energy consumption requirements (Bada et al., 2021). Concerns have grown, particularly over the past few years, as the rising value of Bitcoin has led to increased network usage and intensified the mining process needed to validate transactions. As more participants engage in mining, rewards have diminished, prompting the use of more advanced computational techniques, including application-specific integrated circuits (ASICs), cloud mining, and mining pools, where miners combine resources to solve mathematical puzzles and earn rewards (Abdelrahim et al., 2022).

3.3.4 Centralisation

The evolution of Bitcoin mining pools demonstrates a shift from a more egalitarian system, where individuals could mine coins at home, to a highly concentrated, industrial-scale operation (Caccioli, Livan, and Aste, 2016). This centralisation is evident in Bitcoin and Ethereum mining, where the top miners control over 50% of the network's hash rate, posing a potential risk of a 51% attack that could undermine network security (Efe Gencer et al., 2018). While alternative consensus mechanisms, such as those in permissioned blockchains, offer potential solutions, they introduce their own security and centralisation concerns. Permissioned blockchains require a core group to manage network access, which could lead to arbitrary decision-making and increased costs. Centralisation in the broader blockchain industry raises important questions about the future of decentralised governance and the promise of disintermediation. Despite

these ideals, intermediaries still play a significant role in the market. In the case of Bitcoin, for example, many exchanges facilitate cryptocurrency-to-fiat trades, and digital wallets manage or store users' cryptocurrency accounts, transaction logs, and even private keys (Böhme et al., 2015). These intermediaries act as central hubs for users, making them attractive targets for cybercriminals.

3.3.5 Security

Asymmetric cryptography, also known as public-key cryptography, plays a critical role in authenticating transactions. Blockchain users use public-key cryptography to communicate securely with the Blockchain. Furthermore, hash functions are vital to Blockchains, since they make it possible to create digital signatures and connect blocks.

There is a threat to both hash functions and public-key cryptosystems as a result of the development of quantum computers. Threats posed by quantum computing may quickly recover secure transaction data for public-key cryptosystems. Public-key algorithms such as RSA (Rivest, Shamir, Adleman) (Rivest, Shamir, and Adleman, 1978), ECDSA (Elliptic Curve Digital Signature Algorithm) (Koblitz, 1987), (Miller, 1985), ECDH (Elliptic Curve Diffie-Hellman) (Hellman, 1976) and DSA (Digital Signature Algorithm) (PUB, 2000) are affected by these types of breaches, which can be broken in polynomial time with Shor's algorithm (Shor, 1999) on a sufficiently powerful quantum computer. Quantum computers can also use Grover's technique (Grover, 1996) to speed up the creation of hashes, allowing them to recreate the entire Blockchain. It might also be possible to modify Grover's approach to find hash collisions, which could be used to swap out blocks of data.

To mitigate this potential threat, researchers and developers in the Blockchain space have been exploring and developing quantum-resistant cryptographic algorithms. These algorithms are designed to withstand attacks from quantum computers. Some Blockchain projects have already started implementing quantum-resistant cryptography to future-proof their systems (Fernandez-Carames and Fraga-Lamas, 2020).

A single central figure or middleman cannot compromise a permissionless Blockchain

network. A 51% attack, however, theoretically opens the door to takeovers, manipulations, and collisions (Aponte-Novoa et al., 2021).

Private blockchains, despite their advantages in scalability and energy efficiency, are more susceptible to attacks and collusion. With fewer participants, there is a higher risk of side agreements, transaction reversals, or rule changes. According to (Castillo, 2021), half of the world's biggest companies use HLF, (26 out of 50 companies employ HLF and have at least $1 billion in revenue). In HLF, all Blockchain components are managed exclusively by Peer, Orderer, and Certificate Authority (CA) Admins ; further information can be found in 3.5. Transactors and external users can access Blockchain from the outside via an Application Programming Interface (API), which limits access and lock them out of the system. in addition, the gRPC framework used in HLF for communication between clients, peers, and orderers to exchange blocks, has been subject to a number of medium-severity Common Vulnerability Enumerationss (CVEs)[1]. As a result, insiders have the ability to launch a variety of attacks and use the Blockchain for their own or an organisation's benefit (Putz and Pernul, 2019).

Key management is a major security vulnerability in blockchain systems. Participants are responsible for managing their public and private keys, which can lead to significant problems if keys are lost, such as losing access to funds (ENISA, 2016). It may lead to desperate measures, such as mental breakdowns as a result of lost money (Fröhlich, Gutjahr, and Alt, 2020). As a result of these reasons, and for the ease of use, a lot of people still keep their Blockchain data offline or use a third party in the field of Blockchains, such as mining companies or services that provide digital wallets (Eskandari et al., 2018). Obviously, these companies reintroduce significant security threats if they are hacked, especially those that store the account's private keys.

3.3.6 Privacy

Blockchain's transparency raises concerns about the security of private and confidential data, as all transactions are verified and visible through unique keys or credentials. A key de-

[1] https://nvd.nist.gov/vuln/detail/CVE-2023-32732

bate in the Blockchain community revolves around balancing transparency and privacy (Islam, Rehmani, and J. Chen, 2021). This balance influences whether corporations choose public or private Blockchains. To address privacy concerns, many companies are adopting permissioned Blockchains, where access to data can be restricted at various levels. This allows for private information to be visible to select participants, while other records may be fully public.

Contrary to popular belief, Blockchains are not fully anonymous but pseudo-anonymous (Sas and Khairuddin, 2017). For example, Bitcoin wallet addresses are public, but the identity of the owner remains hidden, giving the appearance of anonymity. Cryptographic keys enable individuals to send Bitcoin without revealing private details to the recipient (Nakamoto, 2008). However, this pseudonymity has made Blockchain technology vulnerable to misuse on the dark web, as seen with Silk Road, negatively affecting its public image (Hiramoto and Tsuchiya, 2020). However, various methods exist for deanonymising Blockchain transactions (Fanti and Viswanath, 2017). Research shows that in over 60% of cases, Personally Identifiable Information (PII) can be linked to Bitcoin addresses used for everyday purchases (Goldfeder et al., 2017). This is largely due to web trackers and cookies, which collect and share user activity with third parties like Google and Facebook. These trackers can expose personal information such as names, locations, and email addresses, making it possible to link transaction data to real-world identities. As Goldfeder and his colleagues in (Goldfeder et al., 2017) have shown, even the use of privacy-enhancing techniques for Blockchain anonymity, like CoinJoin, does not provide a perfect solution to avoid their attacks. Ongoing research is focused on addressing these privacy challenges through advanced cryptographic techniques, such as zero-knowledge proofs, which allow for transactions to be verified without revealing sensitive information (Averin, Samartsev, and Sachenko, 2020; Z. Zhang et al., 2020).

3.3.7 Mutability

The immutability of Blockchain technology is a subject of considerable debate. While Blockchains are designed to prevent unauthorised changes, they can theoretically be manipulated or rolled back, as discussed in 3.3.5. Although no Blockchain has yet suffered such a systemic attack,

immutability becomes a concern when records need to be modified due to errors or inconsistencies. The core issue revolves around Blockchain's ability to allow consensus-based changes, which challenges the idea of immutability. In decentralised networks like Blockchain, decisions and changes rely on consensus among participants. Rather than being entirely unchangeable, Blockchain records are simply difficult to alter (Walch, 2016).

Consensus-based changes, such as altering transaction records, have occurred in the past. One notable example is the "DAO Attack", which sparked significant debate (Mehar et al., 2019). On Ethereum, Distributed Autonomous Organization (DAO) was an experiment in decentralised governance, functioning like an investment fund. Investors contributed Ether to the DAO and were able to vote on venture proposals from Ethereum-based firms, with transactions and decisions executed through smart contracts.

The DAO experiment quickly collapsed due to a bug in the code that left it vulnerable to exploitation. An anonymous hacker took advantage of this flaw, stealing over $50 million worth of Ether from the $168 million invested. In response, the Ethereum community decided to roll back the Blockchain to its pre-hack state and reimburse the affected investors.

Ironically, Bitcoin was created as a response to the 2008 bailout of US banks. This action, however, may have been perceived as a bailout and contradicted the Blockchain community's principles of "code is law" and ledger immutability. Most importantly, this instance led to lively debate regarding what Blockchain systems' immutability actually implies. As Blockchain truly depends on a collection of actors (developers, miners, clients, and other users) who each play a specific role and are able to step in when necessary to address issues, improve the system, or undo undesired results, it highlighted the significance of governance (DuPont, 2017).

3.4 Blockchain Main Concepts

The primary principles implemented by any developed Blockchain network include decentralised infrastructure, ledger structure, consensus mechanism, and smart contracts. Thus, it is typical to use the Blockchain terminology interchangeably to denote ledger structure, decentralised

infrastructure, P2P networks, consensus mechanisms, and similar concepts. This section will provide a more detailed explanation of these concepts.

3.4.1 Decentralisation

The concept of decentralisation has been a topic of interest in many disciplines for a long time, such as in politics, economics, and computer science. Paul Baran was one of the pioneers of the development of computer networks. He was developing a network that could withstand a nuclear attack between the late 1950s and the 1960s. His groundbreaking paper, "On Distributed Communications," published by RAND Corporation in 1964, outlined different structures for networks: centralised, decentralised, and distributed (Baran, 1964). Baran's work in describing these structures, particularly the concept of a distributed network, was foundational to the development of packet switching networks and, ultimately, the Internet as we know it today. However, it wasn't directly about decentralisation in the way we think about it today in terms of Blockchain technology and cryptocurrencies, but more about the resilience of different types of network structures. The way in which Blockchain technology delineated the concept of decentralisation is illustrated in one of the blogs that Vitalic Butern, the Ethereum founder, has published [2]. He categorises decentralisation into:

- **Architectural Decentralisation:** How many actual nodes (computers) are building up the system?
- **Political Decentralisation:** How many persons/ organisations are fully governing theses nodes?
- **Logical Decentralisation:** If the system is cut in half, including both providers and users, will both halves of the system continue to operate independently? i.e., to fulfill the logical decentralisation, users must be able to access and interact with all providers regardless of their location if the system is designed as a swarm. Users must only be able to interact

[2]https://medium.com/@VitalikButerin/the-meaning-of-decentralization-a0c92b76a274

with the providers that are directly connected to the system if it is designed as a monolithic object.

Based on this categorising, Blockchain is politically and architecturally decentralised but, from a logical perspective, Blockchain is always centralised, as it is programmed to behave as if it were a single computer, with a single state and a single set of transactions. This logical centralisation allows for a more secure system since any changes to the state must be approved by all nodes in the network.

Another set of criteria is presented by Jamie Burke (2018) [3], who serves as the CEO of Outlier Ventures, a company that invests in tokenised communities contributing to the emerging decentralised economy. He puts forth this specific framework tailored for Blockchain-based networks:

- **Consensus Establishment:** Who wields control over the network nodes, and what methods are employed to achieve consensus?
- **Protocol Valuation:** To what extent is the value distribution within the network decentralised and evenly spread?
- **Protocol Advancements:** Which entity is responsible for steering the product development roadmap?
- **Dispute Resolution:** What mechanisms are in place for resolving conflicts, and how are the resolutions enforced?
- **Platform Improvements and Developing :** How many individuals or organisations are actively engaged in developing applications or services on top of the network?

Schneide in (Schneider, 2019) examines the term decentralisation in relation to different cultures such as politics culture, network culture, and Blockchain culture. According to him, periods of decentralisation are often followed by periods of recentralisation. Over time, underlying shifts in power and control will become evident, dispelling the illusion of decentralisation.

[3] https://outlierventures.io/research/pathway-to-decentralisation/

For example, centralised Internet monopolies such as Google, Facebook, and Amazon emerged from the initial decentralisation of the Internet. Companies like Airbnb [4] meet Jamie Burke's criteria across all categories, while Google Android [5] is centralised in the first four categories and decentralised in the fifth. Although crypto assets can exhibit decentralisation, centralised exchanges often control their trading, posing risks. This raises the argument that despite decentralisation in several dimensions, centralised groups may ultimately control significant value generation within the Blockchain ecosystem (Brandvold et al., 2015; Gandal et al., 2018). Thus, a technology that appears decentralised in multiple aspects may contradict the conclusion that centralised entities will dominate value creation.

3.4.2 Blockchain Ledger

Blockchain is a type of DLT that maintains a ledger that is replicated on several nodes within the Blockchain network. This ledger continues to grow as new blocks are added to the chain over time, and each block will include a list of verified transactions that interact with the ledger. Following this distributed and decentralised structure will prevent a single point of failure as there is no single node maintaining the ledger; rather, each node has the same copy of the ledger and each node has the right to validate each block before approving its attachment to the chain. This technology was introduced first in 2008 and used well-known techniques such are: the hash function, timestamp, Merkle tree, consensus algorithms, and smart contracts.

An example of block structure is depicted in Figure 3.2, which shows two main parts in each block:

1. **Block Header:** Contains two hash values:
    - The first hash value links the current block to the previous block by including the hash value of the previous block's header.
    - The second hash value represents the root of the Merkle tree, which encompasses the hashes of all transactions within the current block.

[4] airbnb.com
[5] android.com

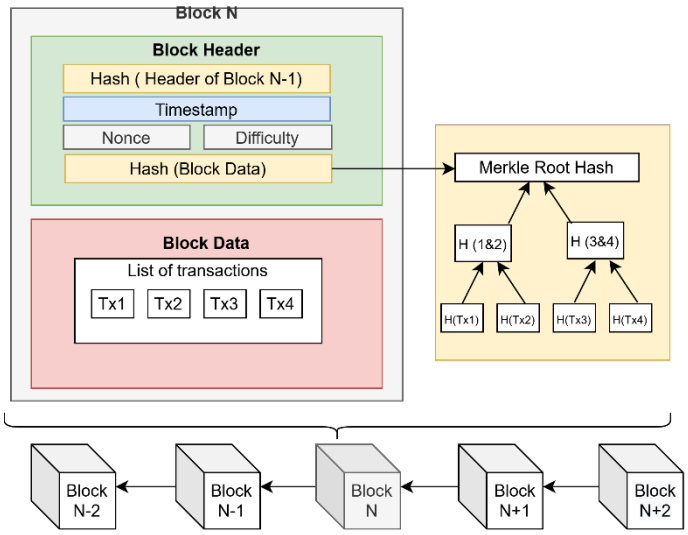

Figure 3.2: Block structure.

2. **Block Body:** Contains a list of transactions.

These two hash values play the main role in securing the block by preventing any tampering attempt and enabling any node to verify the data in any block. If an adversary attempts to modify the content of one of the previously stored transactions, then the hash value of the corresponding block, which will be stored in the next block's header, will no longer be consistent; thus, this attack will be easily exposed to others. Therefore, any change in one block will affect all the subsequent blocks respectively which makes the immutability stronger by increasing the Blockchain length.

As the size of the Blockchain continues to grow over time, storing the entire Blockchain on every node becomes increasingly challenging, especially for nodes with limited storage or computing resources. To address this issue, the Merkle root, a hash value representing the root of the Merkle tree, plays a crucial role in ensuring Blockchain sustainability. It provides a way to verify that a specific transaction is part of a block without needing to store all transactions within the block.

To handle the growing size, Blockchain networks typically use two types of nodes: **full nodes** and **light nodes** (also known as lightweight nodes).

- A full node (or master node) maintains a complete copy of all blocks in the Blockchain. These nodes are typically used on servers with sufficient resources to store the entire chain.

- A light node, on the other hand, stores only a copy of the block headers and downloads only the necessary transaction data. By using the Merkle root hash, light nodes can verify the status of any transaction without needing to store the full Blockchain. They achieve this by fetching only the Merkle tree nodes that connect the Merkle root to the target transaction.

This technique, known as Simplified Payment Verification (SPV), was first described in Satoshi Nakamoto's original Bitcoin white paper (Nakamoto, 2008). It allows lightweight nodes to efficiently verify transactions by holding only a small fraction of the Blockchain's data, reducing the storage and computational demands on the node.

In addition to the Merkle root, the block header contains other important metadata, such as the timestamp, difficulty, and the nonce. These elements are used in the mining process, where miners aggregate valid transactions into new blocks and append them to the Blockchain. Miners follow a predefined consensus algorithm to ensure that all nodes in the network agree on the validity of newly appended blocks.

3.4.3 Consensus Algorithms

Consensus algorithms are essential for ensuring that a group of nodes in a Blockchain network can agree on which new block should be added to the chain. In decentralised networks, where peers do not necessarily know or trust each other, it is critical to prevent situations where different nodes broadcast blocks containing the same transactions, causing confusion and redundancy in the ledger. Consensus mechanisms are designed to prevent such conflicts, ensuring that all peers come to an agreement on which block is valid and should be appended, even if some nodes behave maliciously or unreliably. Consensus algorithms vary in terms of decentralisation, security, and incentives, but all share the same goal: determining block validity, deciding which block to add next, and establishing which node has the right to append the block. Choosing the right consensus algorithm can significantly affect the blockchain's performance, especially in terms of transaction throughput. According to (Nguyen and Kim, 2018), consensus algorithms can be classified into two main categories: voting-based consensus algorithms and proof-based

consensus algorithms.

- **Voting-based consensus algorithms:** In this type, all nodes within the network should be known. In addition, all nodes may participate to verify the transaction or the block, and at least T nodes should agree on doing the appending work (T is a threshold). The conventional methods for tolerating faults that are used in the distributed system are used in voting-based consensus algorithms (Heimerdinger and Weinstock, 1992). Therefore, these consensus mechanisms should resist node failure cases such as crashed nodes and subverted nodes. To prevent crashing cases among f nodes, there should be more than f (at least $f + 1$) nodes operating correctly to make the decision (Lamport, 2001). The subverting problem is presented by a classical problem called Byzantine generals, proposed and solved by Lamport et al. (Lamport, Shostak, and Pease, 2019). They have proved that in order to tolerate f subverted generals, there should be at least another $2f + 1$ at normal generals to reach a consensus.

  Similarly, in Blockchain, when each node tries to execute the consensus work, some other nodes can be subverted and send different responses to other nodes. As a result, the ledger could be different in different nodes. By taking into account the crashing cases, crashed nodes won't be able to send their results to other nodes, which makes it difficult to reach a final decision. Based on these two fault cases, the voting-based consensus algorithms are classified into two categories:

  - Byzantine fault tolerance-based consensus, in which both cases of crashing nodes and subverted nodes are prevented.
  - Crash fault tolerance-based consensus, in which only the case of crashing nodes is prevented.

  In general, in both categories t nodes should work normally where $t < N$ and N is the total number of all nodes. t is equivalent to $[N/2+1]$ in crash fault tolerance-based algorithms, while in Byzantine fault tolerance-based algorithms, t is equivalent to $[2N/3 + 1]$.

- **Proof-based consensus algorithms:** In contrast to voting-based mechanisms, proof-based algorithms do not require all nodes to be known or to participate in every consensus round. Nodes can join and leave the network freely, and certain nodes (typically called miners) are responsible for adding blocks by solving computational challenges or providing proofs of their effort. The Proof of Work (PoW) algorithm is the first algorithm of this type (Wood, 2014). Although it is widely used in cryptocurrencies, it has some drawbacks related to the huge consumption of energy and the low throughput. Therefore, Proof of Stake (PoS) was developed as an energy-saving alternative to the PoW consensus algorithm. In this approach, miners won't need to consume considerable electrical resources to compute hashes. Instead, it depends on how much stake does miner owns to participate in the block creation process proportionately. Although several versions of PoS protocols have been proposed, this algorithm still has some drawbacks such as the way of miner selection (Schuh and Larimer, 2017; King and Nadal, 2012; Pike et al., 2018). Since the selection is based on how wealthy the miner is, this makes the wealthiest node more dominant in the network. As a result, this will cause unfair distribution and unfair centralisation. Thus, PoS is more exposed to malicious attacks than PoW as the effort and the cost demanded in the mining process are much lower, and this low cost in mining causes the Nothing–at–stake problem. When there is a fork in the chain, validators mine the block on both branches to secure their fees from the branch which will win later. Therefore, PoS powered Blockchains only make up less than 2% of the market capitalisation of currently used digital currencies (W. Li et al., 2017). Many other proof-based algorithms have emerged; some of them don't follow the same idea of PoW or PoS and some others are considered to be hybrid, combining elements of both PoW and PoS protocols. One typical example of a hybrid protocol is Proof of Activity, which creates empty blocks by employing PoW and uses PoS to verify blocks and add transactions (Bentov, C. Lee, et al., 2014).

3.4.4 Smart Contracts

Another basic characteristic in Blockchain other than decentralisation, and immutability is the programmable smart contracts. A smart contract is essentially a self-executing piece of code deployed on the Blockchain, designed to automatically enforce agreements or business logic. These contracts run on decentralised infrastructure, ensuring trust and security without needing intermediaries.

In the first generation of Blockchains, such as Bitcoin, the support for smart contracts was very limited. Bitcoin's primary focus was on providing a secure, decentralised digital currency, and while basic scripting functionality existed, it was not robust enough to support complex programmable logic. With the advent of second-generation Blockchains, such as Ethereum, support for more advanced, general-purpose (Turing-complete) smart contracts became a reality (Christidis and Devetsikiotis, 2016b). This allowed developers to build and execute far more sophisticated contracts, enabling a wide range of decentralised applications (dApps) to emerge. In fact, a smart contract is not necessarily smart in the sense of artificial intelligence, nor is it always a legally binding contract. In essence, a smart contract is simply code that can execute predefined actions when certain conditions are met. These contracts are hosted on a Blockchain network and are automatically enforced by the network itself. For example, a smart contract in the Ethereum blockchain will be assigned to a unique address and comprises three main components:

- Private Storage: Used to store data specific to the contract.
- Account Balance: Holds the cryptocurrency balance (in Ether).
- Executable Code: The logic or functions that define how the contract behaves.

Whenever a smart contract is invoked, the Blockchain's consensus mechanism processes the transaction, and miners validate the contract's execution. The transaction might read/write data from the contract's private storage, alter the balance of Ether, exchange data with other contracts or users, or even create new contracts, depending on the function invoked in the contract (Delmolino et al., 2016).

According to Morabito (Morabito, 2017), smart contracts can be classified into two main categories: deterministic and non-deterministic.

- Deterministic Smart Contracts: These contracts operate entirely within the Blockchain. When a deterministic smart contract executes, all the necessary data and information required to make decisions are already present on the Blockchain. No external data is needed, making this type of contract entirely self-contained and predictable.

- Non-Deterministic Smart Contracts: Non-deterministic contracts, on the other hand, depend on external information to execute certain actions. This external data is often provided by third-party services known as Oracles. For example, a smart contract might need current weather data to execute, but since the Blockchain does not store such information, it must rely on an Oracle to provide the necessary data.

Oracles act as intermediaries between the Blockchain and the outside world, providing the contract with the information it needs to make decisions. This reliance on external sources can introduce risks, but when Oracles are properly managed and decentralised, non-deterministic contracts can function securely and effectively against fraud.

3.5 Hyperledger Fabric

As discussed in Section 3.6, HLF has been chosen as the Blockchain platform for the developed testbed framework introduced in Chapter 5. As noted by Androulaki et al. in (Androulaki et al., 2018), HLF is a permissioned Blockchain platform that allows for customisable consensus protocols and accommodates general-purpose programming languages. HLF is a Blockchain platform suited for enterprises, independent of cryptocurrency, aiming to meet industry needs while maintaining essential Blockchain features previously discussed. This section highlights the unique characteristics of a fundamental HLF architecture and the transaction process model.

3.5.1 Hyperledger Fabric Network Elements

The content in this subsection is based on the official documentation of HLF [6] and the article (Androulaki et al., 2018). Assumes a consortium of several recognised organisations that pursue a common purpose or have the same aim. In our case, these organisations can represent any participant in the railway network as depicted in Figure 3.3 and listed in Table 3.1. Hyperledger Fabric allows these organisations to establish a decentralised network utilising Blockchain technology. As an illustration, Figure 3.4 shows a simple network comprising two organisations (ORG1 and ORG2), with each organisation node in the network has several essential parts, which is maintained by the organisation or interact with it as follows:

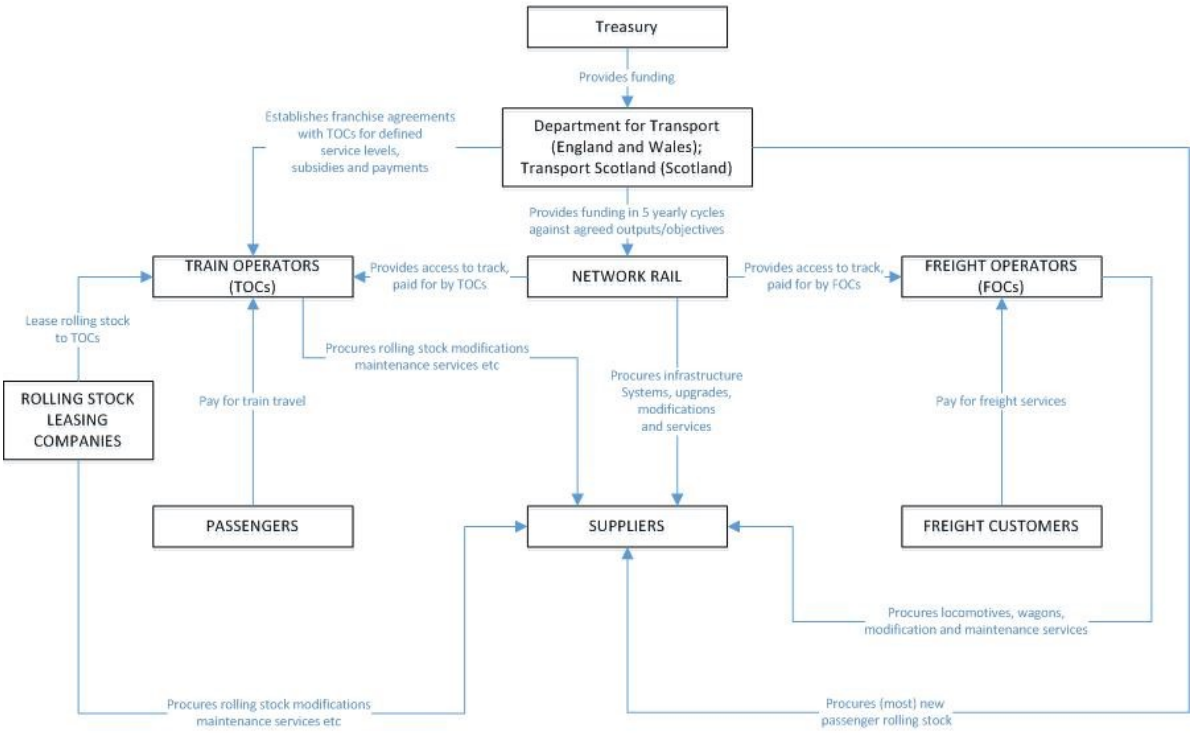

Figure 3.3: High-level structure of the GB rail industry

- **Nodes:** In contrast to Ethereum, Hyperledger Fabric implements an Execute-Order-Validate workflow, as elaborated in Section 3.5.2. As a result, any node can perform one or multiple of these roles:

[6] https://hyperledger-fabric.readthedocs.io/en/latest/index.html

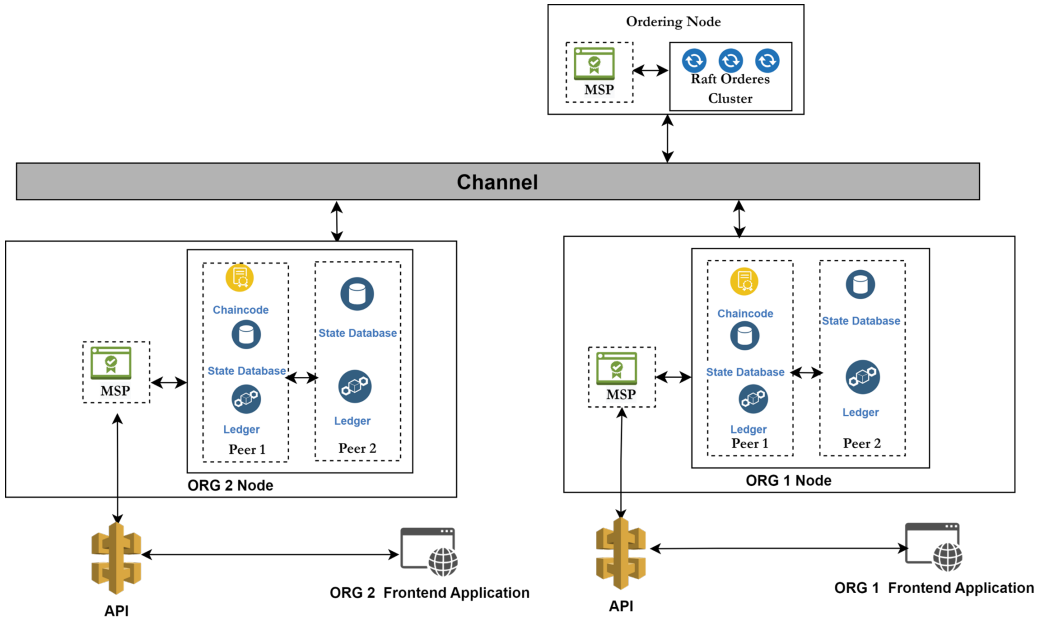

Figure 3.4: Hyperledger Fabric network

Stakeholder	Role/Responsibility
DfT	Sets national rail policy and awards service contracts to train operators.
Network Rail	Publicly owned organisation responsible for owning, maintaining, and upgrading the majority of the railway infrastructure.
TOCs	Private companies that operate passenger rail services under government contracts (e.g., Avanti West Coast, Great Western Railway).
FOCs	Private firms that provide rail-based cargo transport services (e.g., DB Cargo, Freightliner).
Rolling Stock Leasing Companies	Own and lease passenger trains (rolling stock) to TOCs; key players include Angel Trains, Eversholt Rail, and Porterbrook.
Passengers & Advocacy Groups	Represent user interests, campaign for better service and transparency (e.g., Transport Focus).
The Office of Rail and Road (ORR)	Independent regulator overseeing railway safety, economic efficiency, and performance monitoring.

Table 3.1: Major stakeholders in the British railway system.

1. Committing Peer: This node commits the block to their copy of the Blockchain and state storage. The block will contain a list of transactions to validate each transaction

in the list and confirm such transactions as either valid or invalid and then commit them to the block. All such transactions, irrespective of whether they are valid or invalid, are all committed to Blockchain, and this may be used for future audit purposes. This node is not involved in transaction endorsement because it lacks the capability to execute chaincode.

2. Endorsing Peer: This is a special type of committing peers who apart from their regular role, will have an additional responsibility to endorse a transaction in the network. Any request coming from the client's node is endorsed by such a peer. Each of these peers will generally have a copy of the ledger and the installed chaincode. The endorsers are entrusted with the responsibility to simulate the transaction and would generate Read / Write sets, which are then sent to the requesting client. The transaction is not committed to the ledger during such a simulation.

3. Anchor Peer: Anchor peers are responsible for enabling communication between different organisations within a Fabric network. Unlike regular peers, anchor peers can interact with nodes from other organisations via channels. They might also take on roles such as committing or endorsing transactions.

4. Leading Peer: The leading peers relay messages from the ordering service to other peers within the same organisation. Using the Gossip protocol, they ensure smooth communication among peers but are restricted to communicating only within their own organisation.

5. Orderer: The Orderer as the name implies, will be responsible for ordering transactions into a block. Usually, a separate ordering node does this job, which, along with other ordering nodes, forms an ordering service cluster. Based on the application design, the organisation may have its own ordering service to increase the transaction throughput that will then interact with the ordering service.

The modular design of HLF allows different architectures to determine the number of node types each organisation needs to contribute to the network. Each type of node requires

the organisation to allocate the necessary computing power and data storage capacity.

- **Chaincode:** In HLF, smart contracts are referred to as chaincode, which holds the business logic and coordinates the interactions between the application and the ledger. The chaincode can be developed in various programming languages (the currently supported languages are Java, Goplang, and Javascript). In our developed testbed, Golang and Node.js are used to develop chaincode for managing business processes in the railway consortium. A chaincode consists of multiple smart contracts within a single namespace, enabling shared access to the state storage. Accessing the chaincode will be maintained through the endorsing peers on which the chaincode is installed and initiated. A chaincode can embody the business logic along with the data structure as they are stored. For example, a smart contract may define a data model for a data offer and associated agreements. A smart contract may also outline the functions for managing the data model, including Creating, Reading, Updating, and Deleting (CRUD). Figure 5.3 shows a sample of how we utilise the Chaincode feature to model the data structure in the state storage.

- **State Storage:** The state storage is a record-oriented database that organises records in the format (Key, Value, Version). In this format, Key represents a distinct record key, Value is the record's value, and Version is a sequential nonce employed to monitor modifications on the records. As the name suggests, state storage shows the most recent status of an asset according to the distributed ledger. Figure 3.5 provides an example to differentiate the data stored in the state database and the ledger. The state database captures the most recent status of an Offer request in the (Key, Value, Version) format, where Key stands for the offer request ID, Value represents the present status of the Offer request (Initiated, Accepted, Rejected), and Version keeps a record of the updates made to the record. Only the latest record is preserved in the state database, while all generated records are kept in the Blockchain ledger. Further details will be discussed in Chapter 5 proof of concept.

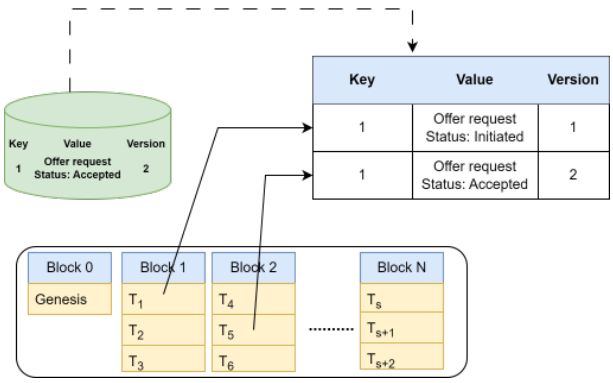

Figure 3.5: Illustration of state database records.

- **Channels:** The channel is the connection which ensures the isolation and the confidentiality of data among the consortium participants. Each organisation will be registered to one or more channels, and the users of that organisation will gain the access to the channel their organisation belongs to. All organisations within a channel share a common ledger, chaincode, and state storage.

- **Certificate Authority:** The CA is responsible for managing identities within the network. It issues and administers the certificates that define each participant's identity and role within the consortium, ensuring that only authorised participants have access to network resources.

3.5.2 Transaction Flow

In contrast to traditional applications, a Blockchain transaction in HLF is deemed valid only after it passes through validation processes like Endorsement System Chaincode (ESCC),Validation System Chaincode (VSCC), and Multi-Version Concurrency Control (MVCC) (Androulaki et al., 2018). This process ensures data integrity and avoids conflicts such as double-spending. The entire transaction lifecycle, from submission to final commitment in the blockchain ledger, is thoroughly outlined in Figure 3.6, which demonstrates the Execute, Order, Validate model used in HLF. The transaction flow can be divided into the following three stages:

1. **Transaction Endorsing :** In the endorsement stage, a client submits a transaction proposal

to invoke a smart contract (or chaincode). This proposal is processed by selected endorsing peers within the network, which each simulate the transaction using their local copy of the state database.

During this simulation, each endorsing peer generates read/write sets:

- The read set is represented as $R\langle \text{Key}, \text{Value}, \text{Version} \rangle$, indicating the record being read, its current value, and its version.
- The write set is represented as $W\langle \text{Key}, \text{Value}', \text{Version}' \rangle$, indicating the updated value and version of the record if the transaction is successful.

Each endorsing peer then signs the generated output and returns it to the client. This ensures that the transaction complies with the endorsement policy set by the network.

2. **Transaction Ordering and Block Generation:** Once a transaction is endorsed, it moves to the ordering service, where it is not yet committed to the ledger. The ordering service groups and sequences transactions into blocks, ensuring they are processed in the correct order. However, the read/write sets are not yet recorded in the state database at this stage.

   For a transaction to proceed, it must satisfy the endorsement policy, which might require, for example, approval from a majority or a specified fraction (such as two-thirds) of endorsing peers.

   The ordering nodes, illustrated in Figure 3.4, play a crucial role in this phase, as they assemble the transactions into a block and determine the sequence of transactions. The consensus mechanism used here is the RAFT protocol (Ongaro and Ousterhout, 2014), which operates in a leader-follower model and ensures crash-fault tolerance. The ordering service collects transactions until they meet predefined criteria, such as the maximum number of transactions per block or a timeout condition, and then organises them into a block for the next stage.

3. **Transaction Validating and Committing**: Once a block is assembled, it is passed on to the validating and committing phase, where transactions are subjected to final checks

before being added to the ledger. During this process, all the endorsing peers and the committing peers evaluate each transaction using the VSCC and MVCC mechanisms.

- The VSCC mechanism ensures that the transaction complies with the endorsement policy—verifying that the necessary number of endorsements is present.
- The MVCC mechanism checks for read-write conflicts. Specifically, it verifies that the version of the records in the read set matches the current version in the state storage. This prevents double-spending and ensures data integrity by ensuring that no other transaction has updated the data since the transaction was endorsed (Javaid, C. Hu, and Brebner, 2019).

If a transaction passes these validation steps, its write set is committed to the state storage, updating the current state of the relevant assets. The transaction is then marked as successful.

If the transaction fails validation, the write sets are not committed, but the transaction is still recorded on the blockchain ledger for transparency and auditing purposes. This guarantees that all transactions, whether successful or failed, are permanently and immutably stored on the Blockchain.

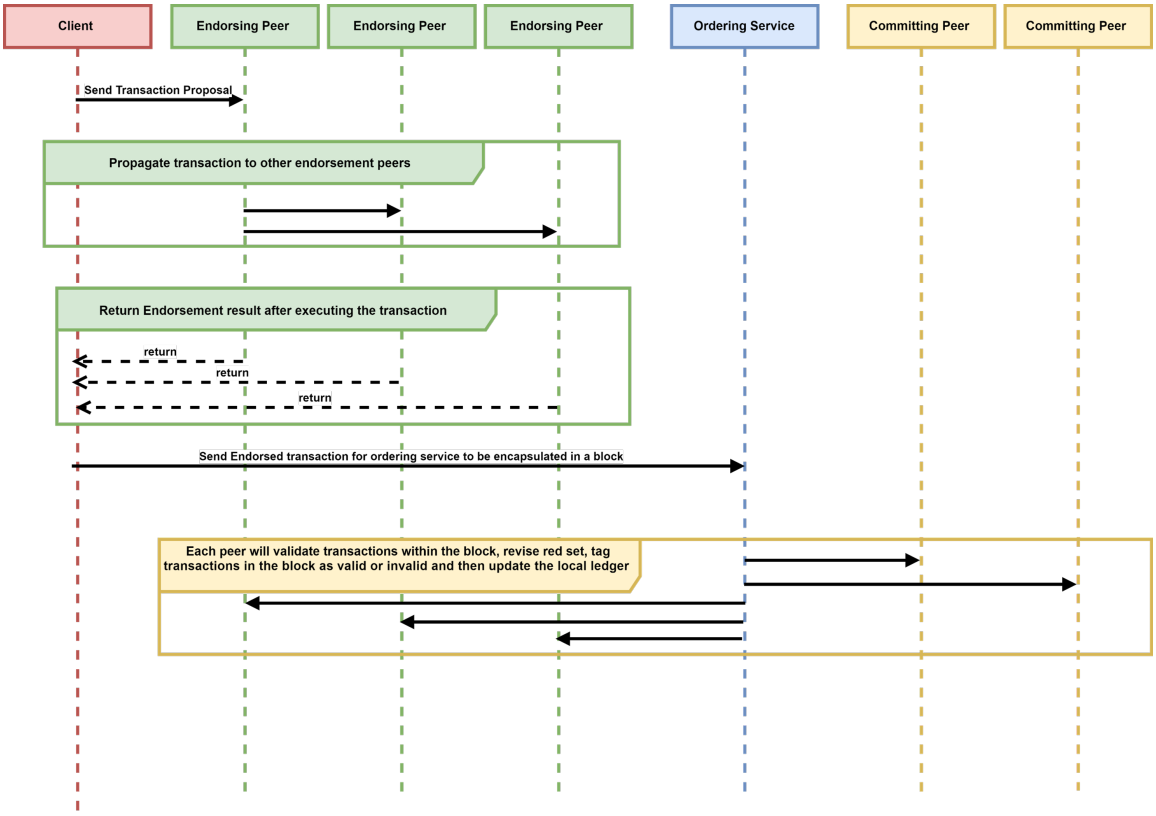

Figure 3.6: Transaction flow in HLF network.

3.6 Blockchain Platform Selection

Instead of constructing a Blockchain platform from the ground up, a typical Blockchain-based solution would usually opt to use a well-established Blockchain platform like Ethereum or HLF as its foundational infrastructure. While Blockchain platforms share core characteristics, they differ significantly in their implementation, affecting the design, performance, and scalability of any decentralised solution built upon them. Table 3.2 provides a side-by-side comparison of the key differences between Ethereum and HLF, particularly in the context of consensus mechanisms, fees, smart contract capabilities, and transaction throughput. Based on these criteria, this thesis selects Hyperledger Fabric for the proposed test-bed in Chapter 5 due to the following factors:

- **Decentralisation or Scalability:** In section 3.3.2, the scalability challenges of permissionless Blockchains are discussed and highlight how predetermined participants in per-

Criteria	**Ethereum (Original Network)**	**Hyperledger Fabric**
Consensus Algorithm	proof of work (PoW)	Crash-Tolerant Protocols (Raft, PBFT, Kafka etc.)
Network Type	Permissionless / public	Permissioned /consortium
Transaction throughput	Low	High
Execution fees	Applicable	Not Applicable
Resource fees	Applicable	Not Applicable
Latency	High	Acceptable
Participant Anonymity	Anonymous with flexible membership	Identified and restricted participation
Smart contract updates	Not feasible	Feasible with endorsement policy support
Energy Efficiency	Low	High
Smart contract capabilities	Adequate but constrained and tailored for specific use (Solidity)	Extensive, widely-adopted, and versatile (Java, node.js, GoLang)
Cryptocurrency	Required	Not Required
Structural Modularity	Not Applicable	Modular Components

Table 3.2: Tradeoff comparison: Ethereum versus Hyperledger Fabric.

missioned Blockchain can enhance network performance. The reduction in scalability and performance is attributed to the decentralised nature of permissionless Blockchains, whereas permissioned Blockchains are more centralised. According to (Altarawneh et al., 2020), the well-known Blockchain Trilemma, a term introduced by Vitalik Buterin, the co-founder of Ethereum, is discussed. The Blockchain trilemma fundamentally concerns the balance of three characteristics: decentralisation, scalability, and security. A Blockchain system can usually achieve excellence in two of these aspects, but this typically comes at the expense of the third characteristic. The work by authors in (Sanka and Cheung, 2021) likewise addresses the Blockchain trilemma, emphasising that private Blockchain applications generally excel in scalability compared to public Blockchain applications, although this comes at the cost of some decentralisation. While the authors in (Altarawneh

et al., 2020) assert that public Blockchain networks preserve strict decentralisation, this results in reduced scalability.

- **Smart Contract Updating:** According to (W. Cai et al., 2018), Ethereum records the installed smart contract in its distributed ledger. Due to the network's permissionless nature, altering the code logic of an active smart contract is impossible. Therefore, to update the smart contract's logic for maintenance or to change its terms, a new smart contract must be deployed, which will render the previous contract either completely or partially obsolete (Wöhrer and Zdun, 2018). Furthermore, it becomes impossible to retrieve the state database of the prior smart contract (Zou et al., 2019). For sustainable operational services, the maintenance needed to enhance the smart contract logic should be guaranteed, as the parameters we used in building the system could be updated for any reason in the future.

  In contrast to Ethereum, Hyperledger Fabric regards smart contracts to be modifiable entities. Given the private structure of Hyperledger Fabric, an endorsement rule enforced by validating nodes (functioning as a polling mechanism) is implemented to manage and oversee the installation and updating of the smart contract. For instance, an endorsement policy might necessitate consent from two-thirds of validating nodes to approve modifications to the code logic of the existing chaincode.

- **Execution Fees:** Given Ethereum's permissionless characteristic, any peer (node) may participate or withdraw at their discretion. Consequently, Ethereum has established an incentivisation system to encourage the public to contribute computational and storage resources. To this end, Ethereum imposes fees on clients for executing smart contracts and allocating storage, which are then paid to miners (Yaga et al., 2019). The fees are determined by the complexity of the smart contract's logic and the amount of storage required to handle the transaction (Aldweesh et al., 2018). Therefore, this thesis contends that utilising Ethereum might present difficulties due to the intricate nature of the smart contract developed for the proposed test-bed framework. In contrast, Hyperledger Fabric

eschews the economic model introduced by Ethereum, using a lightweight consensus mechanism that eliminates the need for mining by anonymous entities. Additionally, Hyperledger Fabric operates as an alliance of identifiable members who are committed to contributing their resources to the Blockchain platform and its setup (Androulaki et al., 2018).

- **Rate of Transactions Throughput:** The speed at which new blocks are generated affects the scalability of the Blockchain platform in terms of queuing and processing of client-side transactions. For example, Ethereum's current block rate is not well suited for the proposed systems, which generate many data offers, data requests, and agreements that need to be managed by smart contracts. Currently, the Ethereum network takes an average of 12 seconds to create a new block [7]. The slow rate of block generation reduces throughput (12-15 transactions per second) and increases latency (Valadares et al., 2023). At the same time, HLF has been shown to manage a considerable volume of transactions with acceptable latency, as shown in Chapter 6. The modular nature of HLF, with its customisable components and configurations, can also help to achieve optimal results.

- **State Storage:** Ethereum assigns individual local storage for every smart contract, which is not only costly but also has a significantly limited capacity (Aldweesh et al., 2018). On the other hand, Hyperledger Fabric offers an improvement in terms of storage capacity and cost-efficiency.

- **Programming Languages:** Ethereum has developed a new programming language known as Solidity. However, Solidity is relatively immature when compared to more established languages like Java, JavaScript, and GoLang, which are supported by Hyperledger Fabric in terms of stability, clarity, available features, and proficiency. As a result, Hyperledger Fabric is a more suitable option for creating high-quality smart contracts for enforcing data-exchanging agreement terms within the Blockchain, as well as for carrying out compliance assessments, enforcing penalties, and executing billing logic.

[7] https://etherscan.io/chart/blocktime

3.7 Blockchain Applications in Industry

Blockchain and other DLTs have begun to disrupt more than just the financial industry's business models and products. This is because DLTs can be used to develop new markets, create new services, and disrupt existing business models. The promise of transparency, security, or decentralised solutions to manage practically all types of digital assets and data has become crucial to many applications where migrating from siloed systems to a shared infrastructure has become essential to keep up with the rapid advancements in technology. These solutions can help organisations overcome the challenges of legacy systems, reduce operational costs, and create secure ecosystems. Additionally, they can help organisations gain a competitive edge by providing access to a wider range of data, greater control over data, and improved security. There are eight industries where Blockchain-based applications are either in an exciting early stage of development or where current issues seem to be ripe for using Blockchain features. This section will provide an overview of Blockchain applications across these eight industries.

3.7.1 Aerial and Space Sciences

In the context of space and aeronautics, Blockchain technology exhibits substantial promise, particularly in augmenting efficiency, transparency, and security across various facets of the industry. As part of these applications, sensitive data can be securely managed and used, encrypted communications can be used to verify quality standards, distributed information processing can be performed, and network administration and security can be overarching (Abdulrahman et al., 2023; Dan et al., 2020). Furthermore, satellite communications can help Blockchain systems increase their performance in scenarios characterised by connectivity to machines or IoT (J. Wang et al., 2021). Through satellites, transactions can be transmitted, and validated blocks can be received, allowing for synchronisation and extending connectivity to Blockchain nodes in remote or hard-to-reach areas where conventional networks are unavailable (Q. Hu et al., 2020).

Blockchain-based technology is being actively explored by the European Space Agency

(ESA) as part of its adaptation to Space 4.0. Authors in (De Filippi and Leiter, 2021), outline how Blockchain technology can improve outer space governance beyond libertarian ideals, supporting a more commons-based approach. While authors in (Drobyazko and T. Hilorme, 2021) illustrated how space objects can be effectively secured with Blockchain technology. With Blockchain technology as a high-tech control system, communications can be combined to create a single whole, provided that programmable artificial intelligence is adapted to each individual's needs (Tetiana Hilorme et al., 2019).

3.7.2 Distributing and Processing Food

Despite the fact that food safety has improved significantly over the past ten years, a significant number of diseases are still linked to food consumption (Franz et al., 2018). The use of Blockchain technology in the food business has recently grown at a rate never before seen and has the potential to address long-standing food safety problems, particularly those involving biological and chemical contaminations, improve food traceability, and boost customer confidence not only in the safety of their food supply but also the ethical standards. Additionally, it makes it possible for information like origin, batch number, and manufacturing date to be shared promptly, which encourages businesses to adopt sustainable business strategies, safeguarding food certificates' veracity and reducing fraud and adulteration threats (Galvez, Mejuto, and Simal-Gandara, 2018). In addition, it is believed that the technology could increase the efficiency of real-time monitoring of food stocks and delivery, as well as, for example, help in figuring out where and why food is thrown out or goes bad, which could reduce food waste (Y. Xu et al., 2022). Other technologies are needed in order for Blockchain technology to be fully applied in the food industry (Yiannas, 2018). Good Manufacturing Practices (GMP), Good Agricultural Practices (GAP), and other standards can be written as smart contract procedures, and food that does not adhere to the necessary standards is not permitted to join the Blockchain (Mao et al., 2018). The issue of high development costs is one that Blockchain technology must overcome in order to deliver the recommended use cases. It also needs to demonstrate the required scalability, speed, and security. Thoughtful application design and implementation can,

however, overcome these difficulties (Cocco, Pinna, and Marchesi, 2017). Also, the speed of data transmission will be significantly increased with the development and use of 5G technology, and the speed limitations of Blockchain technology may be addressed (Z. Chen et al., 2018).

3.7.3 The Transport and Logistics Industry

In the transportation and logistics industries, Blockchain technologies and other DLTs are being used in a wide range of ways for a variety of reasons. Among them are managing the entire supply chain and fleet operations, facilitating the safe transfer and monitoring of assets, ensuring data security, streamlining customs documentation processes, and automating contract execution among various parties. Blockchain can also be applied to track sensitive products or materials, verify their origin, and ensure safety standards are met (Boucher, Nascimento, and Kritikos, 2017). The current problem is that the industry has data silos and fragmented software systems. These include the often-used enterprise resource planning systems for organisations as well as systems for managing transportation, warehousing, and customs. The decentralised network of stakeholders, which includes shippers, freight forwarders, carriers, warehouses, customs officials, governmental organisations, international terminal operators, and road transport businesses, among others, is hindered by these systems from sharing information effectively.

Blockchain has the ability to give the logistics and transportation industries close to real-time data integrity due to its built-in features. This is especially useful in sectors where complex worldwide supply chains involving far-off and unreliable parties like producers, shipping firms, freight forwarders, port operators, and customs agencies are commonplace. Inventory management and supply chain financing are two other promising uses for Blockchain technology and other DLTs in this industry, particularly in areas where small and medium-sized enterprises are important players in warehouse operations, container transportation, and customs clearance services (Vyas, Beije, and Krishnamachari, 2022).

In essence, creating open digital platforms that can standardise and streamline information flows adds a lot of value. These platforms may be linked to global quality tracking systems so international quality assurance efforts can be coordinated and compliance records can be kept for

geographically dispersed parties. With Blockchain systems, diverse assets could be registered in a robust and shared manner. To improve decision-making and expedite reactions to potential infractions, these could be employed by customs officials, rights holders, and logistics operators (Beije, Feyen, and Frijters, 2023; Tagarev, 2023).

To provide an example, if papers like conventional bills of lading for shipments were incorporated into a Blockchain, they could be safely filed, verified, and approved by a variety of parties, such as shipping companies, government agencies, terminal companies, and others. Information on the origin of the items, tariffs, classification information, documents related to import and export, invoices, lists of loadings, and status changes are just a few examples of the information that these stakeholders frequently need to share (Tagarev, 2023). There is no doubt that addressing counterfeit, illegal materials and hazardous compounds is a must in order to reduce the costs and obstacles faced by regulators, industries, and producers globally. There is an urgent need for effective and responsible practices that can be implemented in this field in order to ensure that global supply chains are safe and that health and safety laws are enforced.

3.7.4 Biological Products and Health

Blockchains and similar DLTs have garnered significant interest in the health and biopharmaceutical sectors. They are currently being explored for a wide range of applications, including but not limited to electronic medical record management (Mayer, C. A. d. Costa, and Righi, 2020), identity verification (Liang, 2019), data validation and exchange (Akkaoui, Hei, and Cheng, 2020), establishing payment infrastructures for pre-authorisation, managing insurance claims, and preventing as well as detecting counterfeit drugs, among numerous other possibilities (Mazlan et al., 2020).

Within this context, patients, doctors, hospitals, and other healthcare professionals have the potential to utilise decentralised management systems based on Blockchain for storing electronic health records. In these systems, individuals can encrypt their personal and sensitive information, allowing access solely to authorised parties who possess appropriate credentials (Vora et al., 2018). Patients, in particular, could implement dynamic consents using smart contracts. These

dynamic consents enable patients to specify the terms of data access, including the type of data to be shared, the intended purposes for its use, the entities authorised to access it, and conditions for revocation (Madine et al., 2020). This innovation has the capacity to foster the emergence of novel business models that prioritise privacy preservation, enable personalised medicine, make data sharing easier for drug development, treatment planning, and health services research, or even support the buying, selling, and remarketing of health data among various stakeholders (Jaiman and Urovi, 2020).

3.7.5 Creative Industries

In the expansive realm of creative industries, encompassing digital knowledge and information, Blockchain and other DLTs have the potential to find applications in the management of intellectual property and digital content. This extends to various forms of digital creations such as books, music, art, games, photos, texts, and more. A significant proportion of individuals engaged in creativity industries around the world typically operate either as sole practitioners or within small teams, facing constraints in terms of time and resources when it comes to handling the administrative and legal facets of their endeavors. This leaves little room for activities like research and innovation development, not to mention the acquisition of skills necessary in a rapidly evolving digital landscape. The potential for Blockchain technologies to alleviate this burden holds profound implications (Rennie, Potts, and Pochesneva, 2019).

The establishment of ownership and sublicensing rights, the facilitation of payments and financial transactions, the recording of metadata pertaining to the creation and consumption of content, and the implementation of authentication systems to determine the value and reliability of information constitute a few examples of Blockchain applications. Here, a modular strategy is used in an effort to develop an open, transparent meta-system that works with separate systems. These unique systems are not only created to handle particular problems for which they are intended, but also to work in perfect harmony with one another. By employing open standards and easily accessible data that run on Blockchain platforms, this interoperability is made possible. Furthermore, in a multi-stakeholder paradigm, this strategy is consistent

with independent certification and regulatory frameworks (Patrickson, 2021; Rennie, Potts, and Pochesneva, 2019).

3.7.6 Energy

Numerous applications within the energy industry show promise in the use of Blockchain technology as well as other DLTs. These applications cover a wide range of tasks, including managing smart grids and microgrids, enabling P2P energy trading, facilitating microtransactions and micropayments, tracking energy consumption and production, acquiring renewable energy, and managing the infrastructure for charging electric vehicles (Alladi et al., 2019; P. W. Khan and Byun, 2021). Additionally, the energy industry can benefit from using Blockchain to safely store ownership records for energy flow and company activities. This covers crucial elements like carbon emission permits, certificates for renewable energy, and the condition of devices like smart meters, energy grids, and production facilities. Blockchain can serve as a decentralised coordination infrastructure to benefit microgrid energy markets. With this, individual consumers can directly exchange locally generated green energy, from sources like solar panels or windmills, in these markets with people in their neighborhoods, all while taking advantage of near real-time pricing dynamics (Bao et al., 2021).

