Development of Realtime Data Ingestion and Data Management Software Architecture for Internet of Things Applications

Thesis submitted by

Hiren Dutta

Doctor of Philosophy (Engineering)

Department of Information Technology

Faculty Council of Engineering & Technology

Jadavpur University

Kolkata, India

2025

Under the guidance of:

Dr. Parama Bhaumik

Associate Professor

Department of Information Technology

Jadavpur University

Salt Lake, Sector-3, Kolkata - 700106

JADAVPUR UNIVERSITY

KOLKATA, INDIA

INDEX NO: 185/19/E

1. **Title of the Thesis:** Development of Realtime Data Ingestion and Data Management Software Architecture for Internet of Things Applications

2. **Name, Designation & Institution of the Supervisor/s:**

Parama Bhaumik
Associate Professor,
Department of Information Technology,
Jadavpur University
Kolkata

3. **List of Publications:**

   a. **Journals**

1. Dutta, H., Nagesh, S., Talluri, J., and Bhaumik, P. (2023, May). A Solution to Blockchain Smart Contract Based Parametric Transport and Logistics Insurance. IEEE Transactions on Services Computing, vol. 16, no. 5, pp. 3155-3167, doi: 10.1109/TSC.2023.3281516.
2. Dutta, H., Bhaumik, P. (2018, October). Real-Time Dynamic Data Control Mechanism Using Auto Rebalancing Strategy for IotT Applications. i-manager's Journal on Computer Science, 6(1),9-17, ISSN (Print): 2347-2227, ISSN (Online): 2347-6141, https://doi.org/10.26634/jcom.6.1.14501.
3. Dutta, H., Bhaumik, P. (2020, September). Survey Of Internet of Things (IoT) Systems Architecture. i-manager's Journal on Software Engineering, Volume 15. No.1, ISSN-0973-5151.

Journals (Communicated)

4. Dutta, H., Bhaumik, P. (2024, May). A Novel Approach to Harmonizing Efficiency and Regulation in Decentralized Capital Markets, communicated to the journal

b. Conferences

5. Dutta, H., Bhaumik, P. (2022, September). A Novel Software Architecture of Self-Managed, Reusable, Interoperable IoT