Thus, Blockchain has the potential to reshape renewable energy markets by enhancing trust, improving transparency, and enabling more dynamic pricing models to cope with the inherent unpredictability of renewable energy generation.

3.7.7 Information Technologies

Various information technology applications can benefit from Blockchains and related DLTs. A number of web services and programs are available, including mesh networking, decentralised file systems, cloud storage, peer-to-peer and encrypted communications, and several web services. Decentralised storage protocols connect numerous users who keep copies of files on each other's computers or devices, as opposed to depending on centralised servers for file storage and access. The integration of Blockchain with decentralised file storage systems, which permits

the sharing of data through a peer-to-peer network, is another important use of Blockchain in the IT industry (Kang, W. Yang, and J. Zheng, 2022). Whether their data is stored on their own computers or on particular cloud servers, decentralised web apps like domain names, identity management, and data storage give users control over where their data is stored. They can choose the criteria for accessibility. According to Ali et al., user-generated data can, for instance, be initially saved locally on users' personal computers and then encrypted in order to be stored as a backup in a cloud storage system if it is needed (Ali et al., 2018).

Blockchain's decentralisation and digital key features can also benefit mesh networks. Networks like these consist of local, wireless connections, like antennas, bridges, switches, and other hardware devices, which are all decentralised nodes or wireless connection points. It is common for users to only divulge a small percentage of their profiles or the specific data required for applications and services, particularly if they are built on decentralised platforms (Kabbinale et al., 2020).

3.7.8 Advanced Manufacturing

The possibilities of Blockchain and other distributed ledger technologies for advanced manufacturing are endless. Some of them include asset sharing, the management of distributed value and supply chains, the automation of production processes in agile or smart factories, the tracking of digital representations of products, product life cycle management, certification, and authentication, just to name a few. Manufacturing processes can be improved by combining Blockchain technology with other digital technologies, such as IoT, AI, and robotics in order to support the process.

It is possible to store all kinds of information, such as physical properties, design specifications, used materials, ownership, location of manufacture, maintenance history, certifications, and guarantees, on a Blockchain as an encrypted and immutable digital record. As a tamper-proof record of ownership of digital files in additive and subtractive manufacturing environments, Blockchain can function as a safeguard against unauthorised use, theft, and infringement of digital files (Mandolla et al., 2019; Zuo, 2021). The business might buy a digital file, transmit it

over Blockchain, and confirm the 3D printer vendor and printers closest to the ultimate point of assembly or production. In order to build new business models in the future, it will essentially be a data-driven ecosystem (Klöckner et al., 2020). Through the use of embedded serial numbers and IDs, Blockchain technology can also verify that every produced part was created, installed, and maintained in accordance with all applicable guarantees, licenses, and standards (Westerkamp, Victor, and Küpper, 2020). Manufacturing processes could be made more lean by having parts available at the moment of demand and at the location of need. There is a possibility that smart contracts will one day be able to identify and negotiate the best production facilities automatically based on availability, price, quality, delivery, or location in the near future.

3.8 Blockchain in Railway Industry

The railway sector belongs to the transportation industry, and all the Blockchain solutions discussed in subsection 3.7.3 that are applicable to transport and logistics also hold relevance for railways. In this section, a review of the literature will be provided, highlighting the advantages of Blockchain technology in the railway industry. It will offer an overview of the specific applications of this technology in the railway sector prior to introducing the proposed system in this thesis.

Blockchain is considered advantageous in rail logistics (Naser, 2018; J. Preece and J. Easton, 2018), as well as in supply chain management and the implementation of logistics information products (Levina et al., 2021). Within the realm of rail logistics, Blockchain is expected to not only cut down on time and costs but also to improve transparency (Shirani, 2018). Furthermore, research suggests that the immense potential of Blockchain in rail logistics can be enhanced by integrating it with other technologies such as digital twins (Sahal et al., 2021) and smart wagons (Figueroa-Lorenzo et al., 2021).

One study, for instance, focused on Blockchain's ability to distribute data among different parties and improve interoperability between various IT systems, offering a solution to longstanding data-sharing challenges (J. Preece and J. Easton, 2018).

In the context of railway passenger transport, various studies highlight the potential roles of Blockchain in Mobility as a Service (MaaS) applications, emphasising its beneficial impact on digital ticketing (J. Preece and J. Easton, 2019; Naser, 2018; Xiaodong, Ping, and Xiaoning, 2020). Blockchain is anticipated to enable door-to-door passenger transport through a digital platform, facilitating the creation and payment of a unified electronic ticket as well as the delivery of supplementary services, all without the involvement of intermediaries and operators (Gulyi, 2020; H. Yang et al., 2022). Moreover, Blockchain is projected to enhance operations by providing better information-based solutions for existing traffic management, signalling, and maintenance systems (J. M. Easton, 2021). The study by Zhu et al. in (Zhu et al., 2021), explores the application of Blockchain for identity authentication and its advantages for Communication Based Train Control (CBTC) systems, specifically in replacing the existing centralised key management system that is vulnerable to single-point failures. Within urban rail transit, this technology can establish a foundation for precise rail transit passenger flow predictions (Shen et al., 2020) and help reduce associated supervision and management expenses.

3.8.1 Shift2Rail: Funding Research and Innovation Projects in the Railway Industry

The European Union (EU) created Shift2Rail as a key project to foster innovation and boost competitiveness in the railway industry. The program aims to accelerate the implementation of modern technologies and improve efficiency, capacity, and overall user experience on Europe's rail networks[8]. Shift2Rail's beginnings may be traced back to 2014, when the European Commission introduced the initiative as a component of its Horizon 2020 research effort. The aim was to improve the efficiency, capacity, and all-around appeal of the European rail network. The project was viewed as an essential step in fulfilling the long-term transport goals of the EU, including sustainability, effectiveness, and seamless mobility. Initially, Shift2Rail's Innovation Programme was divided into five main areas, or IPs, which span a variety of research disciplines and technical developments. These IPs are depicted in Figure 3.7 along with five interconnected

[8]https://rail-research.europa.eu/about-shift2rail/

themes that are relevant to each project (Haltuf, 2016).

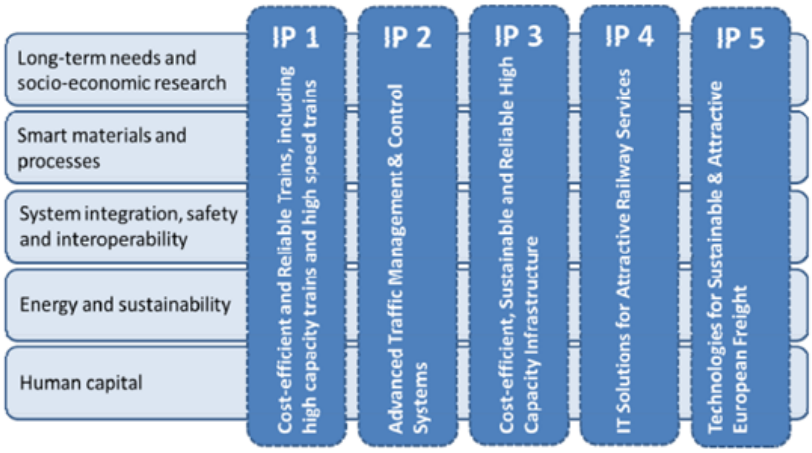

Figure 3.7: The five IPs of the Shift2Rail framework (source:Haltuf, 2016)

IPX is an additional innovation programme that emphasises emerging technologies and concepts with the potential to greatly influence the rail sector [9]. IPX is dedicated to exploring groundbreaking ideas and disruptive technologies that may not align with Shift2Rail's five IPs. By allocating a specific programme for innovative and transformative advancements, Shift2Rail ensures that the European rail industry remains competitive and ready for future developments. Blockchain technology was among the emerging technologies utilised to develop solutions in some projects.

INtelligent solutions 2ward the Development of Railway Energy and Asset Management Systems in Europe (IN2DREAMS): The IN2DREAM project was carried out within the IP3, and all associated documents and deliverables can be accessed on the project website [10]. The project is divided into two primary work streams: WS1, concentrating on energy-related data management, and WS2, focusing on asset-related data management. Blockchain was utilised in the asset management segment with the objective of "managing the maintenance jobs workflow through the application of smart contracts, automatically enforcing the terms and conditions (such as Service Level Agreement (SLA)) of the maintenance agreements between the Infrastructure Manager (IM) and the contractor." They discussed the scenario of monetising data in

[9] https://projects.shift2rail.org
[10] http://www.in2dreams.eu

the railway sector and the absence of legal frameworks for raw data ownership. Blockchain was proposed as a potential solution, albeit without any specific details.

Development of prescriptive AnalYtics baseD on aRtificial intElligence for iAMS (DAYDREAMS): This project is also included under IP3. One of its main objectives was to enhance trust through the application of Blockchain and intelligent technologies derived from IN2DREAMS to track the implementation and usage of Intelligent Asset Management Systems (IAMS) across multiple stakeholder environments (Oneto et al., 2023). The role of Blockchain is to monitor and trace digital artifacts and their utilisation. The Blockchain architecture they proposed consists of two smart contracts that will interact with each other. The interaction between the various parts of the IAMS prototype and the smart contracts is planned to occur via APIs or direct on-chain (Blockchain) communication. One smart contract will be automatically created based on the Business Process Model and Notation (BPMN) diagram that outlines the lifecycle process of the digital artifact. This enables stakeholders to follow the status of a digital artifact and access reliable information about it. The second smart contract is responsible for monitoring and recording all relevant runtime interactions involving digital artifacts during an Intelligent Maintenance Session. This smart contract will be linked or synchronised with the one that specifies the lifecycle of digital artifacts. Additional information is available and detailed in the deliverables and documents published on their website [11].

3.9 Conclusion

In conclusion, this chapter has explored the intricate landscape of Blockchain technology, tracing its evolution from early concepts to its diverse applications across various industries. We initiated the journey by defining Blockchain as a pivotal member of distributed ledger technologies DLTs, highlighting its fundamental characteristics such as decentralisation, immutability, and transparency. The historical context emphasised the foundational ideas that sparked the development of Blockchain, paving the way for innovations like Bitcoin and Ethereum.

[11] http://daydreams-project.eu

We delved into the core concepts crucial to understanding Blockchain: decentralisation, consensus algorithms, and smart contracts, laying the groundwork for grasping its functionality and appeal. The discussion on Hyperledger Fabric illustrated how tailored solutions can meet specific industrial requirements, particularly by emphasising their modular design and capacity for integration.

Moreover, by examining the applications of Blockchain in eight key sectors, including healthcare, logistics, and energy, we revealed their potential to improve efficiency, security, and traceability. The railway industry was notably highlighted, showcasing initiatives within Shift2Rail where Blockchain contributes to smarter asset and energy management, illustrating practical implementations of this technology.

In sum, Blockchain stands as a transformative force that extends beyond mere financial applications into realms that command innovation, transparency, and efficiency. As industries continue to adopt and adapt this technology, the collaborative development of standards and regulatory frameworks will be crucial to unlocking its full potential. Thus, we call for ongoing exploration and dialogue within the ecosystem to navigate challenges and optimise the integration of Blockchain technologies into our increasingly digital world. The future looks promising as we harness the power of Blockchain to reshape industries and improve processes on a global scale.

Chapter 4

IoT Integration and Use Cases

4.1 Introduction

The Blockchain application introduced in Chapter 5 focuses on enabling real-time data exchange. To support this functionality, a simulator has been developed in this chapter, designed to generate real-time data that is seamlessly integrated with the Blockchain application. Section 4.2 provides an overview of the foundational background necessary to understand the system. The use cases selected for the simulation phase are examined in detail in Section 4.3. Subsequently, the system architecture and workflow are presented in Section 4.4. The technical details of the development environment and the implementation process are outlined in Section 4.5. The chapter concludes with a summary of key points and closing remarks in Section 4.6

4.2 Background

Nowadays, integrating IoT in railway digitisation has been increasingly emphasised and considered a main facility to collect informative data that is used to support decision-making. The primary concern is how to exchange this generated data securely and fast to increase the efficiency of rail vehicle and infrastructure maintenance, particularly. Therefore, there is continued encouragement being given to developing systems that enable more accurate maintenance decisions to be made, thereby improving cost distribution and operation reliability in general. To

this end, ongoing condition-based monitoring is being designed to accurately support decision-making procedures by identifying the changes in any monitored asset condition over time. This will lead to more cost-effective and predictive maintenance by using this data to determine if there is an urgent and costly need for interventions to keep the performance and safety. However, the number of units or large areas that should be observed and monitored in a cost-effective way has become viable with the advent of low-cost sensors. The majority of current rail geometry measurement technologies are either manual or dependent on the use of specialised, devoted measurement vehicles. Since manual methods require placing railroad employees on the rail network, they are typically only taken into account where network possessions are necessary for safety. Possessions have a major negative impact on the capacity of rail networks, which makes them generally avoided wherever possible. Devoted measuring vehicles are often less disruptive since they may be planned around other running traffic, but they are expensive to construct, maintain, and run. Therefore, the shortcomings of both methods have a direct impact on the efficiency of the inspection procedure. The existing solution of the aforementioned difficulties is to include the normal traffic as a part of the condition monitoring measuring system. By installing affordable sensors and measuring equipment on in-service rolling stock, a rail geometry data collection system is made feasible to record and monitor the network during ordinary operations. Although this method may not be as accurate as manual inspection or specialised inspection vehicles, it is still possible to accurately represent the track geometry of the network on a daily basis by measuring a number of parameters because an in-service train frequently travels the same track section. Due to resource and expense constraints, it is highly challenging to achieve this using current measurement techniques. In the previous phase in the project, we built and introduced a framework to exchange payments for static datasets based on settled agreements and escrows as a means to protect against a variety of attack vectors and threats in the event of malicious behavior by data consumers or providers (R. A. Alzahrani, Herko, and J. M. Easton, 2020). This chapter focuses on introducing a model that demonstrates how we developed and integrated an IoT simulator with Blockchain technology to simulate real-time data generation. This integration enhances the reliability of condition monitoring, ensures effective and equitable

cost distribution, and promotes data sharing among stakeholders within IoT ecosystems. Also, the new proposed customised solution keeps fulfilling the same advancements we produced in the previous phase, which are

1. **Scalability:** The proposed solution has the ability to handle numerous connected devices in the IoT network without overburdening the Blockchain they are connected to. This is because only the hash value of the dataset generated from the sensors will be appended to the Blockchain network, while the raw data is stored externally.

2. **High throughput:** Our solution is built using a HLF network that is a permissioned Blockchain using RAFT as a distributed consensus algorithm.

3. **Lightweight:** Integrating the IoT devices in the proposed solution doesn't mean including them within the Blockchain network; instead, a RESTful interface will be specified to allow cross platforms communication.

4. **Transparency:** The details of IoT device resources, agreements, payment, and transaction history will be shared between involved parties.

4.3 Use Cases

Two use cases were selected to test the proposed model, both of which are ongoing projects developed by the Birmingham Centre for Railway Research and Education (BCRRE) at the University of Birmingham. These use cases offer practical, real-world scenarios that serve as a foundation for validating the model's effectiveness in remote condition monitoring.

The first use case employs multiple sensors attached to the railway track to remotely monitor the condition of the axle journal bearings (Mani Entezami and Whitehead, 2021). These sensors continuously collect data that can be analysed to predict potential issues before they escalate, thereby enhancing maintenance strategies and improving railway safety. The second use case involves sensors installed on trains that collect data as the train travels over the track. These data

are processed to evaluate the track of health and identify degradation patterns over time (Roberts et al., 2019). Each use case will be explored in greater detail in the subsequent subsections.

4.3.1 Switch and Point Machine Monitoring System (Infrastructure Monitoring Train)

Switches and crossings (S&C) are critical components of the railway infrastructure and are frequently cited as primary causes of point failures and motor malfunctions. In recent in-service monitoring, it was observed that the swing-nose crossing of Point 2076, located on the down line at Stratford, was subjected to some of the highest impact forces on Network Rail's High-Speed NH (HS) managed infrastructure. These impact forces contribute significantly to the deterioration and failure of the track, making it essential to monitor and predict the dynamic forces exerted between the wheel and the rail.

To address this issue, a remote condition monitoring solution was developed, incorporating a range of sensors and measurement devices strategically placed around the swing-nose crossing and associated point-operating equipment. This system facilitates continuous monitoring, enabling the early detection of anomalies that could lead to failure, thus allowing proactive maintenance interventions.

The remote monitoring system is divided into two components. The first part is installed at the swing-nose of the 2076S crossing, where dynamic forces are most intense. The second component is located at the tip of the switch, ensuring comprehensive monitoring coverage. Figure 4.1 illustrates the layout of the system, highlighting the positioning of the sensors and their respective functions in providing real-time data for infrastructure health monitoring. In general, 17 sensors are deployed throughout the monitoring system and divided as follows:

1. **MEMS Accelerometer Sensors:** 10 MEMS Accelerometer sensors are used to measure the vertical movements and vibrations of the bears across the swing-nose. Two accelerometers are positioned on the bearers supporting the swing-nose point machine, while the remaining eight are distributed across the other bearers of the swing-nose crossing. This

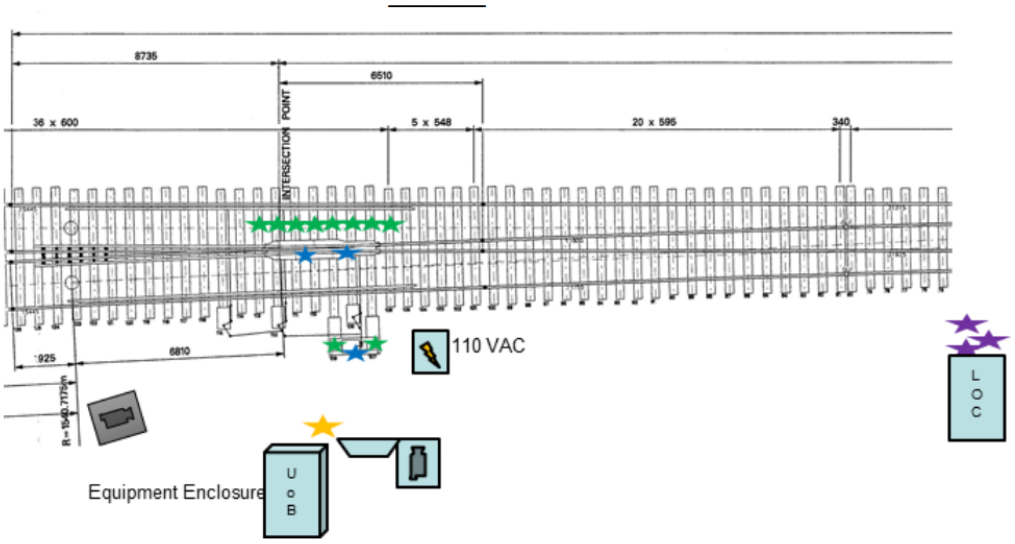

Figure 4.1: Overall system and sensors distribution (Mani Entezami and Whitehead, 2021)

configuration is depicted in Figure 4.1.

2. **Piezoelectric Accelerometers:** 3 piezoelectric accelerometers are employed for monitoring purposes. Two of these sensors are tasked with detecting wheel-rail impacts, while the third records the vibrations of the point machine.

3. **Ruggedized Free-Field Microphone:** A specialised microphone is installed to measure sound levels, facilitating the extraction of audio signatures that could indicate mechanical anomalies.

4. **Miniature AC Current Clamps:** 3 miniature AC current clamps are used to monitor the three-phase electrical currents powering the point machine, providing crucial data on its operational performance.

4.3.2 Railway Track Monitoring Using Onboard Inertial Measurements (Train Monitoring Infrastructure)

The geometry of the railway track may deviate over time from its originally designed configuration due to a variety of factors contributing to track degradation. These factors include

- **Ballast Settlement:** The settlement of ballast may occur after the initial construction of the track-bed or through the repeated passage of rail vehicles over time, leading to changes in track support.
- **Environmental Factors:** Changes in weather conditions, particularly extreme temperatures, can result in infrastructure movement. For instance, the frost heave effect, as discussed in (Wu et al., 2024), occurs when water between the ballast particles freezes, causing the ballast layer to expand and shift the track position.

Uneven ballast settlement often causes certain sections of the track to be supported more than others, creating variations in the vertical track profile. This results in diminished ride quality and is one of the key indicators of track degradation. Additionally, accurate measurement of distance along the track is critical in condition monitoring to assess overall track health and identify potential faults. Several studies have explored the use of Inertial Measurement Unit (IMU) sensors mounted on axle boxes or bogies of rail vehicles to monitor track irregularities, highlighting the sensor's effectiveness in detecting these deviations (Westeon et al., 2007; PF Weston et al., 2007. A research team at the University of Birmingham has developed a track measurement system designed to be installed on in-service rail vehicles. This system incorporates a compact IMU to continuously monitor and collect daily measurements of the track. These real-time measurements are compared against historical data, allowing for rapid detection of any track degradation. The flexibility of the system allows it to be easily integrated into any in-service vehicle, eliminating the need for dedicated and costly measurement trains. Using in-service vehicles, the system can cover extensive sections of the rail network, significantly reducing operational expenses. The IMU sensor used in this system is a MEMS-based device capable of measuring both rotational velocities and accelerations over time. This is achieved through a 3-axis configuration of accelerometers and gyroscopes, offering a six-degree-of-freedom model. In addition to inertial data, the system also integrates Global navigation satellite system (GNSS) and tachograph data when available, storing it locally on an SD card or transmitting it to an onboard computer for further processing. Subsequently, these data are analysed to extract valuable insight into track conditions. The results published from this project

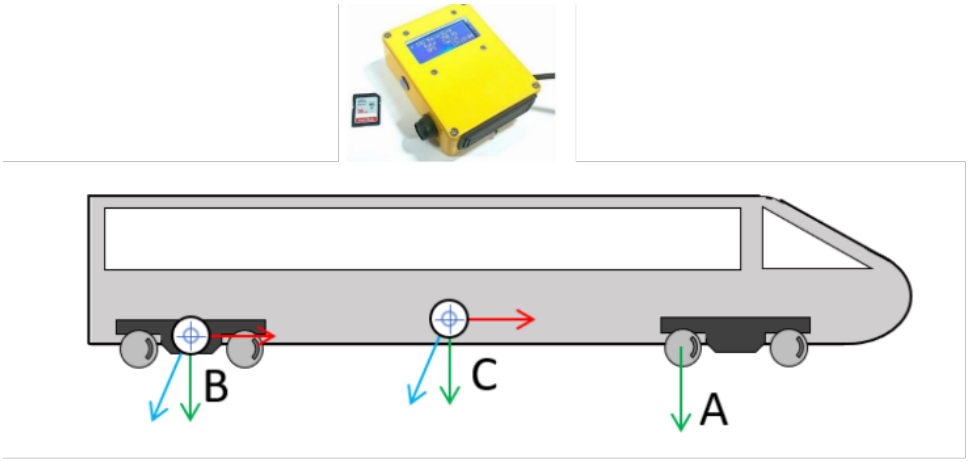

Figure 4.2: The locations on the train to fit the IMU devices (Gonzalo, Entezami, Roberts, Paul Weston, Stewart, et al., 2022)

highlight the critical role of track geometry measurement in maintaining railway infrastructure. Specifically, the study demonstrates how IMU sensors, mounted on bogies or elsewhere on the train, can effectively identify issues related to ride comfort and track degradation (Roberts et al., 2019; Gonzalo, Entezami, Roberts, Paul Weston, Yeo, et al., 2022; Gonzalo, Entezami, Roberts, Paul Weston, Stewart, et al., 2022). The sensors were installed at three different locations on an in-service Class 377 train, as listed below and shown in Figure 4.2.

- Axlebox IMU (A_IMU).
- Bogie and cab GPS (B_GPS , C_GPS).
- Bogie and Cab IMU (B_IMU , C_IMU).

4.4 System Design and Workflow

4.4.1 Conceptual Design of the Proposed IoT Simulator Platform

The connection flow between the end user and IoT devices for obtaining generated data is depicted in Figure 4.3, which outlines the key components of the developed system:

- **Users:** The system users include administrators, data consumers, and device owners (data providers). Consumers and providers participate in data trading, where the data

generated by the IoT devices is exchanged under the control of a Blockchain network. The Blockchain serves as the central coordinator, maintaining an immutable log of all data-sharing transactions, governed by the business logic embedded in the smart contracts. The system administrator is responsible for overseeing and managing the Blockchain network and the smart gateway program.

- **Blockchain:** The Blockchain is the core component of the system, executing the business logic through several smart contracts that regulate the data exchange process as further discussed in Chapter 5.

- **Smart gateway:** Since IoT devices are typically resource-constrained and cannot act as peer nodes in a Blockchain network, the smart gateway serves as an intermediary. It bridges the IoT devices and the Blockchain, ensuring the data generated by the devices is routed to the Blockchain and managing communication to mitigate direct pressure on the IoT devices.

- **IoT devices:** These devices generate data or resources, which are transmitted to the smart gateway for further processing and integration with the Blockchain network.

- **IoT server:** Many services shall be provided through the IoT server, such as interacting with the smart gateway, gathering the generated data from all sensors, hashing data and appending hashes to the Blockchain, storing data to the database, and processing all kind of commands to perform operations on sensors. There are many communication protocols that can be applied by the local bridge to connect devices to the server, such as Bluetooth, ZigBee, WiFi, and 2G/3G/4G cellular.

- **Storage:** The system utilises two types of storage. The first is Blockchain storage, which holds a comprehensive and immutable record of all transactions, data trading operations, and data hashes. The second type is traditional storage, such as hardware (hard drives) or software (databases), used to store the raw data collected from the IoT devices.

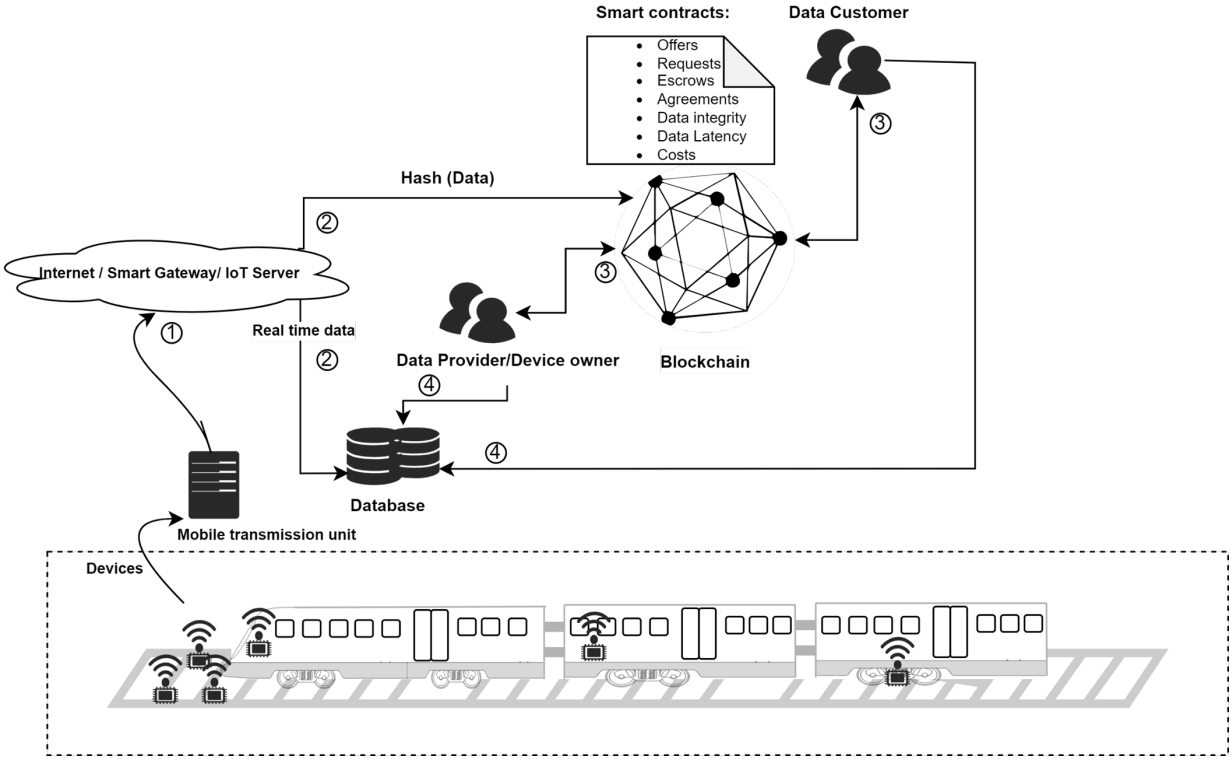

Figure 4.3: Proposed system conceptual design.

4.4.2 Proposed Blockchain-based IoT Platform Interaction Model

The workflow of the proposed Blockchain-based IoT platform is depicted in Figure 4.4. The model outlines a series of steps aimed at facilitating secure data trading and ensuring equitable cost distribution among stakeholders. Building upon the initial framework proposed in (R. A. Alzahrani, Herko, and J. M. Easton, 2020), several modifications have been introduced to adapt the system for real-time monitoring and direct data exchanges within an IoT environment. These refinements focus on enhancing the system's interaction model, improving the interfaces, and optimising operational dynamics to ensure greater efficiency and reliability in the data exchange process. The key steps of the workflow are outlined briefly below:

- **Step 1: Initialisation of the Blockchain Network:** The Blockchain network must first be initialised, and the chaincode must be installed on all endorsing peers. These fundamental tasks are managed by the system administrator. Establishing the fabric network requires configuring the intranet and initialising all fabric components as defined in YAML configuration files. These files contain vital information such as the orderer nodes, the consensus

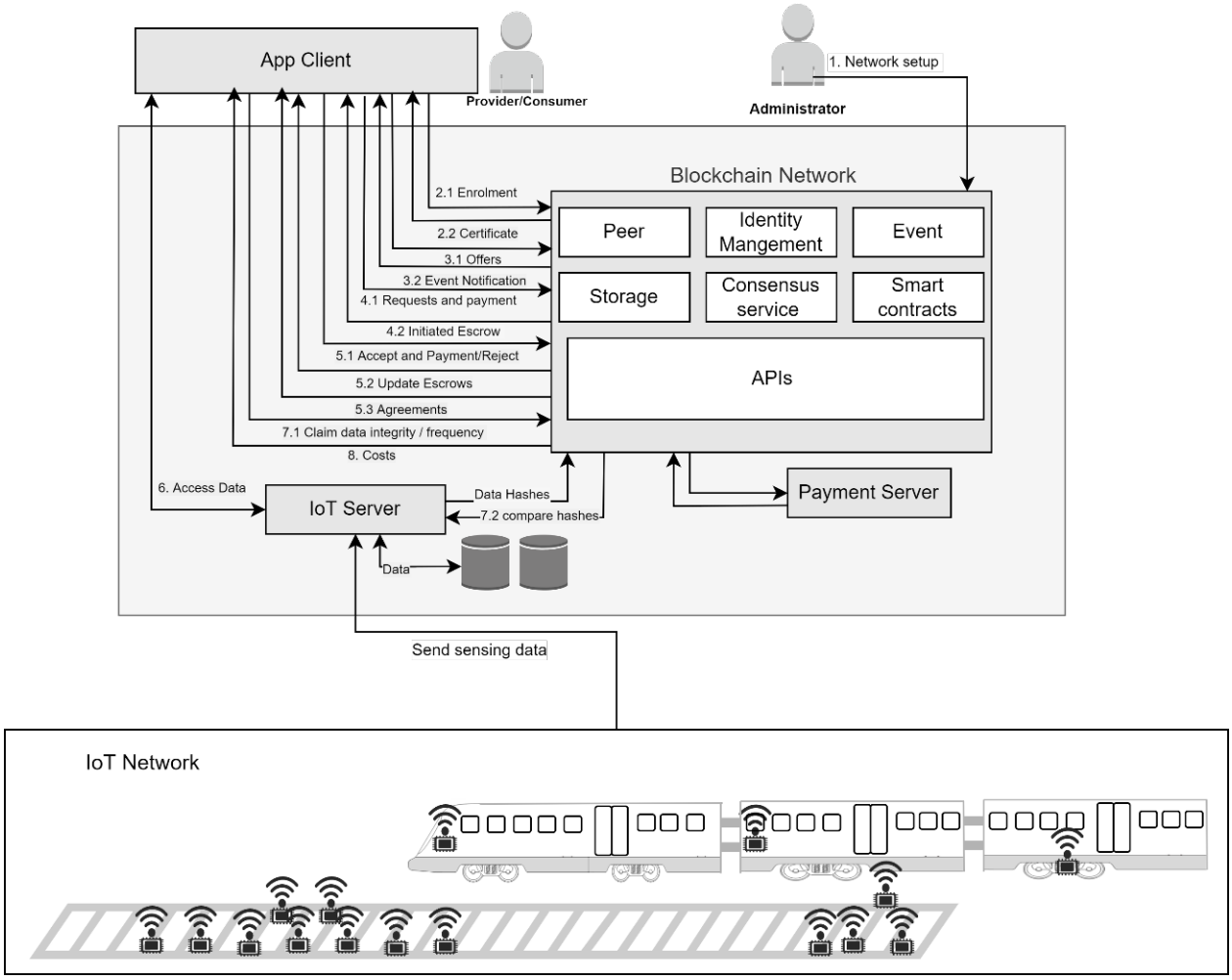

Figure 4.4: System workflow of the proposed platform.

algorithm in use, and the peers. Once the network is configured, the administrator creates a channel and uses the SDK to install and initialise the chaincode that governs the business logic of the system.

- **Step 2: User Enrollment and Certification:** Before users can interact with the Blockchain and submit transactions according to the business logic defined in the chaincode, they must be enrolled in the system. When users enroll, they obtain certificates with a key pair: private keys are necessary for digitally signing transactions, while the public key is openly shared and used by others to verify the user's digital signature.

- **Step 3: Transaction Submission and Service Request:** After enrollment, users can submit transaction proposals to the Blockchain network to consume the services defined in the smart contracts. Data providers specify which sensors they offer for access, while consumers can browse available offers and submit requests for the desired sensors and access duration. These requests remain pending until the escrow is initiated and the payment is processed.

- **Step 4: Payment and Escrow Initiation:** To proceed with the agreement, the consumer initiates the payment process by sending the request. The payment is made to an escrow account monitored by the Blockchain network, with designated accounts for all participating parties. Once the payment is processed, the escrow account holds the funds until the agreement is confirmed.

- **Step 5: Provider Notification and Escrow Lock:** Upon receiving the consumer's request, the Blockchain notifies the provider, who must respond to lock the escrow. The provider has the following options:

  - Accept the request by processing the deposit payment, which activates the agreement.
  - Reject the request, in which case the escrow will refund the payment to the consumer.
  - Neglect the request until the start date passes, which is treated as an implicit rejection.

If the provider processes the payment, the escrow is locked, and subsequent steps proceed. Otherwise, the escrow is released, and the consumer is refunded, terminating further communication between the consumer and the provider.

- **Step 6: Data Access and Sharing:** Upon activation of the agreement, the consumer is allowed access to the sensor data as outlined in the terms of the agreement. Data generated during the active period is shared with the consumer, and the sensor ceases data transmission upon agreement expiration. The sensor then sends the raw data to the provider, while the hash of the data is recorded on the Blockchain. Time stamps are included in the agreement to prevent unauthorised reuse of the same agreement for future sensor access.

- **Step 7: Claims Processing and Cost Distribution:** The platform supports a robust SLA framework, allowing the consumer to claim issues related to the integrity of shared data or delays in transmission that deviate from the expected schedule outlined in the agreement. This ensures a fair cost distribution and transparent resolution of disputes.

The steps described above will be elaborated further in the following Chapter 5, providing a more detailed explanation of the system's operation and interaction model.

4.4.2.1 Data Integrity

To maintain the integrity of the data, the data produced by the sensors should be hashed, and these hashes should be recorded on the Blockchain. Hashing allows for the detection and prevention of data manipulation. In both use cases, the consumer and the provider have direct access to the same sensor data, and both receive identical data streams. By hashing the data on the server prior to committing them to the Blockchain, any external or internal tampering can be traced by comparing the current data against the stored hash values. This ensures that any alterations, whether intentional or due to system breaches, can be identified and allocated for further investigation.

4.4.2.2 Data Frequency

Data transmission delays can be monitored by comparing the time-stamped hash values with the predefined data generation schedule set forth in the smart contract. If the consumer experiences delays in receiving data, the timestamp of the appended hash can be cross-verified against the scheduled transmission times. To account for potential latency in the IoT environment, where sensors could face delays due to data processing and network transmission, a predefined margin for allowable delays must be established in the agreement. This delay buffer ensures that the expected data delivery time is realistic, considering factors such as mobile network latency (Georg et al., 2020).

4.4.2.3 Cost Distribution

Achieving fair cost distribution between the data provider and consumer involves the integration of multiple smart contracts, including claim, escrow, and agreement. Blockchain automation ensures that once an agreement is expired or revoked, the escrow smart contract calculates costs by referencing the claims and agreement smart contracts. The costs are then distributed to both the provider and the consumer.

In the IoT environment, the cost distribution process is rigorously managed by the Blockchain, with adjustments made to account for the scheduled data transmission times defined in the agreement. Data transmission is synchronised with the scheduled train journeys (i.e., train timetables). When the journey schedule is made accessible to all participants in the network, the consumer can estimate the number of data transactions and corresponding requests that can be generated based on the published timetable. The frequency of data capture is directly dependent on the number of scheduled train journeys per day, making the publication of the timetable crucial for forecasting the total volume of transactions between the provider and consumer.

Consequently, cost is calculated based on the actual number of transactions that occur within the specified timeframe, as outlined in the agreed-upon agreements.

4.4.2.4 Historical Data

Handling historical data falls outside the scope of real-time IoT processing, as IoT devices are not designed to store or manage large volumes of historical information. Requests for historical data must therefore be addressed outside the IoT environment, utilising dedicated systems that are capable of long-term data storage and retrieval. The detailed process for handling historical data will be elaborated in the following Chapter 5.

4.5 Simulator Development and Implementation

4.5.1 Development Environment

The platform introduced in this study is composed of three fundamental components: the Blockchain network, the IoT device simulator, and the client application. A version of the code is available on GitHub[1]. In the context of the implementation, the IoT sensors have been simulated to mimic the data collection process and demonstrate how the collected data is stored within a database. Concurrently, the hash value of the generated data is recorded on the Blockchain, ensuring a secure and verifiable mechanism for data integrity.

The development stack utilised for the implementation of the Blockchain network is outlined in Table 4.1. The development environment operates on the Ubuntu Linux 18.04.4 LTS operating system, with hardware specifications including an Intel Core i7-8650U processor clocked at 1.90 GHz and 15.5 GB of memory. The runtime environment is facilitated through Docker Engine version 19.03.8, with the configuration and orchestration of Docker images and containers managed via Docker Compose version 1.23.2.

The platform leverages the open-source Blockchain framework HLF version 2.2. To build the HLF network, the Node.js runtime environment (version 16.20.2) is employed alongside the HLF Software Development Kit (SDK). Smart contracts are implemented using the Go programming language (Golang), which facilitates the execution of automated agreements within the Blockchain framework. Additionally, CouchDB serves as the state database, enabling the

[1] https://github.com/Rahma-Alzahrani/HLFProjectFinal

storage of current state values of the logs and supporting complex querying capabilities within the network. This combination of technologies underscores the robustness and scalability of the platform in ensuring secure and efficient data management in a Blockchain-based IoT environment.

Component	**Description**
CPU	Intel Core i7-8650 @ 1.90 GHz
Memory	15.5 GB
Operating System	Ubuntu Linux 18.04.4 LTS
Docker Engine	Version 19.03.8
Docker-Compose	Version 1.23.2
Node	V16.20.2
Hyperledger Fabric	V 2.2
IDE	Visual Studio code
DBMS	Couch DB, MongoDB
Programming Language	Node.js , GoLang

Table 4.1: Blockchain network development environment.

Table 4.2 outlines the development tools employed in the implementation of the customised IoT platform. The primary Integrated Development Environment (IDE) used for this purpose is the Spring Tool Suite (STS), an open-source Eclipse-based development environment. STS is particularly well-suited for Java-based development, making it highly effective for the creation of web applications and, in this case, for building a customised local IoT platform.

Communication between the simulated IoT devices and the server is facilitated by Redis, a high-performance in-memory data structure store, which ensures efficient and real-time data transmission. Meanwhile, the interaction between the IoT device server and the Blockchain network is achieved through HTTP protocols, ensuring seamless integration of the two systems.

For testing and validation purposes, the IoT simulators were supplied with data samples derived from real-world devices, enabling a rigorous assessment of the platform's performance under realistic conditions. The application provides an intuitive user interface, allowing clients to simulate the process of data generation by the IoT devices. This data is subsequently relayed via a gateway server (implemented using Redis) to the Blockchain network, as depicted in Figure 4.5. This architecture enables a clear demonstration of how device-generated data can

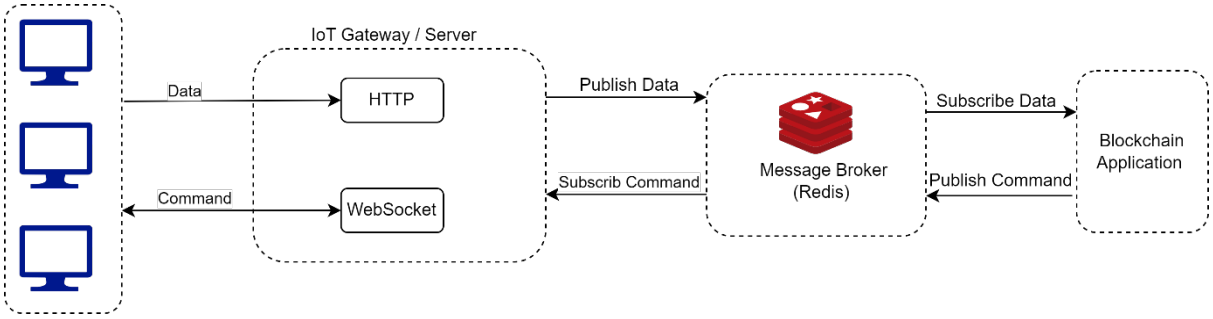

Figure 4.5: The IoT simulator integration with Blockchain

be securely transmitted and verified within a Blockchain framework, further enhancing the platform's applicability in real-world IoT scenarios.

Component	**Description**
Server/Gateway	Redis
Memory	15.5 GB
Operating System	Ubuntu Linux 18.04.4 LTS
IDE	STS
Transmission protocol	HTTP
Programming Language	JAVA

Table 4.2: Development stack of IoT simulator.

The client application is implemented as a web-based solution, consisting of two main components: the back-end and the front-end, developed using the tools specified in Table 4.3. The backend serves as the server responsible for handling the communication between the client-side application and the Blockchain network. It translates communication protocols and routes requests from the web application to the Blockchain and vice versa, acting as the intermediary layer that facilitates the interaction between the user interface and the underlying Blockchain infrastructure.

On the front-end, standard web technologies such as HTML, CSS, and JavaScript are used to create customised Graphical User Interfacess (GUIs). These interfaces are designed to provide an intuitive and user-friendly experience that allows users to interact with the system. The front-end communicates with the REST server through these interfaces, enabling users to invoke relevant APIs and submit transactions to the Blockchain network. By constructing and sending HTTP requests, the frontend ensures the proper execution of transactions, securely transmitting

them to the Blockchain for processing. This architecture allows for seamless interaction between the client application and the Blockchain, enabling users to effectively engage with the system's functionalities through a simple and accessible web-based interface.

Component	**Description**
Browser	Chrome, Firefox
Memory	15.5 GB
Operating System	Ubuntu Linux 18.04.4 LTS
IDE	Angular CLI V11.2.11
Transmission protocol	HTTP
Programming Language	Node.js , HTML ,CSS , JavaScript

Table 4.3: Web app development tools.

4.5.2 Execution Process and Results

The execution sequence illustrating the interaction between the developed IoT simulator and the Blockchain platform is detailed in Figure 4.6. The process begins with the consumer obtaining a processing agreement that grants access to the data generated by the IoT sensors. This agreement is based on predefined offers tied to a scheduled journey, which outline the specific start and end times during which data detection will occur.

The sensor through the simulator in our application will send the data via HTTP request to the IoT server using the POST method with the unique identifier of the device *(Device_id)*, which is used to associate the device with a broker. Once the server receives the data, it publishes the sensor data to the broker, which is implemented using Redis. Redis, acting as a message broker, then forwards the sensing data to the Blockchain nodes that have subscribed to the data stream.

On the Blockchain side, a hash value is computed for the incoming data. This hash is then recorded both on the Blockchain file system and in the state database, ensuring data integrity and traceability. While the raw sensor data itself is stored in an external storage system, the Blockchain maintains the hashed representation to secure the data's authenticity. Additionally, the Blockchain emits notifications to the relevant clients via WebSockets based on the terms of the active agreements. These notifications inform consumers about newly generated data, enabling them to access the information in real time as it is recorded on the Blockchain. This

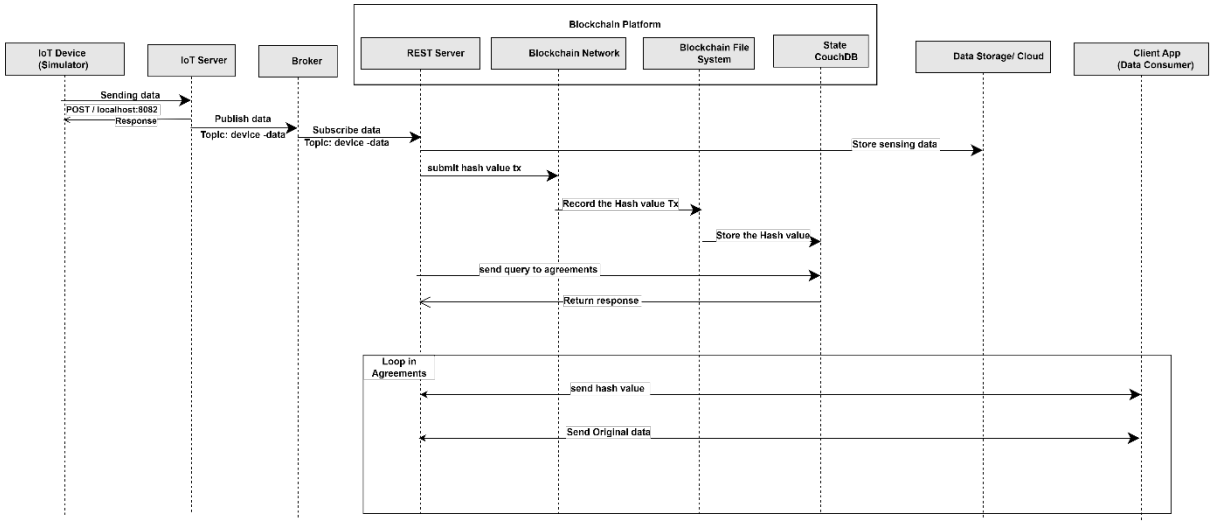

Figure 4.6: Sequence of sending data from sensor to Blockchain.

sequence highlights the efficient and secure exchange of sensor data between the IoT simulator and the Blockchain, ensuring both transparency and data integrity throughout the process.

4.5.3 Use Cases Simulation

In the first use case, a network of 17 distinct sensors operates in collaboration to collect various measurements, which are subsequently analysed to assess the condition of the axle journal bearing. These sensors, positioned strategically along the rail track, are configured to monitor the condition of the axle journal and transmit the data at predefined intervals according to a scheduled time frame, as depicted in Figure 4.7.

The back-end system simulates the operation of these 17 sensors, each of which collects specific condition data relevant to the axle journal bearing. At scheduled times, the collected data are transmitted to the central server, where they are processed and then published to a message broker, as illustrated in Figure 4.8, Figure 4.9, and Figure 4.10. The message broker, which acts as an intermediary, ensures the efficient distribution of the sensor data to the appropriate subscribers, including the Blockchain network, where the data can be securely stored and verified.

This process demonstrates how multiple sensors, working together, provide a comprehensive view of the axle journal bearing's condition. The integration of sensor data through the server

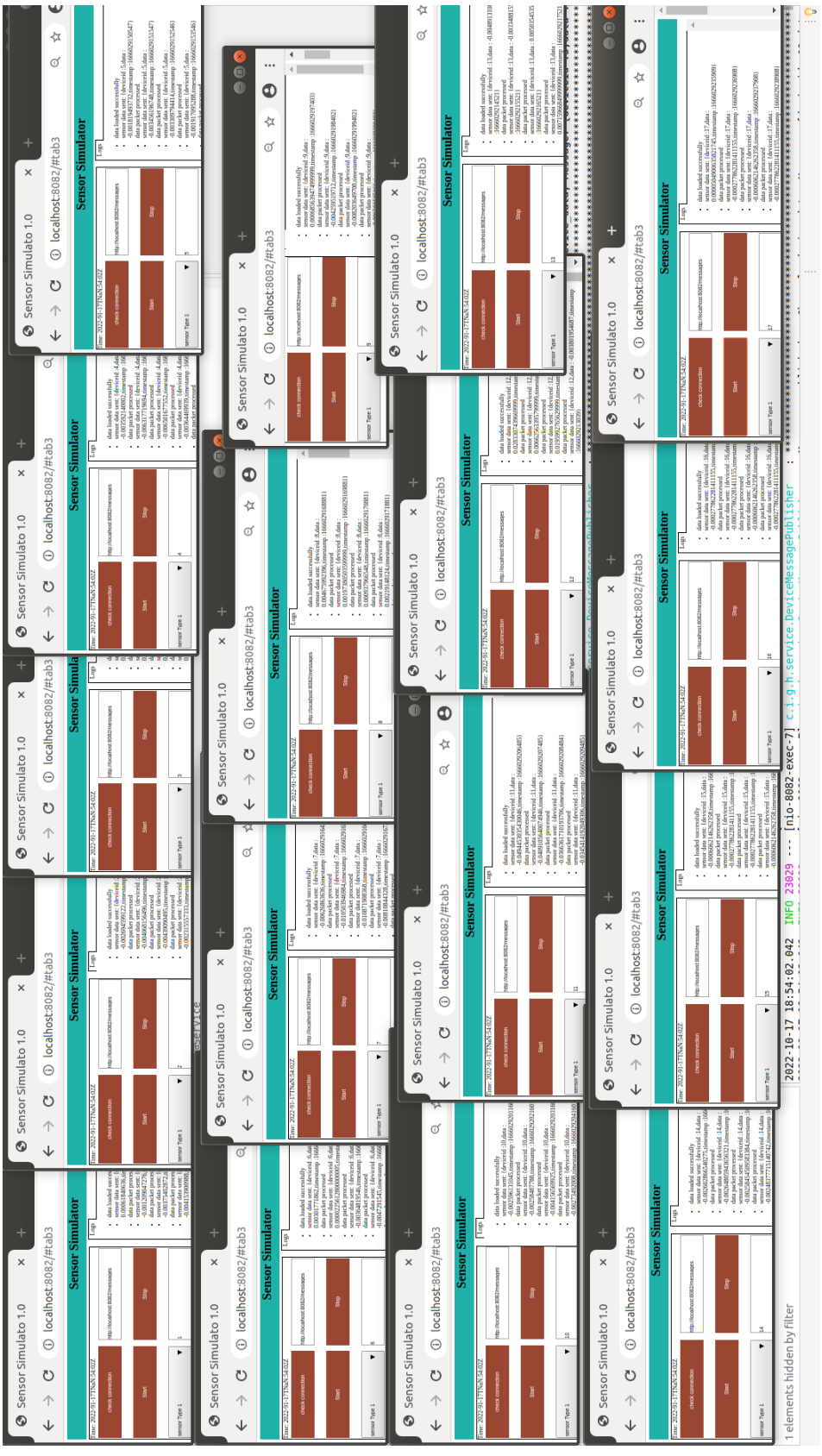

Figure 4.7: First use case sensors.

and broker enables real-time monitoring, ensuring that critical data is accurately transmitted, stored, and processed for further analysis, supporting predictive maintenance and infrastructure management strategies.

In the second use case, a varying number of sensors is employed, each utilising distinct data structures for their respective data streams, as illustrated in Figure 4.11. When these sensors generate data, the data stream triggers the Redis broker, which then forwards the stream to its subscribers, in this case, the Blockchain network. The Blockchain is responsible for calculating the hash value of incoming data and securely storing it in the database for authenticated users, ensuring both data integrity and secure access.

The communication between the developed simulator and the Blockchain is facilitated through the MQTT protocol. In this setup, all device nodes must listen to a specific communication channel and either subscribe to or publish messages/data to a designated topic, specifically named *(device_data)*. On the client side, the Blockchain node is configured to listen to the same channel and subscribe to the *(device_data)* topic, allowing it to receive messages whenever they are generated by the devices.

The messages triggered by the sensors are routed through the Redis broker and subsequently processed by the Blockchain. From there, they are redirected to the appropriate users based on the agreements established between data consumers and providers, as outlined in earlier discussions. Queries embedded within the smart contracts on the Blockchain facilitate the efficient extraction of data from the storage system. These queries allow users to retrieve specific data in accordance with the predefined conditions of their agreements, ensuring secure and transparent access to sensor-generated data through the Blockchain infrastructure.

4.6 Conclusion

In conclusion, this chapter has provided an in-depth analysis of the development and integration of an IoT simulator with Blockchain technology, addressing the critical demand for efficient and secure data exchange within the framework of railway digitisation. Through an exploration of

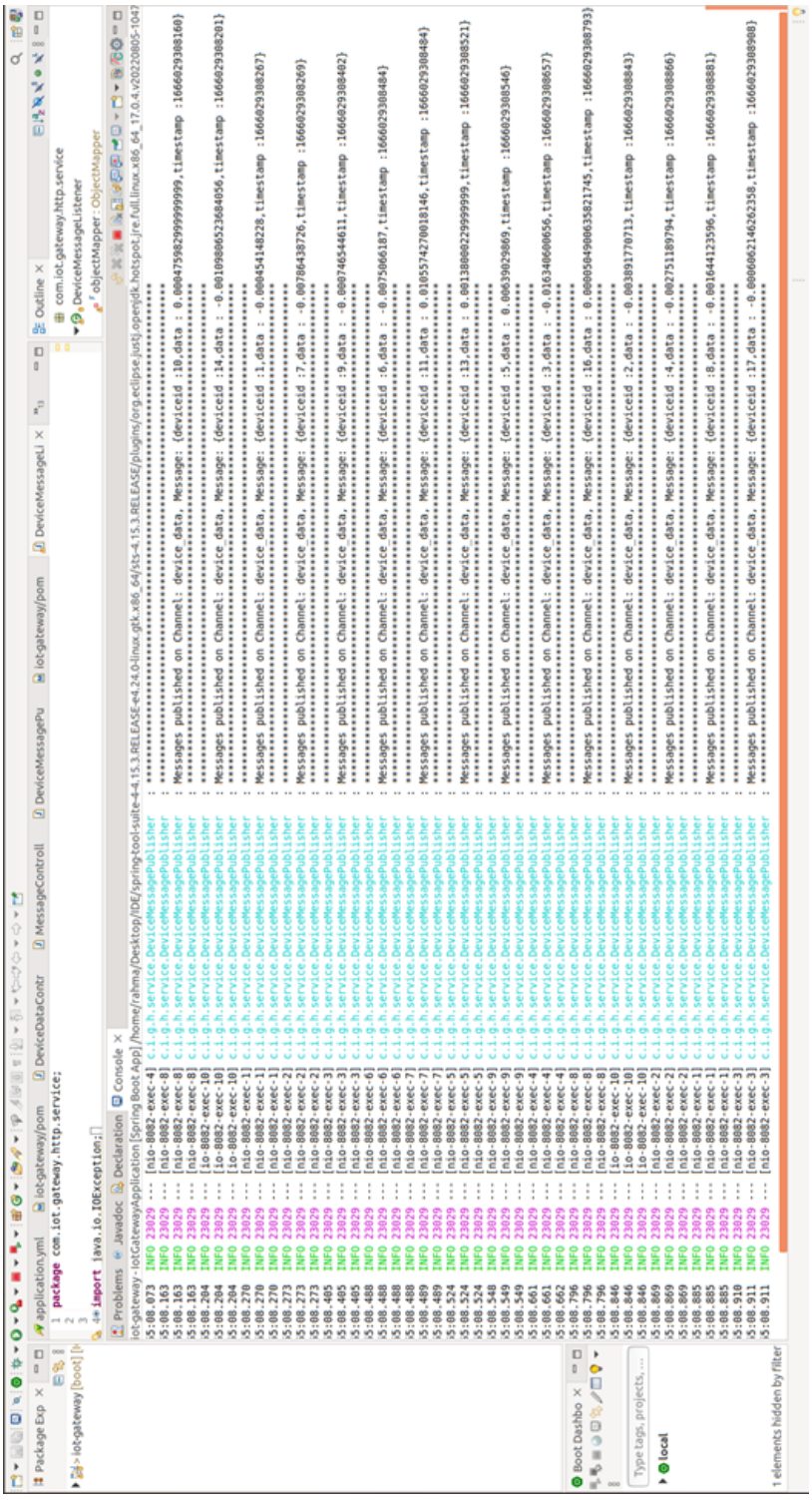

Figure 4.8: Publishing data from sensors in the first use case.

Figure 4.9: Publishing data from sensors in first use case (continued).

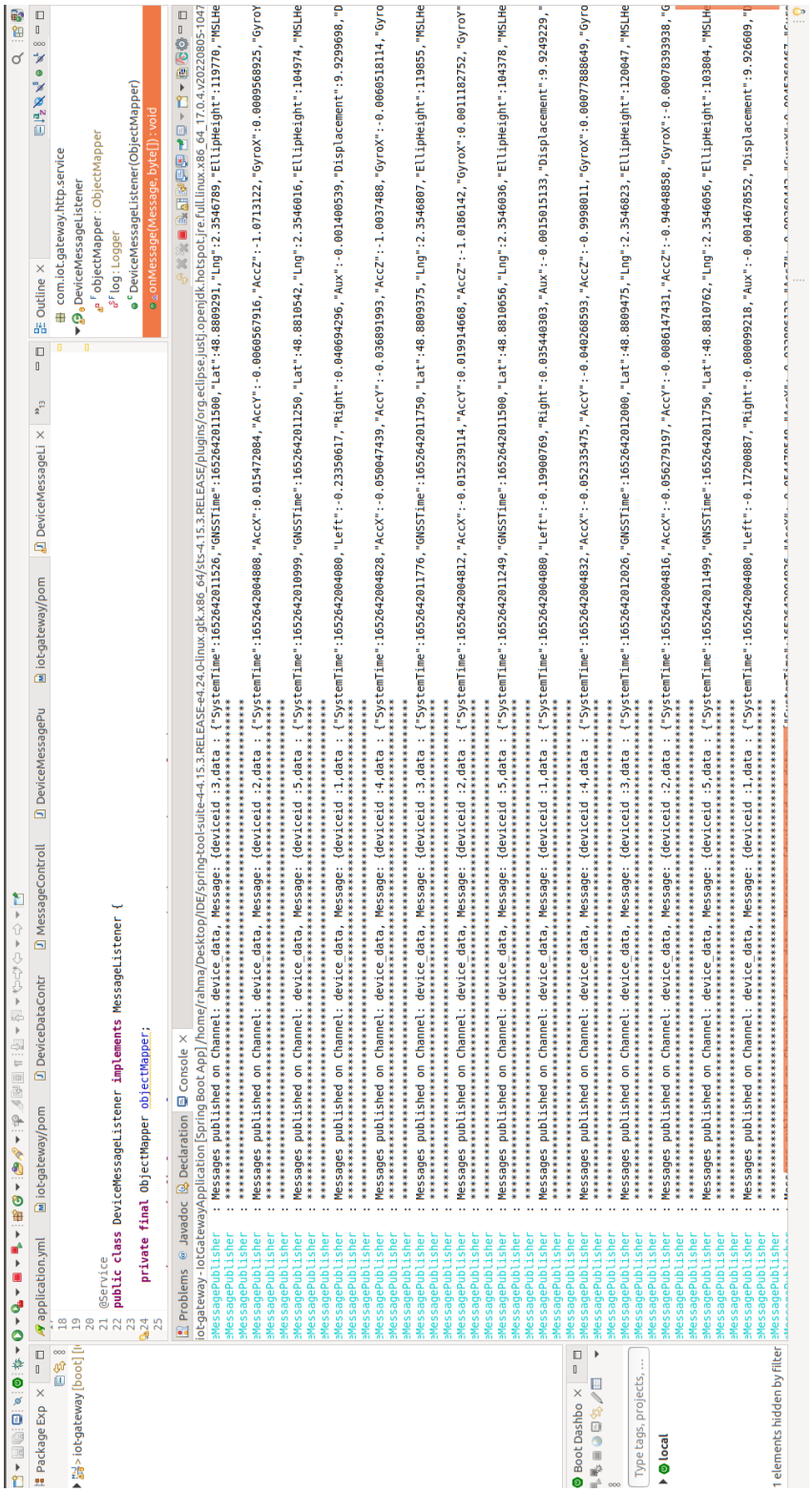

Figure 4.10: Publishing data from sensors in the second use case.

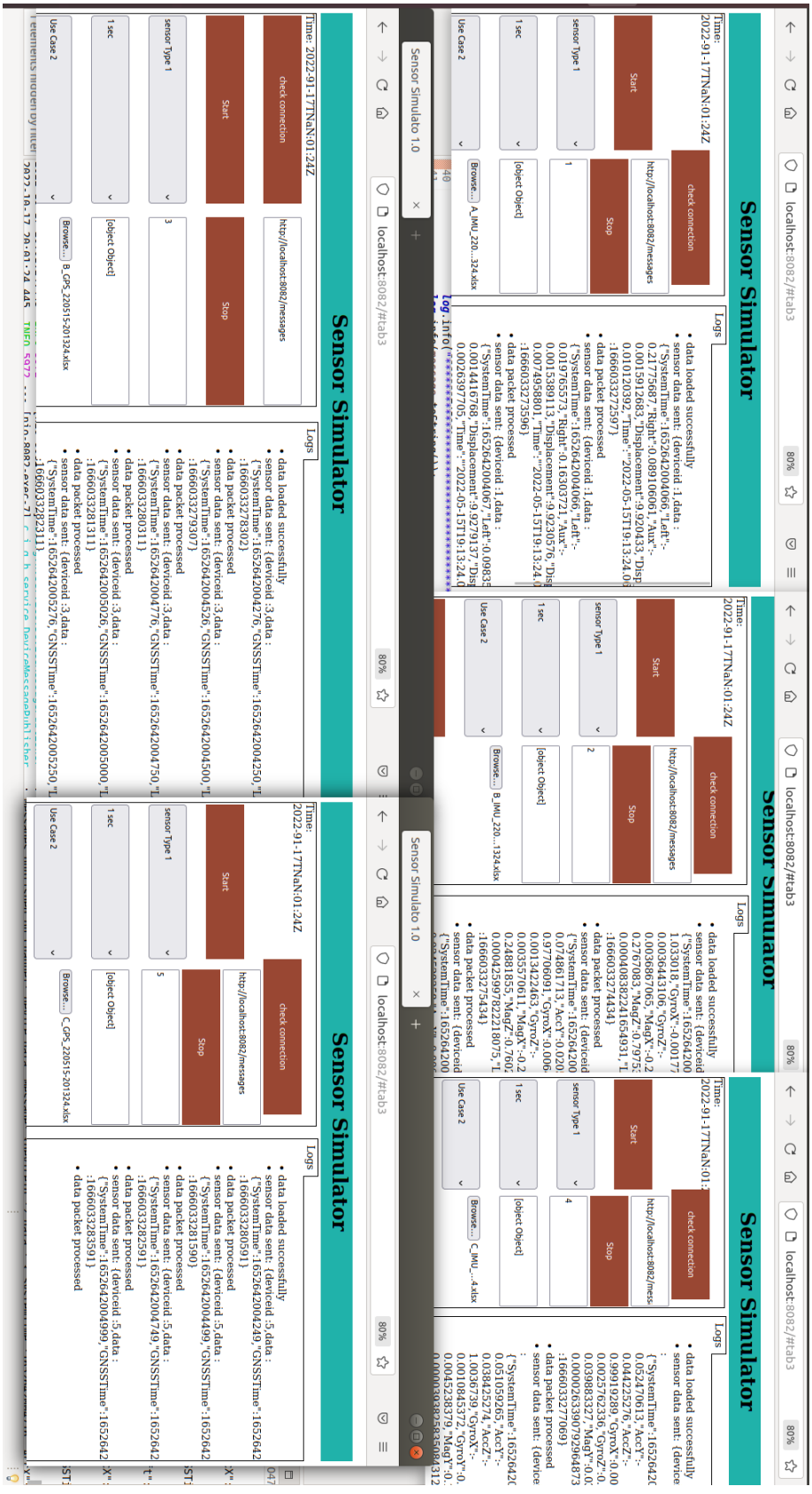

Figure 4.11: Second use case sensors.

pertinent use cases, we have illustrated how the simulator functions in real-time to enable the monitoring of railway infrastructure conditions. The key attributes of the proposed solution, including scalability, high throughput, and transparency, were delineated, demonstrating its ability to address the limitations inherent in conventional monitoring methodologies. Furthermore, the chapter meticulously examined the simulator's architecture, interaction model, and execution processes, emphasising the novel approach employed to enhance data integrity and equitable cost distribution through the utilisation of smart contracts within a permissioned Blockchain network. This innovative framework exemplifies a significant advance in both the technological and operational dimensions of the management of the railway infrastructure. Looking ahead, the insights presented in this chapter lay a strong foundation for future advancements in railway maintenance strategies, suggesting promising directions for continued research and collaboration between academia and industry. These collaborations aim to foster the broader adoption of IoT and Blockchain technologies, ultimately contributing to more reliable, predictive, and cost-effective infrastructure management. As the railway sector undergoes continuous transformation, the integration of such advanced technologies will play an essential role in improving operational efficiency, safety, and sustainability in the long term.

Chapter 5

Proof of Concept and Implementation

5.1 Introduction

This chapter will explore the intricacies of this innovative Blockchain-based platform, detailing its construction, key elements, and the ways in which it redefines the landscape of secure data exchange within the railway industry. Through this exploration, the chapter will provide insight into how Blockchain can be harnessed to address pressing data integrity concerns and create a resilient ecosystem for the future of transportation. Built on the robust foundation of HLF, this platform utilises smart contracts, embedded within chaincode, to ensure a secure and transparent environment for data transactions and fair cost attribution. Section 5.2 provides a comprehensive overview of the system architecture and outlines the process flow that governs its operation. It offers a high-level understanding of how the various components interact and function together within the Blockchain-based platform. A more detailed examination of the implementation process, including a breakdown of the distinct user roles such as Administrator, Data Provider, and Data Consumer will be presented in Section 5.3. Finally, the chapter will conclude in Section 5.4, where key findings and insights derived from the proof of concept will be summarised, highlighting the system's effectiveness and potential impact.

5.2 System Design and Module Overview

The underpinning power of Blockchain technology lies at the heart of this project. Using its capabilities, a fortified infrastructure is erected. This, in turn, empowers data providers to sell data generated from specific sensors for specific train journeys with confidence. Meanwhile, prospective consumers are offered the opportunity to request and acquire sensor data, resulting in an unassailable data marketplace characterised by reliability and immutability. Executing this mission involves a dynamic synergy of technologies. Notably, Node.js and MongoDB assume pivotal roles, harmonising seamlessly with the Blockchain framework. The strategic pairing of these elements yields an agile mechanism for data operations. It ensures the efficient storage, retrieval, and processing of data, ultimately culminating in a solution celebrated for its resilience and scalability. Furthermore, the project's ingenuity extends to incorporating a IoT simulator, which was previously introduced in Chapter 4. Through this integration, the platform orchestrates the reception of real-time sensors' data. This infusion of live data improves the authenticity and pertinence of the information exchanged within the ecosystem. As a result, the "B4CM" project exemplifies a holistic approach that amalgamates cutting-edge technologies to reimagine data exchange with newfound security and transparency. Figure 5.1 illustrates an abstract for the overall system components architecture and integration, respectively.

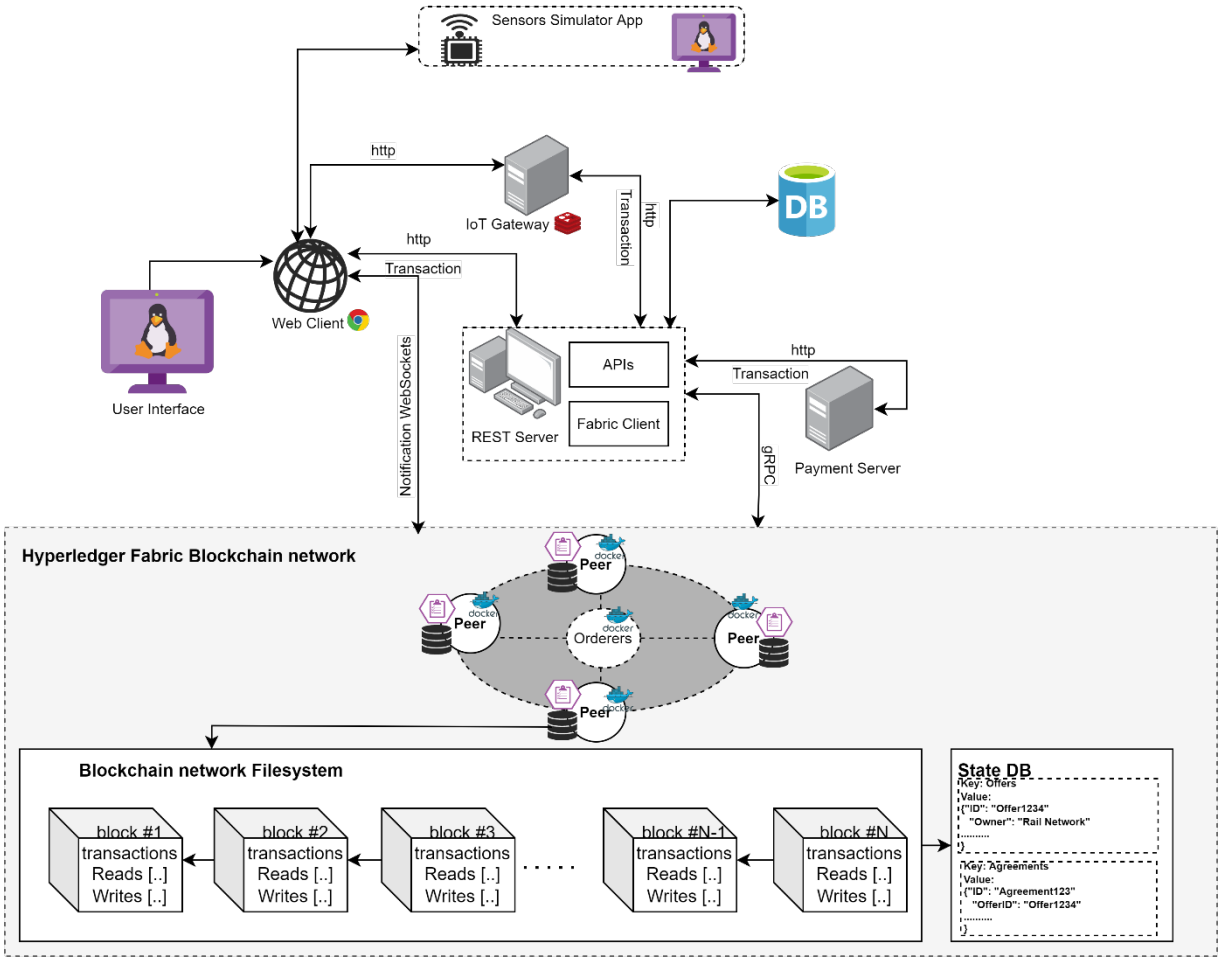

Figure 5.1: System components architecture and integration.

5.2.1 Blockchain Infrastructure

- **Description:** This module establishes a resilient Blockchain-based infrastructure using HLF, creating a decentralised and tamper-proof framework for seamless data transactions and cost distribution. The rationale for selecting HLF is discussed in detail in Section 3.6. The deployed HLF comprises three organisations that symbolise three distinct roles: Admin, Provider, and Consumer, as depicted in Figure 5.2. The development stack used for the implementation of the blockchain network is outlined in Table 4.1, while the specific tools and technologies used for the development of the client application to interact with the developed Blockchain are listed in Table 4.3.

- **Functionality:** It orchestrates the implementation of essential components needed to con-

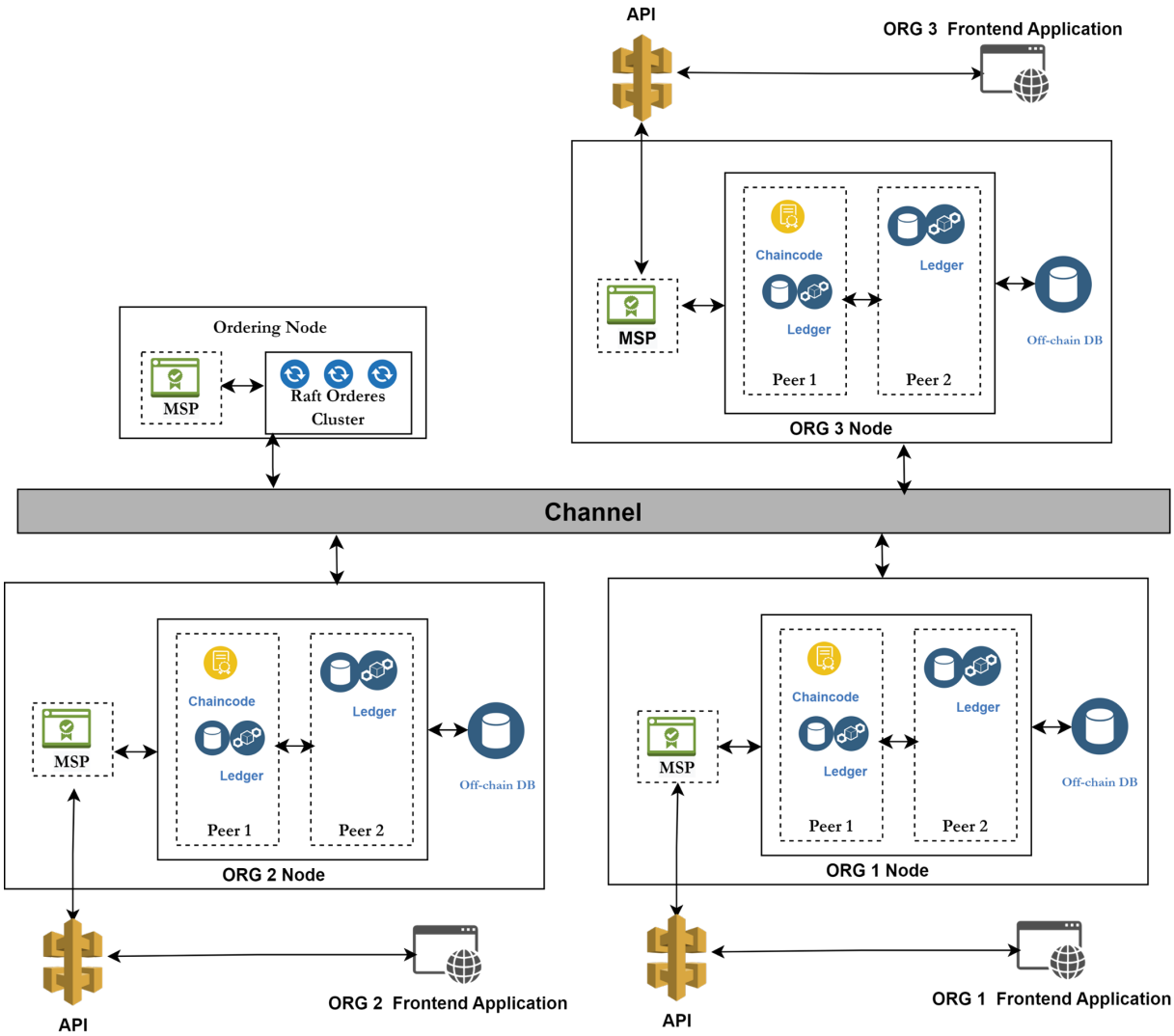

Figure 5.2: Deployed HLF infrastructure network.

struct a private and permissioned Blockchain network. The module effectively manages smart contracts through chaincode, enabling automation.

5.2.2 Smart Contract Management

- **Description:** This module is dedicated to the process of crafting, deploying, and overseeing smart contracts governing data exchange on the Blockchain.
- **Functionality:** Within the scope of Hyperledger Fabric's capabilities, this module is responsible for formulating intricate chaincode. These intelligent contracts are instrumental in preserving data transaction integrity and transparency. Predefined rules within the

contracts further cement the legitimacy of the data exchanges. The architecture of data within smart contracts is illustrated in Figure 5.3.

The relationships between entities in both real-time and historical access are illustrated in Figure 5.4 and Figure 5.5 respectively.

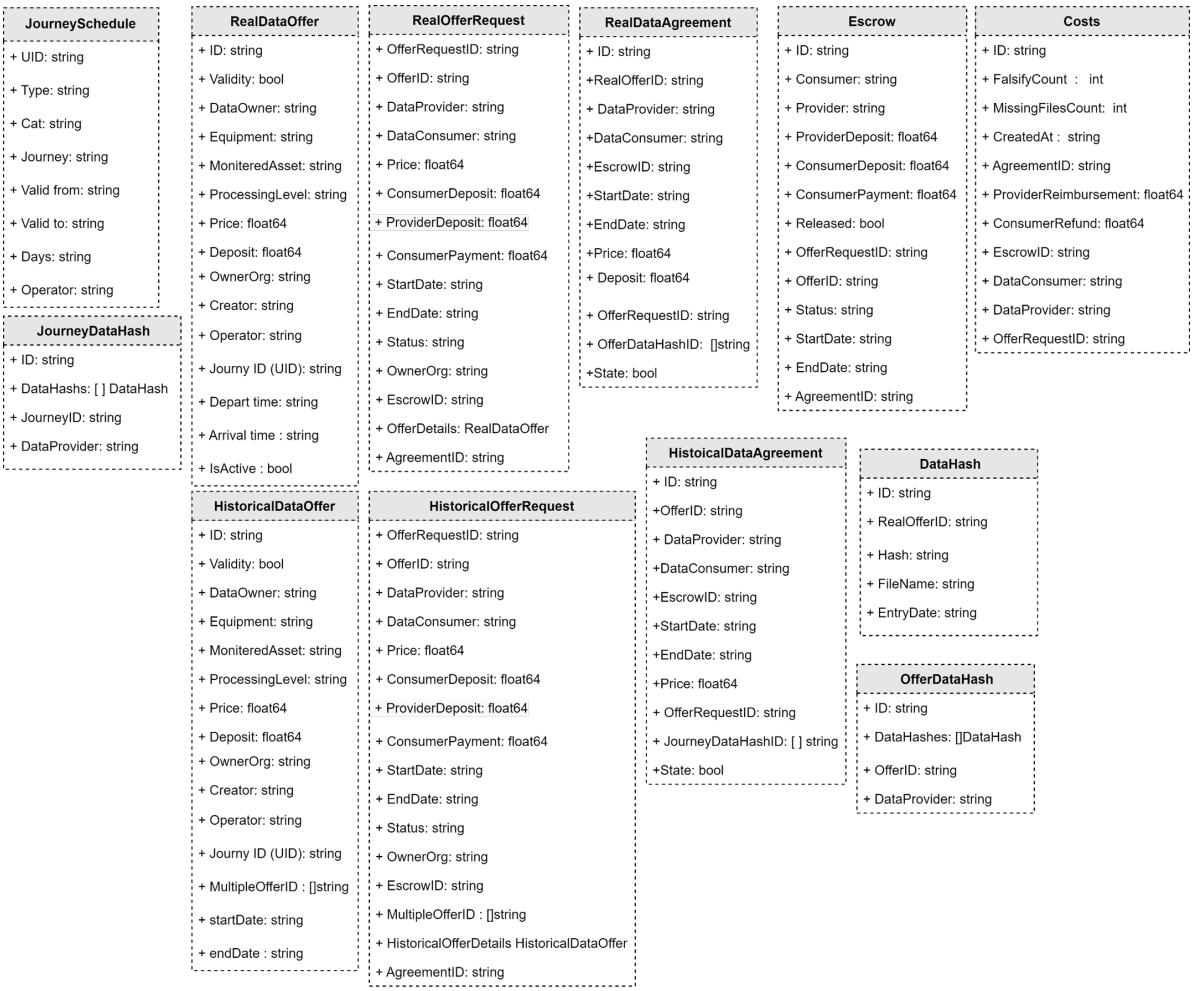

Figure 5.3: Data structure in smart contracts.

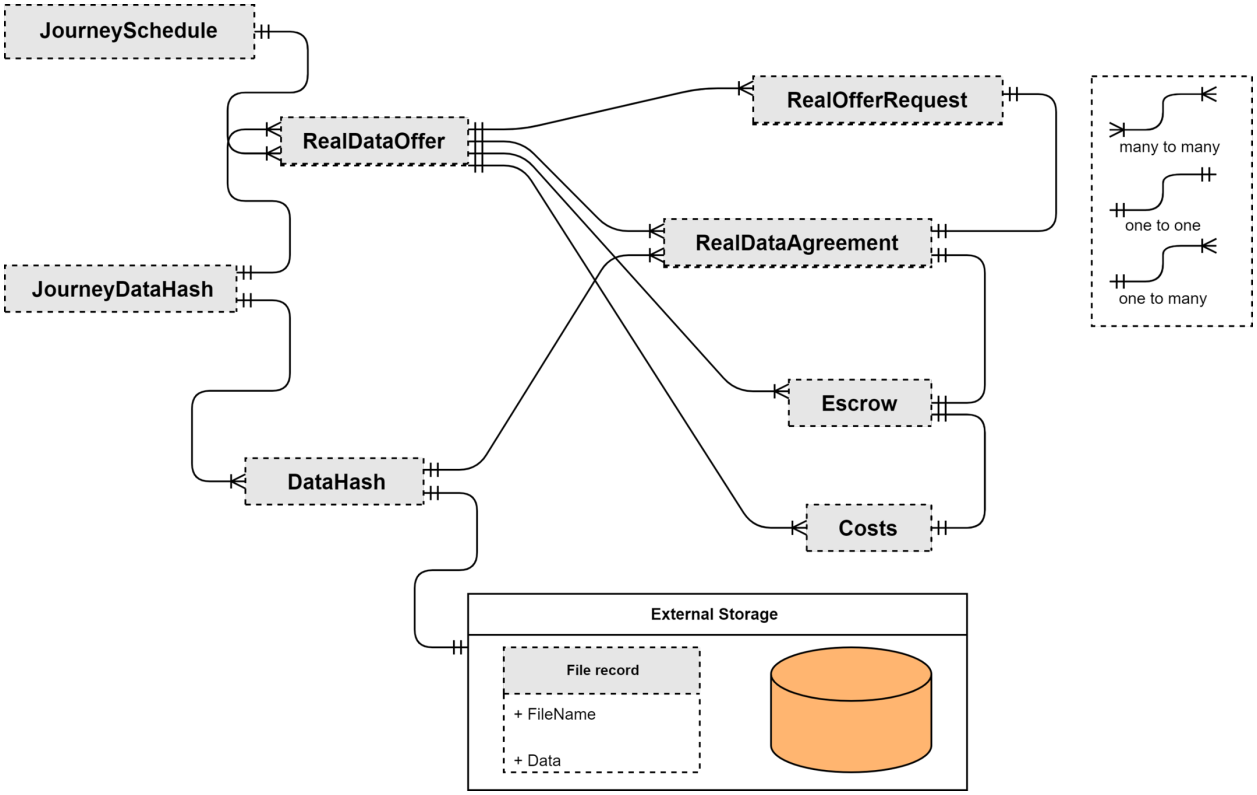

Figure 5.4: Real-time data entities relations.

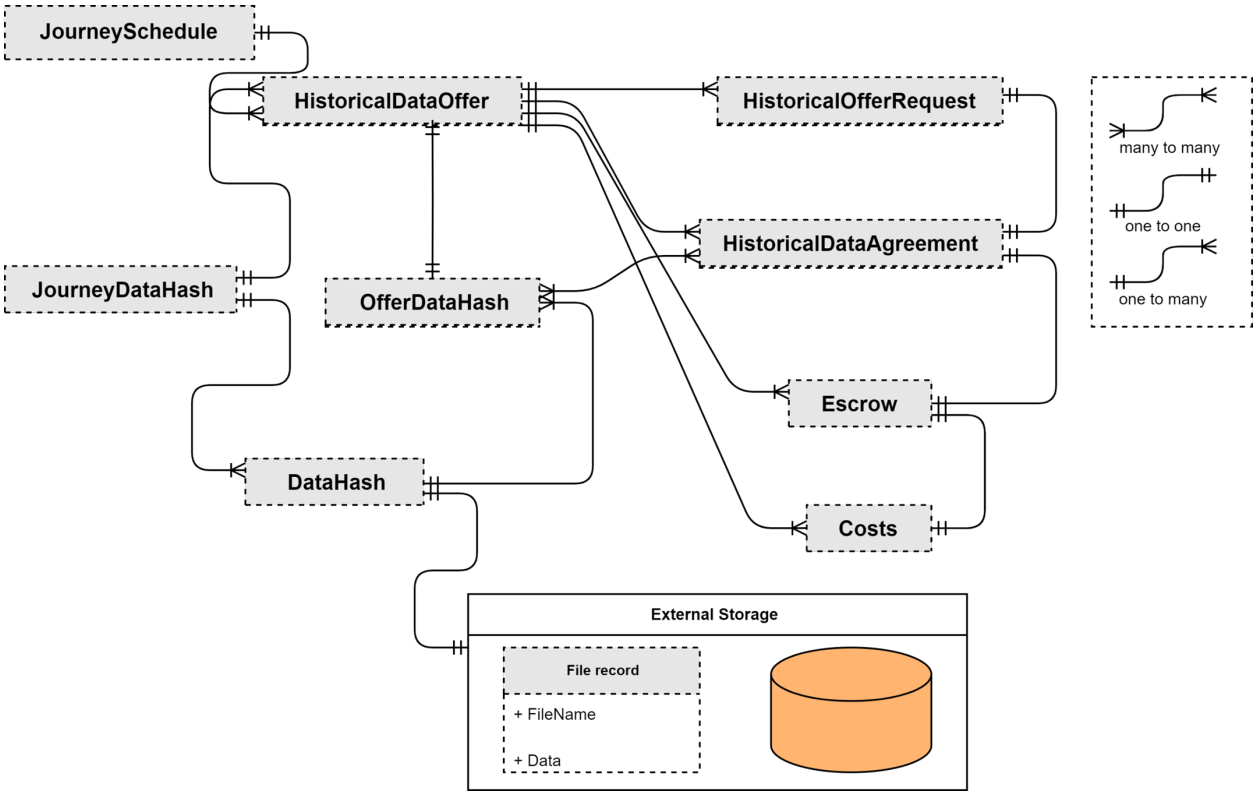

Figure 5.5: Historical data entities relations.

5.2.3 Data Request Workflow

The process for submitting a data request consists of several steps outlined below:

Step 1: Admin generates journey schedule

Administrators possess the capability to formulate journey schedules, outlining the precise time, date, and location of forthcoming events. This establishes a well-organised chronology for the process of sharing data between providers and consumers.

Step 2: Provider Initiates Data Offer

Data providers initiate the data-sharing process by initiating offers in alignment with predefined journey schedules. They specify pricing, data file formats, and other pertinent particulars, facilitating consumer engagement.

Step 3: Consumer Initiates Offer Request

Consumers are afforded the flexibility to explore and initiate inquiries regarding offers presented by providers. By selecting suitable offers, they express their intention to acquire specific sets of data, thereby commencing the negotiation phase.

Step 4: Escrow Mechanism

An escrow mechanism is established to ensure transparent financial transactions. Payments are securely held in escrow and released upon the agreement's revocation or expiration, fostering trust between involved parties.

Step 5: Agreement Generation

Agreements are automatically generated subsequent to offer acceptance. These agreements encapsulate the terms that govern the exchange of data, encompassing payment arrangements, data file specifications, and other pertinent elements. They serve as legally binding records, affirming the validity of the transaction. Depending on this agreement, providers securely transmit data files generated from sensors to consumers. The cost distribution will also be calculated according to the status of the agreements.

In Figure 5.6, a flow diagram is shown for the steps followed after the consumer sends his request.

5.2.4 Data Transmission Workflow

The data transmission process is initiated once a formal agreement has been established between the data provider and the consumer. As illustrated in Figure 5.7, the transmission flow is governed by the terms of the agreement. Upon the acceptance of a data request, a binding contract is created between the two parties, ensuring that the data, along with its associated hash, is securely transmitted. The generated hash file serves as a cryptographic verification, enabling the consumer to confirm the integrity and authenticity of the received data.

At this stage, the claims mechanism embedded within the agreement becomes active, allowing both parties to validate the accuracy and completeness of the transmitted data. This step plays a critical role in ensuring that any discrepancies or integrity concerns are promptly addressed, thus upholding the transparency and reliability of the data exchange process.

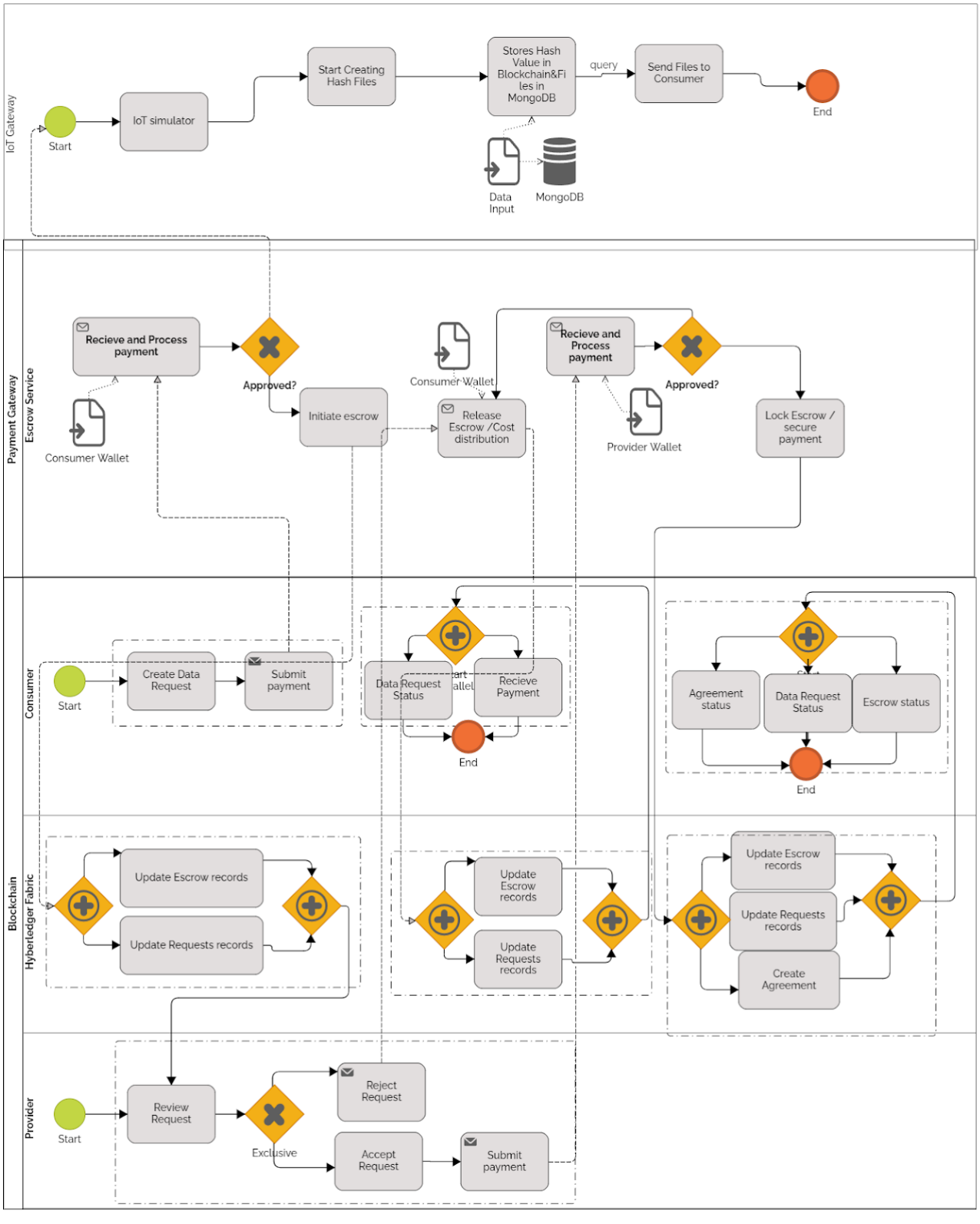

Figure 5.6: Data request workflow.

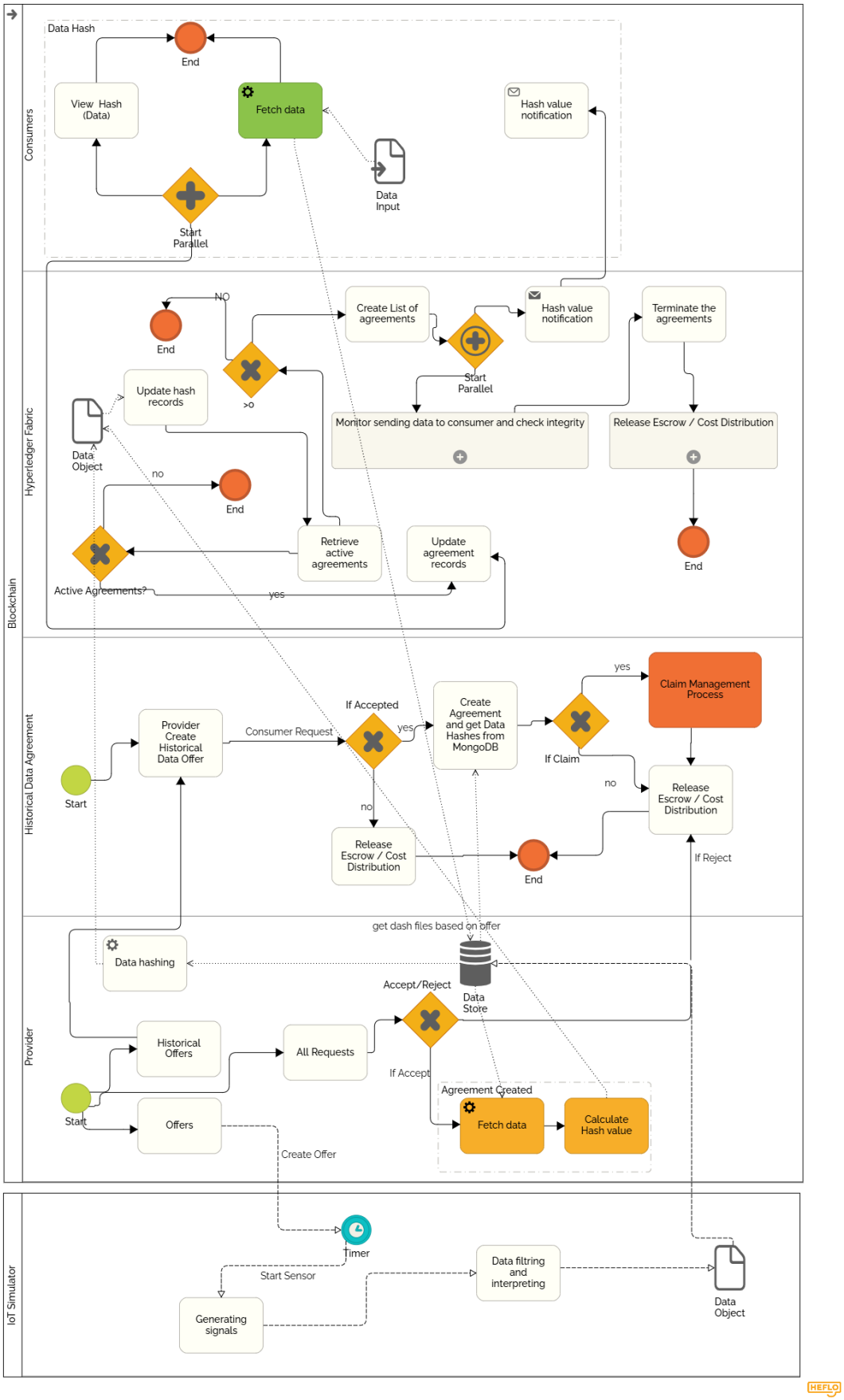

Figure 5.7: Data transmission workflow/ agreement workflow.

5.2.5 Cost Distribution Workflow

Figure 5.8 and Figure 5.9 illustrates the sequence where cost calculations occur post-escrow release. The methods of calculation vary depending on whether claims are present, data integrity verification, and agreement status.

5.2.5.1 Claim Management and Escrow Release

Once the agreement's end date is met, the claim management process begins its countdown. The administrator retains the flexibility to adjust the claim management time as per platform requirements. Following the expiration of both the end date and the claim management time, the escrow linked with the offer is released.

5.2.5.2 Final Distribution of Costs

The conclusive allocation of costs is initiated upon the successful fulfillment of the offer. This distribution meticulously documents all financial interactions between the provider and the consumer.

5.2.5.3 Absence of Consumer Concerns

In cases where the consumer does not raise any concerns regarding the data file or the offer, and the end date, along with the claim management time, has passed, the escrow is released. The consumer is refunded their deposit, and the provider receives their deposit alongside the consumer's payment.

5.2.5.4 Activation of the Claim Management Timer

Upon the conclusion of the end date, the claim management timer springs into action. During this span, consumers can evaluate the received data file for any discrepancies or issues.

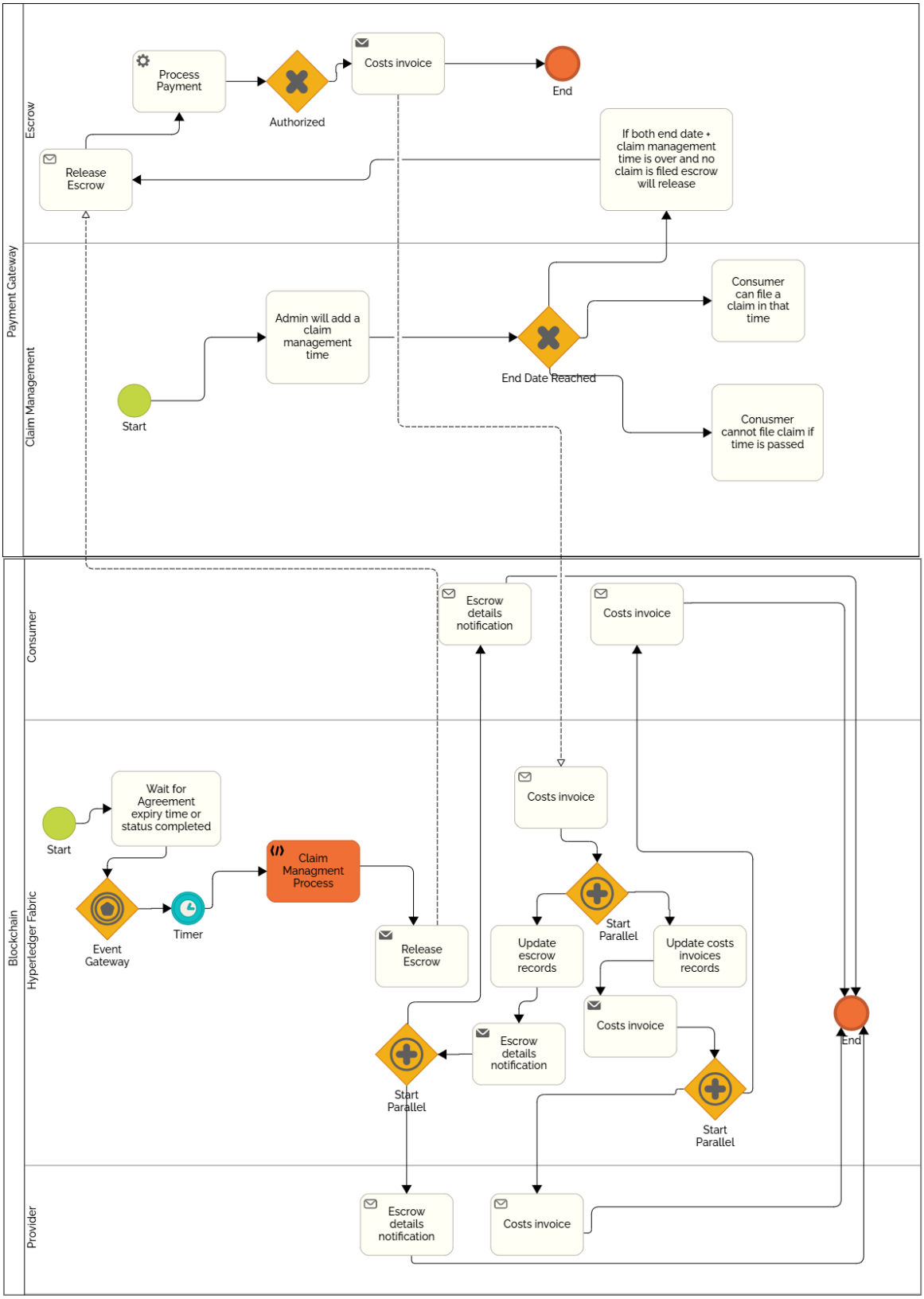

Figure 5.8: Cost distribution workflow.

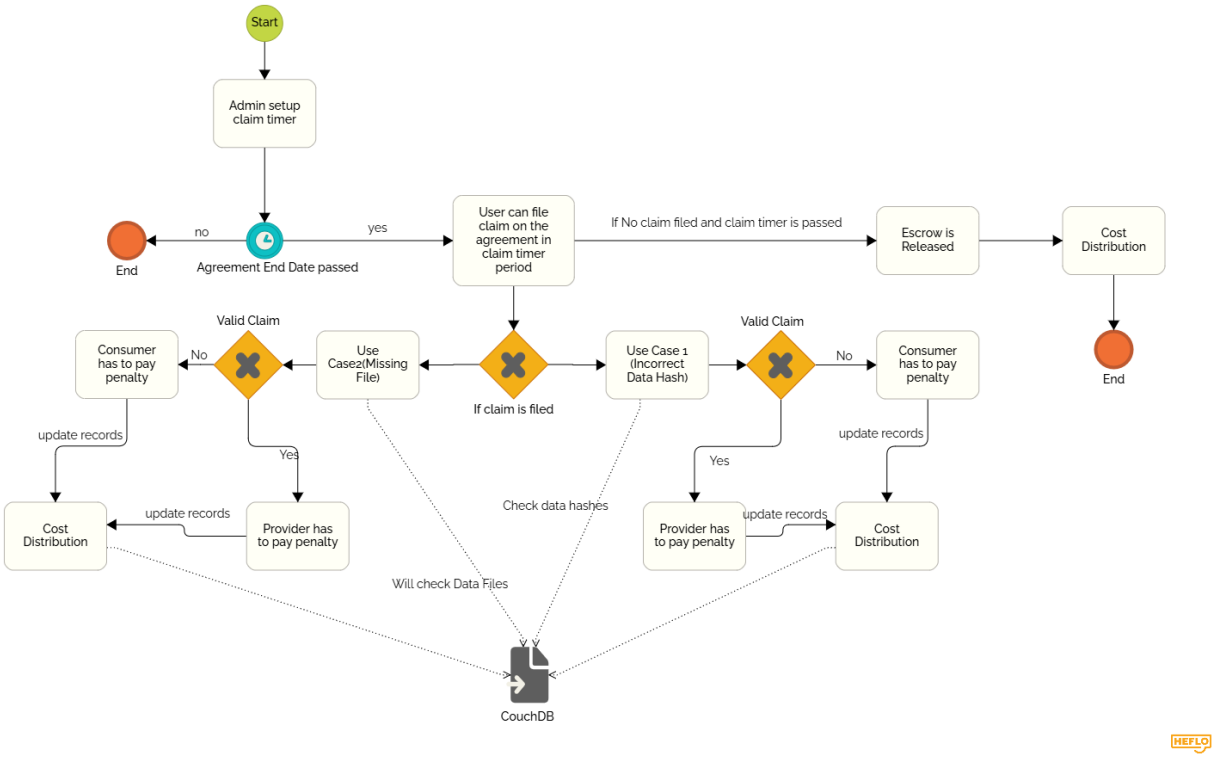

Figure 5.9: Claim management process.

5.2.5.5 Consumer's Claim and Associated Consequences

If the consumer identifies discrepancies or omissions within the data file during the claim management duration, they possess the right to submit a claim. Penalties are imposed on the provider if the data file proves incomplete or inaccurate. Subsequently, the provider's deposit is reimbursed to the consumer, along with their own deposit and the payment received.

5.2.5.6 Verification of Data Integrity

The platform meticulously cross-examines the hash values of the data file against both the Blockchain and the MongoDB database. In the event of disparities, the provider is subjected to penalties. However, if the hash values correspond, the claim process proceeds systematically.

5.2.5.7 Unsubstantiated Claim by the Consumer

Should the consumer raise a claim without presenting substantial proof of issues within the data file, and if the data hash values are verified as accurate, the consumer bears the brunt of a

penalty. Consequently, the provider reclaims both the consumer's deposit and their own, while the platform duly updates the cost distribution record.

5.2.5.8 Revocation of Agreements

While the end date is not yet reached, both the provider and the consumer reserve the option to annul the agreement. A penalty is imposed upon the party initiating the revocation post-acceptance.

- **Agreement Revocation by the Provider:** In the scenario where the provider opts to revoke the agreement after acceptance, the consumer regains their deposit, and the provider loses their own deposit. The consumer also receives their payment.
- **Agreement Revocation by the Consumer:** Should the consumer opt to revoke the agreement after acceptance, the provider acquires both the consumer's deposit and their own. While the consumer receives their payment, their deposit remains unrecovered.
- **Revocation Beyond the End Date:** Revocation of the agreement is rendered infeasible for either party once the end date has lapsed.

5.2.5.9 Resolution of Revocation and Distribution

The entirety of the details associated with agreement revocations is comprehensively documented within the cost distribution record. Serving as a comprehensive summary of financial transactions, penalties, and reimbursements shared between the provider and the consumer, this record encapsulates the complexity of the cost distribution process. It encompasses claim management, penalties, refunds, revocation scenarios, and meticulous recording of pertinent details within the cost distribution record.

5.2.6 Admin

- **Description:**This module handles the intricate financial transactions between data providers and consumers.

- **Functionality:** With precision, the module supervises the secure and transparent processing of payments. It collaborates seamlessly with the Escrow Management module to ensure the release of funds to the provider upon the successful completion of agreements. Diagram 5.10 presents the main APIs developed for interactions between the admin module and the Blockchain on the back-end.

5.2.7 Data Provider

- **Description:** Designed to empower data providers, this module facilitates the offering and sale of sensor data pertaining to specific train journeys.

- **Functionality:**Through a user-centric interface, data providers are empowered to seamlessly upload and present sensor data on the platform. The module takes charge of the intricate details, including data pricing and the initiation of transactions. Diagram 5.11 presents a code map for the main APIs developed for interactions between the provider module and the Blockchain at the back-end in the case of real-time data exchange. While the diagram 5.12 presents a code map for the main APIs developed for interactions between the provider module and the Blockchain at the back-end in the case of historical data exchange.

5.2.8 Data Consumer

- **Description:** Focusing on data consumers, this module streamlines the process of requesting and acquiring sensor data from available listings.

- **Functionality:** With a user interface designed for ease, consumers effortlessly navigate through the available listings, searching for and requesting relevant sensor data related to train journeys. The module further activates the initiation of data purchase transactions via smart contracts. The diagram 5.13 presents a code map for the main APIs developed for interactions between the consumer module and the Blockchain at the back-end in the

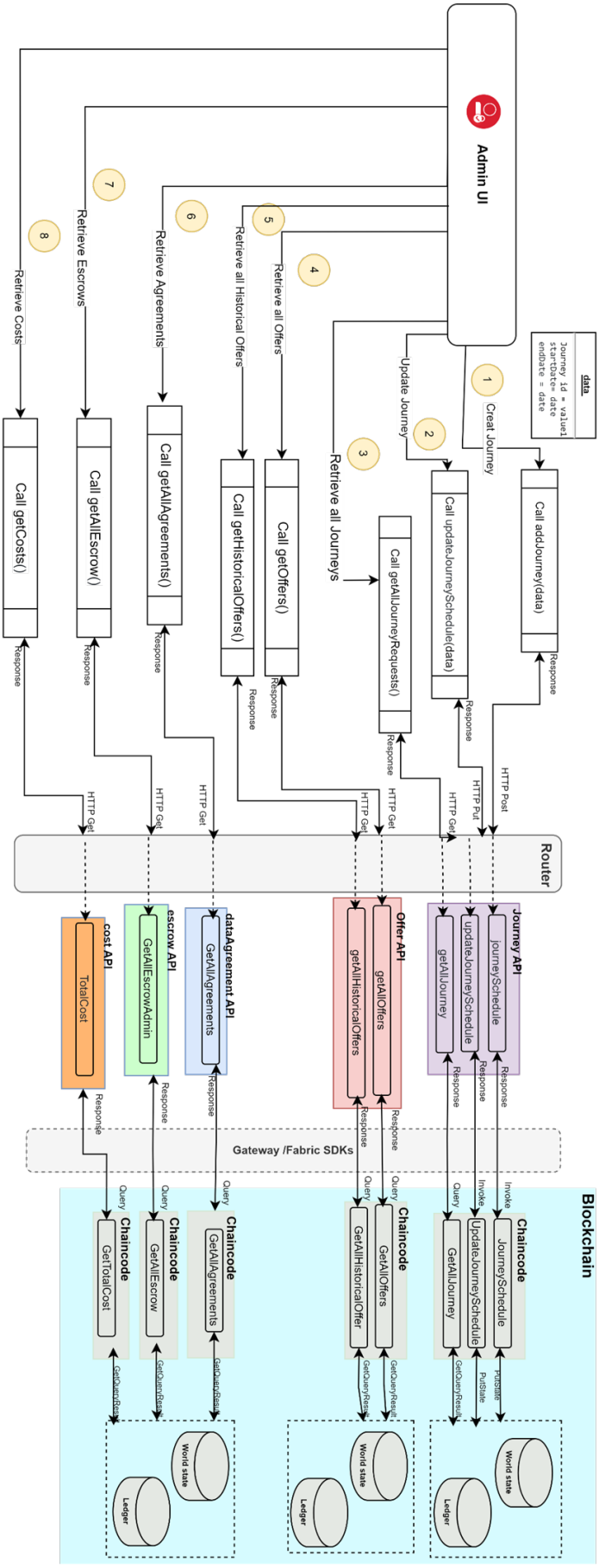

Figure 5.10: Admin's APIs to interact with chaincode.

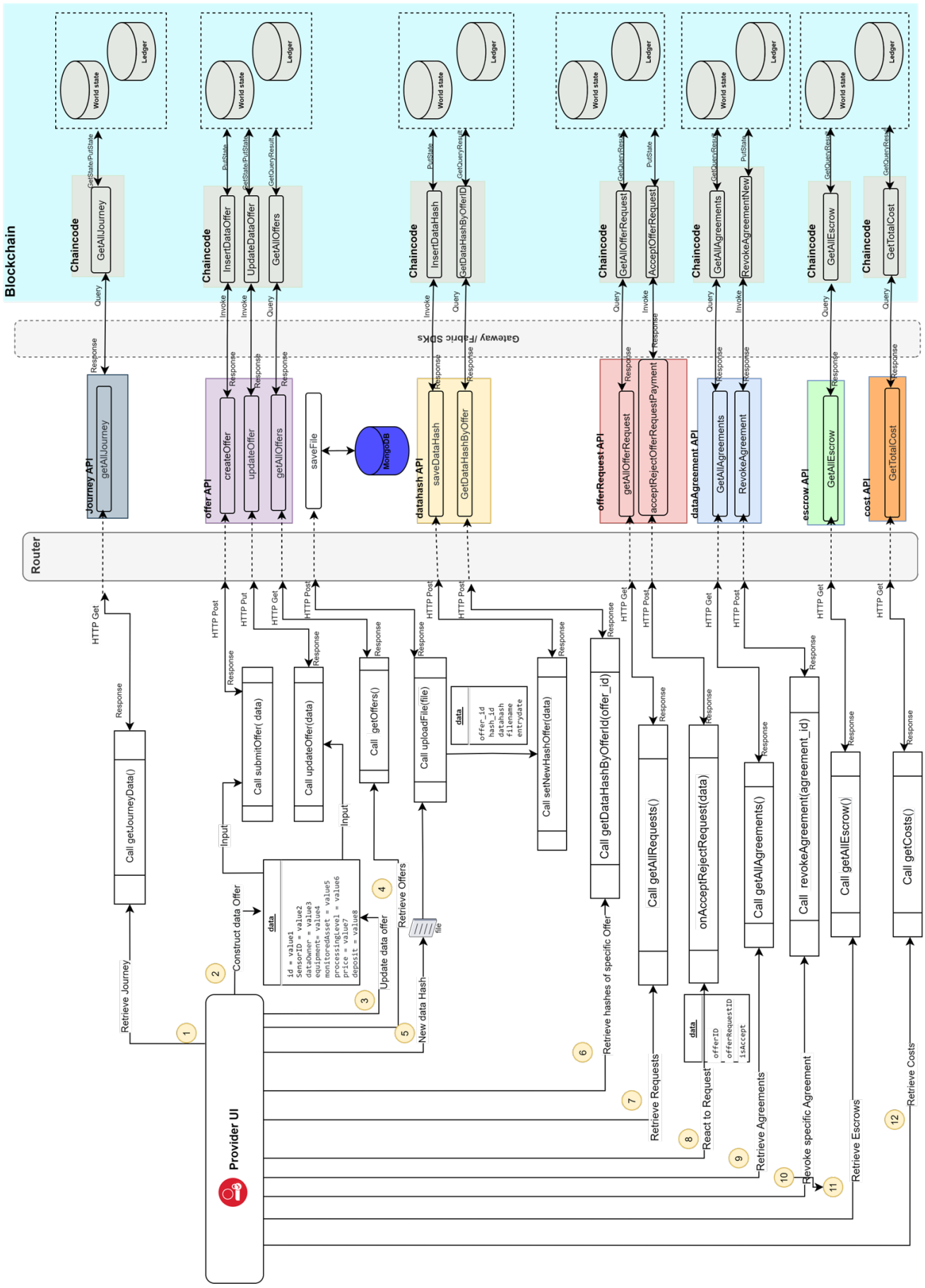

Figure 5.11: Provider's APIs to interact with chaincode in real-time data exchange.

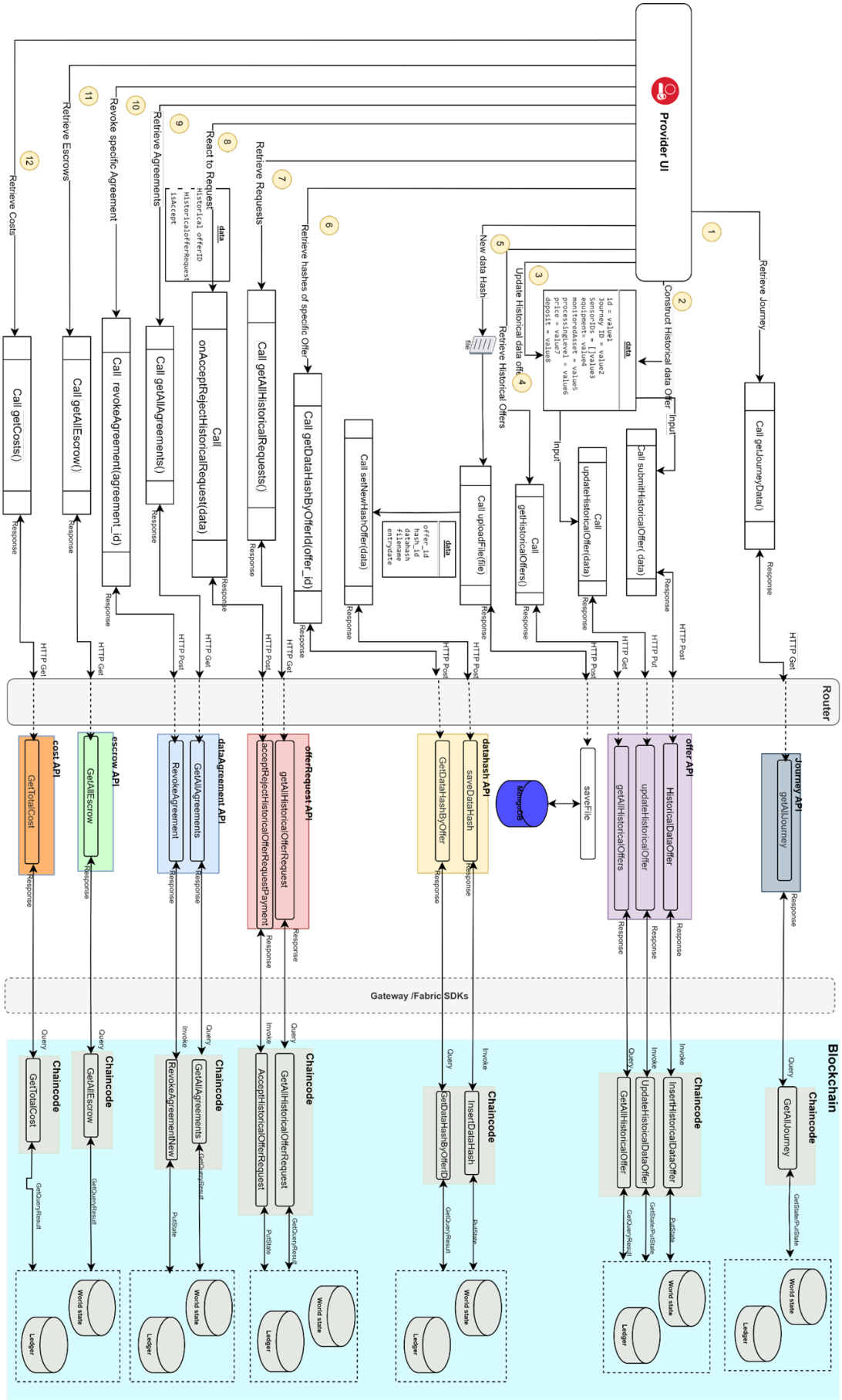

Figure 5.12: Provider's APIs to interact with chaincode in historical data exchange.

case of real-time data exchange. The diagram 5.14 presents a code map for the main APIs developed for interactions between the consumer module and the Blockchain at the back-end in the case of historical data exchange.

5.2.9 Data Integrity and Validation

- **Description:** This module is devoted to upholding data integrity and validation, ensuring the reliability of exchanged sensor data within the ecosystem.
- **Functionality:**Through the implementation of robust validation mechanisms, the authenticity and accuracy of sensor data are rigorously confirmed. The module actively monitors the Blockchain, vigilant for any unauthorised modifications to preserve data fidelity.

5.2.10 Payment Processing

- **Description:** This module handles the payment transactions between data providers and consumers.
- **Functionality:** With precision, the module oversees the secure and transparent processing of payments. It collaborates seamlessly with the Escrow Management module to ensure the release of funds to the provider upon the successful completion of agreements.

The process flow depicted in Figure 5.15

5.2.11 Payment Gateway

When a consumer initiates a payment, the payment gateway, such as mollie[1] in our case, initiates a hosted checkout page. This page presents an intuitive interface, allowing the consumer to select their preferred payment method. If the consumer opts to pay via a card, they will be prompted to input essential card details, including the card number, expiry date, and CVV code.

[1] https://www.mollie.com/

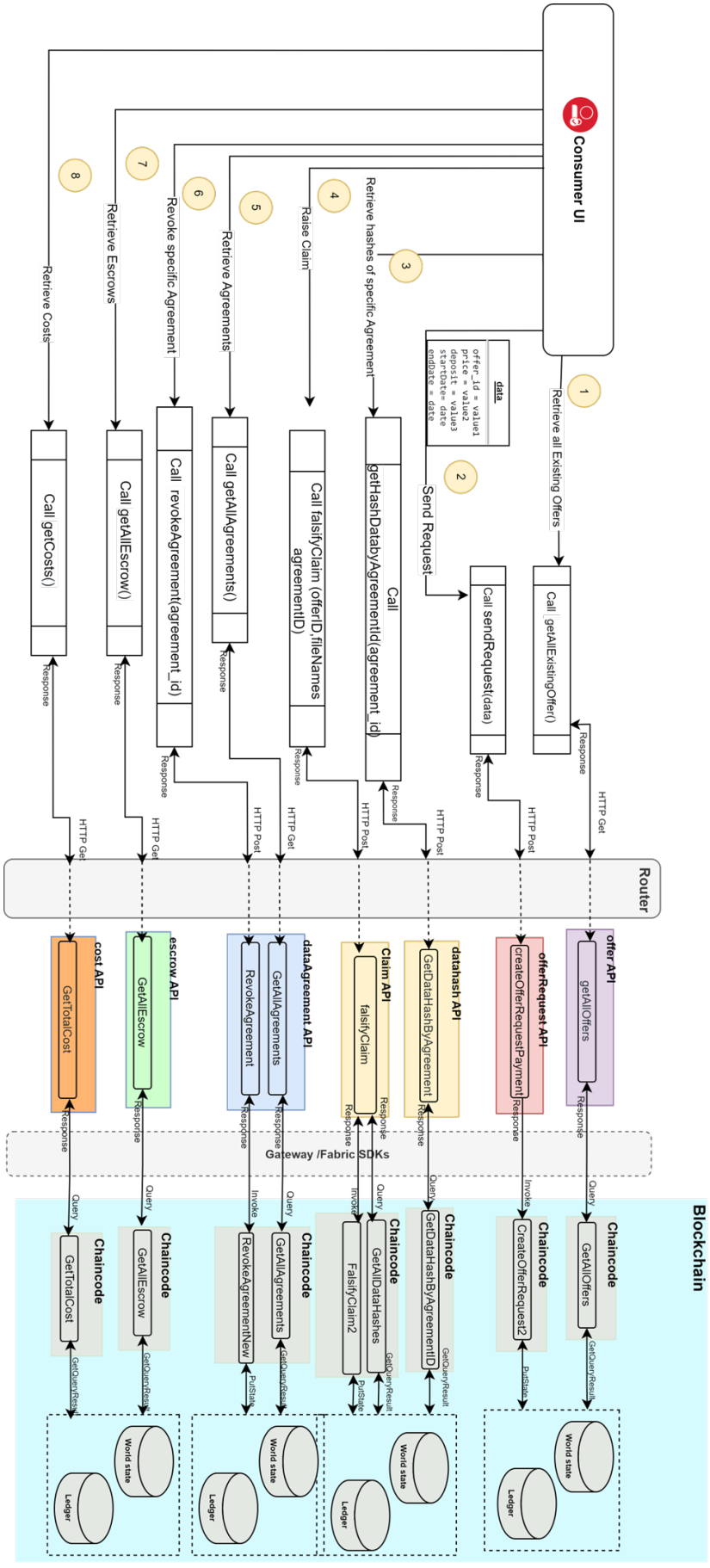

Figure 5.13: Consumer's APIs to interact with chaincode in real-time data exchange.

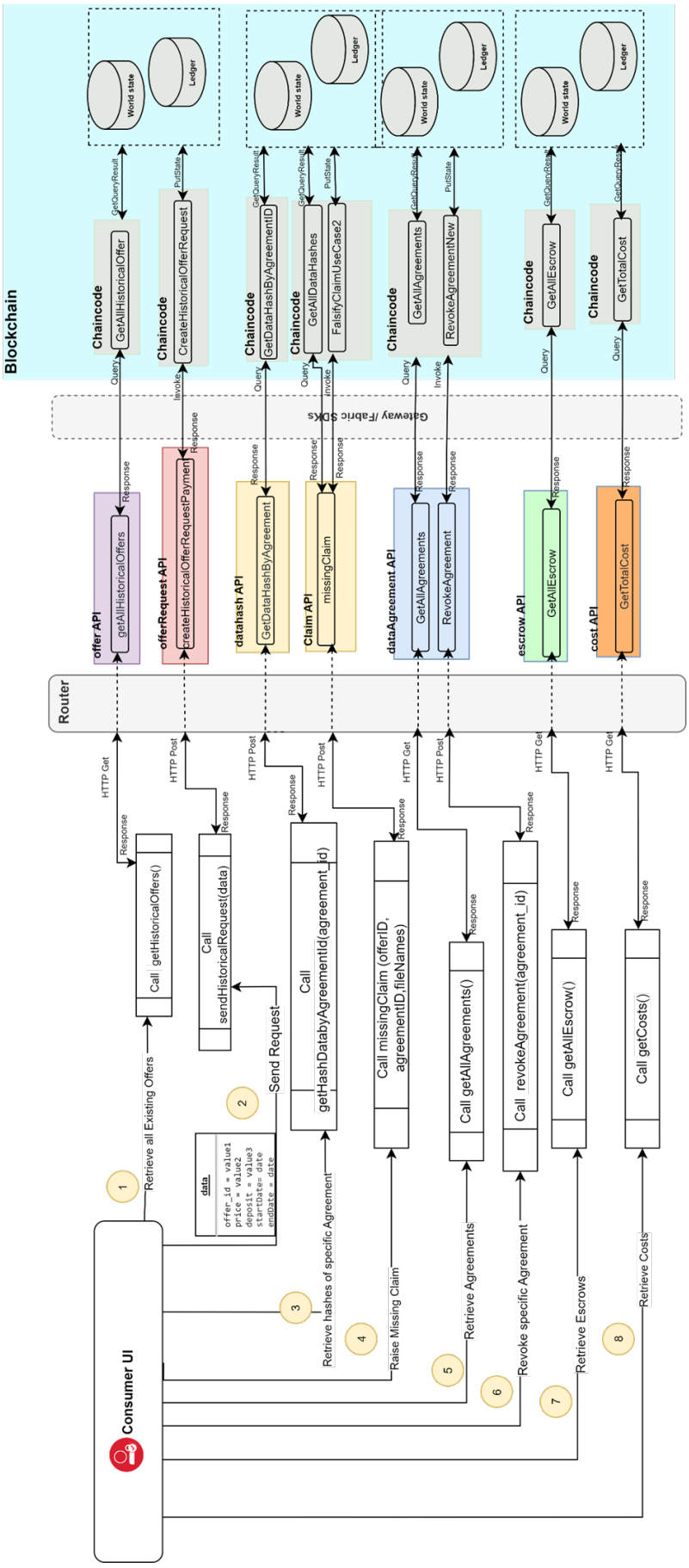

Figure 5.14: Consumer's APIs to interact with chaincode in historical data exchange.

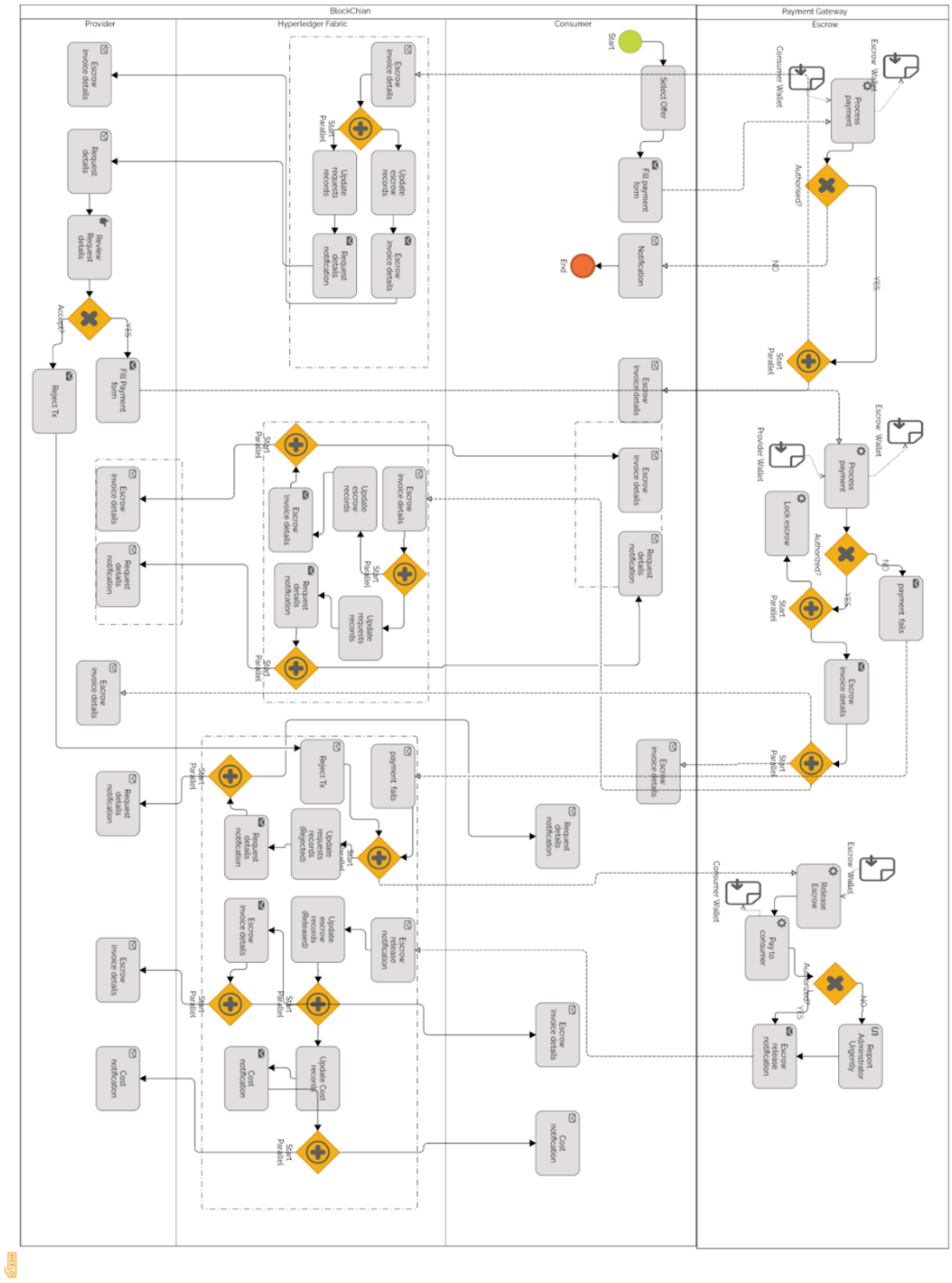

Figure 5.15: Payment process workflow.

Upon inputting the card details, the payment gateway undertakes a validation process to verify the precision and legitimacy of the provided information. If the card details are accurate and the transaction goes through successfully, the payment gateway presents a message confirming the successful transaction, signifying that the payment has been processed seamlessly.

Following the display of the success message, the payment gateway redirects the consumer back to the merchant's website, signaling the successful completion of the product purchase. At this juncture, the consumer gains access to the purchased data if the provider accepts the consumer's request. The flow of the process is depicted in Figure 5.16.

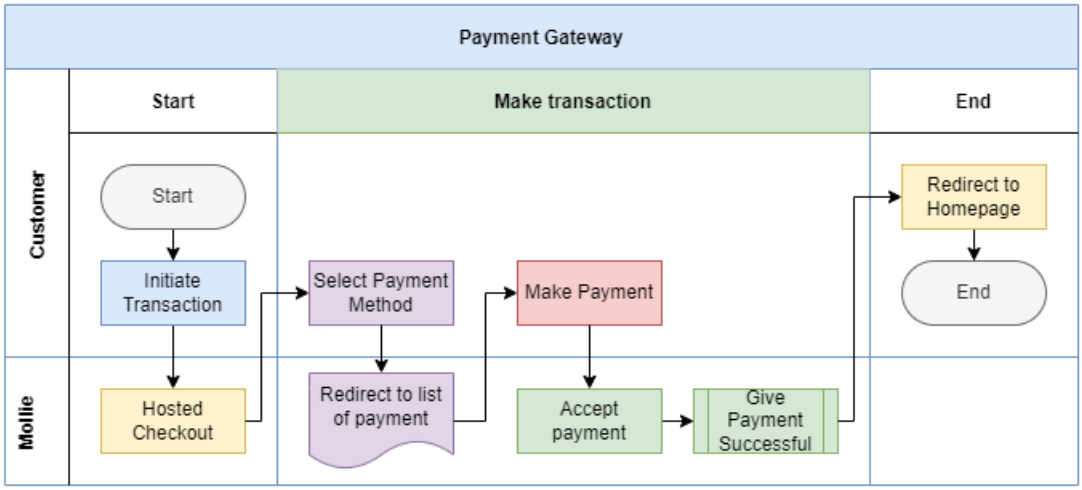

Figure 5.16: Payment gateway.

5.2.12 Escrow

- **Description:** This module expertly administers the escrow process, safeguarding funds between data providers and consumers until agreements conclude.
- **Functionality:** Acting as a vigilant custodian, the module meticulously manages the holding of funds during transactions. It operates with precision to release funds to the provider once the agreement has been fulfilled to satisfaction.

5.2.13 IoT Simulator

- **Description:** With the incorporation of an IoT simulator, this module enriches the platform by enabling real-time sensor data reception.
- **Functionality:** The module orchestrates the connection between the IoT simulator and the platform, facilitating the exchange of real-time sensor data. This dynamic infusion of live data enhances both the authenticity and relevance of the data shared.

5.2.14 SHA-3 Hash Function (Data File Hashing)

- **Description:** This module is dedicated to generating and retaining the SHA-3 hash of data files on the Blockchain.
- **Functionality:** Employing the SHA-3 hash function, the module constructs a unique and verifiable fingerprint for each data file. This fingerprint serves as an unassailable testament to the integrity and authenticity of the data.

5.3 Implementation, User Responsibilities and Role-Based Interactions

This project employs the organisational structure of HLF, with three distinct roles: provider, consumer, and admin, denoted as org1, org2, and org3, respectively. The core objective of this Blockchain-driven project is to facilitate the secure and transparent exchange of sensor data associated with train journeys. Each role is endowed with specific functionalities and privileges as depicted in Figure 5.17. The main functions of each user are as follows:

1. **Provider (ORG1)**
    - Possesses the capability to generate offers concerning journey schedules, outlining the sensor data from the train to be encompassed in the offer through an IoT gateway.

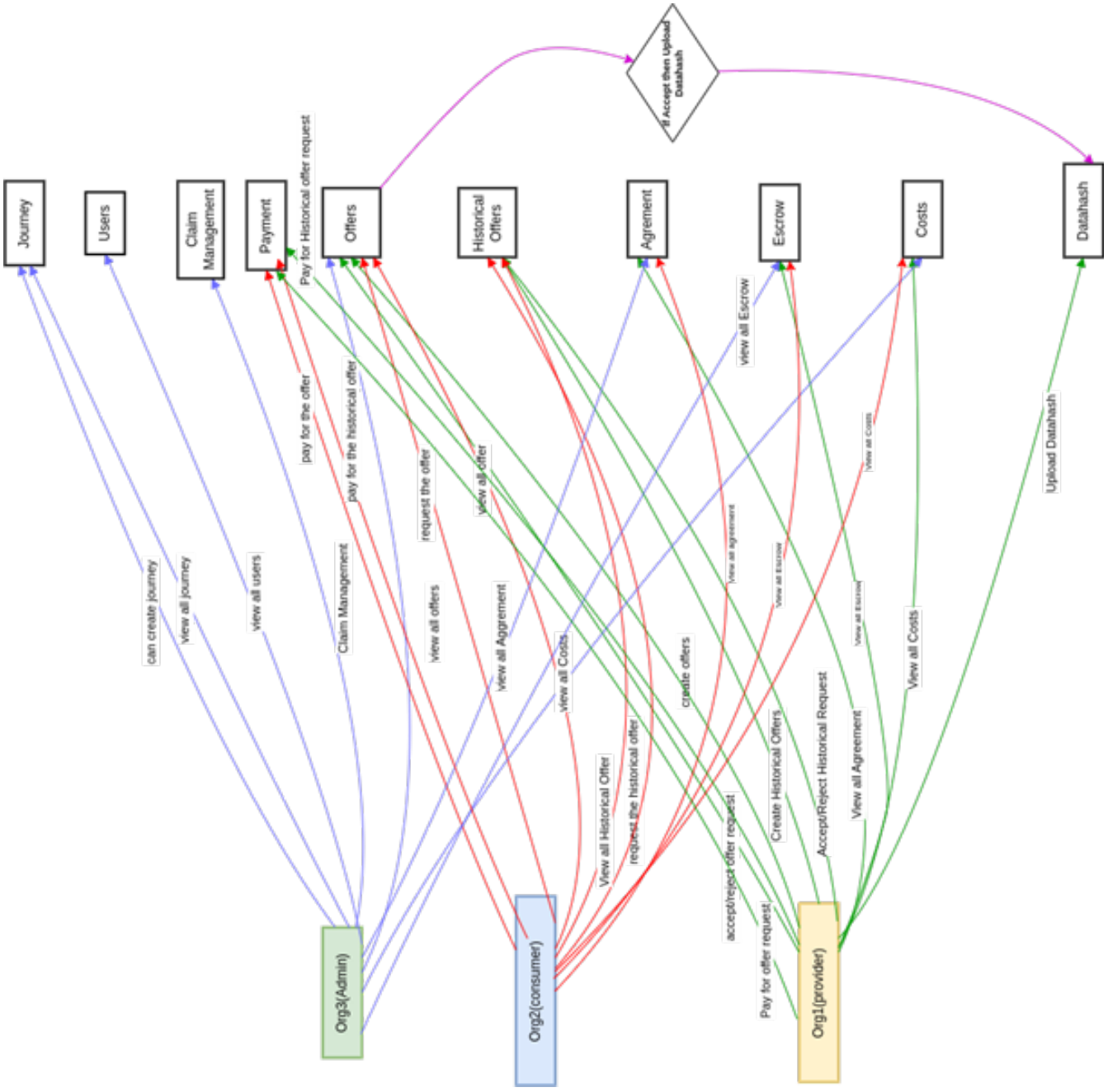

Figure 5.17: Users' roles.

- Establishes the price for the offer and presents it to potential consumers.

2. **Consumer (ORG2)**

   - Can access available offers on the platform.
   - Holds the choice to request offers from providers based on their preferences and needs.
   - Once interested in an offer, consumers can make payments for the stipulated offer price.
   - Following successful payment, consumers receive the sensor data from the chosen offer.

3. **Admin (ORG3)**

   - Wields administrative privileges within the system.
   - Assumes responsibility for crafting and updating journey schedules.
   - Possesses the ability to view all available offers on the platform.

5.3.1 Admin Responsibilities

Administrators hold a comprehensive set of responsibilities aimed at ensuring the smooth functioning of the platform:

5.3.1.1 User Management

Administrators have the authority to access a comprehensive user listing, allowing them to view essential information about the platform's users. This includes the number of providers and consumers within each organisation. Through this centralised view, administrators gain insights into user demographics and distribution across various organisations.

- **Provider:** Administrators can navigate through a comprehensive provider listing, which provides a detailed overview of each individual provider. See Figure 5.18 and 5.19.

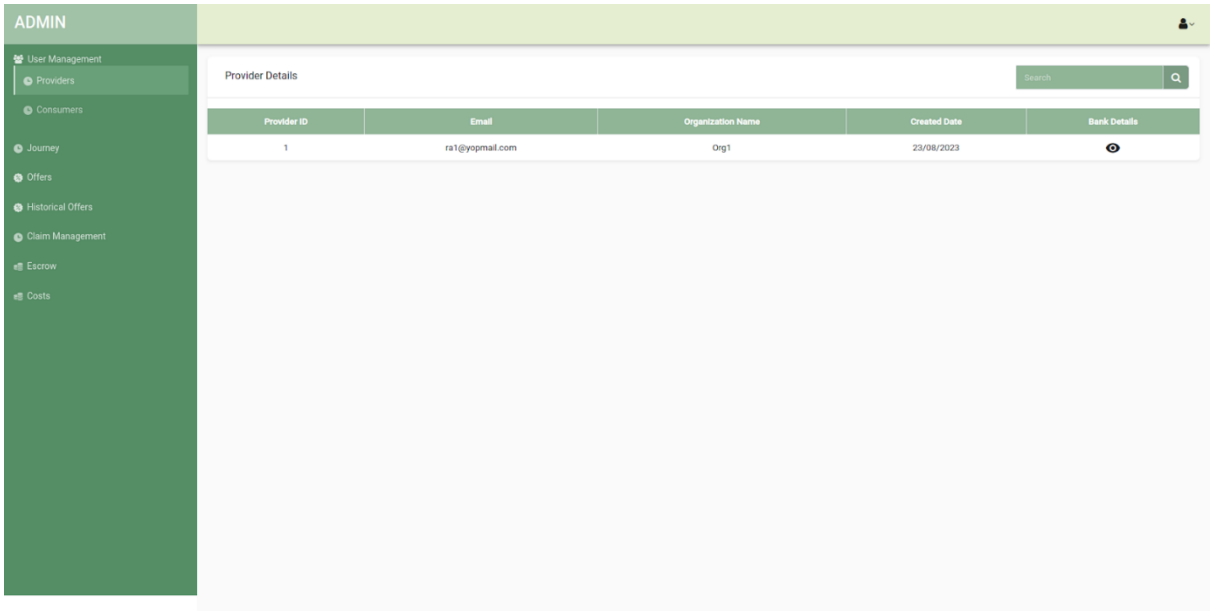

Figure 5.18: Provider user details.

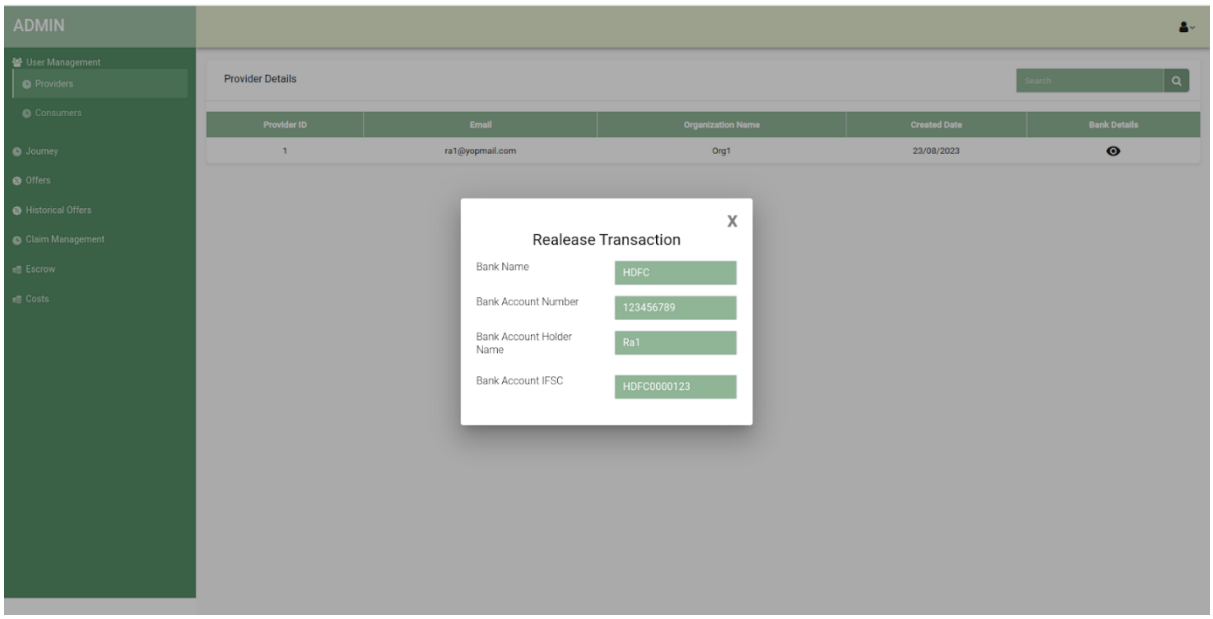

Figure 5.19: Provider bank details.

- **Consumer:** Administrators can navigate through an exhaustive consumer listing, which provides a detailed overview of each individual consumer. See Figure 5.20 and 5.21

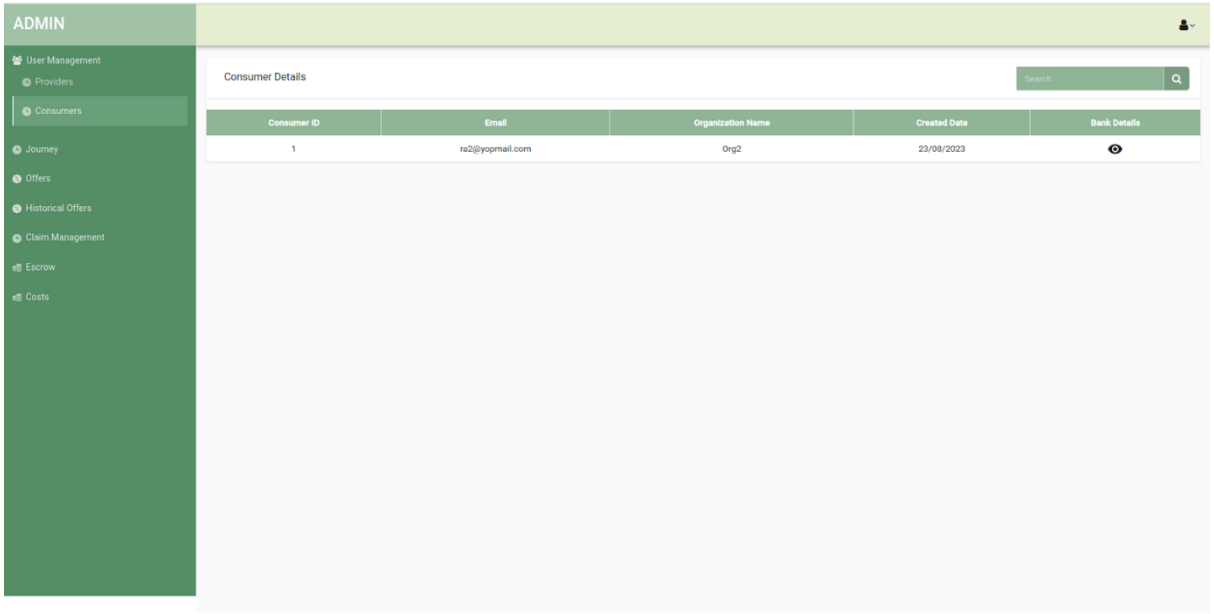

Figure 5.20: Consumer user details.

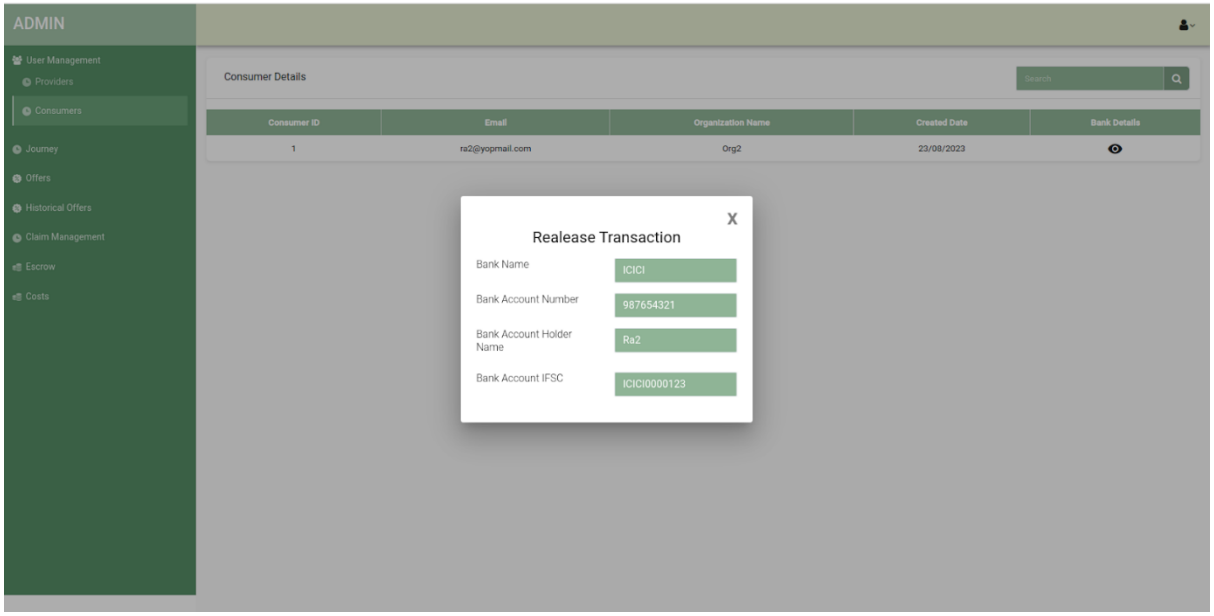

Figure 5.21: Consumer bank details.

5.3.1.2 Journey

Administrators have the ability to create journey schedules, defining the time, date, and location of upcoming events. This ensures a structured timeline for data-sharing activities between providers and consumers.

- **List of Journey:** Additionally, administrators have the capability to oversee journeys within the platform. They can create, modify, and monitor journey schedules, ensuring that data sharing activities are structured and organised. See Figure 5.22.

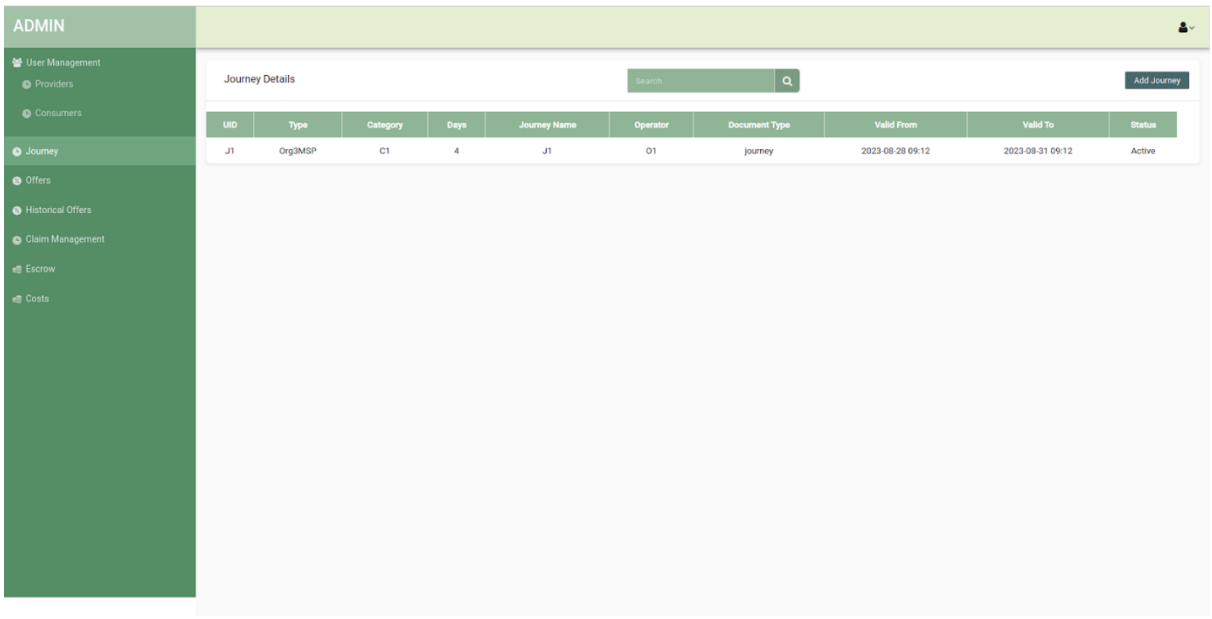

Figure 5.22: List of journeys.

- **Create Journey:** Administrators hold the authority to create journey schedules within the platform. This functionality empowers administrators to define the specifics of upcoming events.See Figure 5.23.

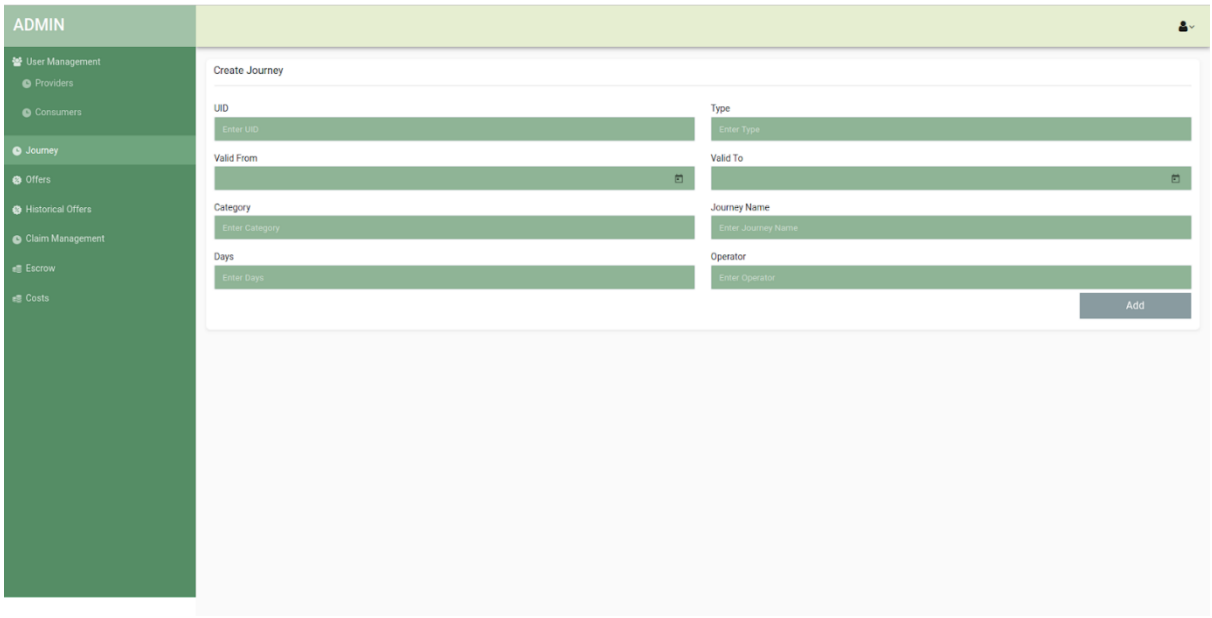

Figure 5.23: Create journey.

5.3.1.3 Offers

If providers create offers, all the details of these offers are listed and made available to administrators. See Figure 5.24.

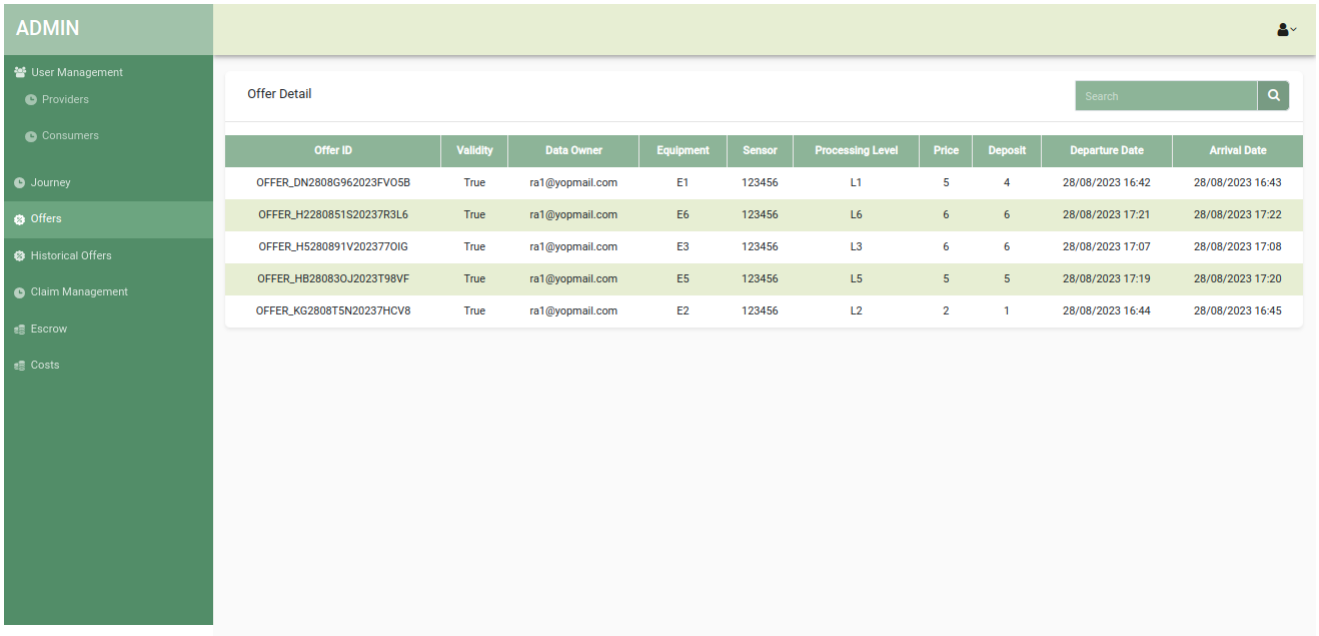

Figure 5.24: List of offers.

5.3.1.4 Historical Offers

For historical offers that are initiated by providers, all the relevant information regarding these historical offers is also listed and accessible to administrators. See Figure 5.25.

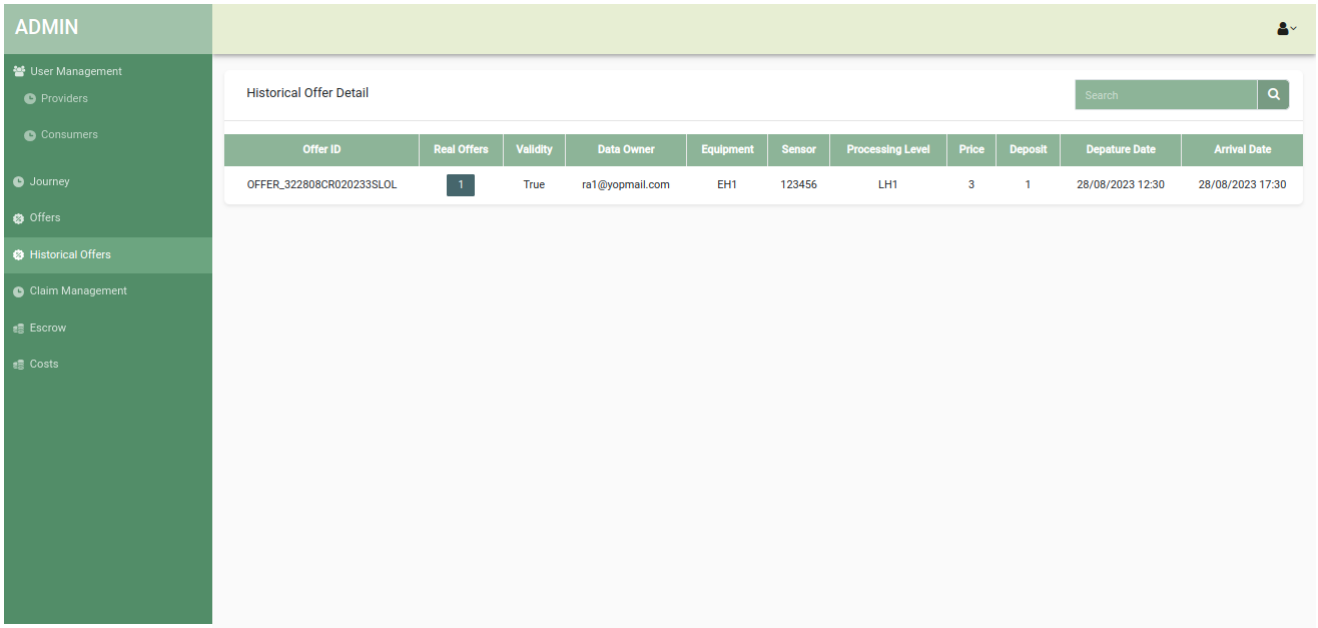

Figure 5.25: List of historical offers.

5.3.1.5 Claim Management

In the context of claim management, administrators have the ability to set a specific time frame within the platform. This designated time frame allows users to resolve and settle claims during that predefined period. See Figure 5.26.

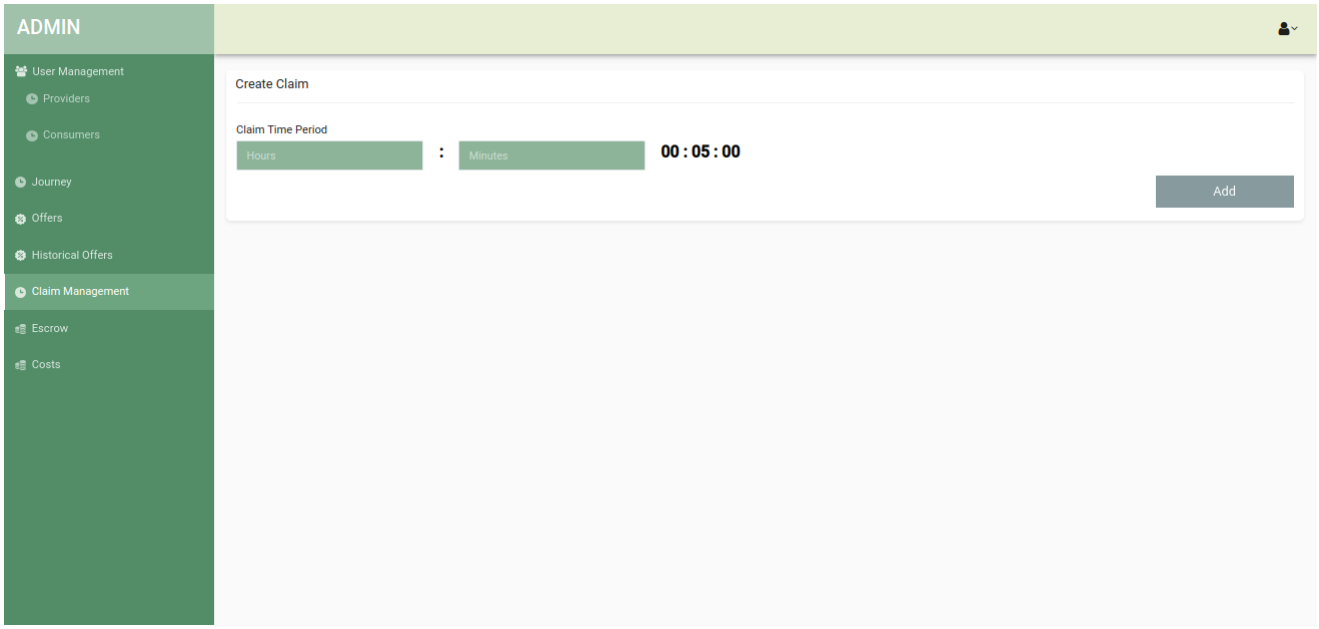

Figure 5.26: Claim management.

5.3.1.6 Escrow Management

This module oversees the escrow process as detailed earlier, which entails retaining funds between the data provider and the consumer until the agreement is successfully completed. Administrators are granted access to a list that contains all ongoing escrow transactions. This access enables administrators to monitor and oversee the financial interactions occurring between parties, ensuring transparency and accountability throughout the process. See Figure 5.27.

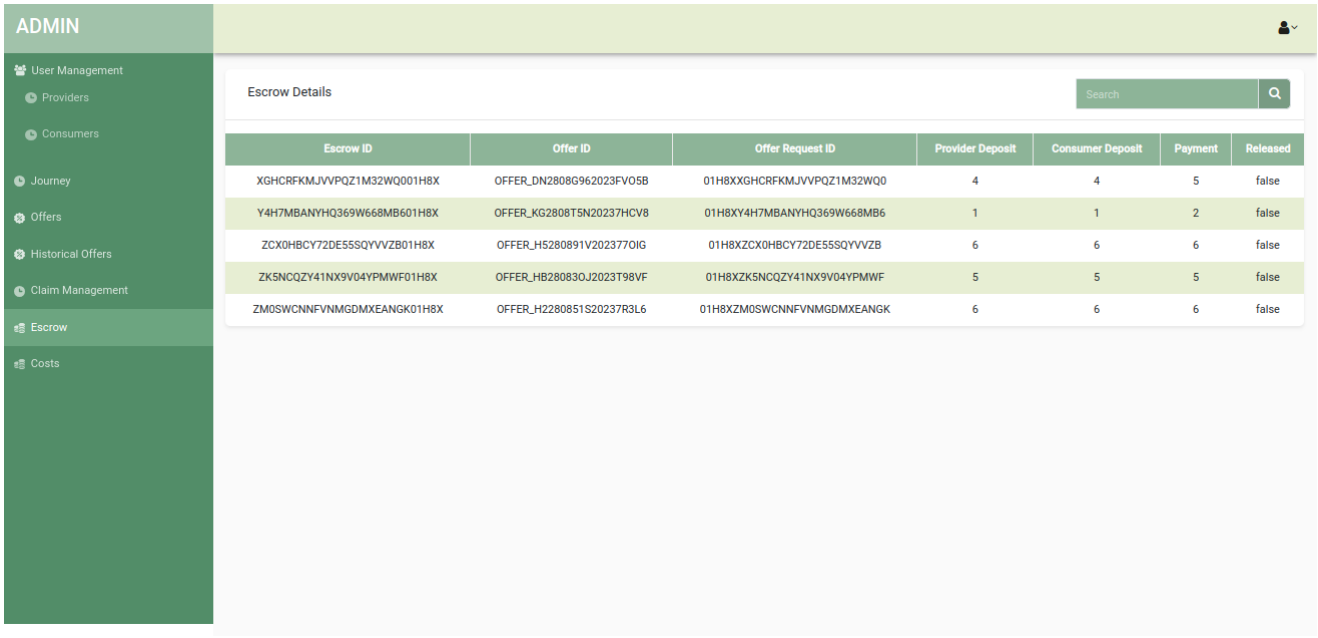

Figure 5.27: Escrow management.

5.3.1.7 Cost Management

When the escrow is released, either at the end date of the agreement or upon offer rejection, a detailed "Cost Distribution" record is generated. This record provides a comprehensive breakdown of how the funds are distributed between the consumer and the provider. It accurately reflects the payments made by each party involved in the transaction. This informative record is accessible to administrators, allowing them to review and verify the financial distribution of the transactions. See Figure 5.28.

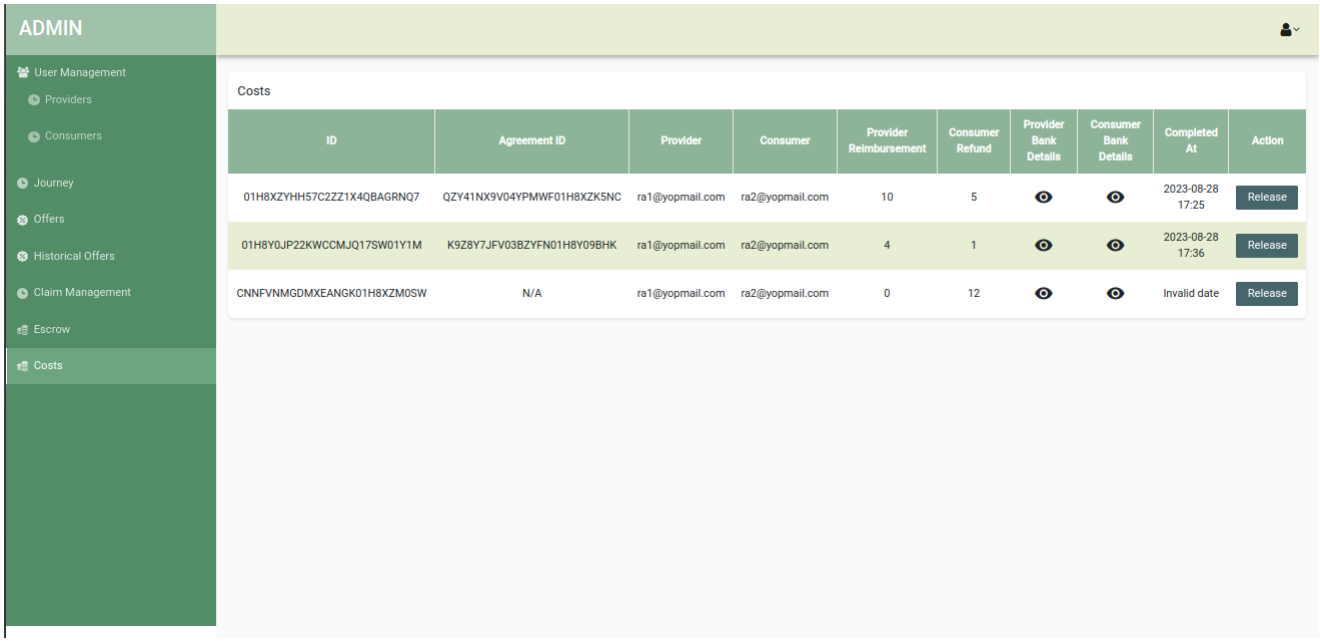

Figure 5.28: Cost management.

5.3.2 Provider Responsibilities

The provider also holds a set of responsibilities as follows:

5.3.2.1 Journey

Providers are able to view a list of available journeys and select a specific one for which they wish to create an offer. See Figure 5.29.

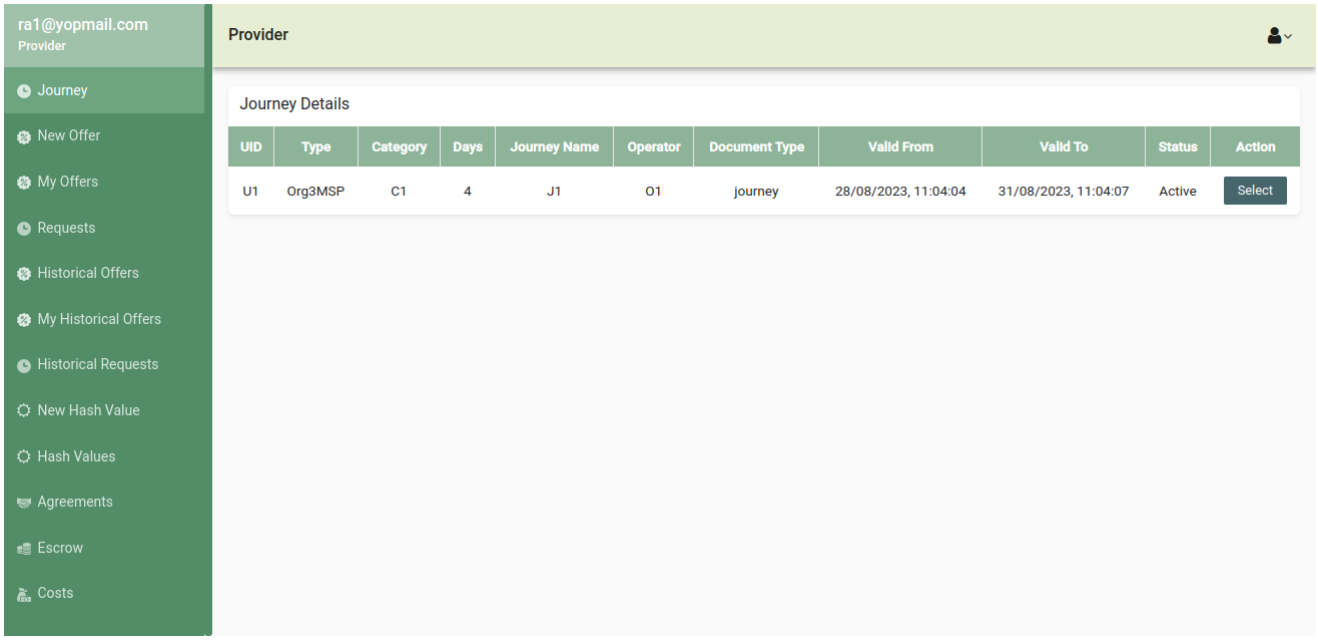

Figure 5.29: Provider list of journey.

5.3.2.2 New Offer

- Data providers can initiate the data sharing process by creating offers based on predefined journey schedules and other relevant details, making it easier for consumers to engage. See Figure 5.30.

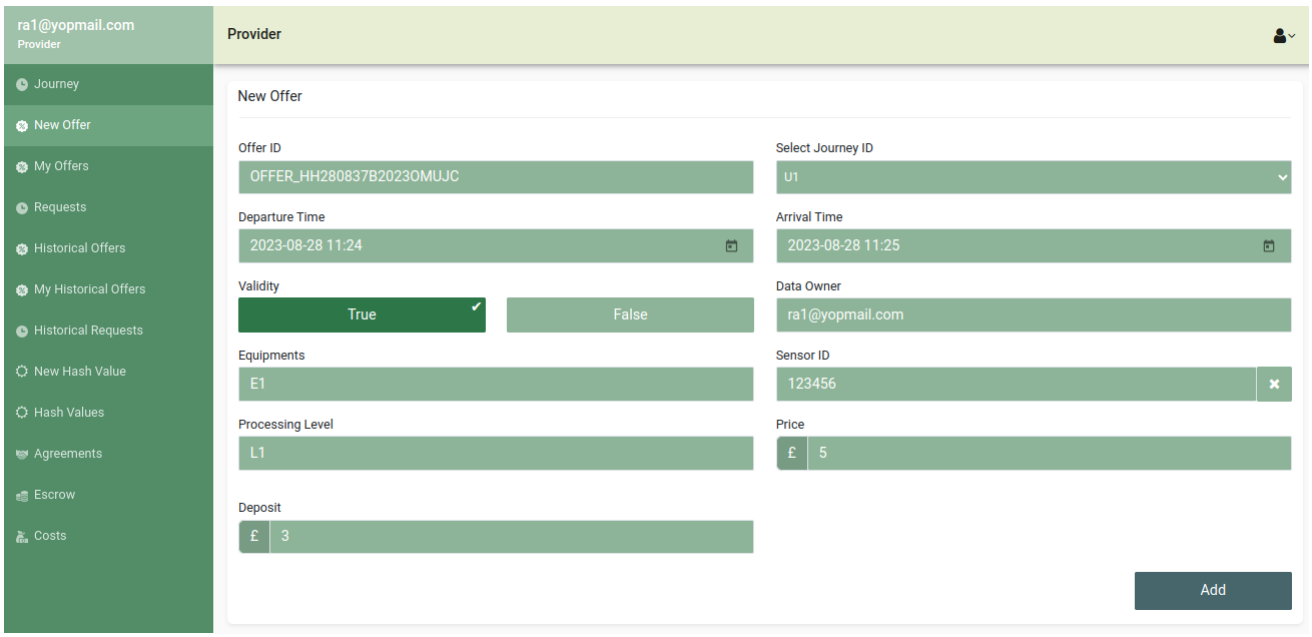

Figure 5.30: Provider create new offer.

- When creating an offer, the simulator takes requests for the start time and end time of the offer in order to schedule it. Subsequently, it begins collecting sensor data in accordance with the scheduled timeframe. See Figure 5.31 and 5.32.

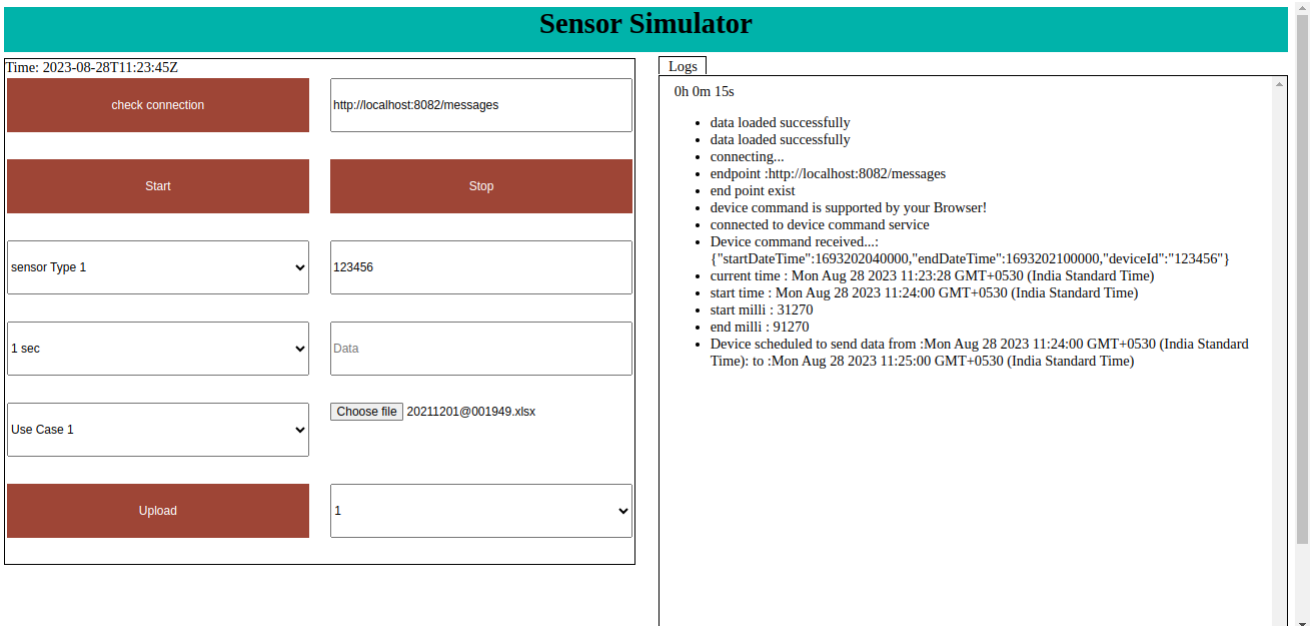

Figure 5.31: Scheduled timeframe.

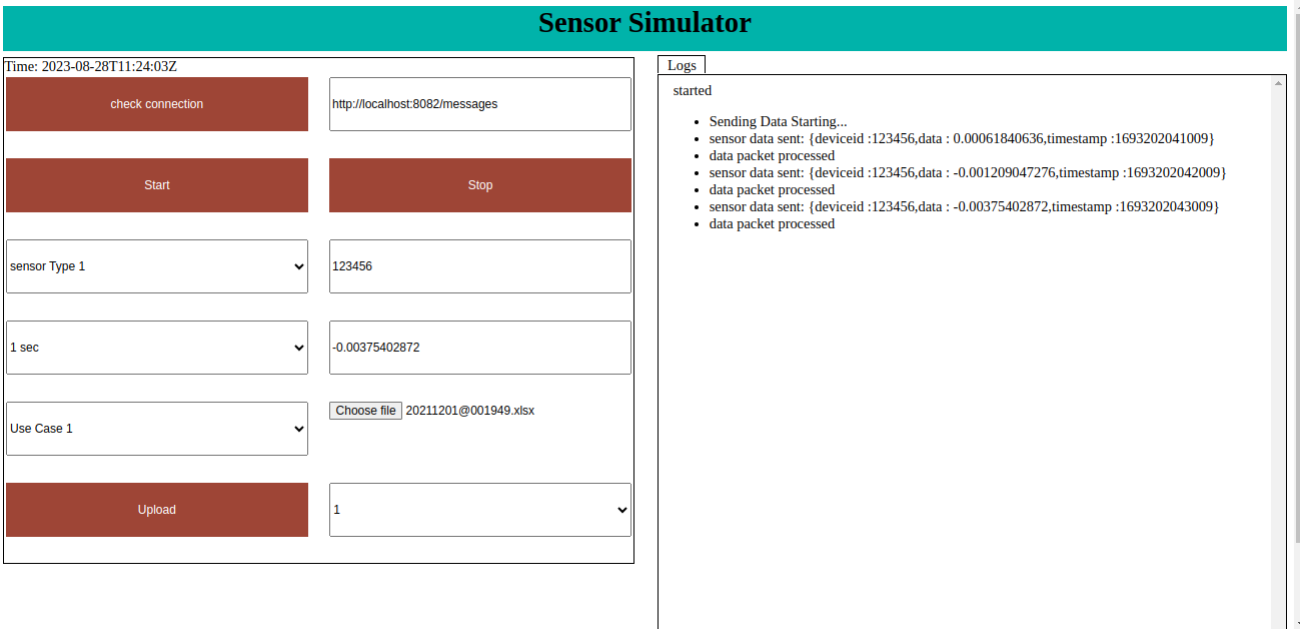

Figure 5.32: Collecting data.

5.3.2.3 My Offer

My Offer listing provides comprehensive details about the offers that providers have created. This listing includes all the pertinent information that providers need in order to craft their offers. See Figure 5.33.

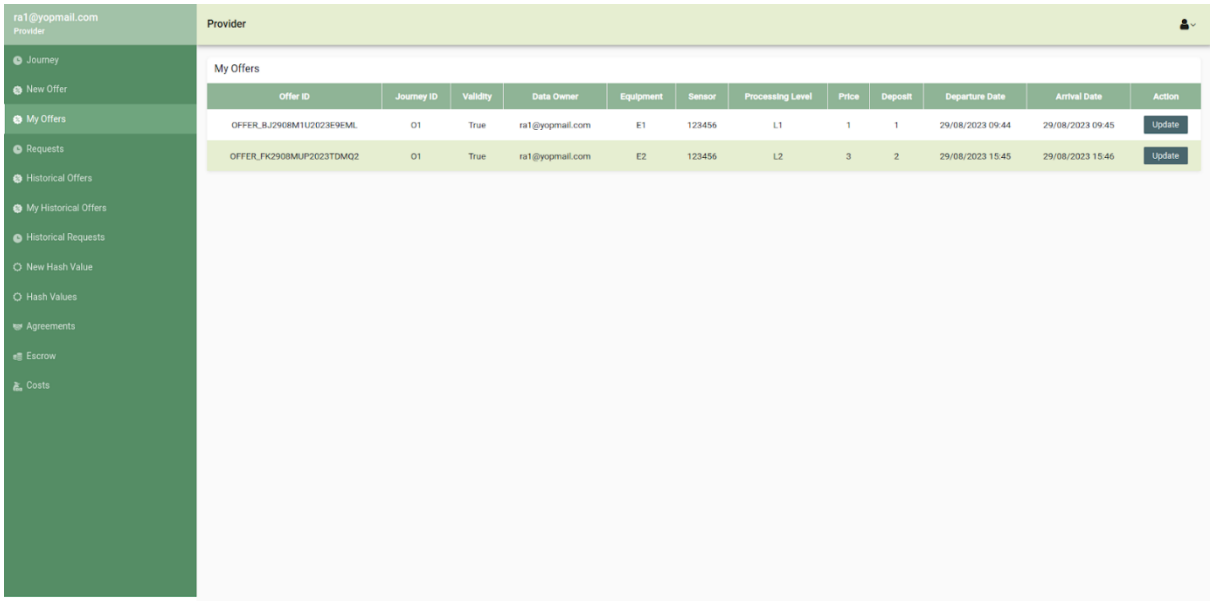

Figure 5.33: List of offers.

5.3.2.4 Requets

- The provider needs to review incoming requests and verify the details. If the details are accurate, they can accept the request; otherwise, they have the option to reject it. See Figure 5.34.

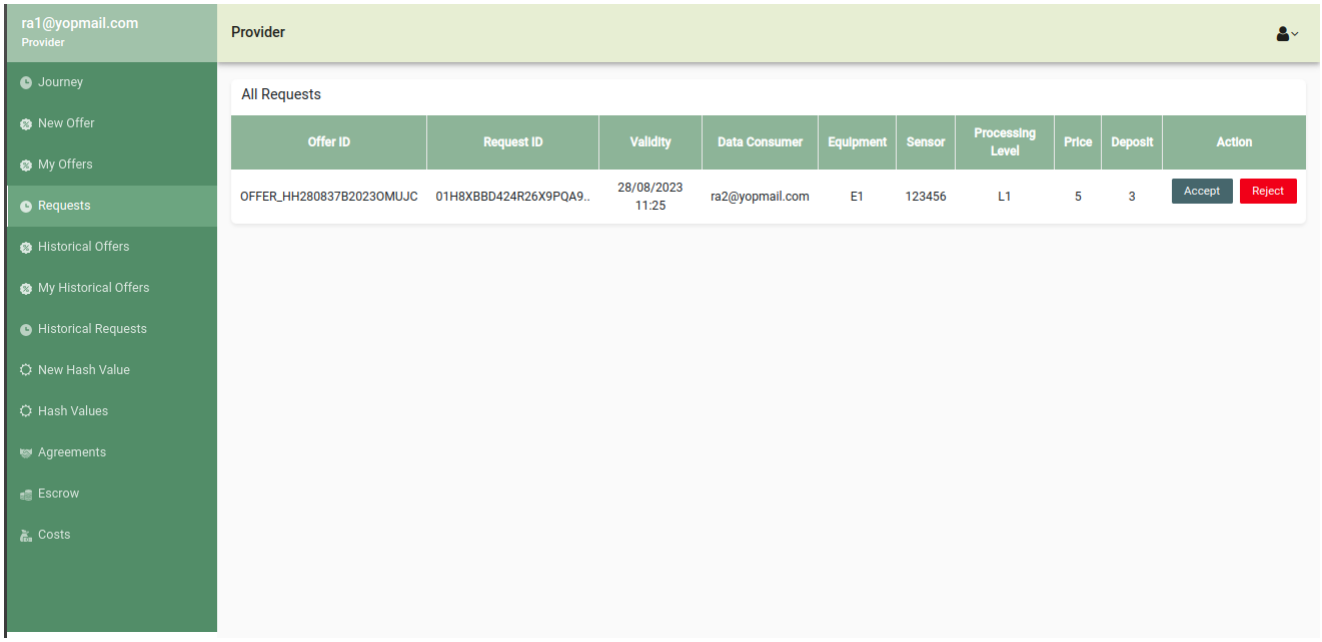

Figure 5.34: List of requests.

- If an offer is accepted, it signifies the creation of an agreement, and subsequently, the payment gateway page will be presented to facilitate the completion of the payment for the specific agreement. Additionally, the data hash will be added to the Blockchain, streamlining the process for the consumer to access and download the associated data file. See Figure 5.35.

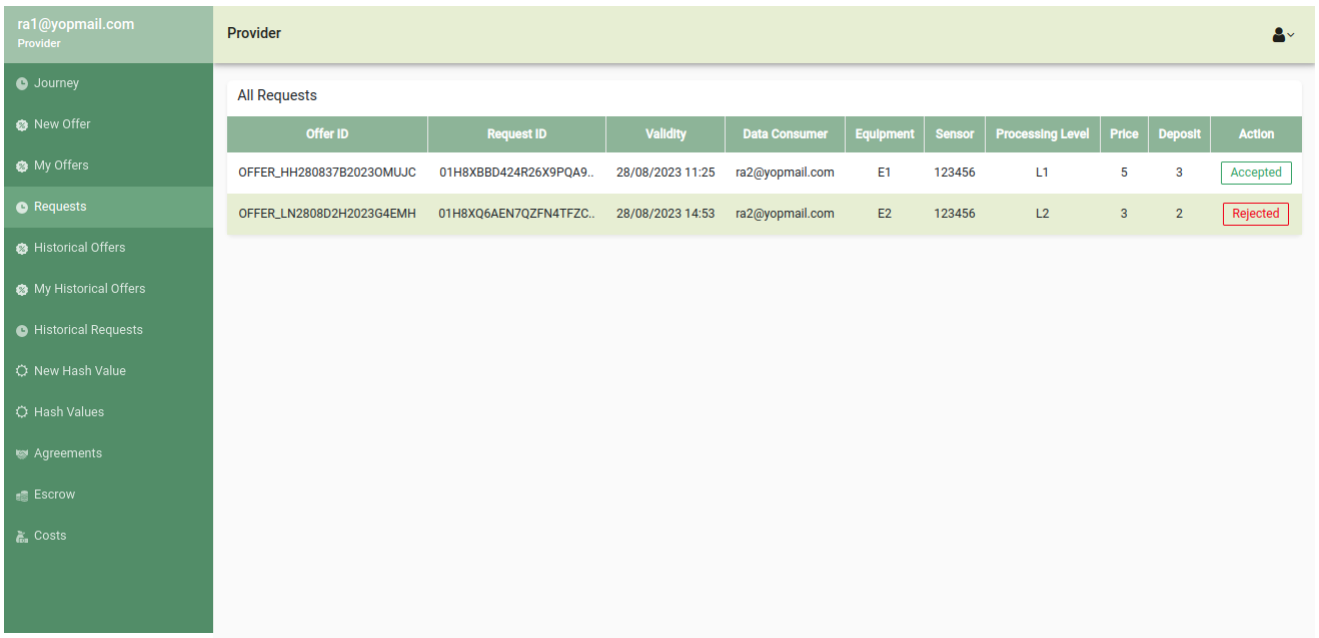

Figure 5.35: List of accepted and rejected requests.

- In the event that an offer request is rejected, the system will initiate a refund for the entirety of the consumer's deposit and fees.

5.3.2.5 Historical Offers

- Historical offers pertain to data collected previously, which the provider intends to sell to interested consumers.

- To create historical offers, a form needs to be completed; see Figure 5.36. The key elements of the historical offer form are the selection of a sensor ID, which allows for the filtering of offers based on the chosen sensor, and the specification of a time range. Offers that match the criteria within this time range are considered eligible for the historical offer category. Additionally, the form requires the selection of a journey ID for further refinement. Once these filters are applied, the resulting list provides offer IDs that match the selected criteria. See Figure 5.37.

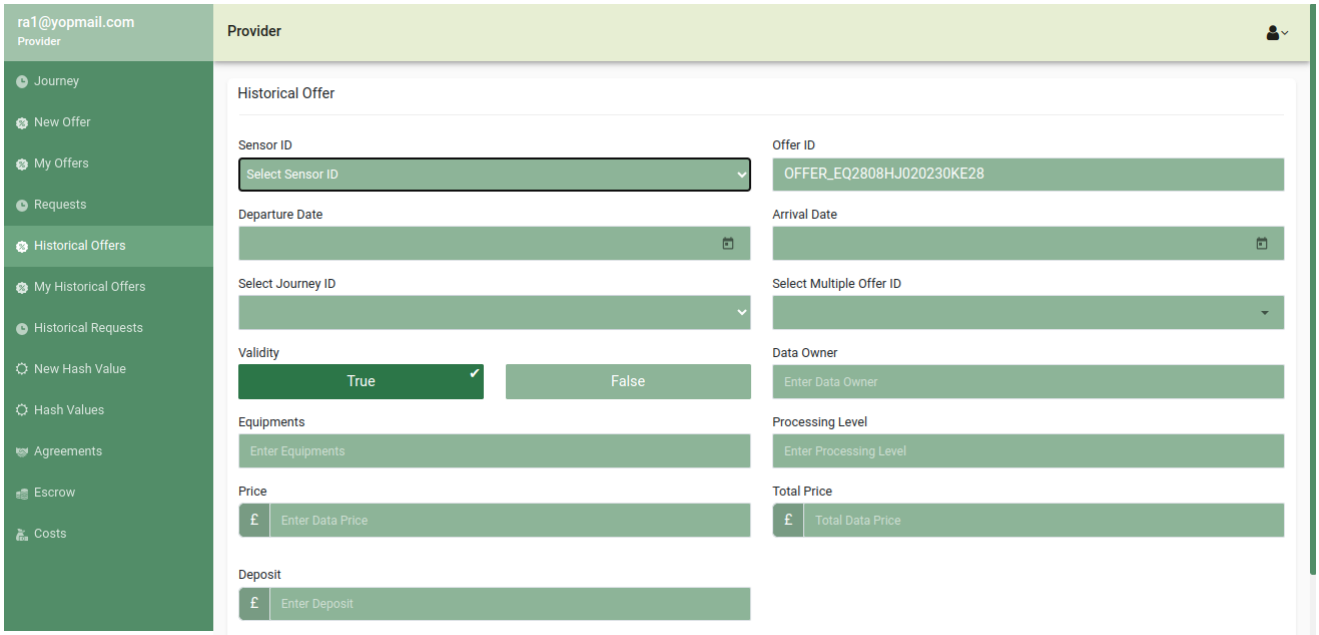

Figure 5.36: Historical offer form.

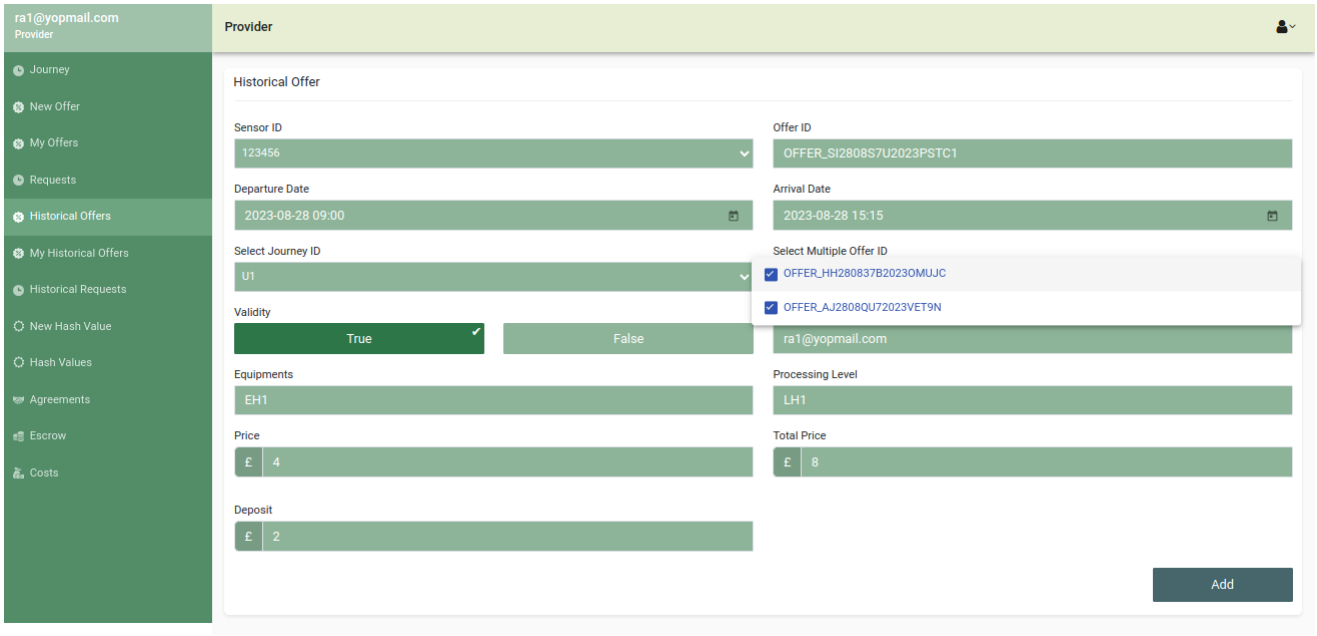

Figure 5.37: Multiple historical offer IDs.

- Subsequently, the provider selects offer IDs from this list and fills in all the necessary fields. Notably, in the "Price" section, the provider inputs the base price value, with the total price being calculated based on this value. The calculation formula is represented in Equation (5.1). See Figure 5.38.

$$\text{Total Price} = \text{Price} \times \text{Number of Selected Offers} \tag{5.1}$$

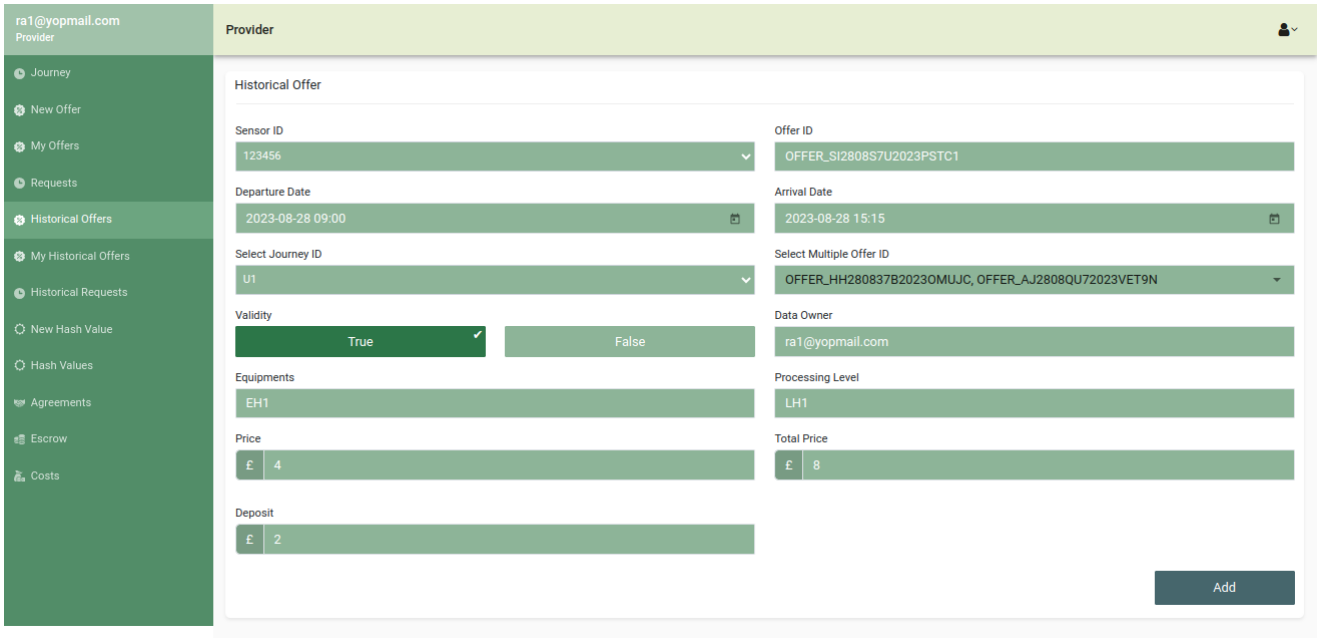

Figure 5.38: Price calculation.

5.3.2.6 My Historical Offers

Upon successful creation of historical offers, all the generated historical offers are listed in the "My Historical Offer" section. This enables providers to conveniently view and access the historical offers they have created within this specific category. See Figure 5.39.

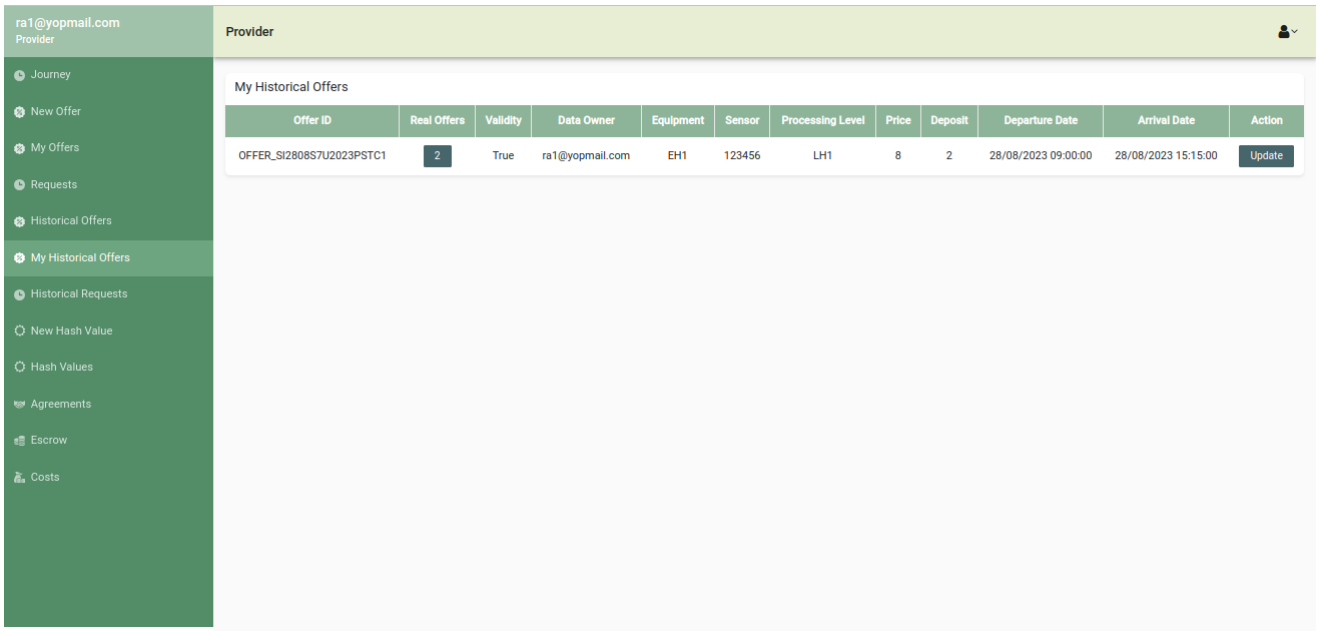

Figure 5.39: List of historical offers.

5.3.2.7 Historical Requests

- For historical offers, the provider must review incoming requests and validate the details. If the details are correct, they can proceed to accept the request. Alternatively, they retain the option to decline the request if the provided information is inaccurate. See Figure 5.40.

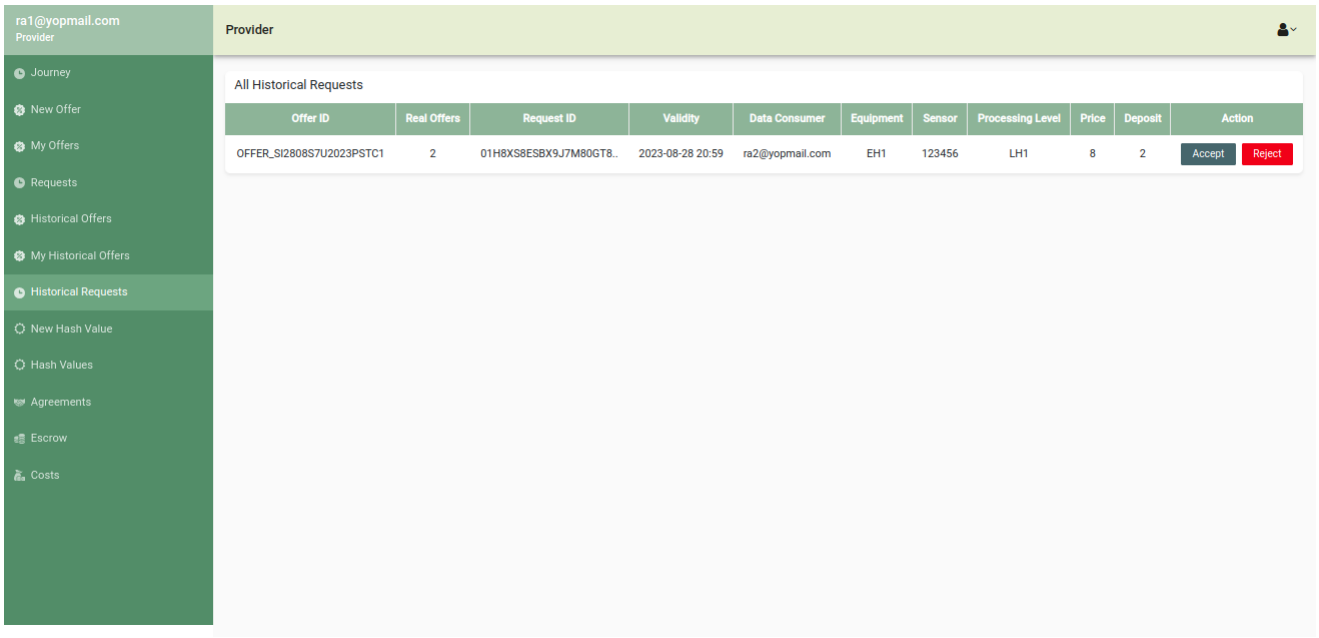

Figure 5.40: List of historical requests.

- If a historical offer request is accepted, it indicates the establishment of an agreement. Subsequently, the payment gateway page will appear to streamline the payment completion process for the specific agreement; see Figure 5.41 and Figure 5.42. Furthermore, the system can retrieve the associated offer's data hash and store its value in our database under the historical offers section, while simultaneously creating the agreement.

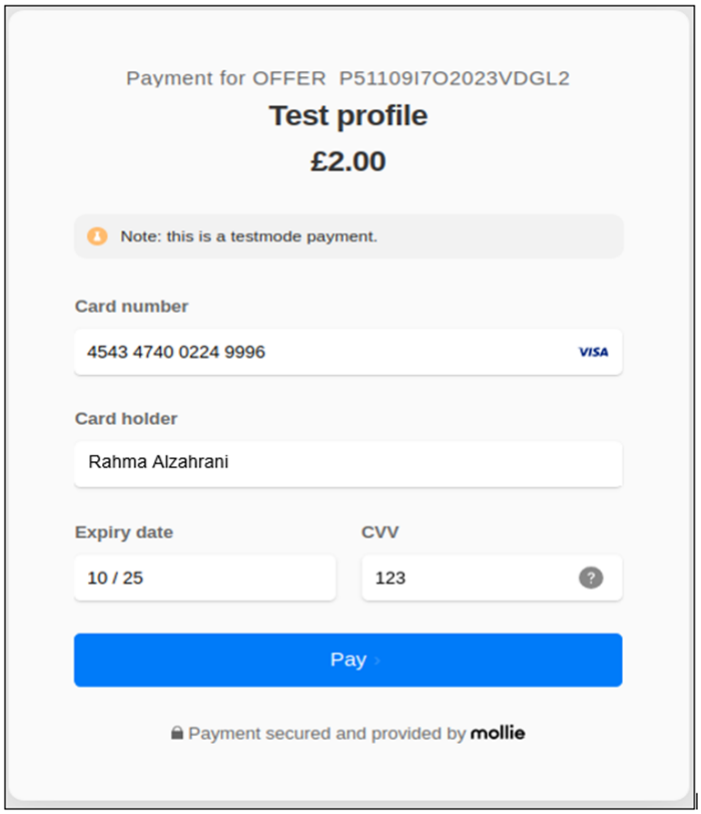

Figure 5.41: Provider payment gateway.

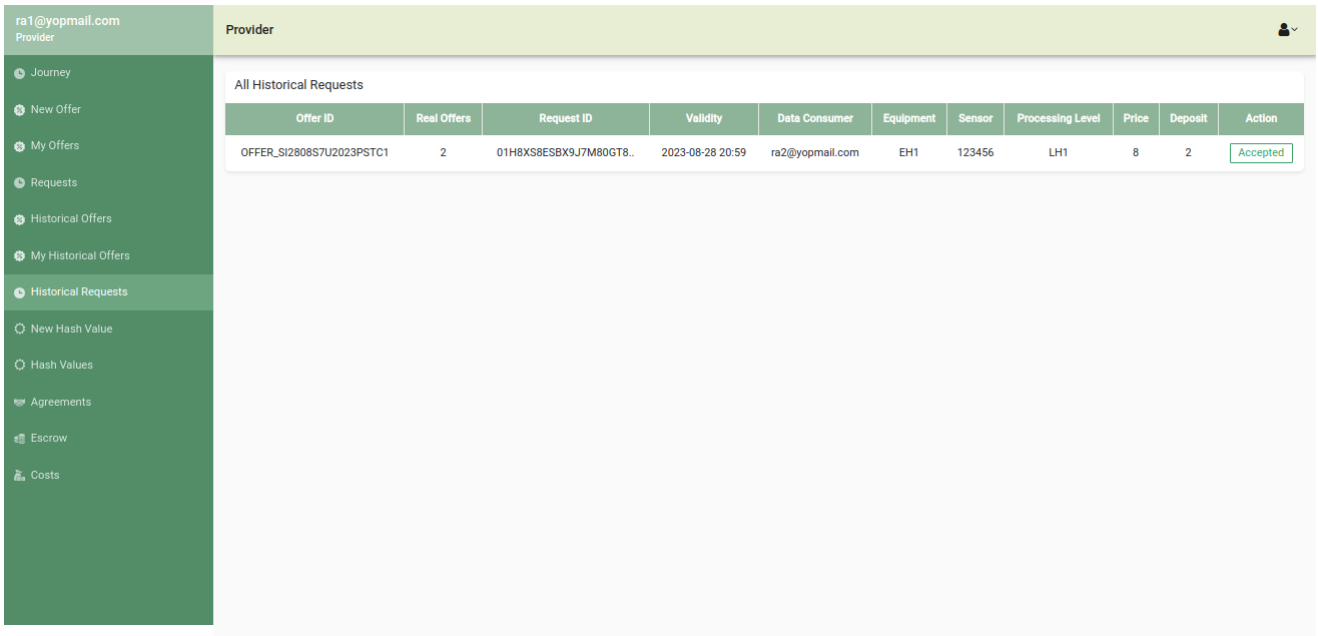

Figure 5.42: List of accepted and rejected historical requests.

5.3.2.8 New Hash Value

On the 'New Hash Value Page,' if hash data is not generated automatically, there is also a manual option available for creating a hash based on the data received from the sensor. In this case, the provider simply needs to click the 'Create' button to generate a new hash. See Figure 5.43.

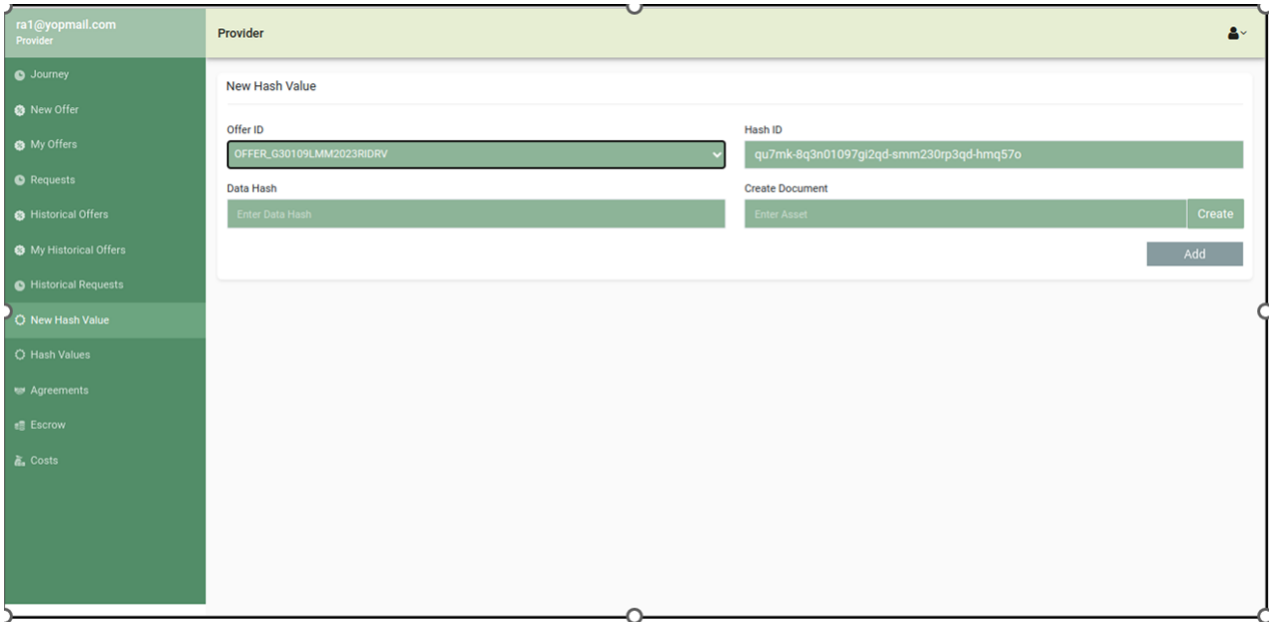

Figure 5.43: Insert new hash value.

5.3.2.9 Hash Value

On the Hash Value Page, providers have access to essential data information, including the hash ID, hash value, date of creation, and offer ID. This comprehensive data overview empowers providers to verify the authenticity and completeness of the generated data. Additionally, providers can use the offer ID as a reference point for selecting the corresponding agreement, streamlining the process of managing and confirming data integrity. See Figure 5.44.

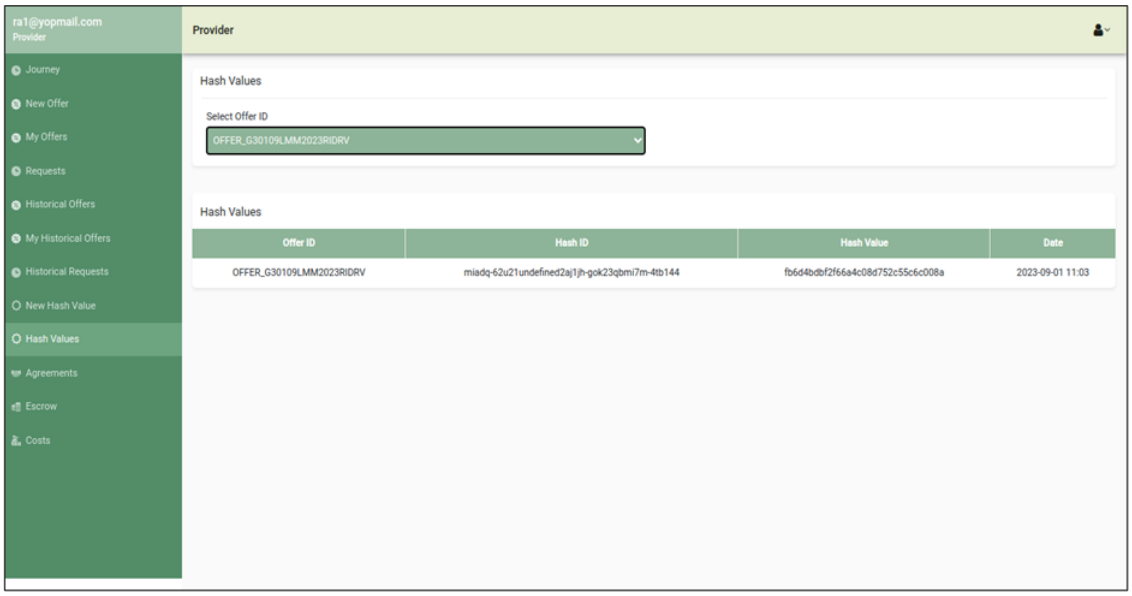

Figure 5.44: Retrieve hash values.

5.3.2.10 Agreements

Within the Agreement page, providers are afforded the ability to access a comprehensive list of all agreements established with consumers, including all relevant details. Moreover, providers possess the authority to revoke agreements when necessary. This revocation process ensures that consumers are reimbursed in full, encompassing both the deposit and the agreed-upon price, thereby facilitating a straightforward and transparent resolution process. See Figure 5.45.

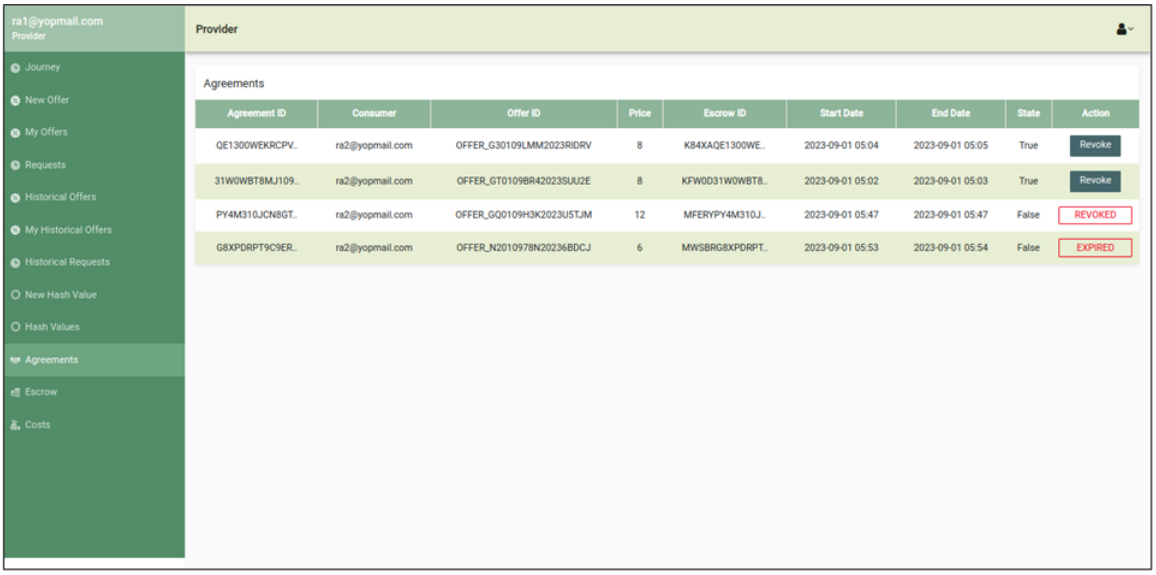

Figure 5.45: Provider agreements list.

5.3.2.11 Escrow

On the Escrow page, once an agreement is successfully completed and data is delivered to the consumer, the escrow system records all pertinent payment information. This includes the specific amounts that the consumer owes and what the provider is entitled to receive. The system also displays the release status of the escrowed funds. If the release status is marked as false, indicating that the payment has not been released, the consumer has the option to initiate a claim. However, if the release status is marked as true, indicating that the payment has been successfully released, the consumer no longer has the option to file a claim. See Figure 5.46.

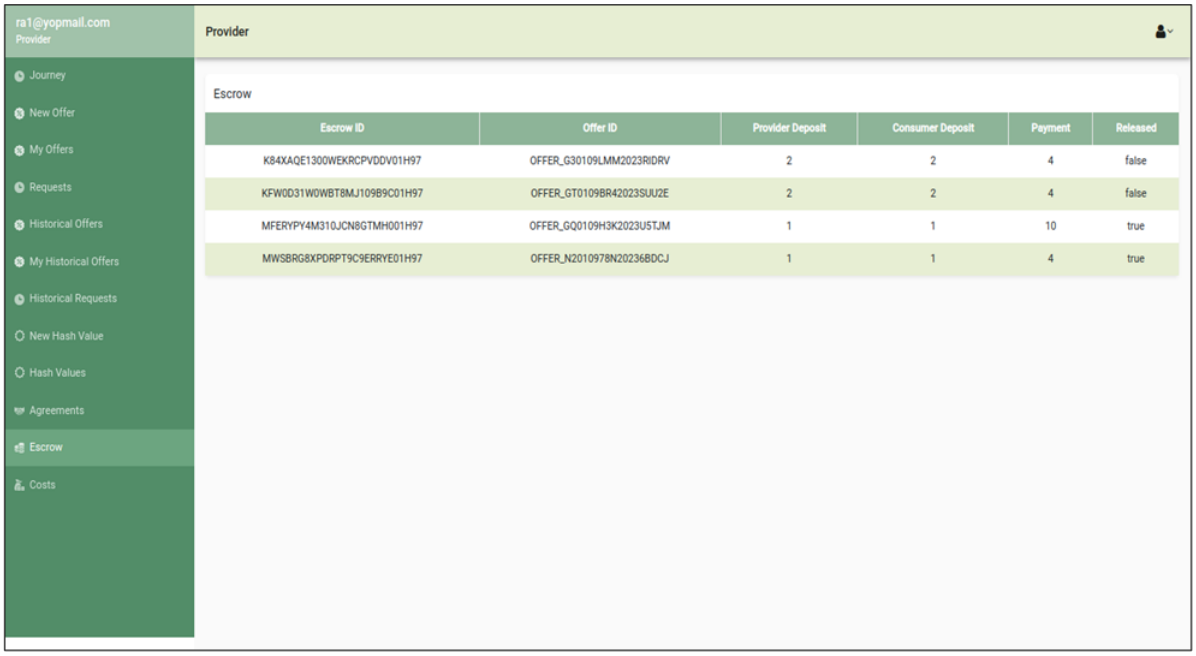

Figure 5.46: Provider escrows list.

5.3.2.12 Costs

On the Cost page, all the settlement transactions between the provider and the consumer are meticulously recorded and listed. This comprehensive record includes details regarding the amounts to be reimbursed to the provider and the refunds to be provided to the consumer, offering a clear and transparent overview of the financial interactions between the two parties. See Figure 5.47.

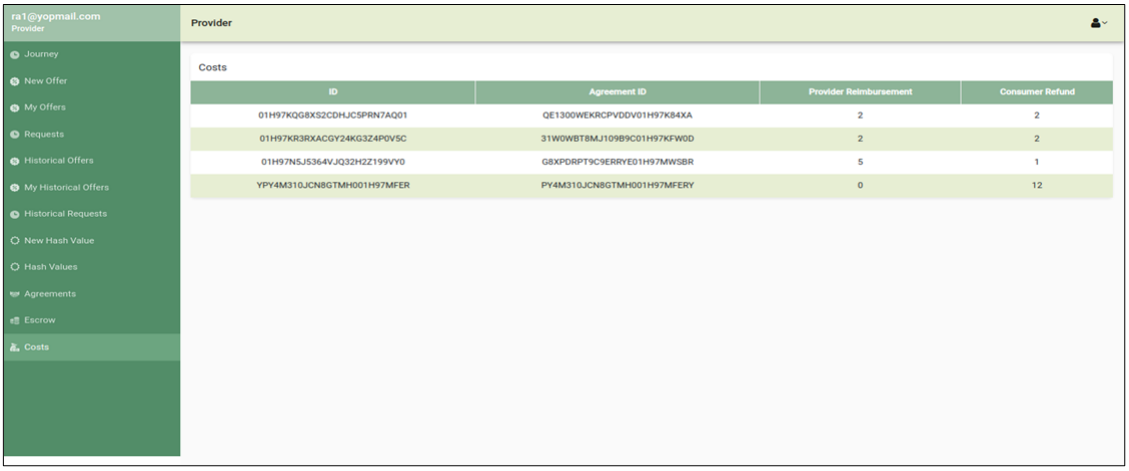

Figure 5.47: Provider costs list.

5.3.3 Consumer Responsibilities

Consumer responsibilities are as follow:

5.3.3.1 All Offer

Consumers have the flexibility to explore and request offers made by providers. By selecting suitable offers, they express their interest in acquiring specific data sets, initiating the negotiation process. See Figure 5.48.

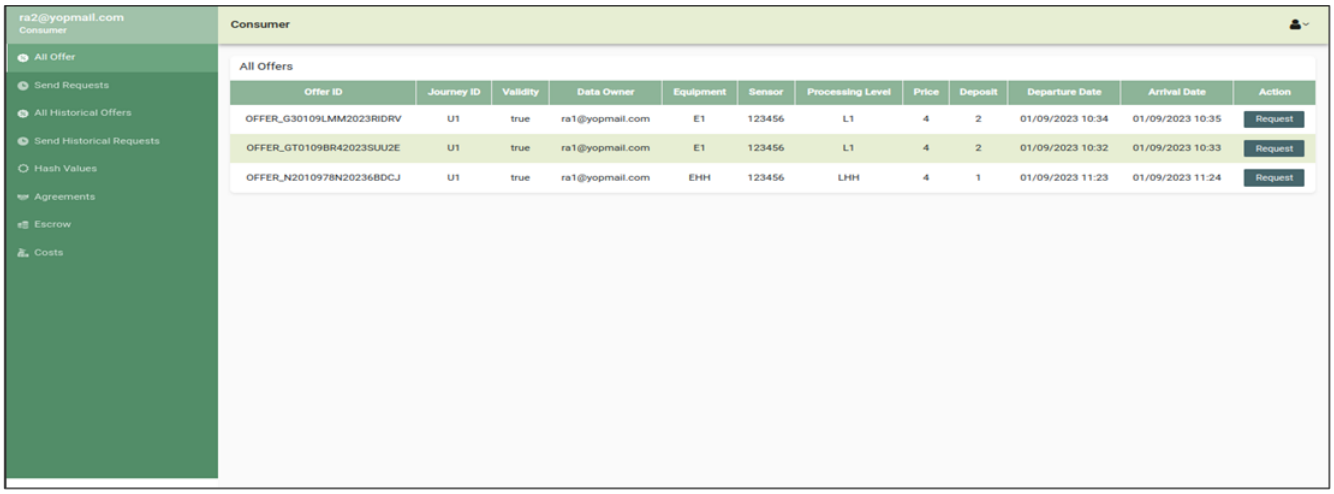

Figure 5.48: Offers list on consumer side.

5.3.3.2 Send Requests

When a consumer selects an offer of interest, they are directed to the 'Send Request' page. On this page, consumers can review all the information related to the selected offer. This enables consumers to make an informed decision. Once satisfied, they can proceed to select the offer and make the required payment. See Figure 5.49.

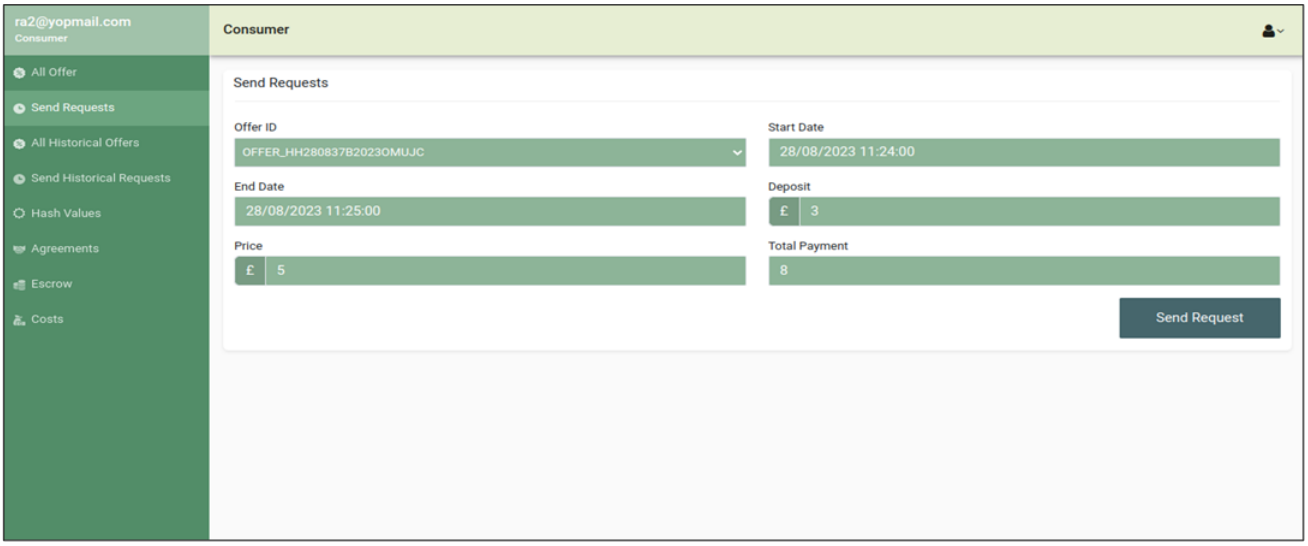

Figure 5.49: Send data request from the consumer side.

After requesting the selected offer, the payment gateway page will appear, allowing the consumer to proceed with making the payment for the offer. See Figure 5.50.

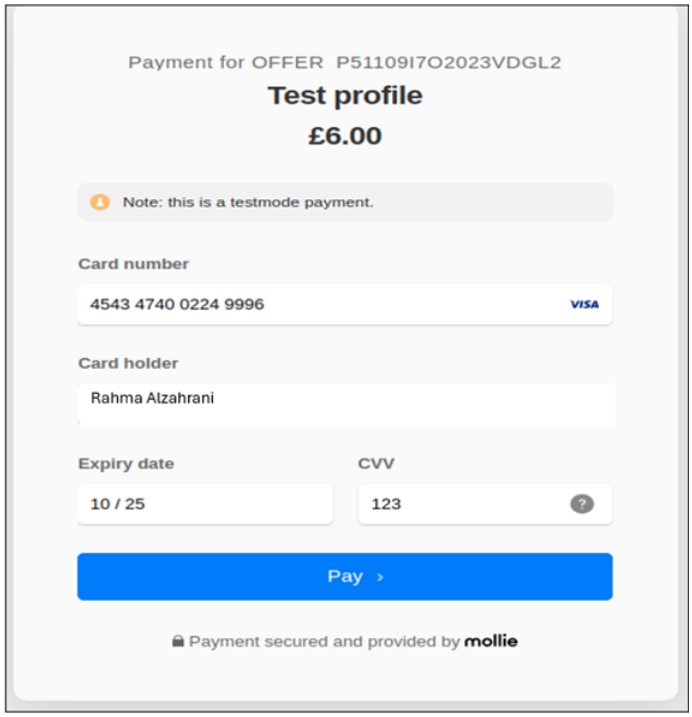

Figure 5.50: Consumer payment gateway.

5.3.3.3 All Historical Offers

Consumers have the flexibility to explore and request historical offers provided by the providers. By choosing an appropriate historical offer, they demonstrate their interest in acquiring certain datasets, thus beginning the negotiation process. See Figure 5.51 and Figure 5.52.

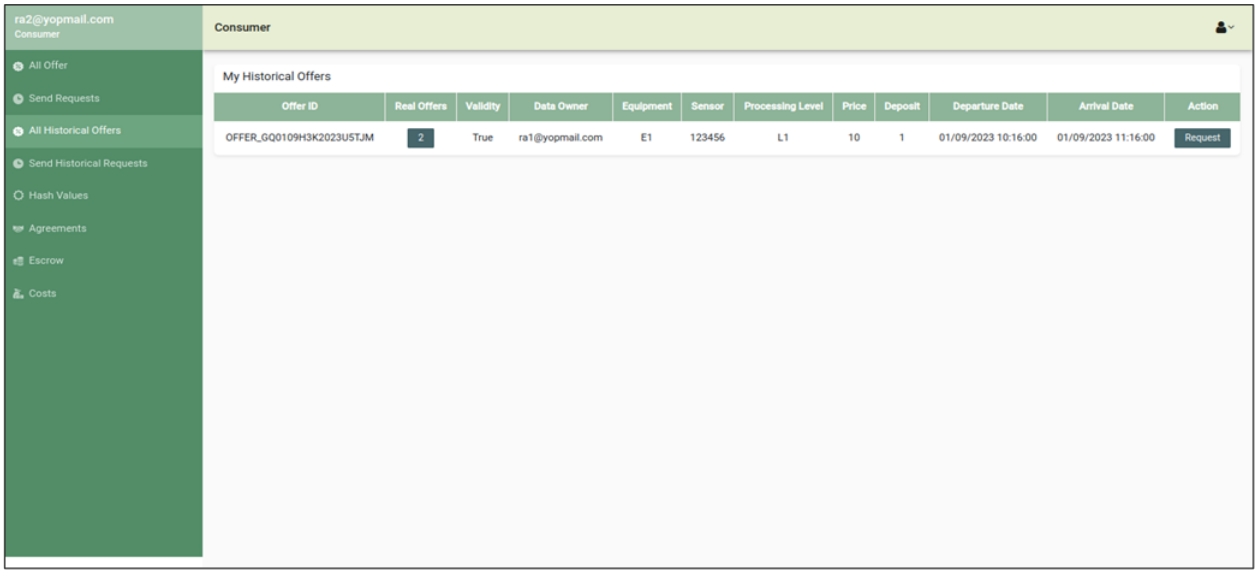

Figure 5.51: List of all historical offers on the consumer side.

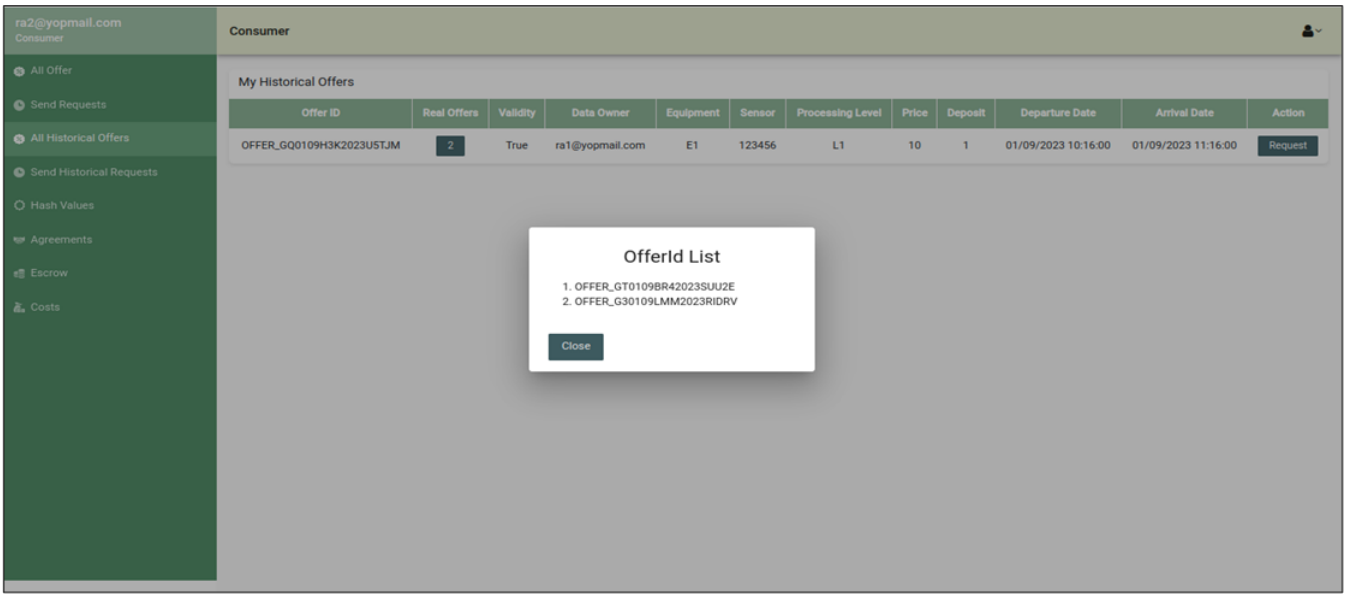

Figure 5.52: List of all historical offers on the consumer side -selected.

5.3.3.4 Send Historical Requests

When a consumer chooses a historical offer of interest, they are directed to the 'Send Historical Request' page. On this page, consumers have the opportunity to review all the pertinent information regarding the selected offer. This empowers consumers to make an informed decision. Consumers are free to choose the offer that aligns with their needs. Additionally, the pricing is recalculated based on the chosen offers. Once satisfied, consumers can proceed to select the historical offer and complete the necessary payment. See Figure 5.53 and Figure 5.54.

After requesting the selected historical offer, the payment gateway page will appear, allowing the consumer to proceed with making the payment for the historical offer.

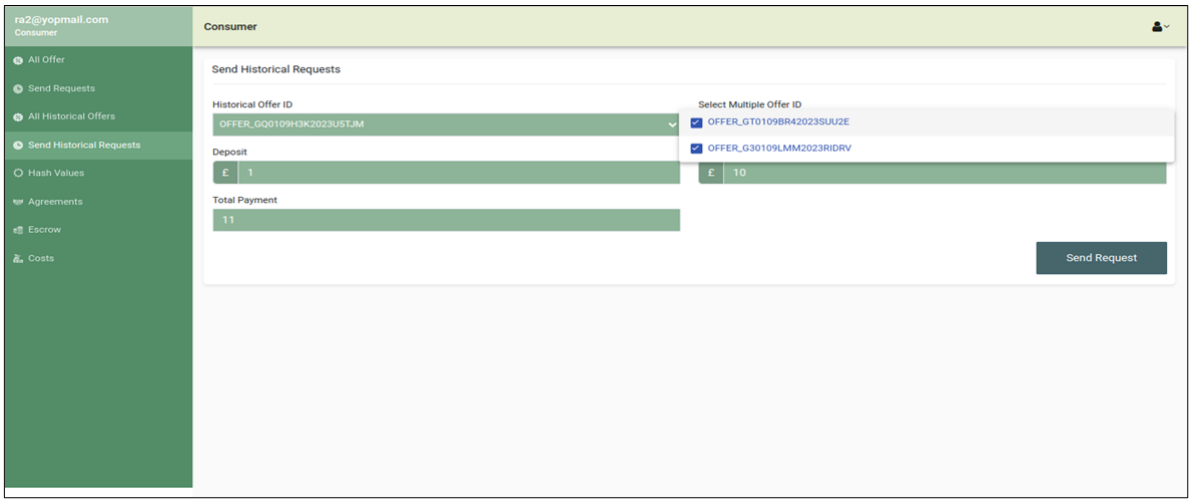

Figure 5.53: Send historical data request form.

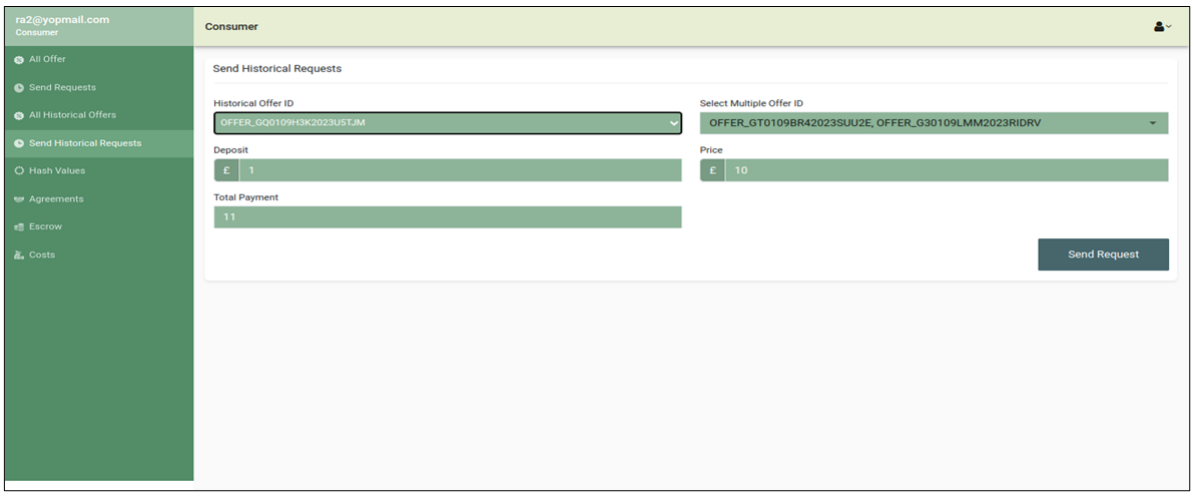

Figure 5.54: Send historical data request form -selected.

5.3.3.5 Hash Value

- On the Hash Value Page, consumers are provided with access to crucial data information, comprising the hash ID, hash value, date of creation, and offer ID. Furthermore, consumers have the option to download a file as needed. Additionally, consumers can use the Agreement ID as a convenient reference point for selecting the corresponding agreement. This simplifies the process of managing and verifying the integrity of the data associated with that agreement. See Figure 5.55.

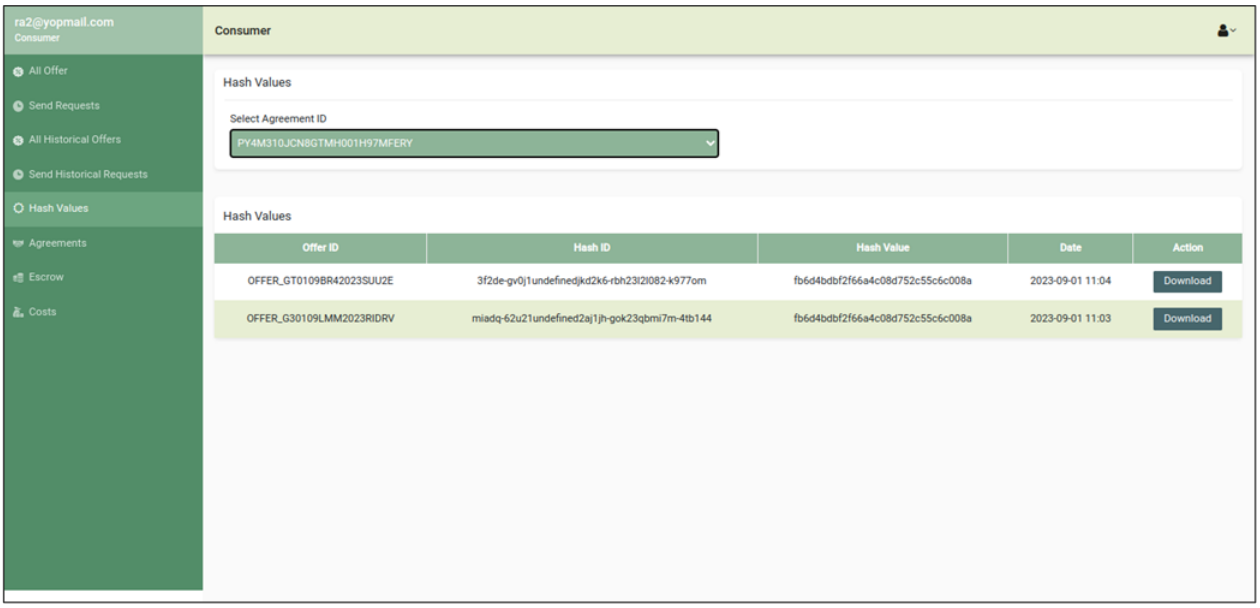

Figure 5.55: Hash values list based on agreement.

- **Claim Management:** In cases where the file is missing and the message 'No file found' is displayed, the consumer has the option to initiate a claim for a refund of the amount paid for the offer. The consumer's claim will be assessed, and costs will be determined according to the evaluation's outcome. See Figure 5.56.

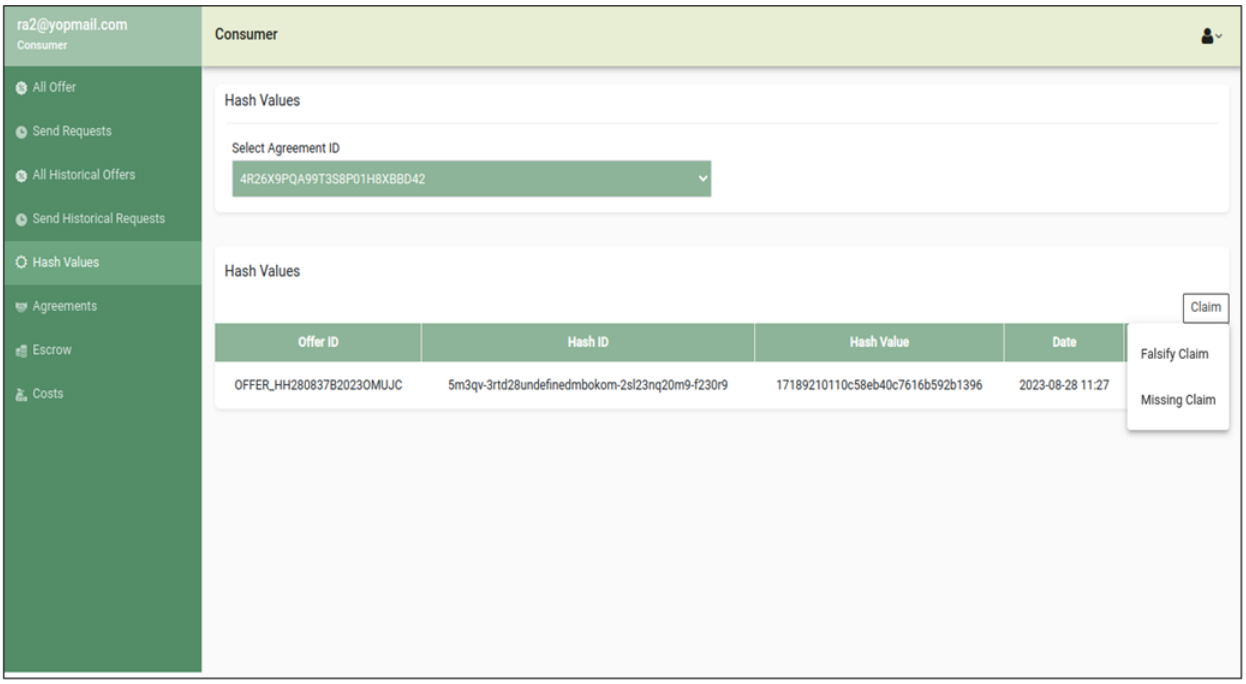

Figure 5.56: Hash values claim options.

However, if the claim is deemed incorrect, the provider will retain the consumer's deposit amount. As depicted in Figure 5.56, Claim has two options:

- Falsify Claim
- Missing File Claim

In Claim Management, as depicted in Figure 5.26, the **administrator** establishes a specific **time window** during which consumers can file claims for issues like receiving false data or missing files. Within this designated time period, consumers are permitted to initiate claims. However, once this time frame has elapsed, consumers will no longer have the option to file claims for such issues.

- **Real-time Data Claims:** In instances where a consumer fails to receive a file or suspects that a falsified file has been delivered, the consumer has the right to initiate a claim. In real-time data transactions, the verification process is more straightforward, as it involves the assessment of a single file and its corresponding hash value linked to the agreement between the provider and the consumer. If the claim is validated, meaning the consumer indeed received an incorrect file, the provider is obligated to reimburse the consumer's deposit. The cost distribution in such cases will be calculated in accordance with Equation (5.2).

$$\begin{gathered} Consumer_{\text{Refund}} = Provider_{\text{Deposit}} + Consumer_{\text{Deposit}} + Consumer_{\text{Price}} \\ Provider_{\text{Refund}} = 0£ \end{gathered} \tag{5.2}$$

.

However, if the claim is found to be invalid, the consumer is required to forfeit their deposit to the provider. In this scenario, the cost distribution is calculated according to Equation

(5.3).

$$\begin{aligned} Provider_{\text{Refund}} &= Provider_{\text{Deposit}} + Consumer_{\text{Deposit}} + Consumer_{\text{Price}} \\ Consumer_{\text{Refund}} &= 0£ \end{aligned} \tag{5.3}$$

Data files become available to users after the specified contract end date has passed, at which point the Claims Feature for Data Files becomes active. If no claim is raised within the specified time period, the cost distribution will follow the standard calculation outlined in Equation (5.4).

$$\begin{aligned} Consumer_{\text{Refund}} &= Consumer_{\text{Deposit}} \\ Provider_{\text{Refund}} &= Consumer_{\text{Price}} + Provider_{\text{Deposit}} \end{aligned} \tag{5.4}$$

This framework ensures a fair and transparent approach to resolving disputes concerning the receipt of incorrect or missing files, fostering trust and accountability between consumers and providers in real-time data exchanges.

- **Historical Data Claims:** Consumers possess the right to submit claims when requesting historical data if they identify that certain files are either missing or falsified. Addressing claims in the context of historical data requests is inherently more intricate than in real-time data transactions, as each agreement may involve multiple files, each linked to a distinct hash value. In situations where a consumer has paid for several offers and subsequently finds that the relevant files were not delivered or were falsified, the consumer is entitled to file a claim for those particular offers.

  If the claim is found to be invalid, the consumer's deposit will be deducted and the cost distribution will be calculated according to Equation (5.3). However, if the consumer's claim is deemed valid according to the validation process detailed in Algorithm 1, they will receive a refund for the amount paid, and the cost distribution will be calculated according to Equation (5.5).

$$\begin{aligned} Consumer_{\text{Refund}} &= Provider_{\text{Deposit}} + Consumer_{\text{Deposit}} + Penalty \\ Provider_{\text{Refund}} &= Total\ File\ Cost - Penalty \end{aligned} \tag{5.5}$$

Example Scenario

Total Expected Data File Hashes = 4

Data Files Received = 3

Missing Files = 1

Provider's Deposit = £20

Consumer's Deposit = £20

Cost of Each File = £25

Total File Cost = £25 * 4 = £100

Penalty Calculation = $\frac{\text{Missing Files}}{\text{Total Files}} = \frac{1}{4}$

Penalty = Total File Cost * Penalty = £100 * $\frac{1}{4}$ = £25

The cost distribution will be calculated as follows:

$$\text{Consumer}_{\text{Refund}} = \text{Provider's Deposit} + \text{Consumer's Deposit} + \text{Penalty} =$$

$$20 + 20 + 25 = 65$$

$$\text{Provider}_{\text{Refund}} = \text{Total File Cost} - \text{Penalty} = 100 - 25 = 75$$

Algorithm 1 offers a comprehensive framework for detecting instances of missing files within the scope of a data exchange, as governed by the smart contract function *FalsifyClaimUseCase2*. This algorithm rigorously analyses the data hashes associated with a specific agreement by comparing them with the files actually received. Through this verification process, discrepancies such as missing or falsified files are identified, and appropriate penalties are calculated. These

Algorithm 1 Calculate Falsify Count and Penalties

```
1: function FalsifyClaimUseCase2(dataAgreement, offerDataHashes, hashes)
2:     missingDataHashes ← Create an empty list
3:     for each hash in dataAgreement.OfferDataHashID do
4:         found ← false
5:         for each offerDataHash in offerDataHashes do
6:             if hash == offerDataHash.DataHashes[0].ID then
7:                 found ← true
8:                 break
9:             end if
10:        end for
11:        if !found then
12:            missingDataHashes.append(hash)
13:        end if
14:    end for
15:    FalsifyCount ← len(hashes) - len(missingDataHashes)
16:    if len(missingDataHashes) > 0 then
17:        logger.Info("Missing data hash(es) - applying penalty")
18:        refundAmount ← 0
19:        missingFilesRefund ← len(missingDataHashes) / len(hashes)
20:        penalty ← Round(dataAgreement.Price × missingFilesRefund)
21:        refundAmount ← Round(penalty + dataAgreement.ConsumerDeposit + dataAgreement.ProviderDeposit)
22:        cost.ProviderReimbursement ← dataAgreement.Price - penalty
23:        cost.ConsumerRefund ← refundAmount
24:    else
25:        logger.Info("No missing data hashes")
26:        cost.ProviderReimbursement ← dataAgreement.ProviderDeposit + dataAgreement.ConsumerDeposit + dataAgreement.Price
27:        cost.ConsumerRefund ← 0
28:    end if
29:    return FalsifyCount, missingDataHashes, cost
30: end function
```

penalties are then allocated according to established rules, ensuring transparency and equity in the reimbursement and cost-sharing processes between the consumer and the provider.

The key components of the algorithm are as follows:

- **Smart Contract Definition:** The smart contract *FalsifyClaimUseCase2* is defined with three parameters: dataAgreement, offerDataHashes, and hashes.
- **Initialisation of missingDataHashes:** An empty list is created to store any missing hashes.
- **Looping Through Hashes:** The outer loop checks each hash in dataAgreement.OfferDataHashID, and the inner loop checks for matches in offerDataHashes.
- **Appending Missing Hashes:** If a match is not found, the hash is added to missingDataHashes.
- **Falsify Count Calculation:** FalsifyCount is calculated as the difference between the lengths of hashes and missingDataHashes. Calculating the difference between the total number of expected hashes and the number of missing hashes, reflecting the extent of falsified or missing data.
- **Penalty Calculation:** If there are missing hashes, the penalty and refund amounts are calculated based on the specified logic. If there are no missing hashes, reimbursement is calculated based on deposits.
- **Return Statement:** The function returns the FalsifyCount, the list of missingDataHashes, and the cost.

5.3.3.6 Agreements

In the Agreements section, consumers have the ability to access a comprehensive list of all agreements established between themselves and the provider. This list provides critical information related to each individual agreement, facilitating clear understanding and management

of their contractual engagements. In addition, consumers are given the option to revoke agreements when necessary. Initiating this action triggers a process to release the escrow and finalise the cost distribution. This mechanism promotes a transparent and efficient resolution process, safeguarding the consumer's interests. Refer to Figure 5.57.

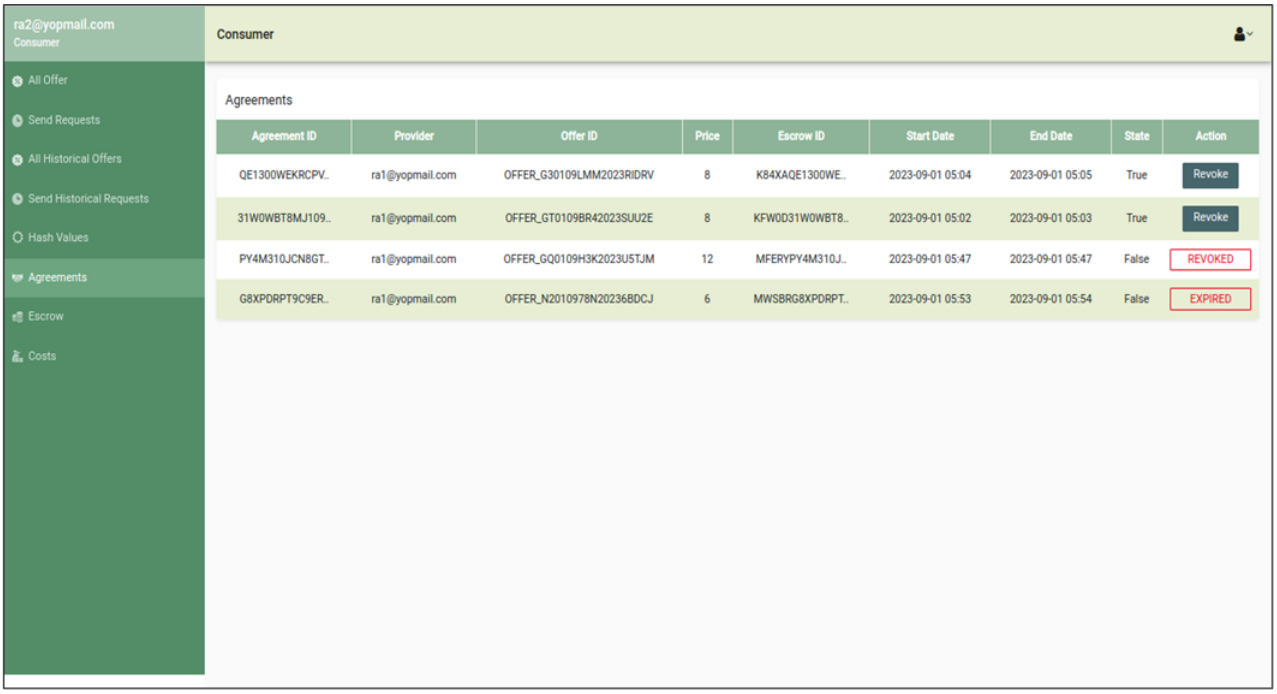

Figure 5.57: List of agreements on the consumer side.

5.3.3.7 Escrow

The Escrow page serves as a detailed repository of all critical payment-related information, capturing the precise amounts owed by the consumer and the corresponding entitlements of the provider. The system also transparently displays the release status of the escrowed funds. If the release status is marked as "false," indicating that the payment has not yet been disbursed, the consumer is presented with the option to initiate a claim. However, when the release status is marked as "true," confirming that the payment has been successfully disbursed, the claim option is disabled, as the funds have already been released. For further reference, see Figure 5.58.

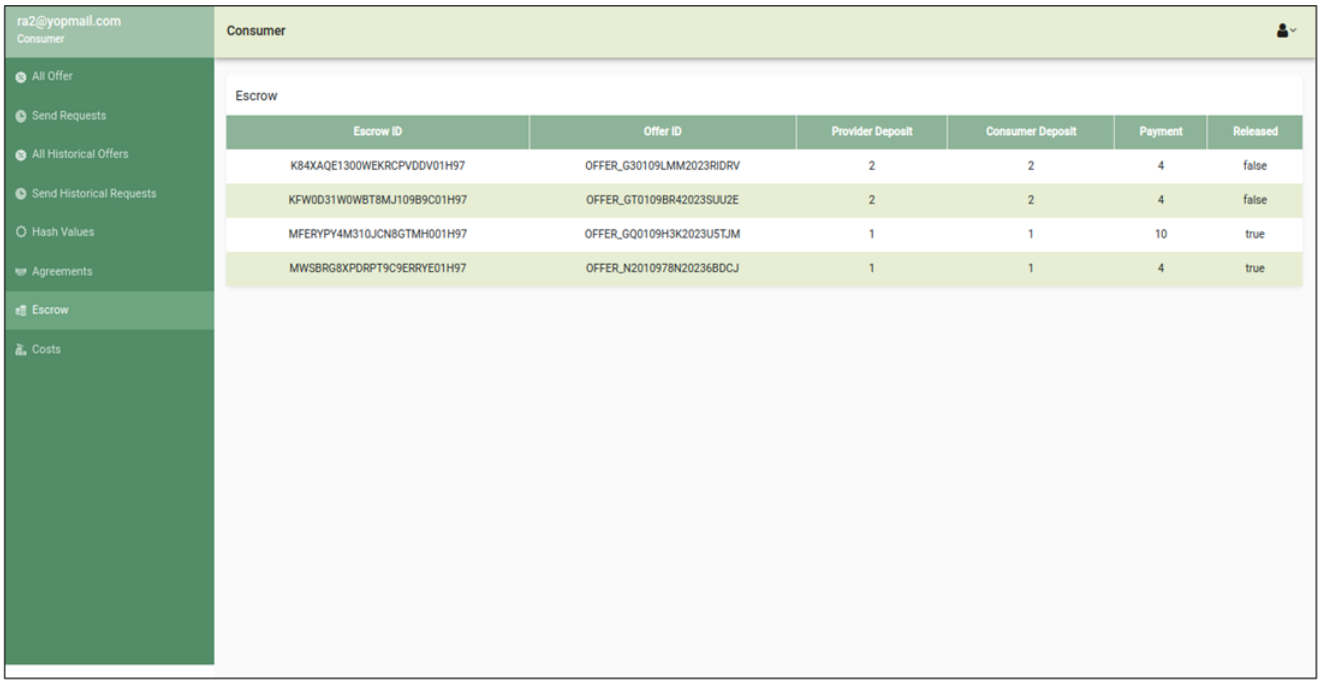

Figure 5.58: List of escrows on consumer side.

5.3.3.8 Costs

The Cost page serves as a comprehensive repository for all settlement transactions between the provider and the consumer, meticulously documenting each financial interaction. This detailed log presents an exhaustive account of the reimbursements owed to the provider and the refunds due to the consumer. By offering this extensive record, the system acts as a vital reference point, facilitating a clear and transparent overview of the financial exchanges between the two parties. This transparency is essential to ensure accountability and foster trust in the transactional process. Refer to Figure 5.59.

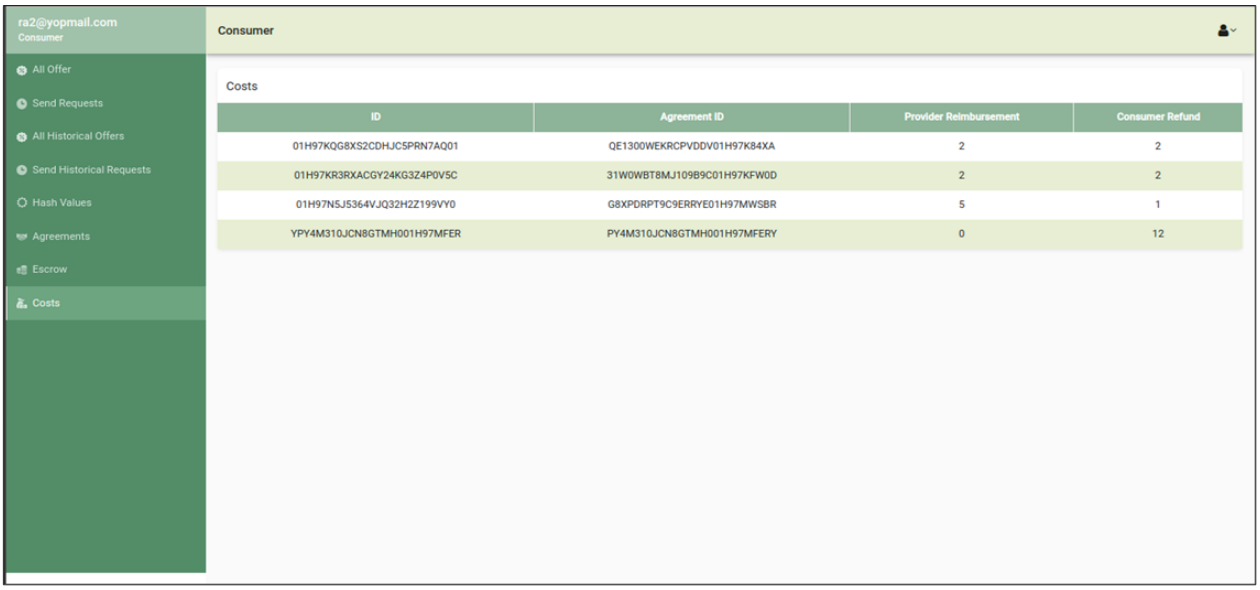

Figure 5.59: List of costs on the consumer side.

5.4 Conclusion

This chapter presents a comprehensive proof of concept for an innovative project that uses Blockchain technology to transform the exchange of historical and sensor data within the railway sector. Using the advanced capabilities of HLF, the project establishes a secure and decentralised platform that facilitates transparent and reliable data transactions. The chapter provides an in-depth exploration of the three primary user roles: Administrator, Data Provider, and Data Consumer. The roles and interactions of each role within the system are examined in detail, demonstrating how they contribute to the overall functionality and integrity of the platform.

Moreover, the chapter outlines two distinct data request scenarios: real-time sensor data and historical data. It describes how each type of request is handled, from initiation to completion, with a focus on the mechanisms in place to ensure data accuracy, security, and integrity. The methodology for calculating and distributing the costs associated with data transactions is also explained, offering insights into the platform's economic model and its approach to incentivising data sharing.

Additionally, the chapter introduces the escrow mechanism, which acts as a safeguard for transactions by holding payments until the conditions of the data exchange are fulfilled. The

integration of a payment gateway is also discussed, highlighting the steps taken to ensure that financial transactions are seamless and aligned with the decentralised nature of the platform. This chapter sets the stage for understanding the platform's key components, technical intricacies, and its potential to reshape data exchange in the railway industry.

Chapter 6

Benchmarking The Developed Application Performance

6.1 Introduction

In the railway sector, where data integrity and timely access to information are of utmost importance, the implementation of a robust and reliable blockchain solution is crucial. The system developed in this study, which uses Hyperledger Fabric, is designed to meet these requirements by enabling secure and efficient data sharing among various stakeholders within the rail network. However, to ensure that the system can withstand the stringent demands of a real-world railway environment, it is imperative to conduct a thorough performance evaluation. Performance testing is a critical phase in the system development lifecycle, as it allows for the assessment of the system's capabilities under various operational conditions and the identification of potential bottlenecks or areas for optimisation.

In this chapter, Hyperledger Caliper, a widely recognised benchmarking tool to evaluate blockchain networks, was used to measure the performance of the system developed and introduced in Chapter 5. Section 6.2 provides an introduction to Hyperledger Caliper, discussing its relevance as a performance benchmarking tool within the blockchain domain. The overall performance of the network is influenced by decisions made during the development process,

including configuration settings and the selection of Hyperledger Fabric components. Managing various factors, such as the number of channels, chaincode implementations, and transaction policies, becomes particularly challenging when multiple participating organisations contribute their own networking and hardware infrastructures to the system.

To address these complexities, the configuration of the testing environment used in this pilot study is discussed in Section 6.3. This section outlines the specific testing configuration adopted for performance evaluation. The test cases developed to evaluate the performance of the system are detailed in Section 6.4. A general discussion of the results and their implications will be presented in Section 6.5. In addition, the limitations of the study and suggestions for future research will be addressed in Section 6.6. Finally, the chapter concludes with a summary of key findings in Section 6.7.

6.2 Caliper Components and Performance Metrics

Hyperledger Caliper[1] is a scalable, distributed, and extendable testing framework to benchmark the developed network by generating and predefining different workloads/requests that target the back-end system. By Caliper, the request response time is measured and reported consequently once the result is aggregated. Accordingly, Caliper is not used as a deployment tool and doesn't manage the backend system in any way; rather, it only measures the system under testing response time. Also, Caliper is not an infrastructure monitoring solution by itself, but integrated and bonded with many enterprise tools such as HLF and Besu. The two main client components in Caliper are

- **Manager:** The manager in Hyperledger Caliper is a central coordinating component responsible for overseeing and controlling the benchmarking process. It orchestrates the interactions between the different components involved in the performance evaluation of a blockchain network, ensuring that the benchmark tests are executed according to the specified configurations and parameters.

[1] https://hyperledger.github.io/caliper

- **Workers:** In Hyperledger Caliper, workers are specialised components responsible for generating and submitting transaction requests to the Blockchain network under test. Each worker simulates a client interacting with the blockchain, and their primary role is to emulate realistic workloads to evaluate the network's performance metrics, such as transaction throughput, latency, and resource utilisation. Workers can be configured to perform a variety of transaction types, depending on the test scenarios defined by the user. They can execute predefined smart contract functions, issue queries, and send transactions with different parameters to test various aspects of the Blockchain's performance under load.

Caliper's findings are meant to serve as a guide for determining the configuration values necessary for implementing the Blockchain system. The following performance indicators will be included in the generated performance report (Hyperledger and Scale Working Group, 2021):

- **Transaction success/fail count:** The total success (committed) and failed (uncommitted) transactions when the testing rounds are completed.

- **Transactions throughput (TPS):** Transaction throughput signifies the pace at which valid transactions are executed by the Blockchain System Under Test (SUT) within a specified duration. It's worth noting that this rate isn't specific to a single node but encompasses transactions committed across the entirety of the SUT, spanning all nodes within the network. This measurement is commonly expressed as Transactions Per Second (TPS) relative to the network's size.

  Equation 6.1 illustrates how throughput is computed by dividing the total number of successful transactions (T_{succ}) by subtracting the first submitting time from the last committing time. In equation 6.2 the send rate is computed by dividing the total number of successful transactions (T_{succ}) and failed transactions (T_{fail}) by subtracting the first submitting time from the last submitting time.

- **Transaction latency:** Transaction latency provides a network-wide perspective on the duration required for a transaction's impact to become functional across the network.

This metric encompasses the duration from when the transaction is submitted to when its outcome becomes widely accessible within the network. It encompasses propagation time and any settling time influenced by the consensus mechanism. Equation 6.3 shows how the average latency is computed by dividing the total latency for successful transactions by the total number of successful transactions. Equation 6.4 explains that L_{succ} is found by first computing the latency for each individual transaction and then adding together the latencies of all these transactions.

- **Resource Consumption:** The consumed CPU, used memory, and network IO.

$$Throughput = \frac{T_{\text{succ}}}{(Time_{\text{last\_committing}} - Time_{\text{first\_submitting}})} \tag{6.1}$$

$$Send\_rate = \frac{(T_{\text{succ}} + T_{\text{fail}})}{(Time_{\text{last\_submitting}} - Time_{\text{first\_submitting}})} \tag{6.2}$$

$$Average\_Latency = \frac{L_{\text{succ}}}{T_{\text{succ}}} \tag{6.3}$$

$$L_{\text{succ}} = \sum_{i=1}^{n} Tx_{\text{i}_{\text{commiting\_time}}} - Tx_{\text{i}_{\text{submitting\_time}}} \tag{6.4}$$

6.3 Hyperledger Fabric Configuration Setup

Preparing a testing environment using Hyperledger Caliper involves meticulous planning and configuration to ensure accurate and reliable performance assessments of Blockchain applications. The process begins with defining the specific performance metrics to be evaluated, such as transaction throughput, latency, resource utilisation, and network scalability. These metrics guide the configuration of the test network, including the selection of appropriate hardware, network topology, and the number of peers and orderers within the Hyperledger Fabric network. Furthermore, careful attention must be paid to the deployment and configuration of smart contracts (chaincode), as their design and execution logic directly impact performance outcomes. Hyperledger Caliper requires precise configuration files that define workload scenarios, includ-

ing the rate of transaction submissions, the size of data payloads, and the duration of the testing period, and this will be detailed more in Section 6.4. This comprehensive setup process ensures providing meaningful insight into the system performance under various conditions and facilitating the identification of potential bottlenecks or optimisation opportunities.

Performance benchmarks for HLF encompass various aspects of network setup as detailed in (Hyperledger Fabric, 2024), including hardware considerations, peer configurations, orderer setups, and overall application design. In Table 6.1, condensed overview of these considerations and the testing environment that was used to generate the reported tests, which are provided in the following Section 6.4.

Table 6.1: Testing environment considerations and description.

Platform	**Optimal Characteristics**	**Testing Environment Description**
1. Hardware Configuration:	• Utilise the fastest available disk storage due to high disc I/O demands. • Allocate ample CPU and memory resources for peer and ordering service nodes, crucial for stability and network speed.	Testing local machine has: **Memory:** 15.5.GiB. **Processor:** Intel(R) Core(TM) i7-8650 CPU @ 1.90GHz, 8 GB running Ubuntu 18.04 LTS. **OS type:** 64-bit. **Disk:** 202.9 GB.

Table 6.1: Testing environment considerations and description (continued).

Platform	**Optimal Characteristics**	**Testing Environment Description**
2. Peer Considerations:	• Optimise the number of peers per organisation for better load balance. • Manage the number of channels per peer to maintain CPU usage below 65-70%.	Number of organisations: 2. Number of total peers: 2. Number of channels per peer: 1.
3. Orderer Considerations:	• The number of orderers impacts network performance due to Raft consensus; multiple nodes enhance fault tolerance. • Adjust SendBufferSize to prevent message loss, with default settings improved in Fabric v2.5.	Number of Orderer: 1. Consensus: Raft. SendBufferSize: 10.

Table 6.1: Testing environment considerations and description (continued).

Platform	**Optimal Characteristics**	**Testing Environment Description**
4. Channel Configuration:	• Tailor batch size, timeout, and message count to balance throughput and latency. • Configure block cutting parameters for efficient transaction processing.	Number of channels:1. Batch size: 2MB. Batch time out: 2s. Message count: 500.
5. Application Design:	• Avoid CouchDB for high-throughput applications; consider range queries or off-chain solutions. • Manage payload size and choose appropriate chaincode language for optimal performance. • Design endorsement policies and channel architecture to enhance speed and resource utilisation.	World state database: CouchDB. Ledger database: LevelDB. Endorsement Policy: OR. Chaincode Language: Golang.

Table 6.1: Testing environment considerations and description (continued).

Platform	**Optimal Characteristics**	**Testing Environment Description**
6. CouchDB Considerations	• Monitor CouchDB resource usage and optimise cache settings for better performance. • Use indexes for efficient query execution and optimise query complexity to reduce latency.	Using Indexes: True. Query format: JSON.

In the proposed testing environment, the majority of the parameters described in Table 6.1, including batch timeout and block size within the orderer component of the Hyperledger Fabric (HLF) framework, will be predefined and configured prior to initiating system tests. Peers and orderer nodes will be hosted within Docker containers, with all nodes interconnected through a dedicated channel. The chaincode, which specifies the operations subject to testing, will be discussed in more detail in the following section.

The primary objective of the testing process in this pilot study will be to evaluate the system's throughput and average latency, as previously stated, to benchmark the performance of SUT. To facilitate this evaluation, a simplified test network, as illustrated in Figure 6.1, will be utilised to deploy the smart contracts under consideration. The HLF network will define two organisations, each comprising peer, channel, and orderer nodes. Each peer node will employ LevelDB to store a replica of the ledger. Chaincode, written in Golang, will be used to manage access to these peer organisations. Within the HLF network, the ordering service is responsible for generating blocks of transactions and distributing them to the committers, ensuring the integrity and consistency of the blockchain. The following processes can be used to describe the system's

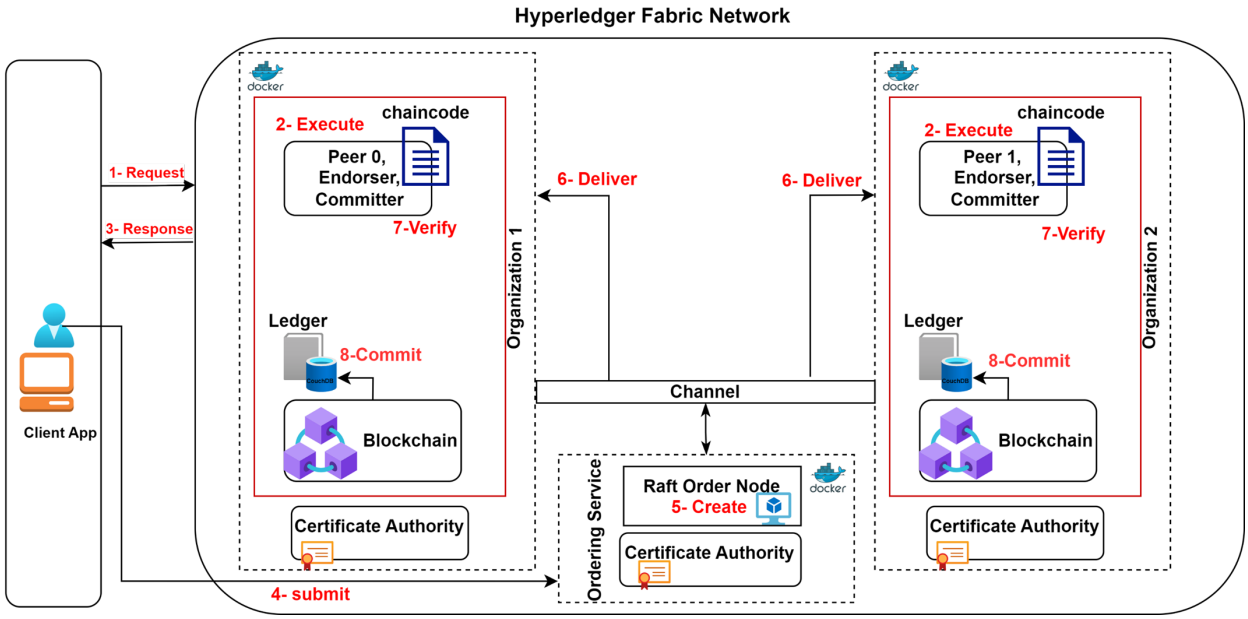

Figure 6.1: Hyperledger Fabric network architecture used for testing.

transaction flow, which is detailed before in 3.5.2 and Figure 3.6:

1. Users request to use a blockchain-based system.
2. The endorser runs the chaincode to grant authorisation.
3. Peers respond to users' request.
4. The ordering service receives the transaction from the users.
5. The transaction is created as a block by the orderer node.
6. Peers receive blocks from the orderer node.
7. The committer verifies every block transaction.
8. Every block transaction is committed to a ledger by the committer.

Sections	Test 1	Test 2	Test 3	Test 4	Test 5
Function Under Test	ScheduleJourney	~	~	~	~
Total Workers	5	~	~	~	~
Control Rate	TPS:15	TPS:20	TPS: 25	TPS:30	TPS:35
Execution timeout	300 sec	~	~	~	~
ByteSize	100	1000	5000	10000	15000

Table 6.2: Schedule journey testing cases.

6.4 Performance Test Cases

In this section, eight performance evaluations were conducted utilising Hyperledger Caliper, and the full source code has been made accessible via a GitHub repository[2]. The primary emphasis will be placed on evaluating the most resource-intensive functionalities within the developed framework, particularly those that directly handle requests from external clients during the performance testing with Hyperledger Caliper.

The strategy used involves starting with low test case values and gradually increasing them to assess potential bottlenecks. We will vary the byte size, total workers and TPS for each test case to gauge their impact on the report values. In certain scenarios, we will maintain the same number of workers to focus on incrementally increasing TPS while keeping the control rate constant.

6.4.1 Schedule Journey

Schedule Journey using the smart contract *"ScheduleJourney"* is the first step in our scope, and all other steps are related to this, so it is important to check how it performs under certain circumstances. Only administrators can create the journey, so setting workers to five in all test cases will be enough to check performance. In five test cases, we gradually increase TPS from 15 to 35 TPS to check how many transactions it can handle in the 300 second execution timeout. See Table 6.2.

The results presented in Table 6.3 show the following:

[2]https://github.com/Rahma-Alzahrani/HyperledgerCaliper

- **Latency:** All five test cases show that latency is calculated in seconds. Observations reveal a consistent average latency across all cases, suggesting robust functionality. Each test case illustrates a notable discrepancy between the minimum and maximum latency values, attributable to the rapid commit rate on the ledger network. Consequently, numerous transactions experience delays in being included in the current block, necessitating their deferment to subsequent blocks.

- **Success/Fail rates:** In all five test cases, there is no failure on the transactions, which means the functionality is running smoothly without any issues. As shown in Figure 6.2, the chart reveals that the success rate remains consistently high under normal load conditions.

- **Transactions Per Second:** All five test cases show that the Caliper successfully ran TPS near the value provided in the configuration. The steady rise in throughput corresponding to latency is depicted in Figure 6.2.

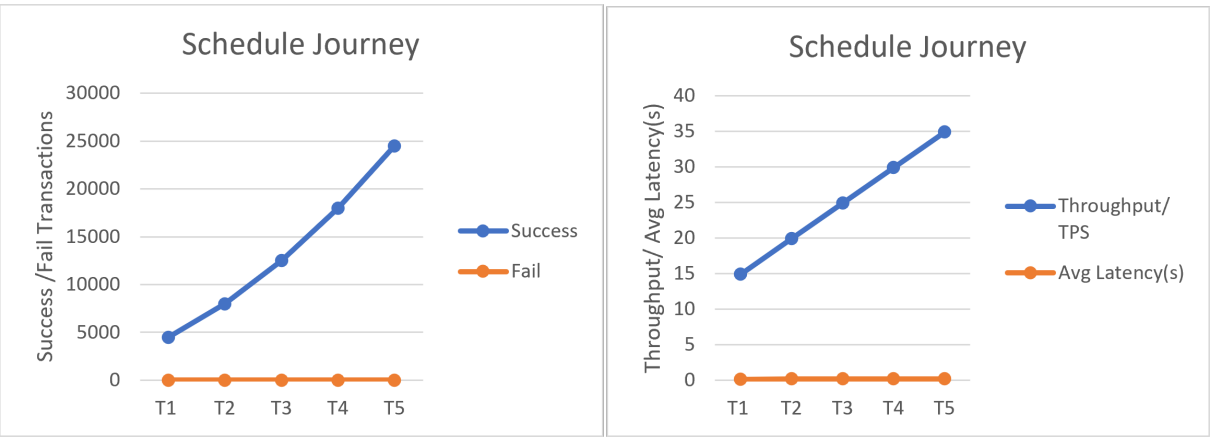

Figure 6.2: Success and fail transactions rates (left side). Throughput and average latency (right side) of *"ScheduleJourney"* smart contract.

According to the passenger rail usage statistics provided by ORR, a total of 1.61 billion rail journeys were completed in the United Kingdom between April 2023 and March 2024 (Office of Rail and Road, 2024). This equates to an average of approximately 4.4 million journeys per day, underscoring the significant reliance on rail transport within the UK. Such a high volume of daily

	T1	T2	T3	T4	T5
Success	4505	8005	12505	18005	24505
Fail	0	0	0	0	0
Max Latency(s)	2.06	2.08	2.03	2.12	2.14
Min Latency(s)	0.06	0.07	0.07	0.07	0.07
Avg Latency(s)	0.17	0.18	0.18	0.18	0.18
TPS (Transactions Per Second)	14.9	19.9	24.9	29.9	34.9

Table 6.3: Schedule journey testing results.

Sections	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8
Function Under Test	InsertDataOffer	~	~	~	~	~	~	~
Total Workers	10	~	~	~	~	~	~	~
Control Rate	startingtps :1	~	~	~	~	~	~	~
Transaction load	15	20	30	40	50	70	90	100
Execution timeout	300 sec	~	~	~	~	~	~	~
Bytesize	1000	2000	3000	4000	8000	16000	32000	64000

Table 6.4: Insert real data offer testing cases.

journeys exemplifies the crucial role that rail systems play in the country's overall transportation infrastructure.

In a parallel context, if we consider a TPS rate of 35, as demonstrated in Table 6.3, the SUT has the capability to process $35 \times 86400 = 3,024,000$ transactions per day. This calculation emphasises the system's operational efficiency in managing substantial transaction loads over a 24-hour period.

It is important to highlight that the testing infrastructure used in this evaluation represents a minimal Blockchain network configured on a local machine with constrained computational resources. Despite these limitations, the performance results are particularly encouraging, as they suggest that even a simplified setup is capable of achieving a high throughput. This finding is significant as it implies that the SUT could scale effectively in a more robust, resource-rich environment, providing a strong foundation for future enhancements in transaction processing capabilities.

	T1	T2	T3	T4	T5	T6	T7	T8
Success	460	2043	2217	2170	2342	2348	2387	2422
Fail	0	0	0	0	0	0	0	0
Max Latency(s)	0.58	2.06	2.07	0.54	2.07	2.19	2.08	2.08
Min Latency(s)	0.06	0.07	0.07	0.09	0.11	0.13	0.10	0.08
Avg Latency(s)	0.26	0.17	0.23	0.31	0.44	0.57	0.70	0.77
TPS (Transactions Per Second)	15.1	67.1	73.2	76.2	77.0	77.2	78.1	77.1

Table 6.5: Insert real data offer testing results.

6.4.2 Insert Real Data Offers

Following the creation of journeys by the administrator, the provider can now generate real data offers associated with those journeys, accessible for consumer requests. The tests were carried out on the *"InsertDataOffer"* smart contract in eight distinct scenarios to evaluate its performance under varying conditions, as presented in Table 6.4. Employing 10 workers in total and initiating at a starting TPS of 1, each test case involves a progressive increase in both transaction load and byte size. This comprehensive testing aims to gauge performance across a spectrum of numbers and scenarios.

The results presented in Table 6.5 show the following:

- **Latency:** All eight test cases show the measured latency in seconds. We observe that the average latency is less than one second in each testing case, suggesting robust functionality. We can see from all test cases that there is a clear fluctuation between minimum and maximum latency. We can attribute this to the high rate of commits on the ledger across the network. This leads many transactions to miss the chance of being included in the current block, and thus wait for the next one.

- **Success/Fail rates:** In all eight test cases there is no failure on the transactions, which means the functionality is running smoothly without any issues. There was a significant discrepancy in the success rate between the initial test and the subsequent tests, as illustrated in Figure 6.3.

- **Transactions Per Second:** In the initial test scenario, a rate limit of 15 TPS was imposed, yielding a slightly higher actual rate of 15.1 TPS, meeting the specified criteria. Subse-

quent tests, with an initial TPS set at 1 and varying transaction loads, were successfully executed without any transaction failures, showing a greater variance in results compared to the initial test, as illustrated in Figure 6.3. Across all test cases, the observed TPS consistently ranged between 65 and 78, aligning with the anticipated outcomes.

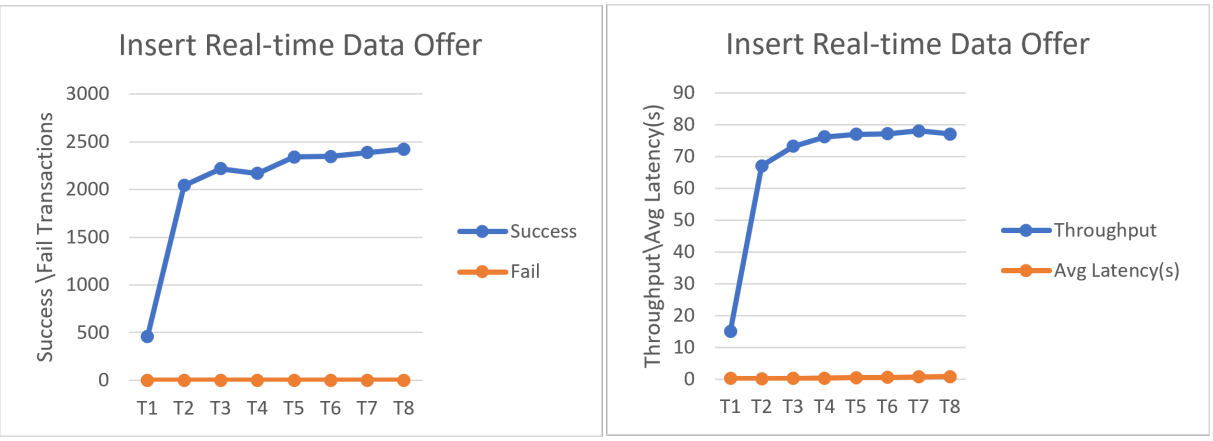

Figure 6.3: Success and fail transactions rates (left side). Throughput and average latency (right side) of *"InsertDataOffer"* smart contract.

6.4.3 Insert Real Data Hash

This functionality is crucial for testing to determine potential bottlenecks and performance in various scenarios as presented in Table 6.6 to evaluate the smart contract *"InsertDataHash"*. The testing environment is configured with 10 workers, starting with a TPS of 1 transaction load and increasing the byte size in each test case to assess its performance. Each test case is given a 30-second execution timeout to determine whether this duration is adequate for assessing its performance under the specified conditions.

The results presented in Table 6.7 show the following:

- **Latency:** All eight test scenarios indicate latency measurements in seconds. Notably, there is a consistent average latency across all cases, underscoring the stability of the functionality under examination. A discernible fluctuation between minimum and maximum latency values is evident across all test instances. This variance can be ascribed to the elevated commit rate observed on the ledger network. Consequently, numerous

Sections	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8
Function Under Test	InsertDataHash	~	~	~	~	~	~	~
Total Workers	10	~	~	~	~	~	~	~
Control Rate	startingtps :1	~	~	~	~	~	~	~
Transaction load	10	20	30	40	50	70	90	100
Execution timeout	30 sec	~	~	~	~	~	~	~
Bytesize	1000	2000	3000	4000	8000	16000	32000	64000

Table 6.6: Insert real data hash testing cases.

	T1	T2	T3	T4	T5	T6	T7	T8
Success	910	2604	2690	2842	2840	2946	3019	3009
Fail	0	0	0	0	0	0	0	0
Max Latency(s)	0.59	2.09	0.88	2.09	1.05	2.10	2.11	2.43
Min Latency(s)	0.07	0.08	0.08	0.09	0.09	0.12	0.08	0.10
Avg Latency(s)	0.20	0.29	0.39	0.49	0.59	0.65	0.81	0.87
TPS (Transactions Per Second)	15.1	43.1	45.9	47.0	48.3	48.6	49.9	49.8

Table 6.7: Insert real data hash testing results.

transactions encounter delays in their inclusion within the current block, necessitating their deferral to subsequent blocks.

- **Success/Fail rates:** In all eight test cases, there is no failure on the transactions, which means the functionality is running smoothly without any issues.

- **Transactions Per Second:** In the initial test case, a rate limit of 15 TPS was implemented, yielding an actual TPS slightly exceeding the target at 15.1, thus meeting the criteria. Subsequent tests initiated with a starting TPS of 1, varying transaction loads, and encountered no transaction failures. Across all test cases, the observed TPS consistently ranged between 40 and 50, aligning with the expected outcomes only in some cases. Figure 6.4 illustrates the significant increase in both success rate and throughput when the transaction load exceeds 10.

To contextualise the results in relation to industry applications, we refer to Section 4.3 in Chapter 4, which details two use cases. The use case, "Railway Track Monitoring Using Onboard Inertial Measurements," involves the utilisation of 5 sensors, while the other use case, "Switch and Point Machine Monitoring System," employs 17 sensors.

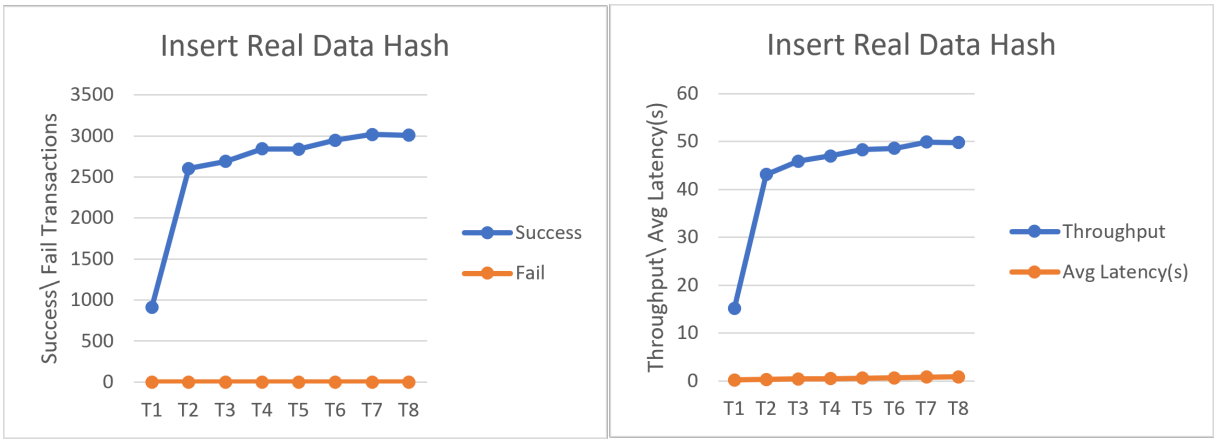

Figure 6.4: Success and fail transactions rates (left side). Throughput and average latency (right side) of *"InsertDataHash"* smart contract.

When assessing the performance of a Blockchain network for the real-time storage of sensor data hashes across these two scenarios, the reported TPS rate of 40 to 50 is appropriately aligned with the specific demands of these applications.

- **Use Case 1: Railway Track Monitoring Using Onboard Inertial Measurements**

  This use case involves 5 sensors; each sensor typically generates a constant stream of data that is periodically hashed and stored in the Blockchain. The total transaction load from 5 sensors would be relatively low, meaning that a TPS of 40 to 50 would be more than sufficient to handle the data in real-time. Assuming each sensor transmits data once per second (or at even slower intervals depending on the scheduled journey), the Blockchain would need to process only 5 transactions per second, far below the network's reported capacity. This makes the TPS rate of 40-50 optimal for this scenario, providing significant headroom for scalability or future increases in sensor activity.

- **Use Case 2: Switch and Point Machine Monitoring System**

  In the second case, with 17 sensors, the demand for transaction processing is naturally higher, but even here, the system's capacity remains adequate. If each of the 17 sensors sends data at intervals of one transaction per second, the Blockchain would need to process 17 transactions per second. Given the network's reported TPS range of 40 to 50, this scenario would utilise approximately 34% to 42.5% of the system's maximum

capacity. This suggests that the Blockchain is more than capable of managing the data from these 17 sensors without encountering performance bottlenecks.

In real-world industrial settings, sensor networks can vary significantly in scale. For smaller setups such as those with 5 or 17 sensors, blockchain-based storage systems with a TPS rate of 40 to 50 provide robust performance, particularly in ensuring data integrity and security through decentralised mechanisms. Industrial systems with similar or even larger sensor networks often rely on centralised databases, where TPS rates can be much higher, but these systems may lack the immutability and security features offered by Blockchain. Thus, for small- to medium-scale applications such as the two cases previously investigated, where data integrity is critical but the transaction load is moderate, the achieved TPS rate is ideal. It offers sufficient processing power without overwhelming system resources, while still maintaining the decentralised advantages of Blockchain technology.

6.4.4 Insert Historical Data Offer

The *"InsertHistoricalDataOffer"* smart contract holds significant importance in creating offers on existing data. Testing this function under extreme conditions is crucial to understanding its performance within specific parameters. Optimal results were achieved by configuring five workers with a consistent starting TPS rate of 1 for each test case, ensuring optimal performance and reliable outcomes. Throughout the testing process, a constant transaction load of 10 was maintained, alongside an execution timeout set at 300 seconds. Additionally, varying byte sizes were applied in each test case to assess performance across different data loads. See Table 6.8. The test results presented in Table 6.9 show the following:

- **Latency:** In all eight test cases, latency measurements are consistently recorded in seconds. Notably, there's minimal deviation in average latency across these cases, highlighting the stability of the functionality under scrutiny. Analysis of all test instances reveals a discernible fluctuation between minimum and maximum latency values. This variation can be linked to the rapid rate of commits observed across the ledger network. Conse-

Sections	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8
Function Under Test	InsertHistoricalDataOffer	~	~	~	~	~	~	~
Total Workers	5	~	~	~	~	~	~	~
Control Rate	startingtps :1	~	~	~	~	~	~	~
Fixed Rate	transactionload: 10	~	~	~	~	~	~	~
Execution timeout	300 sec	~	~	~	~	~	~	~
Bytesize	1000	2000	3000	4000	8000	16000	32000	64000

Table 6.8: Insert historical data offer testing cases.

	T1	T2	T3	T4	T5	T6	T7	T8
Success	16360	15610	15310	16062	15681	15548	15653	16090
Fail	0	0	0	0	0	0	0	0
Max Latency(s)	1.21	1.18	1.17	2.07	2.07	2.09	2.13	1.19
Min Latency(s)	0.06	0.07	0.07	0.07	0.07	0.07	0.07	0.07
Avg Latency(s)	0.17	0.18	0.18	0.18	0.18	0.18	0.18	0.18
TPS (Transactions Per Second)	54.6	52.2	50.6	53.3	52.0	51.6	52.0	53.8

Table 6.9: Insert historical data offer testing results.

quently, numerous transactions are unable to be included in the current block, necessitating their deferment to subsequent blocks.

- **Success/Fail rates:** In all eight test cases, there is no failure on the transactions, which means the functionality is running smoothly without any issues.
- **Transactions Per Second:** In all eight test cases, the Caliper effectively sustained a TPS of approximately 50 or slightly more, indicating satisfactory performance. Figure 6.5 illustrates that the throughput and success rate remain consistent around similar values.

Figure 6.5: Success and fail transactions rates (left side). Throughput and average latency (right side) of *"InsertHistoricalDataOffer"* smart contract.

6.4.5 Insert Historical Data Hash

The purpose of testing the smart contract *"InsertHistoricalDataHash"* is to examine potential bottlenecks in various scenarios. To perform this analysis, the configuration is established with

five workers, initiating at 1 TPS, and a transaction load of 10. Each test iteration involves an incremental increase in byte size and transaction load. The evaluation allows for a comprehensive assessment under varying loads, accompanied by an execution timeout set at 60 seconds. See Table 6.10.

The test results presented in Table 6.11 show the following:

- **Latency:** All eight test scenarios reveal latency measurements recorded in seconds. Notably, there is a discernible rise in average latency as the transaction load increases. This is attributed to the complexity of the function within the developed chaincode. The intricate nature of this function, particularly the necessary checks and time required for querying other functions like retrieving hash values linked to data stored in external databases, anticipates it to operate at a lower efficiency, yielding a lower TPS output. Consequently, meeting the required parameters consumes more time, resulting in fewer transactions being accommodated in each generated block.

- **Success/Fail rates:** The initial seven test cases, as depicted in Table 6.11 and Figure 6.6, demonstrate flawless transaction execution, indicating smooth functionality operation without encountering any issues, provided that the transaction load should remain below 120 transactions per second. The final test scenario indicates a substantial number of failed transactions, totaling 209, when the transaction load is set to 120.

- **Transactions Per Second:** As depicted in Table 6.11 and Figure 6.6, the TPS decreases as the number of sent requests increases.

6.4.6 Get All Journeys

This test aims to retrieve information from HLF about all journeys recorded in the ledger using the smart contract *"GetAllJourney"*. It will consist of eight test cases to evaluate the functionality, utilising five workers and starting with a TPS of 1. A fixed transaction load rate of 50 and an execution timeout of 300 seconds will be adequate to assess the benchmark performance by progressively increasing the Bytesize in each test case. See Table 6.12.

Sections	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8
Function Under Test	InsertHistoricalDataHash	~	~	~	~	~	~	~
Total Workers	10	~	~	~	~	~	~	~
Control Rate	startingtps :1	~	~	~	~	~	~	~
Transaction Load	10	15	20	40	60	80	100	120
Execution timeout	60 sec	~	~	~	~	~	~	~
Bytesize	100	100	1000	4000	8000	16000	32000	64000

Table 6.10: Insert historical data hash testing cases.

	T1	T2	T3	T4	T5	T6	T7	T8
Success	910	458	328	248	340	250	200	10
Fail	0	0	0	0	0	0	0	209
Max Latency(s)	17.76	3.41	7.23	13.10	20.82	32.95	62.92	2.01
Min Latency(s)	0.17	0.41	0.41	0.58	1.41	1.27	0.90	1.90
Avg Latency(s)	3.47	1.80	3.90	8.49	14.91	23.23	48.48	1.95
TPS (Transactions Per Second)	13.3	7.2	5.1	3.9	4.6	3.6	2.7	1.6

Table 6.11: Insert historical data hash testing results.

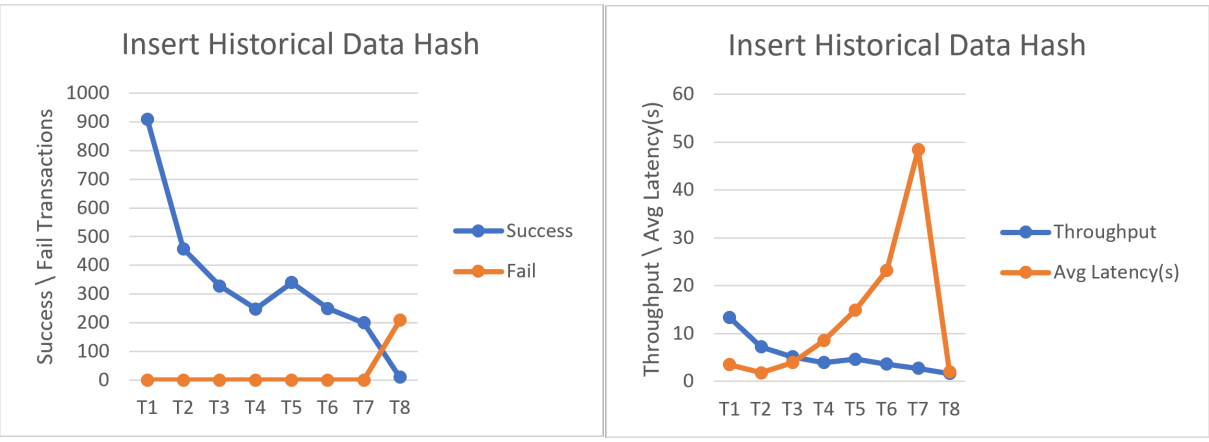

Figure 6.6: Success and fail transactions rates (left side). Throughput and average latency (right side) of *"InsertHistoricalDataHash"* smart contract.

The outcomes detailed in Table 6.13 and Figure 6.7 reveal the following aspects:

- **Latency:** Each of the eight test scenarios records latency values in seconds. Remarkably, the average latency remains under 0.25.
- **Success/Fail rates:** All transactions querying data on HLF were successful, with no failures recorded.
- **Transactions Per Second:** The TPS ranged from 138.5 to 174.5, and given the consistent load of 50 transactions for each test case, this result is satisfactory.

6.4.7 Get All Historical Offers

When historical offers are published, it is crucial to evaluate how HLF queries this data on the platform. Using eight test cases with 10 workers and a transaction load of 50, each assigned a

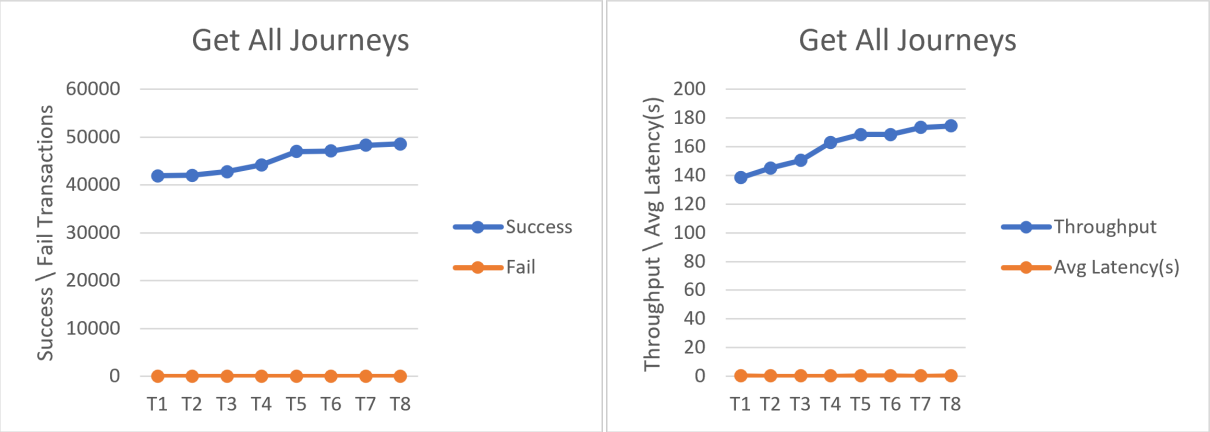

Figure 6.7: Success and fail transactions rates (left side). Throughput and average latency (right side) of *"GetAllJourney"* smart contract.

Sections	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8
Function Under Test	GetAllJourney	~	~	~	~	~	~	~
Total Workers	5	~	~	~	~	~	~	~
Control Rate	startingtps :1	~	~	~	~	~	~	~
Fixed Rate	transactionload: 50	~	~	~	~	~	~	~
Execution timeout	300 sec	~	~	~	~	~	~	~
Bytesize	100	1000	2000	4000	8000	16000	32000	64000

Table 6.12: Get all journeys testing cases.

different byte size, allows us to assess its performance under varying conditions for the smart contract *"GetAllHistoricalOffer"*. See Table 6.14.

Table6.15 and Figure 6.8 illustrate the following outcomes:

- **Latency:** Each of the eight test scenarios records latency values in seconds with an average latency of 0.16 and minimal spikes in maximum latency.

- **Success/Fail rates:** All transactions querying data on HLF were successful, with no

	T1	T2	T3	T4	T5	T6	T7	T8
Success	41928	42030	42803	44222	47003	47113	48330	48566
Fail	0	0	0	0	0	0	0	0
Max Latency(s)	0.75	0.65	0.77	0.77	0.77	0.77	0.77	0.76
Min Latency(s)	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Avg Latency(s)	0.24	0.23	0.23	0.23	0.24	0.24	0.23	0.24
TPS (Transactions Per Second)	138.5	145.2	150.5	163.1	168.5	168.6	173.4	174.5

Table 6.13: Get all journeys testing results.

Sections	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8
Function Under Test	GetAllHistorcalOffer	~	~	~	~	~	~	~
Total Workers	10	~	~	~	~	~	~	~
Fixed Rate	transactionload: 50	~	~	~	~	~	~	~
Execution timeout	60 sec	~	~	~	~	~	~	~
Bytesize	100	1000	2000	4000	8000	16000	32000	64000

Table 6.14: Get all historical offers testing cases.

	T1	T2	T3	T4	T5	T6	T7	T8
Success	12910	12839	13212	13290	13313	13764	13570	13565
Fail	0	0	0	0	0	0	0	0
Max Latency(s)	0.34	0.35	0.33	0.35	0.34	0.36	0.33	0.34
Min Latency(s)	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Avg Latency(s)	0.16	0.16	0.15	0.15	0.15	0.15	0.15	0.15
TPS (Transactions Per Second)	221.8	220.1	226.7	228.1	228.5	235.6	232.6	232.0

Table 6.15: Get all historical offers testing results.

Figure 6.8: Success and fail transactions rates (left side). Throughput and average latency (right side) of *"GetAllHistoricalOffer"* smart contract.

failures recorded.

- **Transactions Per Second:** The TPS in all tests consistently falls within the range of 220 to 232, indicating robust performance.

6.4.8 Sensor Query

The objective is to query all available data hashes on Blockchain (HLF) to evaluate its performance under high volumes of simultaneous queries using the smart contract *"GetAllDataHashes"*. To accomplish this, a configuration of ten workers is employed with a fixed-rate transaction load of 50 in eight test cases. The ninth test case is specifically designed with a fixed transaction load of 350 to provide further insight into performance. Each test case includes incrementally increasing byte sizes up to 64,000, with an execution timeout of 60 seconds applied to ensure a comprehensive evaluation. See Table 6.16.

Sections	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 9
Function Under Test	GetAllDataHashes	~	~	~	~	~	~	~	~
Total Workers	10	~	~	~	~	~	~	~	~
Fixed Rate	transactionload: 50	~	~	~	~	~	~	~	TPS:350
Execution timeout	60 sec	~	~	~	~	~	~	~	~
Bytesize	100	1000	2000	4000	8000	16000	32000	64000	8000

Table 6.16: Sensor query testing cases.

Table 6.17 and Figure 6.9 illustrate the following:

- **Latency:** Each of the first eight test scenarios records latency values in seconds with an average latency between 0.15 and 0.17 and minimal spikes in maximum latency. In the ninth test case, the average latency rose as the transaction load was adjusted to 350, resulting in longer processing times.

- **Success/Fail rates:** All transactions querying data on HLF were successful, with no failures recorded in the first eight test cases. In the final test scenario, failure was induced intentionally to identify bottlenecks. A transaction load of 350 was applied using ten workers, causing the Caliper to attempt 3,500 transactions per second. This surpassed the specified concurrency limit, leading to the failure of some transactions in their queries. This prioritised the preservation of data integrity within the module.

- **Transactions Per Second:** The sensor query yielded favorable results through the first eight test cases, consistently maintaining a TPS range of 200-220. The throughput in the final test case climbed to 289.1 despite a concurrent rise in average latency.

Concurrency Limit: During data querying from HLF with transactions exceeding 2500 per second, we encountered a concurrency limit error from Hyperledger. This error occurred when attempting to query more than 2500 transactions, causing subsequent transactions to fail. In Fabric v2.1.0 and above, the number of concurrent requests to peer services is capped by default to mitigate the risk of poorly programmed or malicious clients causing Denial of Service (DoS) attacks on peers (T. B. d. S. Costa et al., 2022). This restriction can be removed, or increased by modifying these values in the "core.yaml" file [3]. To remove the restriction, set the limit to 0, or increase it to any value that makes sense for the environment.

[3]https://github.com/hyperledger/fabric/blob/2fdbafbed9fc1213a28e5980b8c334ce04c27dcb/sampleconfig/core.yaml#L438-L448

	T1	T2	T3	T4	T5	T6	T7	T8	T9
Success	12028	12172	12509	12826	12764	12673	12584	12658	4721
Fail	0	0	0	0	0	0	0	0	16284
Max Latency(s)	0.47	0.42	0.37	0.41	0.39	0.40	0.39	0.37	59.95
Min Latency(s)	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02
Avg Latency(s)	0.17	0.16	0.16	0.16	0.15	0.16	0.16	0.16	24.00
TPS (Transactions Per Second)	205.7	208.3	214.2	219.8	218.6	217.2	216.0	216.9	289.1

Table 6.17: Sensor query testing results.

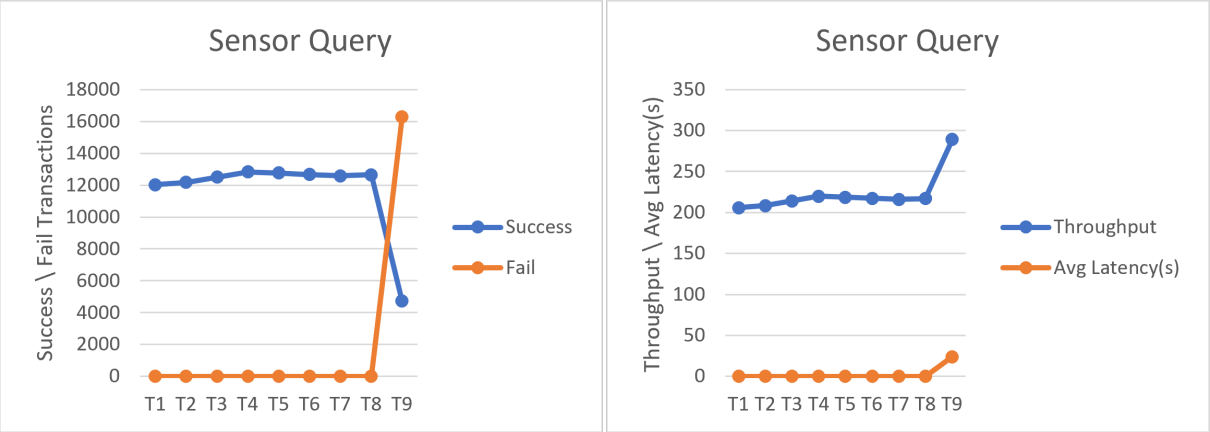

Figure 6.9: Success and fail transactions rates (left side). Throughput and average latency (right side) of *"GetAllDataHashes"* smart contract.

6.5 Discussion

In this chapter, a comprehensive performance assessment of the developed Blockchain application was conducted using Hyperledger Caliper. The primary objective of this evaluation was to analyse the functional characteristics of the system under various operating conditions. The test results revealed several critical insights into the system's performance dynamics.

First, the analysis demonstrated a marked difference between read and write operations, with read operations consistently outperforming write operations in terms of speed. This disparity can be attributed to the inherent nature of Blockchain transactions, where writing typically involves more complex operations, such as consensus mechanisms and data propagation across the network, which introduces additional latency. Furthermore, throughput, defined as the rate at which transactions are processed successfully, was observed to increase significantly with a higher transaction arrival rate, especially when the number of concurrent users (workers) was minimised. This suggests that the system is more efficient when handling larger batches of transactions at a time with fewer parallel processing demands, which could otherwise introduce competition for resources and potential bottlenecks. In addition, the test results indicated that the byte size of transactions plays a crucial role in performance optimisation. Larger transaction sizes were correlated with improved system performance, probably due to more efficient use of block space and reduced overhead per transaction.

However, it was also observed that the number of concurrent users and the total number of transactions within each block significantly influenced the transaction delay. Specifically, higher levels of concurrency and larger transaction counts per block were associated with increased delays, which can be attributed to the additional processing and validation time required to handle simultaneous transactions and larger data loads.

To optimise performance and reduce the appearance of timeouts and transaction delays, it was found that maintaining transaction arrival rates within the range of 10 to 100 transactions per block was the most effective. This range appears to strike a balance between maximising throughput and minimising latency, ensuring that the system operates within its optimal parameters.

These findings underscore the importance of carefully tuning transaction arrival rates, concurrency levels, and transaction sizes to achieve the desired performance results in Blockchain deployments. Future studies may build on these insights by exploring the interplay of these variables under different network conditions and with varying consensus algorithms.

The primary functionality of our application centers on the management of offers and the handling of data hashes. Given the critical role these components play within the system, it is essential to rigorously assess their performance. The efficient operation of these functionalities is not only fundamental to the overall effectiveness of the application but also directly influences user experience, particularly for consumers engaged in the exchange of data for offers.

A key aspect of our performance evaluation strategy involves examining the impact of querying these data sets on system performance. Specifically, the focus is on understanding how the retrieval of offers and data hashes affects the consumer experience. Consumers, who rely on timely and accurate data in exchange for offers, represent a crucial user base, and their satisfaction is contingent upon the system's ability to handle such queries efficiently.

Through targeted performance testing using Hyperledger Caliper, our aim is to quantify the effect of these operations on transaction processing times, system throughput, and overall responsiveness. By doing so, we can identify potential bottlenecks and optimise the system to ensure that the querying process does not hinder the flow of data or diminish the perceived value

of the offers from the consumer's perspective.

6.6 Limitations and Future Work

The results discussed in Section 6.4 represent the highest performance outcomes observed during a carefully controlled series of test trials using Hyperledger Caliper. Each test case was executed with specific configurations, which were fine-tuned to achieve the most favorable performance metrics under the given conditions. However, it is important to note that these results should be interpreted with caution when considering reproducibility. The variability of available system resources on the device used for testing plays a significant role in the performance of the Blockchain network. Factors such as CPU load, memory availability, disk I/O, and network bandwidth can fluctuate due to concurrent processes or environmental changes, leading to potential inconsistencies in resource allocation during repeated test runs. As a result, even when identical test settings are applied in subsequent trials, the performance outcomes may differ.

This phenomenon underscores a critical aspect of performance benchmarking in distributed systems: the influence of resource variability on test results. Although the data presented reflect good performance under controlled conditions, it is not indicative of the performance that can be observed in every trial. Therefore, while the results provide valuable insight into the potential capabilities of the Blockchain network under ideal conditions, they should not be viewed as absolute benchmarks.

Future studies and practical applications must consider the inherent variability in system resources and its impact on performance. To mitigate this, researchers and practitioners should ensure a controlled and consistent test environment or conduct a larger number of trials to average out the performance variations. This approach would lead to more robust and generalisable conclusions regarding the performance characteristics of Blockchain networks.

To sum up, while the results presented in Section 6.4 offer a snapshot of the optimal performance of the Blockchain network, they do not fully capture the potential range of performance

variations that may occur due to fluctuating system resources. This highlights the need for careful consideration of resource management and consistency in performance testing environments.

A further limitation to be noted in this benchmarking is the smart contracts that were tested. The Hyperledger Caliper benchmarking tool, while robust in its ability to measure and report on the performance of individual chaincode functions, encounters significant limitations when dealing with functions that are interdependent or operate within complex workflows. Specifically, Caliper is designed to generate performance reports primarily for chaincode operations that function independently, without the need for interactions between multiple functions. This inherent design constraint poses challenges for comprehensive performance testing, particularly in scenarios where the application's functionality relies on a sequence of interconnected operations.

Our testing approach, which focuses on evaluating performance in a largely unidirectional manner, highlights these limitations. The linear nature of such tests makes it difficult to accurately assess the performance of functions that depend on the outcomes of prior operations or that must be executed in conjunction with other functions. For instance, in scenarios where functions like creating offer requests and subsequently accepting them are tested, the dependence on unique request IDs introduces a layer of complexity. Accurately tracking and rerunning transactions across these interlinked functions within Caliper becomes a highly intricate task, often bordering on impracticality. To illustrate, the process of creating an offer request and then accepting it is not merely a sequence of isolated operations but a workflow that requires the retention and correct usage of request IDs generated during the initial step. Caliper's current capabilities do not easily support the continuous tracking and integration of such interdependent transactions into cohesive performance reports. As a result, these interlinked functions can typically be invoked only once within the testing framework, complicating efforts to merge and analyse their performance in a unified report. However, despite these challenges, we have identified certain functionalities that can be tested more effectively within the constraints of Caliper. For example, within a single journey, it is possible to create multiple offers and attach data hashes to those offers. These operations, being less dependent on sequentially generated identifiers or prior

states, can be repeated across multiple iterations, allowing us to derive meaningful performance metrics. This aspect of our testing enables the generation of more reliable performance data, albeit within a limited scope.

In summary, while Hyperledger Caliper provides valuable insight into the performance of discrete Blockchain operations, its limitations in handling interconnected functions necessitate a cautious interpretation of results. Future work may involve developing or integrating additional tools or methodologies that can better accommodate the complex and interdependent nature of Blockchain applications, allowing more comprehensive and accurate performance assessments.

6.7 Conclusion

In this chapter, we present a comprehensive pilot study aimed at evaluating the performance of some of the smart contracts introduced in Chapter 5. The chapter began by introducing Hyperledger Caliper and highlighted its relevance in assessing the performance of Blockchain networks. The study focused on critical performance metrics such as transaction throughput and latency, which are essential to understanding the behavior of the system under varying load conditions.

Through carefully designed test configurations and scenarios, the study provided valuable insights into the performance of the developed system. The results demonstrated that the system is generally effective in meeting the operational demands of the railway industry, with specific observations regarding the relative speed of read versus write operations, the impact of transaction arrival rates on throughput, and the influence of byte size on overall performance.

However, the study also identified significant limitations of the Caliper tool, particularly its challenge in generating performance reports for chaincode functions that are interdependent or linked through complex workflows. Although Caliper excels in evaluating functions that operate independently, it struggles with accurately tracking and reporting on interconnected functions, such as those that require request IDs to create and accept offer requests. This limitation underscores the difficulty of conducting comprehensive performance evaluations for Blockchain

applications with intricate dependencies, as rerunning and merging transactions across these functions can be highly complex and often impractical.

Despite these challenges, the testing revealed that certain operations, such as managing offers and data hashes, could be effectively assessed within the constraints of the Caliper tool. The ability to repeatedly perform actions like creating offers within a journey and attaching data hashes allowed meaningful performance results, particularly in terms of how data querying impacts consumers who exchange data for offers.

In conclusion, this chapter underscores the importance of rigorous performance testing in the development and deployment of Blockchain applications, especially in critical infrastructure settings such as the railway industry. The findings not only validate the system's design but also contribute to a broader understanding of the potential and limitations of Blockchain in complex, real-world environments. Moving forward, there is a need for improved tools and methodologies that can better accommodate the interdependencies inherent in many Blockchain applications, ensuring that performance evaluations are as comprehensive and accurate as possible.

Chapter 7

Discussion

7.1 Introduction

The following section 7.2 presents a detailed examination of the research questions addressed in this thesis, highlighting the role of Blockchain technology in overcoming key challenges within the railway industry. As the sector grapples with issues such as data centralisation, stakeholder mistrust, and the complexities of cost attribution, Blockchain offers a promising solution through its decentralised and transparent nature. By exploring these research questions, this thesis uncovers how Blockchain can revolutionise data management, improve trust among stakeholders, and streamline cost processes in the rail industry. Next, Section 7.3 provides insight into the implications of Blockchain integration, demonstrating its capacity to transform the sector's operations and contribute to a more efficient, secure, and equitable industry. Finally, Section 7.4 discusses the limitations and threats that Blockchain poses and provides a collection of topics for further investigation.

7.2 Response to Research Questions

Research Question 1: Considering that data silos are typical in the rail sector, can Blockchain technology address the fundamental issues of data centralisation and stakeholder mistrust?

Data monetisation in the railway industry has the potential to incentivise stakeholders to

invest in the data generated by railway operations, thus creating new revenue streams. Using data analytics and insights, companies can optimise operational efficiency, improve passenger services, and develop targeted marketing strategies. This, in turn, can lead to the creation of value-added services such as real-time tracking, personalised travel experiences, and predictive maintenance. As stakeholders recognise the economic benefits of using data, they are likely to invest more in data infrastructure and analytics capabilities, further driving innovation and revenue growth within the industry. The Blockchain in this part would play a crucial role in ensuring that stakeholders who engage in data payment transactions can establish an open, transparent, and competitive data marketplace. The developed system provided in Chapter 5, employs the Blockchain to create an accountant record of all transactions of purchasing data from providers that manage sensors monitoring specific assets.

Research Question 2: Can Blockchains improve the transparency of the data cost attribution process and compliance with agreement clauses and therefore reduce the need for a third party to enforce the terms of the agreement?

One of the drawbacks identified in Project T1010 is the lack of a clear enforcement strategy regarding the agreements between stakeholders. In addition, data consumers face a long and cumbersome process in locating the appropriate data provider. The developed system proposes utilising Blockchain technology as an efficient solution for tracking agreement statuses and enforcing SLAs with defined attributes. Smart contracts can encode detailed SLAs that specify the quality, delivery time, and other conditions of data services. This ensures that payments are only released when the data provider meets the agreed-upon standards, incentivising compliance and high-quality service delivery. Consequently, the necessity for intermediaries, such as billing agents, is reduced. This accelerates the cost-attribution process while also cutting down on intermediary-related expenses, thus decreasing overall transaction costs.

Research Question 3: How can the efficiency of the data cost attribution process be improved via smart contracts?

The data cost attribution process can be significantly improved by using smart contracts on a Blockchain platform. In addition to keeping all cost attribution processes saved in an immutable

ledger, which enhances the auditability and accountability of the transactions, making it easier to verify and trace payment histories, here are several ways in which this has been achieved in the developed testbed that is introduced in Chapter 5:

- **Automated and Transparent Cost Calculation:** Smart contracts are automating the cost distribution mechanism based on predefined criteria such as escrow releasing or agreement status (revoked/expired). This ensures transparency and fairness in distributed costs, as the rules are coded in the contract and executed automatically.

- **Dispute Resolution Mechanisms:** Smart contracts include built-in mechanisms for dispute resolution, automatically enforcing penalties or refunds if disputes arise, based on the terms of the contract.

- **Real-Time Payments and Settlements:** In the proposed accounting framework, a smart contract designed to function as an escrow facilitates instant payment processing and settlement. When the agreement's conditions are satisfied, the payment is automatically initiated, minimising delays and conflicts related to manual invoicing systems.

Research Question 4: In light of the fact that Blockchains aren't meant to be used for storing huge amounts of data, how can the features of Blockchain technology be used to ensure the integrity of the data that is exchanged?

The responsibility for data storage lies with the provider, and this research does not address the solutions for off-chain data storage. Instead, we monitor provider enrollment by submitting the hash value of the generated data to the Blockchain. Using this hash value along with other metadata, such as data creation, modifications, request events, and related sensors, we can ensure the integrity of the data at any time by comparing the stored hash with the hash of the data retrieved. Since the hash of the data is stored on the Blockchain, any unauthorised changes to the data can be detected by comparing the current data hash with the Blockchain-recorded hash. This mechanism provides tamper-proof and verification of data integrity. By integrating the Blockchain-based solution developed in this thesis with secure distributed storage such as the one suggested in (J. D. Preece and J. M. Easton, 2018), a robust system can be achieved

that provides data integrity, transparency, and secure access control, while also leveraging the scalability and performance benefits of off-chain storage solutions. This hybrid approach ensures that the strengths of Blockchain in providing auditability and security are combined with the efficiency and capacity of distributed storage systems.

Research Question 5: What are the potential applications of Blockchains in the context of the railway sector?

Blockchain technology has the potential to disrupt many industries, including the rail industry, as reviewed in Chapter 3. Although its adoption in the rail sector is slower compared to other industrial sectors such as finance or logistics, technology could improve efficiency, transparency, and security in the rail industry. As Blockchain matures and becomes more scalable and interoperable, its use in the rail industry is expected to grow, benefiting operators, passengers, and stakeholders. This thesis shows clearly how a Blockchain has the potential to distribute trust among stakeholders and enable automated marketisation of rail industry data to reach fair cost attribution between parties.

7.3 Implication of Blockchain Integration

Within railway applications, the thesis examines how the Blockchain could resolve trust concerns about data sharing and cost attribution in the IoT setting. The following subsections outline some suggestions and lessons discovered:

7.3.1 Smart Contracts and Condition Monitoring Data

This thesis emphasises the importance of smart contracts aligned with the transaction flow protocols specific to the Blockchain platform on which they operate. Leading Blockchain platforms like Ethereum and HLF implement event-driven smart contracts that are designed to function strictly within the Blockchain's ecosystem, thereby avoiding direct interaction with external systems. These contracts are intended to remain inactive until they are triggered by an authorised entity, which could be an external source or another smart contract within the same

network.

To ensure deterministic outcomes and reach a consensus on the transaction's integrity and immutability, Blockchain platforms typically impose a well-defined transaction flow. By following this structured approach, Blockchain systems can ensure the consistency and reliability of transactions within the network. This, in turn, supports key Blockchain principles such as immutability, transparency, and security. Such a framework is vital in fostering trust among participants in a decentralised network, as it ensures that smart contracts execute consistently and predictably, thereby maintaining the network's overall stability and functionality.

In Chapter 5, the automation of smart contracts related to agreements, escrows, and cost distribution is implemented to mitigate the risk of external threats by ensuring that these contracts operate independently of the direct management of the network participants. This automation is critical for maintaining the integrity and security of the Blockchain network, as it minimises the potential for manipulation or interference by unauthorised entities. Additionally, the smart contract responsible for adding data hashes to the Blockchain must retain a digital fingerprint of the data stored externally. This mechanism serves as a crucial validation tool, allowing the tracking and verification of data integrity in instances of suspected misbehavior or discrepancies.

However, a significant challenge facing Blockchain technology is its limited capacity to store the vast amounts of data generated by various applications directly on the Blockchain. Storing extensive data on-chain can severely impact the Blockchain's scalability, leading to inefficiencies and performance issues. As of the writing of this thesis, the prevailing solutions involve securing data while storing them on external storage systems, with the Blockchain serving primarily as a ledger for the hashes or references to these data. This approach is necessary to balance the need for data security and integrity with the technical constraints of Blockchain scalability. In recent literature, the integration of off-chain storage solutions, such as InterPlanetary File System (IPFS) or cloud-based systems, with Blockchain networks has been explored as a means of addressing these scalability challenges while ensuring data security (Shafagh et al., 2017; Shrivastava and Patel, 2023; Razzaq et al., 2023).

7.3.2 Integrating Blockchain Technology Into Existing Railway Systems

The two use cases presented in Section 4.3 demonstrate the feasibility of integrating the proposed Blockchain-based framework into current operational systems within the railway domain. However, when it comes to legacy systems, further investigation is required to fully assess the potential and challenges of such integration.

Legacy systems often exhibit characteristics that complicate seamless Blockchain adoption. One of the key issues is inconsistent data formatting. Many of these systems still rely on manual data entry processes, which can introduce human errors such as typographical mistakes or inconsistencies in data structure. This becomes particularly critical in Blockchain applications, where data integrity is ensured through cryptographic hashing. Since the hash value functions as a digital fingerprint of the stored data, even minor alterations, such as extra spaces, misspellings, or inconsistent formatting, can result in different hash outputs. This undermines the reliability of the data verification process.

Therefore, integrating the proposed Blockchain framework into legacy systems must be approached with caution, especially in environments where data modification is frequent or not tightly controlled. Ensuring data consistency and standardisation is essential before hashing and storing records on the Blockchain. To evaluate the practical application of blockchain in RCM, it is essential to consider both the challenges and benefits, as detailed in the following subsection.

7.3.3 Challenges and Benefits of Applying Blockchain Technology in RCM

Integrating Blockchain into RCM practices within the railway industry introduces several technical and organisational challenges:

- Data Quality Issues: Legacy systems often contain poorly structured or incomplete data. Manual entry increases the risk of errors, which can break the immutability and verification functionality of Blockchain systems.

- Integration Complexity: Existing maintenance platforms and databases may not be readily

compatible with Blockchain technology. Interfacing these systems requires middleware, custom APIs, or system reengineering.

- Scalability Limitations: Blockchain platforms may introduce latency or limited throughput, particularly when large volumes of maintenance data need to be processed in near real-time.
- Cost of Implementation: The initial cost of adopting Blockchain, including infrastructure upgrades, system redesign, and staff training, can be substantial.
- Organisational Resistance: Personnel may be hesitant to adopt new technologies, especially if perceived as disruptive to existing maintenance workflows.
- Data Privacy Concerns: Ensuring the confidentiality of sensitive maintenance data on an immutable and shared ledger can be difficult, especially in public or consortium Blockchains.

Despite these challenges, Blockchain offers several potential advantages that align well with the principles and objectives of RCM:

- Enhanced Data Integrity: Blockchain ensures that once data is recorded, it cannot be altered without detection, thereby improving the reliability.
- End-to-End Traceability: Shared data, agreements, and cost distribution can be securely tracked.
- Support for Predictive Maintenance: Blockchain can complement IoT and AI technologies by securely recording sensor data and predictive insights, thereby enabling more proactive maintenance strategies.
- Operational Efficiency: By automating validation processes and reducing data redundancy, Blockchain can streamline maintenance planning and execution.
- Decentralised Trust and Collaboration: Blockchain facilitates secure data sharing among multiple stakeholders without reliance on centralised control.

7.3.4 Sharing Data and Accountant Models in IoT

In the framework proposed in Chapter 5, an accounting model is introduced through the concept of Data as a Product (DaaP). This model encompasses the sale of raw data, processed data, or insights derived from such data to consumers, irrespective of the format. While other accounting models exist, they fall beyond the scope of this thesis. It is important to note that the data provider retains full authority over pricing strategies for the data being offered. However, this control may be subject to regulation, necessitating the establishment of legal frameworks within the railway industry to ensure compliance and standardisation. In the context of IoT, the integration of IoT with Blockchain technology is examined in greater detail in Chapter 4 and Chapter 5. This examination focuses on identifying the essential attributes required to establish robust data agreements and enforce penalties between the involved parties. The proposed system not only meets functional requirements, such as the recording and verification of agreements but also addresses key quality criteria, including the availability of data hashes and data integrity.

The detection of violations in non-functional criteria can often be facilitated through the monitoring of functional criteria. For example, a sensor failure is particularly critical and less acceptable than other types of failure, as it directly impacts the integrity of data transmitted to the Blockchain. Maintaining comprehensive visibility throughout the IoT system is therefore crucial, requiring the implementation of effective monitoring mechanisms. Sensor failure can be easily detected when no hash values are transmitted to the Blockchain, allowing for immediate identification of issues related to the associated data agreements. However, failures within external storage systems present a more complex challenge. In such instances, although data hashes may be successfully recorded on the Blockchain, the corresponding data might become unavailable, resulting in issues that extend beyond the control of the Blockchain. Although external storage management falls outside the scope of this thesis, addressing this challenge requires the integration of external storage monitoring with Blockchain technology. This integration is essential to improve the accuracy of cost attribution and facilitate the development of a more precise accounting model.

7.3.5 Reliability and Performance

The MVCC protocol employed by HLF is designed to reject all transactions except one in the event of a conflict in the read-write sets, as previously discussed in Section 3.5.2. Notably, in the testing of the smart contracts conducted in Chapter 6, we did not encounter any instances of MVCC conflicts. This absence of conflicts suggests that the smart contracts were well designed, demonstrating their dependability and resilience under testing conditions. Specifically, the design of these smart contracts effectively avoids read-write set conflicts by ensuring that each transaction processes distinct records. Consequently, it is essential to emphasise that the logic underpinning smart contract design plays a critical role in influencing Blockchain network performance. This thesis underscores the importance of smart contract design in throughput assessment, providing results that validate the effectiveness of our approach. However, due to the novelty of the application, there was no opportunity to compare our results with other existing applications.

Moreover, the experiments detailed in Chapter 6 achieved significant throughput and maintained an acceptable average latency, although the development and testing were carried out on a limited local machine. This achievement highlights the ability of the system to handle a large volume of transactions from the client application end. Furthermore, our findings emphasise the need to verify the reliability of the smart contract in both production and testing environments. It should be emphasised that passing validation tests in a testing environment does not automatically ensure dependable performance in production scenarios.

With respect to scalability, it remains a significant challenge for Blockchain-based high-throughput applications. Therefore, it is imperative that system architecture considers scalability issues to avoid overloading the Blockchain with unnecessary transactions. This approach is vital to maintain the performance and efficiency of Blockchain systems in real-world applications.

7.3.6 Thesis Genarlisation

Although the primary objective of this thesis is to develop a decentralised cost distribution application, it also extends to decentralising other related processes, including the formulation, enforcement, and termination of SLAs. The findings presented in this thesis, while primarily focused on IoT within railways, have broader applicability across various industry sectors employing cloud computing, traditional IT infrastructure, and communications. The decentralised approach to SLAs proposed here can be adapted to these diverse contexts, demonstrating the versatility of the solution.

Regarding the Blockchain platform selected for this research, HLF offers a unique approach to Blockchain implementation, particularly in its support for consensus mechanisms, permissioned networks, transaction lifetimes, and smart contract lifetimes. It is important to note that the conclusions drawn from this thesis are specific to HLF and may not be directly applicable to different Blockchain platforms. Nevertheless, the research illustrates that Blockchain technology, overall, has the potential to enhance trust mechanisms and decentralise the responsibilities associated with data trading and cost distribution, regardless of the underlying platform.

Moreover, while the proposed methods are designed with HLF in mind, there is a strong indication that these methods could be adapted for use with other Blockchain systems. Consequently, future research should explore the extent to which the concepts and methodologies developed in this thesis can be applied to alternative Blockchain platforms, such as Ethereum. This will help to assess the generalisability and flexibility of the proposed solutions in different Blockchain environments.

The generic and adaptable sensor simulator described in Chapter 4 facilitates the seamless incorporation of Java-based IoT simulators with Blockchain environment. Implemented as a bridging component using the Java programming language, this simulator is designed to operate with any smart contract installed on the Blockchain network. Its flexibility lies in its loosely coupled architecture, which allows it to interact with the logic of the smart contract without being tightly bound to it. This modularity ensures that the simulator can be effectively utilised in various scenarios, regardless of the specific smart contract in use, as long as HLF remains

the core Blockchain infrastructure. Moreover, this design approach makes the sensor simulator adaptable to use cases involving different numbers of sensors, thereby enhancing its applicability across a range of testing environments.

7.4 Threats to Validity and Future Trends

The main motivation for this research project is the need to foster trust within the railway industry, particularly in the exchange of monitoring data, while ensuring a fair cost distribution among the parties involved. This objective led to the development of smart contracts that operate independently and in a decentralised manner, effectively removing reliance on potentially untrustworthy intermediaries. Empirical testing indicates that the integration of Blockchain technology can significantly enhance existing trust mechanisms, thereby enabling an equitable allocation of costs to achieve net benefits. For example, the use of autonomous smart contracts for tasks related to cost management and incident handling can reduce the need for manual intervention, thus streamlining operations and minimising the risk of disputes. This improvement is largely attributable to the inherent qualities of Blockchain technology, such as its traceability, transparency, and the immutability of its ledger. HLF embodies these attributes in alignment with its unique design philosophy. However, the deployment, maintenance, and operation of permissioned Blockchain platforms like HLF introduce a distinct set of challenges. Unlike platforms such as Ethereum, where the focus could predominantly be on smart contract design, HLF requires considerable attention to networking and infrastructure. Consequently, the presumed trustworthiness of the Blockchain is significantly influenced by its deployment architecture and operational framework. In Figure 5.2, you can see an example of a Blockchain network. Examine the subsequent cases as a result:

- A permissioned Blockchain network is fully governed by a single authority, which in this context is the DfT, serving as the franchising authority for the national rail network. The DfT is tasked with the design and procurement of new and replacement rail franchise services, which makes its role in managing the entities allowed to participate in the

Blockchain network crucial. Even though the ORR, which acts as Britain's independent regulator for railway safety and economics, supervises adherence to health and safety regulations among railway operators and functions as the proper entity for resolving regulatory disputes within the Blockchain network, it is still considered a single authority.

- The concept of decentralisation refers to the distribution of control and decision-making across multiple participants in the network. However, in some implementations of HLF, the management of CAs, which is responsible for issuing and validating the digital certificates that authenticate network participants, may be centralised under a single operator.

  This means that while the Blockchain network itself operates in a decentralised manner, with no single entity having control over the entire system, the critical function of managing identities and access through certificate authorities is governed by one central authority. This approach can simplify administration and enhance security, but it also introduces a potential single point of control within an otherwise decentralised network architecture.

- In HLF, the smart contract, also known as chaincode, can be modified or upgraded by a participant if the existing endorsement policy does not align with the desired requirements. The endorsement policy dictates which participants must endorse a transaction for it to be considered valid. If the current policy is found to be inadequate or incorrect, authorised participants have the ability to propose an upgrade to the smart contract, ensuring that the endorsement policy meets the network's operational and security needs. This capability allows for the continuous improvement and adaptation of smart contracts to better serve the goals of the Blockchain network.

- In HLF, orderers are responsible for the ordering and dissemination of transactions across the network. When most of these orderers are controlled by a single participant within the Blockchain network, this participant has significant influence over the transaction ordering process. This centralisation of control may raise concerns regarding the fairness and impartiality of the consensus mechanism, as it could potentially compromise the decentralised nature of the Blockchain.

Therefore, it is unrealistic to anticipate that HLF will achieve complete decentralisation. To increase trust in the smart contract of the application, it is essential that the contract is designed with care and precision. The thoughtful creation of the smart contract is crucial to ensure its reliability, security, and effectiveness within the framework of HLF, given the inherent structure and governance model of the platform.

Another concern is maintaining data ownership via Blockchain applications. Blockchain technology has significant capabilities for capturing and monitoring data from external sources through intermediaries such as oracles, but it also has limitations. Although Blockchain can securely record and track data transactions within its ledger, it lacks control over the creation, copying, or processing of data outside its network, as well as the handling of data once it has been exchanged on the Blockchain. In terms of data exchange, Blockchain effectively prevents double-spending of tokens representing raw data within its network. However, it cannot stop the original data holder from distributing the same data or a copy through other means outside the Blockchain, making data exchange on the Blockchain non-exclusive. Furthermore, the recipient's ability to duplicate data reveals that data on the Blockchain is not scarce, challenging the notion that the Blockchain enables complete control over tokenised data. Therefore, presenting Blockchain as a comprehensive tool for data control may overstate its capabilities. While it provides secure and transparent data transactions, it does not fully address issues of data exclusivity and control beyond its network. Legally, using Blockchain for data exchange does not change that data, due to its volatility, non-excludability, and non-rivalry, making it unsuitable for traditional ownership. Instead, this scenario should be seen as various contractual arrangements for making data accessible, with Blockchain and smart contracts automating and enforcing specific aspects of these agreements, rather than altering the legal status of data.

Moreover, the following topics are recommended for future research:

7.4.1 Relevance to Ethereum

Despite being Blockchain-based, this thesis's contributions are heavily influenced by the HLF ideology. For a comparison of Ethereum and HLF, see section 3.6. Future research will

look more closely at the benefits and drawbacks of the suggested methods as they relate to Ethereum. Future research will focus on how Ethereum affects the solutions suggested by this thesis in relation to the **Open nature.** Ethereum is an example of a public network that has to provide incentives for anonymous nodes to engage in the network infrastructure, in contrast to HLF. Because of this, transactions need execution fees to pay for the involvement of these nodes. Transactions from automated smart contracts, such as generating agreements, building escrows, storing and retrieving hashes, and cost calculations, would be expensive under this arrangement. Who must pay the bill is the most important question. Furthermore, concerns are raised regarding the industry and government sectors' readiness to embrace a model in which everyone can contribute to the infrastructure, and transparency is a right that applies to everyone, not just authorised participants. Furthermore, the majority of public networks are permissionless, necessitating the use of strong consensus procedures like the PoW protocol, which affects overall performance. Thus, in the future, research will empirically investigate whether the permissionless Blockchain paradigm is feasible for the proposed application.

7.4.2 Enhancement of Performance at Blockchain Infrastructure Level

The Blockchain network evaluated in Chapter 6 was hosted on a local infrastructure. Future research could focus on assessing its performance within a scalable environment, such as cloud hosting. This line of inquiry could lead to significant advancements in optimising the HLF infrastructure. For example, identifying optimal block configurations is crucial to minimising latency and maximising throughput. Given HLF's modular architecture, it would also be valuable to explore how different variables, such as the network size in terms of participating organisations, peer commitment, and endorsement policies, impact overall performance.

Moreover, the performance of the Blockchain network could be influenced by various implementation factors, including the choice of programming languages for smart contracts, chaincode configurations, ordering services, and consensus mechanisms. The physical characteristics of the network, such as the processing power of individual nodes and whether they are co-located in the same geographical region or data center, could also have a significant impact on perfor-

mance. These areas present valuable opportunities for future research that aims to improve the efficiency and scalability of HLF in various operational contexts.

7.4.3 Interoperability

Blockchain networks and platforms often operate in isolation, creating silos that hinder seamless interaction between them. For example, the HLF and Ethereum networks are inherently incompatible by default. However, through interoperability, it is possible to create a hybrid architecture that leverages the strengths of both networks. This approach enables the integration of distinct Blockchain ecosystems, allowing them to work together and capitalise on their respective advantages.

7.4.4 Payment and Cryptocurrency

The payment gateway utilised in this study poses certain security risks, which highlights an area for future research. Investigating the integration of cryptocurrency for micropayments is particularly promising. This approach not only aligns with the previous suggestion on interoperability but also addresses a key limitation of HLF, which, unlike Ethereum, does not natively support payments or has an integrated cryptocurrency. Additionally, the use of cryptocurrency for micropayments would be advantageous because of the elimination of intermediary fees, making transactions more cost-effective.

References

A. Alabdulkarim, Abdullah, Peter D. Ball, and Ashutosh Tiwari (2014). “Influence of resources on maintenance operations with different asset monitoring levels: A simulation approach”. In: *Business Process Management Journal* 20.2, pp. 195–212.

Abdelrahim, Malik et al. (2022). “Crypto Currency Cloud Mining”. In: *2022 International Conference on Artificial Intelligence in Everything (AIE)*. IEEE, pp. 488–492.

Abdulrahman, Yusra et al. (2023). “AI and Blockchain Synergy in Aerospace Engineering: An Impact Survey on Operational Efficiency and Technological Challenges”. In: *IEEE Access* 11, pp. 87790–87804.

Akkaoui, Raifa, Xiaojun Hei, and Wenqing Cheng (2020). “EdgeMediChain: A Hybrid Edge Blockchain-Based Framework for Health Data Exchange”. In: *IEEE Access* 8, pp. 113467–113486.

Alaswad, Suzan and Yisha Xiang (2017). “A review on condition-based maintenance optimization models for stochastically deteriorating system”. In: *Reliability engineering & system safety* 157, pp. 54–63.

Aldweesh, Amjad et al. (2018). “Performance benchmarking of smart contracts to assess miner incentives in Ethereum”. In: *2018 14th European Dependable Computing Conference (EDCC)*. IEEE, pp. 144–149.

Alemi, Alireza, Francesco Corman, and Gabriel Lodewijks (2016). “Condition monitoring approaches for the detection of railway wheel defects”. In: *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit* 231.8, pp. 961–981.

Alexopoulos, Kosmas, Sotiris Makris, et al. (2016). "A concept for context-aware computing in manufacturing: the white goods case". In: *International Journal of Computer Integrated Manufacturing* 29.8, pp. 839–849.

Alexopoulos, Kosmas, Konstantinos Sipsas, et al. (2018). "An industrial internet of things based platform for context-aware information Services in Manufacturing". In: *International Journal of Computer Integrated Manufacturing* 31.11, pp. 1111–1123.

Ali, Saqib et al. (2018). "A Blockchain-Based Decentralized Data Storage and Access Framework for PingER". In: *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE, pp. 1303–1308.

Alladi, Tejasvi et al. (2019). "Blockchain in Smart Grids: A Review on Different Use Cases". In: *Sensors* 19.22, p. 4862.

AlOdat, Zeyad, Assad Abbas, and Samee U Khan (2019). "Randomness analyses of the secure hash algorithms, SHA-1, SHA-2 and modified SHA". In: *2019 International Conference on Frontiers of Information Technology (FIT)*. IEEE, pp. 316–3165.

AlOdat, Zeyad and Samee Khan (2019). "Constructions and attacks on hash functions". In: *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, pp. 139–144.

Altarawneh, Amani et al. (2020). "Buterin's scalability trilemma viewed through a state-change-based classification for common consensus algorithms". In: *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, pp. 0727–0736.

Alzahrani, Fahad Ahmad (2020). "Subscription-based data-sharing model using blockchain and data as a service". In: *IEEE Access* 8, pp. 115966–115981.

Alzahrani, Rahma A, Simon J Herko, and John M Easton (2020). "Blockchain application in remote condition monitoring". In: *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, pp. 2385–2394.

Amadi-Echendu, Joe E et al. (2010). *Definitions, Concepts and Scope of Engineering Asset Management (Engineering Asset Management Review)*. Springer Verlag London Limited.

Androulaki, Elli et al. (2018). “Hyperledger fabric: a distributed operating system for permissioned blockchains”. In: *Proceedings of the thirteenth EuroSys conference*, pp. 1–15.

Anwar, Muhammad Rehan, Desy Apriani, and Irsa Rizkita Adianita (2021). “Hash algorithm in verification of certificate data integrity and security”. In: *Aptisi Transactions on Technopreneurship (ATT)* 3.2, pp. 181–188.

Anwar, Sidra et al. (2020). “Generation Analysis of Blockchain Technology: Bitcoin and Ethereum.” In: *International Journal of Information Engineering & Electronic Business* 12.4.

Aponte-Novoa, Fredy Andres et al. (2021). “The 51% attack on blockchains: A mining behavior study”. In: *IEEE Access* 9, pp. 140549–140564.

Asokan, Nadarajah, Matthias Schunter, and Michael Waidner (1997). “Optimistic protocols for fair exchange”. In: *Proceedings of the 4th ACM Conference on Computer and Communications Security*, pp. 7–17.

Asokan, Nadarajah, Victor Shoup, and Michael Waidner (1998). “Optimistic fair exchange of digital signatures”. In: *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, pp. 591–606.

Averin, Andrey, Alexsandr Samartsev, and Nikolay Sachenko (2020). “Review of methods for ensuring anonymity and de-anonymization in blockchain”. In: *2020 International Conference Quality Management, Transport and Information Security, Information Technologies (IT&QM&IS)*. IEEE, pp. 82–87.

Avoine, Gildas and Serge Vaudenay (2004). “Optimistic fair exchange based on publicly verifiable secret sharing”. In: *Information Security and Privacy: 9th Australasian Conference, ACISP 2004, Sydney, Australia, July 13-15, 2004. Proceedings 9*. Springer, pp. 74–85.

Bada, Abigael Okikijesu et al. (2021). “Towards a green blockchain: A review of consensus mechanisms and their energy consumption”. In: *2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, pp. 503–511.

Bao, Jiabin et al. (2021). “A Survey of Blockchain Applications in the Energy Sector”. In: *IEEE Systems Journal* 15.3, pp. 3370–3381.

Baran, Paul (1964). *On Distributed Communications: I. Introduction to Distributed Communications Networks*. RAND Corporation. URL: https://www.rand.org/pubs/research_memoranda/RM3420.html (visited on 09/03/2023).

Beije, Aljosja, Feyen, and Frijters (2023). *Decentralised management of logistics documentation*. White paper. Progress towards Federated Logistics through the Integration of TEN-T into A Global Trade Network (PLANET), p. 29. URL: https://www.etp-logistics.eu/wp-content/uploads/2023/05/PLANET_WHITE-PAPER_Decentralised-management-of-logistics-documentation.pdf (visited on 08/31/2023).

Bentov, Iddo and Ranjit Kumaresan (2014). "How to use bitcoin to design fair protocols". In: *Annual Cryptology Conference*. Springer, pp. 421–439.

Bentov, Iddo, Charles Lee, et al. (2014). "Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract] y". In: *ACM SIGMETRICS Performance Evaluation Review* 42.3, pp. 34–37.

Bernal, Esteban, Maksym Spiryagin, and Colin Cole (2018). "Onboard condition monitoring sensors, systems and techniques for freight railway vehicles: a review". In: *IEEE Sensors Journal* 19.1, pp. 4–24.

Bhebhe, M and PN Zincume (2020). "Maintenance strategies in the rail environment". In: *SAIIE31 Proceedings, 5th–7th October*.

Bilal, Muhammad et al. (2016). "Big Data in the construction industry: A review of present status, opportunities, and future trends". In: *Advanced engineering informatics* 30.3, pp. 500–521.

Binder, Mario, Vitaliy Mezhuyev, and Martin Tschandl (2023). "Predictive maintenance for railway domain: A systematic literature review". In: *IEEE Engineering Management Review* 51.2, pp. 120–140.

Biswas, Baidyanath and Rohit Gupta (2019). "Analysis of barriers to implement blockchain in industry and service sectors". In: *Computers & Industrial Engineering* 136, pp. 225–241.

Böhme, Rainer et al. (2015). "Bitcoin: Economics, technology, and governance". In: *Journal of economic Perspectives* 29.2, pp. 213–238.

Boucher, Philip, Susana Nascimento, and Mihalis Kritikos (2017). *How blockchain technology could change our lives: in depth analysis.* URL: https://data.europa.eu/doi/10.2861/926645 (visited on 08/31/2023).

Brandvold, Morten et al. (2015). "Price discovery on Bitcoin exchanges". In: *Journal of International Financial Markets, Institutions and Money* 36, pp. 18–35.

Brezulianu, Adrian et al. (2020). "Active Control Parameters Monitoring for Freight Trains, Using Wireless Sensor Network Platform and Internet of Things". In: *Processes* 8.6, p. 639.

British Transport Police (2018). *Policing Great Britain's Rail Network*. Tech. rep. British Transport Police Authority. URL: https://btpa.police.uk/wp-content/uploads/2015/06/8271_BTPA_Policing-Plans_AW_web.pdf (visited on 03/23/2023).

Buterin, Vitalik (2014). "A next-generation smart contract and decentralized application platform". In: *white paper* 3.37, pp. 2–1.

Caccioli, Fabio, Giacomo Livan, and Tomaso Aste (2016). "Scalability and egalitarianism in peer-to-peer networks". In: *Banking Beyond Banks and Money: A Guide to Banking Services in the Twenty-First Century*, pp. 197–212.

Cai, Wei et al. (2018). "Decentralized applications: The blockchain-empowered software system". In: *IEEE access* 6, pp. 53019–53033.

Carretero, Jesús et al. (2003). "Applying RCM in large scale systems: a case study with railway networks". In: *Reliability engineering & system safety* 82.3, pp. 257–273.

Castillo, Michael del (2021). *Blockchain 50 2021*. Forbes. Section: Crypto & Blockchain. URL: https://www.forbes.com/sites/michaeldelcastillo/2021/02/02/blockchain-50/ (visited on 09/15/2023).

Chaum, David (1983). "Blind signatures for untraceable payments". In: *Advances in Cryptology: Proceedings of Crypto 82*. Springer, pp. 199–203.

— (1985). "Security without identification: Transaction systems to make big brother obsolete". In: *Communications of the ACM* 28.10, pp. 1030–1044.

Chen, Zhonglin et al. (2018). "A Security Authentication Scheme of 5G Ultra-Dense Network Based on Block Chain". In: *IEEE Access* 6, pp. 55372–55379.

Christidis, Konstantinos and Michael Devetsikiotis (2016a). "Blockchains and smart contracts for the internet of things". In: *IEEE Access* 4, pp. 2292–2303.

— (2016b). "Blockchains and smart contracts for the internet of things". In: *IEEE Access* 4, pp. 2292–2303.

Cocco, Luisanna, Andrea Pinna, and Michele Marchesi (2017). "Banking on Blockchain: Costs Savings Thanks to the Blockchain Technology". In: *Future Internet* 9.3, p. 25.

Costa, Thiago Bulhões da Silva et al. (2022). "Blockchain-based architecture design for personal health record: development and usability study". In: *Journal of Medical Internet Research* 24.4.

Crosby, Michael et al. (2016). "Blockchain technology: Beyond bitcoin". In: *Applied innovation* 2.6-10, p. 71.

Cui, Hao et al. (2019). "Real-time inspection system for ballast railway fasteners based on point cloud deep learning". In: *IEEE Access* 8, pp. 61604–61614.

Dai, Wei (1998). *b-money | Satoshi Nakamoto Institute*. URL: https://nakamotoinstitute.org/b-money/ (visited on 09/07/2023).

Dan, Wu et al. (2020). "Research on aerospace data transaction platform and method based on blockchain". In: *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*. 2020 2nd International Conference on Information Technology and Computer Application (ITCA), pp. 410–413.

Danezis, George and Sarah Meiklejohn (2015). "Centrally banked cryptocurrencies". In: *arXiv preprint arXiv:1505.06895*.

De Filippi, Primavera and Andrea Leiter (2021). "Blockchain in Outer Space". In: *AJIL Unbound* 115, pp. 413–418. ISSN: 2398-7723. DOI: 10.1017/aju.2021.63. URL: https://www.cambridge.org/core/product/identifier/S2398772321000635/type/journal_article (visited on 08/28/2023).

Debnath, Santanu, Abir Chattopadhyay, and Subhamoy Dutta (2017). "Brief review on journey of secured hash algorithms". In: *2017 4th International Conference on Opto-Electronics and Applied Optics (Optronix)*. IEEE, pp. 1–5.

Delmolino, Kevin et al. (2016). "Step by Step Towards Creating a Safe Smart Contract: Lessons and Insights from a Cryptocurrency Lab". In: *Financial Cryptography and Data Security*. Ed. by Jeremy Clark et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 79–94.

Desai, Harsh et al. (2018). "Adjudicating Violations in Data Sharing Agreements Using Smart Contracts". In: *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, pp. 1553–1560.

DigiCash - company brochure (1997). URL: https://web.archive.org/web/19970610023426/http://www.digicash.com/publish/digibro.html (visited on 09/07/2023).

Dinh, Tien Tuan Anh et al. (2018). "Untangling blockchain: A data processing view of blockchain systems". In: *IEEE transactions on knowledge and data engineering* 30.7, pp. 1366–1385.

Drobyazko, S. and T. Hilorme (2021). "Influence of Sustainable Development of Space Activities on Earth Ecology". In: *IOP Conference Series: Earth and Environmental Science* 940.1, p. 012014.

DuPont, Quinn (2017). "Experiments in algorithmic governance: A history and ethnography of "The DAO," a failed decentralized autonomous organization". In: *Bitcoin and beyond*. Routledge, pp. 157–177.

Durand, Antoine, Emmanuelle Anceaume, and Romaric Ludinard (2019). "Stakecube: Combining sharding and proof-of-stake to build fork-free secure permissionless distributed ledgers". In: *Networked Systems: 7th International Conference, NETYS 2019, Marrakech, Morocco, June 19–21, 2019, Revised Selected Papers 7*. Springer, pp. 148–165.

Easton, John M (2021). "Blockchains: A distributed data ledger for the rail industry". In: *Research Anthology on Blockchain Technology in Business, Healthcare, Education, and Government*. IGI Global, pp. 1267–1279.

Efe Gencer, Adem et al. (2018). "Decentralization in Bitcoin and Ethereum Networks". In: *arXiv e-prints*, arXiv–1801.

ENISA (2016). *Distributed Ledger Technology & Cybersecurity - Improving information security in the financial sector*. European Union Agency For Network And Information Security. URL: https://www.enisa.europa.eu/publications/blockchain-security (visited on 09/15/2023).

Eskandari, Shayan et al. (2018). "A first look at the usability of bitcoin key management". In: *arXiv preprint arXiv:1802.04351*.

Faiz, RB and Eran A Edirisinghe (2009). "Decision making for predictive maintenance in asset information management". In: *Interdisciplinary Journal of Information, Knowledge, and Management* 4, p. 23.

Fanti, Giulia and Pramod Viswanath (2017). "Deanonymization in the bitcoin P2P network". In: *Advances in Neural Information Processing Systems* 30.

Fayyaz, Muhammad Asad Bilal and Christopher Johnson (2020). "Object detection at level crossing using deep learning". In: *Micromachines* 11.12, p. 1055.

Fernandez-Carames, Tiago M and Paula Fraga-Lamas (2020). "Towards post-quantum blockchain: A review on blockchain cryptography resistant to quantum computing attacks". In: *IEEE access* 8, pp. 21091–21116.

Fernández-Caramés, Tiago M and Paula Fraga-Lamas (2018). "A Review on the Use of Blockchain for the Internet of Things". In: *Ieee Access* 6, pp. 32979–33001.

Figueroa-Lorenzo, Santiago et al. (2021). "Alarm collector in smart train based on ethereum blockchain events-log". In: *IEEE Internet of Things Journal* 8.17, pp. 13306–13315.

Franz, CMAP et al. (2018). "Microbial food safety in the 21st century: Emerging challenges and foodborne pathogenic bacteria". In: *Trends Food Sci. Technol* 81, pp. 155–158.

Friedlmaier, Maximilian, Andranik Tumasjan, and Isabell M. Welpe (2018). "Disrupting Industries with Blockchain: The Industry, Venture Capital Funding, and Regional Distribution of Blockchain Ventures". In: *Proceedings of the 51st Hawaii International Conference on System Sciences*.

Fröhlich, Michael, Felix Gutjahr, and Florian Alt (2020). "Don't lose your coin! Investigating Security Practices of Cryptocurrency Users". In: *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, pp. 1751–1763.

Galar, Diego, Dammika Seneviratne, and Uday Kumar (2018). "Big data in railway O&M: A dependability approach". In: *Innovative applications of big data in the railway industry*. IGI Global, pp. 1–26.

Galar, Diego, Adithya Thaduri, et al. (2015). "Context awareness for maintenance decision making: A diagnosis and prognosis approach". In: *Measurement* 67, pp. 137–150.

Galvez, Juan F., J.C. Mejuto, and J. Simal-Gandara (2018). "Future challenges on the use of blockchain for food traceability analysis". In: *TrAC Trends in Analytical Chemistry* 107, pp. 222–232.

Gandal, Neil et al. (2018). "Price manipulation in the Bitcoin ecosystem". In: *Journal of Monetary Economics* 95, pp. 86–96.

Gbadamosi, Abdul-Quayyum et al. (2019). "The role of internet of things in delivering smart construction". In: *CIB world building congress*. June, pp. 17–21.

Georg, Jean-Michael et al. (2020). "Sensor and actuator latency during teleoperation of automated vehicles". In: *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, pp. 760–766.

Gerhátová, Zuzana, Vladislav Zitrickỳ, and Vladimir Klapita (2021). "Industry 4.0 implementation options in railway transport". In: *Transportation Research Procedia* 53, pp. 23–30.

Goldfeder, Steven et al. (2017). "When the cookie meets the blockchain: Privacy risks of web payments via cryptocurrencies". In: *arXiv preprint arXiv:1708.04748*.

Gonzalo, Alfredo Peinado, Mani Entezami, Clive Roberts, Paul Weston, Edward Stewart, et al. (2022). "Railway track location estimation using onboard inertial sensors". In: *Vehicle System Dynamics* 60.10, pp. 3631–3649.

Gonzalo, Alfredo Peinado, Mani Entezami, Clive Roberts, Paul Weston, Graeme Yeo, et al. (2022). "Observations on track evolution from onboard inertial measurements". In: *2022 UKACC 13th International Conference on Control (CONTROL)*. IEEE, pp. 18–23.

Grover, Lov K (1996). "A fast quantum mechanical algorithm for database search". In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pp. 212–219.

Gubbi, Jayavardhana et al. (2013). "Internet of Things (IoT): A vision, architectural elements, and future directions". In: *Future generation computer systems* 29.7, pp. 1645–1660.

Gulyi, Ilia (2020). "Economic assessment of the implementation of distributed data registry platforms in multimodal transport". In: *E3S Web of Conferences*. Vol. 220. EDP Sciences, p. 01068.

Haber, Stuart and W Scott Stornetta (1991). *How to time-stamp a digital document*. Springer.

Haltuf, Miroslav (2016). "Shift2Rail JU from Member State's Point of View". In: *Transportation Research Procedia*. Transport Research Arena TRA2016 14, pp. 1819–1828.

He, Kaiming et al. (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Heimerdinger, Walter L. and Charles B. Weinstock (1992). *A Conceptual Framework for System Fault Tolerance*. Fort Belvoir, VA: Defense Technical Information Center. DOI: 10.21236/ADA258467. URL: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=6ae162d41e5c447d6f2be21d8fef822f141cd958 (visited on 09/05/2023).

Hellman, Martin (1976). "New directions in cryptography". In: *IEEE transactions on Information Theory* 22.6, pp. 644–654.

Herzberg, Amir (2003). "Micropayments". In: *Payment technologies for E-commerce*, p. 344.

Hilorme, Tetiana et al. (2019). "Decision making model of introducing energy-saving technologies based on the analytic hierarchy process". In: *Journal of Management Information and Decision Sciences* 22.4, pp. 489–494.

Hiramoto, Naoki and Yoichi Tsuchiya (2020). "Measuring dark web marketplaces via Bitcoin transactions: From birth to independence". In: *Forensic Science International: Digital Investigation* 35, p. 301086.

Hoelzl, Cyprien et al. (2022). "On-board monitoring for smart assessment of railway infrastructure: A systematic review". In: *The Rise of Smart Cities*, pp. 223–259.

Hu, Qin et al. (2020). "Sync or Fork: Node-Level Synchronization Analysis of Blockchain". In: *Wireless Algorithms, Systems, and Applications*. Ed. by Dongxiao Yu, Falko Dressler, and Jiguo Yu. Vol. 12384. Cham: Springer International Publishing, pp. 170–181.

Hyperledger and Scale Working Group (2021). *Hyperledger blockchain performance metrics white paper*. URL: https://www.lfdecentralizedtrust.org/learn/publications/blockchain-performance-metrics (visited on 05/14/2024).

Hyperledger Fabric (2024). *Performance considerations — Hyperledger Fabric Docs main documentation*. URL: https://hyperledger-fabric.readthedocs.io/en/release-2.5/performance.html (visited on 05/14/2024).

Islam, Muhammad, Mubashir Husain Rehmani, and Jinjun Chen (2021). "Transparency-privacy trade-off in blockchain-based supply chain in industrial internet of things". In: *2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*. IEEE, pp. 1123–1130.

Jaiman, Vikas and Visara Urovi (2020). "A Consent Model for Blockchain-Based Health Data Sharing Platforms". In: *IEEE Access* 8, pp. 143734–143745.

Jamshidi, Ali et al. (2018). "A decision support approach for condition-based maintenance of rails based on big data analysis". In: *Transportation Research Part C: Emerging Technologies* 95, pp. 185–206.

Javaid, Haris, Chengchen Hu, and Gordon Brebner (2019). "Optimizing validation phase of hyperledger fabric". In: *2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*. IEEE, pp. 269–275.

Jin, Jiong et al. (2014). "An information framework for creating a smart city through internet of things". In: *IEEE Internet of Things journal* 1.2, pp. 112–121.

Jin, Long et al. (2017). "Self-powered wireless smart sensor based on maglev porous nanogenerator for train monitoring system". In: *Nano Energy* 38, pp. 185–192.

Jwo, Jung-Sing et al. (2021). “Intelligent system for railway wheelset press-fit inspection using deep learning”. In: *Applied Sciences* 11.17, p. 8243.

Kabbinale, Aniruddh Rao et al. (2020). “Blockchain for economically sustainable wireless mesh networks”. In: *Concurrency and Computation: Practice and Experience* 32.12, e5349.

Kang, Peng, Wenzhong Yang, and Jiong Zheng (2022). “Blockchain Private File Storage-Sharing Method Based on IPFS”. In: *Sensors* 22.14, p. 5100.

Karakose, Mehmet and Orhan Yaman (2020). “Complex fuzzy system based predictive maintenance approach in railways”. In: *IEEE Transactions on industrial informatics* 16.9, pp. 6023–6032.

Karim, Ramin et al. (2016). “Maintenance analytics–the new know in maintenance”. In: *IFAC-PapersOnLine* 49.28, pp. 214–219.

Khan, Prince Waqas and Yung-Cheol Byun (2021). “Blockchain-Based Peer-to-Peer Energy Trading and Charging Payment System for Electric Vehicles”. In: *Sustainability* 13.14, p. 7962.

King, Sunny and Scott Nadal (2012). “PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake”. In: *self-published paper, August* 19.1.

Klöckner, Maximilian et al. (2020). “Does Blockchain for 3D Printing Offer Opportunities for Business Model Innovation?” In: *Research-Technology Management* 63.4, pp. 18–27.

Koblitz, Neal (1987). “Elliptic curve cryptosystems”. In: *Mathematics of computation* 48.177, pp. 203–209.

Kochovski, Petar and Vlado Stankovski (2018). “Supporting smart construction with dependable edge computing infrastructures and applications”. In: *Automation in Construction* 85, pp. 182–192.

Koutsos, Vlasis et al. (2022). “Agora: A Privacy-Aware Data Marketplace”. In: *IEEE Transactions on Dependable and Secure Computing* 19.6, pp. 3728–3740.

Lamport, Leslie (2001). “Paxos Made Simple”. In: *ACM SIGACT News (Distributed Computing Column) 32, 4 (Whole Number 121, December 2001)*, pp. 51–58.

Lamport, Leslie, Robert Shostak, and Marshall Pease (2019). "The Byzantine generals problem". In: *Concurrency: the works of leslie lamport*, pp. 203–226.

Landscheidt, Steffen and Mirka Kans (2016). "Automation practices in wood product industries: Lessons learned, current practices and future perspectives". In: *The 7th Swedish Production Symposium SPS, 25-27 October, 2016, Lund, Sweden*. Lund University.

Le, Tuyen and H David Jeong (2016). "Interlinking life-cycle data spaces to support decision making in highway asset management". In: *Automation in construction* 64, pp. 54–64.

Lee, In and Kyoochun Lee (2015). "The Internet of Things (IoT): Applications, investments, and challenges for enterprises". In: *Business horizons* 58.4, pp. 431–440.

Lee, Jay (2003). "E-manufacturing—fundamental, tools, and transformation". In: *Robotics and Computer-Integrated Manufacturing* 19.6, pp. 501–507.

Levina, Anastasia et al. (2021). "Logistic blockchain platform project: railways case study". In: *XIV International Scientific Conference "INTERAGROMASH 2021" Precision Agriculture and Agricultural Machinery Industry, Volume 1*. Springer, pp. 647–655.

Li, Wenting et al. (2017). "Securing Proof-of-Stake Blockchain Protocols". In: *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. Ed. by Joaquin Garcia-Alfaro et al. Cham: Springer International Publishing, pp. 297–315. ISBN: 978-3-319-67816-0.

Liang, Yiheng (2019). "Identity Verification and Management of Electronic Health Records with Blockchain Technology". In: *2019 IEEE International Conference on Healthcare Informatics (ICHI)*, pp. 1–3.

Lohman, Clemens, Leonard Fortuin, and Marc Wouters (2004). "Designing a performance measurement system: A case study". In: *European journal of operational research* 156.2, pp. 267–286.

Lomotey, Richard K and Ralph Deters (2013). "RSenter: terms mining tool from unstructured data sources". In: *International Journal of Business Process Integration and Management* 6.4, pp. 298–311.

Madhuravani, B and DSR Murthy (2013). "Cryptographic hash functions: SHA family". In: *Int J Innov Technol Explor Eng* 2, pp. 326–9.

Madine, Mohammad Moussa et al. (2020). "Fully Decentralized Multi-Party Consent Management for Secure Sharing of Patient Health Records". In: *IEEE Access* 8, pp. 225777–225791.

Malakar, Bidhan and Binoy K Roy (2018). "Adaptive multisensor data fusion technique for train localisation and detection of accidental train parting". In: *IET Radar, Sonar & Navigation* 12.8, pp. 853–861.

Mandolla, Claudio et al. (2019). "Building a digital twin for additive manufacturing through the exploitation of blockchain: A case analysis of the aircraft industry". In: *Computers in Industry* 109, pp. 134–152.

Mani Entezami, Mat Rippin and David Whitehead (2021). *Switch and Point Machine Monitoring Network Rail High Speed -Stratford*. Tech. rep. University of Birmingham, T243 RCM Track SC.

Mao, Dianhui et al. (2018). "Credit Evaluation System Based on Blockchain for Multiple Stakeholders in the Food Supply Chain". In: *International Journal of Environmental Research and Public Health* 15.8, p. 1627.

Martino, Raffaele and Alessandro Cilardo (2019). "A flexible framework for exploring, evaluating, and comparing SHA-2 designs". In: *IEEE Access* 7, pp. 72443–72456.

Mayer, André Henrique, Cristiano André da Costa, and Rodrigo da Rosa Righi (2020). "Electronic health records in a Blockchain: A systematic review". In: *Health Informatics Journal* 26.2, pp. 1273–1288.

Mazlan, Ahmad Akmaluddin et al. (2020). "Scalability Challenges in Healthcare Blockchain System—A Systematic Review". In: *IEEE Access* 8, pp. 23663–23673.

Mcmahon, P., T. Zhang, and R. Dwight (2020). "Requirements for Big Data Adoption for Railway Asset Management". In: *IEEE Access* 8, pp. 15543–15564.

Mehar, Muhammad Izhar et al. (2019). "Understanding a revolutionary and flawed grand experiment in blockchain: the DAO attack". In: *Journal of Cases on Information Technology (JCIT)* 21.1, pp. 19–32.

Meijers, James et al. (2021). "Cost-Effective Blockchain-based IoT Data Marketplaces with a Credit Invariant". In: *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, pp. 1–9.

Miller, Victor S (1985). "Use of elliptic curves in cryptography". In: *Conference on the theory and application of cryptographic techniques*. Springer, pp. 417–426.

Mitchell, Billy F and Robert J Murry (1995). "Predictive maintenance program evolution-lessons learned". In: *Annual Reliability and Maintainability Symposium 1995 Proceedings*. IEEE, pp. 7–10.

Morabito, Vincenzo (2017). *Business innovation through Blockchain: The B ^{3} perspective*. 1st ed. Vol. 20. Springer. 188 pp.

Mujica, Gabriel, Javier Henche, and Jorge Portilla (2021). "Internet of Things in the railway domain: Edge sensing system based on solid-state LIDAR and fuzzy clustering for virtual coupling". In: *IEEE Access* 9, pp. 68093–68107.

Nabilou, Hossein (2019). "How to regulate bitcoin? Decentralized regulation for a decentralized cryptocurrency". In: *International Journal of Law and Information Technology* 27.3, pp. 266–291.

Nakamoto, Satoshi (2008). "Bitcoin: A Peer-to-Peer Electronic Cash System". In: *Decentralized business review* 4.2, p. 15.

Narayanan, Arvind and Jeremy Clark (2017). "Bitcoin's Academic Pedigree: The concept of cryptocurrencies is built from forgotten ideas in research literature." In: *Queue* 15.4, pp. 20–49.

Naser, Feras (2018). "The potential use of blockchain technology in railway applications: an introduction of a mobility and speech recognition prototype". In: *2018 IEEE international conference on big data (Big Data)*. IEEE, pp. 4516–4524.

Nash, Chris (2008). "Passenger railway reform in the last 20 years–European experience reconsidered". In: *Research in Transportation Economics* 22.1, pp. 61–70.

Natarajan, Harish, Solvej Krause, and Helen Gradstein (2017). "Distributed ledger technology and blockchain". In.

Network Rail (2018). *A better railway for a better Britain-strategic business plan 2019-2024*. Tech. rep. URL: https://www.networkrail.co.uk/wp-content/uploads/2018/02/Strategic-business-plan-high-level-summary.pdf (visited on 03/23/2023).

Nguyen, Giang-Truong and Kyungbaek Kim (2018). "A survey about consensus algorithms used in blockchain". In: *Journal of Information processing systems* 14.1, pp. 101–128.

Nichols, Mary L (2017). "Survey of Multisensor Data Fusion Systems". In: *Handbook of multisensor data fusion*. CRC Press, pp. 711–720.

Niebel, Thomas, Fabienne Rasel, and Steffen Viete (2019). "BIG data–BIG gains? Understanding the link between big data analytics and innovation". In: *Economics of Innovation and New Technology* 28.3, pp. 296–316.

Office of Rail and Road (2024). *Passenger Rail Usage Statistics*. URL: https://dataportal.orr.gov.uk/statistics/usage/passenger-rail-usage (visited on 10/02/2024).

Office of Rail Regulation (2013). *Policing Great Britain's Rail Network*. Tech. rep. British Transport Police Authority. URL: https://www.orr.gov.uk/sites/default/files/om/long-term-regulatory-statement.pdf (visited on 03/23/2023).

Oneto, L et al. (2023). "DAYDREAMS-Development of Prescriptive Analytics based on Artificial Intelligence for Railways Intelligent Asset Management Systems". In: *Transportation Research Procedia* 72, pp. 478–485.

Ongaro, Diego and John Ousterhout (2014). "In search of an understandable consensus algorithm". In: *2014 USENIX annual technical conference (USENIX ATC 14)*, pp. 305–319.

Özyilmaz, Kazim Rifat, Mehmet Doğan, and Arda Yurdakul (2018). "IDMoB: IoT data marketplace on blockchain". In: *2018 crypto valley conference on blockchain technology (CVCBT)*. IEEE, pp. 11–19.

Patrickson, Bronwin (2021). "What do blockchain technologies imply for digital creative industries?" In: *Creativity and Innovation Management* 30.3, pp. 585–595.

Pike, Douglas et al. (2018). *PoST White Paper*. URL: https://cdn.vericonomy.com/documents/VeriCoin-Proof-of-Stake-Time-Whitepaper.pdf (visited on 09/06/2023).

Pintelon, L, N Nagarur, and F Van Puyvelde (1999). "Case study: RCM–yes, no or maybe?" In: *Journal of Quality in Maintenance Engineering* 5.3, pp. 182–192.

Preece, Joseph and John Easton (2018). "A review of prospective applications of blockchain technology in the railway industry". In: *Preprint submitted to Int. J. Railw. Technol*, pp. 1–22.

— (2019). "Blockchain technology as a mechanism for digital railway ticketing". In: *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, pp. 3599–3606.

Preece, Joseph D and John M Easton (2018). "Towards encrypting industrial data on public distributed networks". In: *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, pp. 4540–4544.

PUB, FIPS (2000). "Digital signature standard (DSS)". In: *FIPS PUB*, pp. 186–192.

Putz, Benedikt and Günther Pernul (2019). "Trust Factors and Insider Threats in Permissioned Distributed Ledgers: An Analytical Study and Evaluation of Popular DLT Frameworks". In: *Transactions on Large-Scale Data-and Knowledge-Centered Systems XLII*, pp. 25–50.

Qian, Yu et al. (2019). "Railroad infrastructure 4.0: Development and application of an automatic ballast support condition assessment system". In: *Transportation Geotechnics* 19, pp. 19–34.

Ran, Yunfeng et al. (2021). "High-accuracy on-site measurement of wheel tread geometric parameters by line-structured light vision sensor". In: *IEEE Access* 9, pp. 52590–52600.

Razzaq, Abdul et al. (2023). "IoT data sharing platform in web 3.0 using blockchain technology". In: *Electronics* 12.5, p. 1233.

Rennie, Ellie, Jason Potts, and Ana Pochesneva (2019). *Blockchain and the creative industries: a provocation paper*. RMIT Blockchain Innovation Hub. DOI: 10.25916/5DC8A108DC471. URL: https://apo.org.au/node/267131 (visited on 09/01/2023).

Risius, Marten and Kai Spohrer (2017). "A Blockchain Research Framework". In: *Business & Information Systems Engineering* 59.6, pp. 385–409.

Rivest, Ronald L, Adi Shamir, and Leonard Adleman (1978). "A method for obtaining digital signatures and public-key cryptosystems". In: *Communications of the ACM* 21.2, pp. 120–126.

Roberts, Clive et al. (2019). "Continuous railway track monitoring using passenger trains". In: *World Congress on Railway Research: Railway Research to Enhance the Customer Experience*.

Sahal, Radhya et al. (2021). "Blockchain-empowered digital twins collaboration: smart transportation use case". In: *Machines* 9.9, p. 193.

Sanka, Abdurrashid Ibrahim and Ray CC Cheung (2021). "A systematic review of blockchain scalability: Issues, solutions, analysis and future research". In: *Journal of Network and Computer Applications* 195, p. 103232.

Sas, Corina and Irni Eliana Khairuddin (2017). "Design for trust: An exploration of the challenges and opportunities of bitcoin users". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 6499–6510.

Schneider, Nathan (2019). "Decentralization: an incomplete ambition". In: *Journal of cultural economy* 12.4, pp. 265–285.

Schuh, Fabian and D. Larimer (2017). "BITSHARES 2.0: GENERAL OVERVIEW". In: URL: https://www.semanticscholar.org/paper/BITSHARES-2.0%3A-GENERAL-OVERVIEW-Schuh-Larimer/c5c6eefc414d32637890dbe40a1440e46f68e10f (visited on 09/06/2023).

Seebacher, Stefan and Ronny Schüritz (2017). "Blockchain technology as an enabler of service systems: A structured literature review". In: *Exploring Services Science: 8th International Conference, IESS 2017, Rome, Italy, May 24-26, 2017, Proceedings 8*. Springer, pp. 12–23.

Selcuk, Sule (2017). "Predictive maintenance, its implementation and latest trends". In: *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 231.9, pp. 1670–1679.

Shafagh, Hossein et al. (2017). "Towards blockchain-based auditable storage and sharing of IoT data". In: *Proceedings of the 2017 on cloud computing security workshop*, pp. 45–50.

Sharma, Arvind K and SK Mittal (2019). "Cryptography & network security hash function applications, attacks and advances: A review". In: *2019 Third International Conference on Inventive Systems and Control (ICISC)*. IEEE, pp. 177–188.

Shen, Chunzi et al. (2020). "A blockchain based federal learning method for urban rail passenger flow prediction". In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 1–5.

Shirani, Ashraf (2018). "Blockchain for global maritime logistics." In: *Issues in Information Systems* 19.3.

Shor, Peter W (1999). "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer". In: *SIAM review* 41.2, pp. 303–332.

Shrivastava, Gaurav and Sachin Patel (2023). "Secure Storage and Data Sharing Scheme Using Private Blockchain-Based HDFS Data Storage for Cloud Computing". In: *International Journal of Computer Networks and Applications* 10.1, pp. 28–38.

Smith, Andrew SJ and Chris A Nash (2023). "Will the latest British reforms to rail passenger service procurement work?" In: *Research in Transportation Economics* 100, p. 101321.

Sober, Michael et al. (2023). "A blockchain-based IoT data marketplace". In: *Cluster computing* 26.6, pp. 3523–3545.

Sparkrail (2014). *Cross-industry remote condition monitoring (T1010)*. URL: http://www.sparkrail.org/Lists/Records/DispForm.aspx?%20ID=8096.

— (2016). *Cross-industry remote condition monitoring, Commercial, Final report Appendix E Standard Form (Template) (T1010 Report Appendix)*. Tech. rep. RSSB. URL: https://www.rssb.co.uk/-/media/Project/RSSB/RssbWebsite/Documents/Registered/Research-Projects/2020/07/06/17/45/2015-03-report-t1010-RCM-commercial-report.pdf.

Stenström, Christer, Aditya Parida, and Diego Galar (2012). "Performance indicators of railway infrastructure". In: *The international Journal of railway technology* 1.3, pp. 1–18.

Swanson, Laura (2001). "Linking maintenance strategies to performance". In: *International journal of production economics* 70.3, pp. 237–244.

Szabo, Nick (1997). "Formalizing and securing relationships on public networks". In: *First monday*.

Tagarev, Andrey (2023). *Towards an Integrated System for Global Transport Tracking*. Progress towards Federated Logistics through the Integration of TEN-T into A Global Trade Network

(PLANET), p. 20. URL: https://www.etp-logistics.eu/wp-content/uploads/2023/05/PLANET_WHITE-PAPER_System-For-Global-Transportation-Tracking.pdf (visited on 08/31/2023).

Tasca, Paolo and Claudio J Tessone (2017). "Taxonomy of blockchain technologies. Principles of identification and classification". In: *arXiv preprint arXiv:1708.04872*.

Tucker, G.J. and A. Hall (2014). "Breaking down the barriers to more cross-industry Remote Condition Monitoring (RCM)". In: *6th IET Conference on Railway Condition Monitoring (RCM 2014)*. IET. Institution of Engineering and Technology.

Valadares, Dalton Cézane Gomes et al. (2023). "Privacy-preserving blockchain technologies". In: *Sensors* 23.16, p. 7172.

Velmurugan, Rama S and Tarun Dhingra (2015). "Maintenance strategy selection and its impact in maintenance function: A conceptual framework". In: *International Journal of Operations & Production Management* 35.12, pp. 1622–1661.

Vora, Jayneel et al. (2018). "BHEEM: A Blockchain-Based Framework for Securing Electronic Health Records". In: *2018 IEEE Globecom Workshops (GC Wkshps)*. 2018 IEEE Globecom Workshops (GC Wkshps), pp. 1–6.

Vu, Thang X, Symeon Chatzinotas, and Björn Ottersten (2019). "Blockchain-based content delivery networks: Content transparency meets user privacy". In: *2019 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, pp. 1–6.

Vyas, Nick, Aljosja Beije, and Bhaskar Krishnamachari (2022). *Blockchain and the Supply Chain*. Second. Kogan Page Publishers. 289 pp. ISBN: 978 1 3986 0522 0.

Walch, Angela (2016). "The path of the blockchain lexicon (and the law)". In: *Rev. Banking & Fin. L.* 36, p. 713.

Wang, Biao et al. (2020). "Design of auto disturbance rejection controller for train traction control system based on artificial bee colony algorithm". In: *Measurement* 160, p. 107812.

Wang, Jiaheng et al. (2021). "Blockchain-enabled wireless communications: a new paradigm towards 6G". In: *National Science Review* 8.9, nwab069.

Wang, Wennan et al. (2023). "A Blockchain-Based Continuous Micropayment Scheme Using Lockable Signature". In: *Mathematics* 11.16, p. 3472.

Ward, C P et al. (2011). "Condition Monitoring Opportunities Using Vehicle-Based Sensors". In: *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit* 225.2, pp. 202–218.

Westeon, PF et al. (2007). "Monitoring vertical track irregularity from in-service railway vehicles". In: *Proceedings of the institution of mechanical engineers, Part F: Journal of Rail and Rapid Transit* 221.1, pp. 75–88.

Westerkamp, Martin, Friedhelm Victor, and Axel Küpper (2020). "Tracing manufacturing processes using blockchain-based token compositions". In: *Digital Communications and Networks* 6.2, pp. 167–176.

Weston, PF et al. (2007). "Monitoring lateral track irregularity from in-service railway vehicles". In: *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit* 221.1, pp. 89–100.

Wirdum, Aaron van (2018). *The Genesis Files: If Bitcoin Had a First Draft, Wei Dai's B-Money Was It*. Bitcoin Magazine - Bitcoin News, Articles and Expert Insights. URL: https://bitcoinmagazine.com/technical/genesis-files-if-bitcoin-had-first-draft-wei-dais-b-money-was-it (visited on 09/07/2023).

Wöhrer, Maximilian and Uwe Zdun (2018). "Design patterns for smart contracts in the ethereum ecosystem". In: *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, pp. 1513–1520.

Wood, Gavin (2014). "Ethereum: A secure decentralised generalised transaction ledger". In: *Ethereum project yellow paper* 151.2014, pp. 1–32. URL: https://cryptodeep.ru/doc/paper.pdf.

Wright, Aaron and Primavera De Filippi (2015). "Decentralized blockchain technology and the rise of lex cryptographia". In: *Available at SSRN 2580664*.

Wu, Xuyang et al. (2024). "Cumulative frost heave and hydrothermal process of the high-speed railway subgrade under extreme climate conditions in northwest China". In: *Transportation Geotechnics*, p. 101297.

Xiaodong, Zhang, Li Ping, and Ma Xiaoning (2020). "Research on Technical Architecture and Overall Scheme of Railway Block Chain Service Platform". In: *Proceedings of the 2020 3rd International Conference on Blockchain Technology and Applications*, pp. 55–60.

Xu, Jianxi et al. (2018). "A VR-based the emergency rescue training system of railway accident". In: *Entertainment Computing* 27, pp. 23–31.

Xu, Yan et al. (2022). "Application of blockchain technology in food safety controlcurrent trends and future prospects". In: *Critical Reviews in Food Science and Nutrition* 62.10, pp. 2800–2819.

Yaga, Dylan et al. (2019). "Blockchain technology overview". In: *arXiv preprint arXiv:1906.11078*.

Yang, Haochun et al. (2022). "Key technologies of low-carbon-oriented intelligent travel service for urban rail transit based on MaaS". In: *International Conference on Intelligent Traffic Systems and Smart City (ITSSC 2021)*. Vol. 12165. SPIE, pp. 172–176.

Yiannas, Frank (2018). "A New Era of Food Transparency Powered by Blockchain". In: *Innovations: Technology, Governance, Globalization* 12.1, pp. 46–56.

Zamani, Mahdi, Mahnush Movahedi, and Mariana Raykova (2018). "Rapidchain: Scaling blockchain via full sharding". In: *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pp. 931–948.

Zellagui, Amine, Naima Hadj-Said, and Adda Ali-Pacha (2019). "Comparative Study Between Merkle-Damgård And Other Alternative Hashes Construction". In: *Proceedings of the Second Conference on Informatics and Applied Mathematics IAM, Guelma, Algeria*, pp. 24–25.

Zhang, Zongyang et al. (2020). "A refined analysis of zcash anonymity". In: *Ieee Access* 8, pp. 31845–31853.

Zheng, Xu, Zhipeng Cai, and Yingshu Li (2018). “Data linkage in smart internet of things systems: a consideration from a privacy perspective”. In: *IEEE Communications Magazine* 56.9, pp. 55–61.

Zhu, Li et al. (2021). “Joint security and train control design in blockchain-empowered CBTC system”. In: *IEEE Internet of Things Journal* 9.11, pp. 8119–8129.

Zikopoulos, Paul and Chris Eaton (2011). *Understanding big data: Analytics for enterprise class hadoop and streaming data*. McGraw-Hill Osborne Media.

Zou, Weiqin et al. (2019). “Smart contract development: Challenges and opportunities”. In: *IEEE transactions on software engineering* 47.10, pp. 2084–2106.

Zuo, Yanjun (2021). “Making smart manufacturing smarter – a survey on blockchain technology in Industry 4.0”. In: *Enterprise Information Systems* 15.10, pp. 1323–1353.

Zyskind, Guy, Oz Nathan, et al. (2015). “Decentralizing privacy: Using blockchain to protect personal data”. In: *2015 IEEE security and privacy workshops*. IEEE, pp. 180–184.

Appendix A

Agreement Templates in Project T1010

This part presents various examples of the condition monitoring agreements suggested within the "RCM T1010 cross-industry" initiative. All subsequent examples are documented in (Sparkrail, 2016).

Schedule 2

Specification

[PARTIES TO COMPLETE]

Equipment – Specifications and Use

To include:

- Specification of any Equipment – what is being fitted and where; who is installing such Equipment – NB different parties may be installing different Equipment
- Any Relevant Approvals that need to be sought e.g. from ROSCOs, DfT, ORR etc., who will be responsible for obtaining them and by when
- Parties also to consider whether they can list the applicable Industry Standards for clarity
- Who owns the Equipment e.g. if it is being provided by a third party manufacturer etc.
- Who can use any Equipment – including any third parties
- Are there circumstances in which the Equipment may need to be removed? E.g. for maintenance?
- Parties may also wish to provide for any safety/operational issues in respect of any Equipment – for example, any liabilities if any Equipment fails and has knock-on repercussions

Data Collection, Processing, Hosting Transmission

Parties to provide details of who is responsible for the collection and handling of the Data produced from the Equipment. To include details of:

- Data to be collected;
- Information lifecycle (initial processing, cleansing, transmission etc);
- Frequency of collection and provision;
- Reporting provisions
- Storage provisions
  - Who, if any Party, shall be entitled to store the Data, for what period of time and on what terms?
  - Who, if any Party, shall be entitled to keep records of the collected Data? If so, on what terms and for what period of time?

Figure A.1: Schedule 2 from Appendix E in T1010 Project

Schedule 4

Data Ownership and Intellectual Property

Use of Data

The Parties shall be entitled to use, develop and modify the Data in accordance with the following provisions:

[PARTIES TO SPECIFY FURTHER DETAILS AS TO HOW DATA, OR ANY DERIVED DATA[9], MAY BE USED OR MODIFIED]

Intellectual Property Rights

Title to Data

[OPTION 1 – to be used where a particular Party retains all IPR to the Data as it is recorded – perhaps with licence to another Party for various purposes (see below)]

1. Title to any Intellectual Property Rights in relation to the Data that is recorded by the Equipment shall be vested in and owned by the Party that produces such Data, being, for the purpose of this Agreement, [Facilitator/Beneficiary/third party].

[OPTION 2 – title to the original Data remains vested in a particular Party, but another Party has title to any modifications or improvements or processed output made in respect of the Data]

2. Title to any Intellectual Property Rights in relation to the Data that is recorded by the Equipment shall remain vested in and owned by the Party that produces such Data, being, for the purposes of this Agreement, [Facilitator/Beneficiary/third party]. Title to any Intellectual Property Rights in relation to improvements and/or modifications made to such Data shall be vested in and owned by the Party who has made such improvements and/or modifications immediately on such improvement and/or modification.

[OPTION 3 – title to IPR developed or created in the course of the Agreement, to be owned by the party that developed or created the IPR]

3. All Intellectual Property Rights arising, developed or created by or on behalf of a Party to this Agreement in the course of or as a consequence of performance of this Agreement shall vest in and shall be owned by that Party immediately upon creation.

Licensing of IPR

4. [The Facilitator/Data owner] grants or agrees to grant to [the Beneficiary/Data user] a non-exclusive, irrevocable, royalty-free licence (with[out] a right to sub-licence) of any Intellectual Property Rights in relation to the Data that is generated by the Equipment for the Permitted Purpose.

[OPTION 1 – license to use the Data for any and all purposes]

[9] Parties may wish to consider what they want to do with the Data once modified and/or derived, including

Figure A.2: Schedule 4 from Appendix E in T1010 Project

Schedule 5

Payments

[PARTIES TO COMPLETE]

Part A – Payment for Equipment

The total cost of supply and installation of the Equipment by the [Facilitator] to the [Beneficiary] shall be £[ ], to be made on completion of the following milestones:

Payment Number	Milestone	Amount
1		
2		
3		
4		

Part B – Payment for the maintenance and operation of the Equipment

[PARTIES TO AGREE HOW PAYMENT FOR THE OPERATION AND MAINTENANCE OF EQUIPMENT WILL WORK. SOME OPTIONS ARE INCLUDED BELOW – DELETE THOSE NOT RELEVANT:

[OPTION 1] The Parties agree that the following payments shall be made in respect of the operation and maintenance of the Equipment in accordance with the times, frequency and amounts set out below:

Payment Number	Time/Frequency	Amount
1		
2		

[OPTION 2 - PARTIES TO AGREE ON AN ALTERNATIVE BASIS FOR PAYMENT IN RESPECT OF THE MAINTENANCE AND OPERATION OF THE EQUIPMENT.]

Part C – Payment for Data

[PARTIES TO AGREE HOW PAYMENT FOR DATA WILL WORK. SOME OPTIONS INCLUDED BELOW – DELETE THOSE NOT RELEVANT:

Figure A.3: Schedule 5 from Appendix E in T1010 Project

Schedule 6

Service Level Agreement

[PARTIES TO COMPLETE (if an SLA is agreed). These are likely to be highly bespoke to each project[11]]

The contents of Schedule 2 are likely to impact on what is included in Schedule 6 as the Specification should drive the output and both must be achievable

To include details of the following in relation to the Data:

- Availability
- Timeliness (or frequency)
- Quality (integrity, precision, accuracy)
- Transfer dependability
- Security
- Fault tolerances
- Response times
- Steps that are to be taken in the event of service delivery issues
- Escalation procedure
- Compensation for downtime
- Disaster recovery in the event of system failure
- Parties to consider whether a performance regime may be appropriate and what remedies there may need to be for any failure to meet the Specification – for example, liquidated damages, increased monitoring, remedial plans, liability caps for breach etc

Figure A.4: Schedule 6 from Appendix E in T1010 Project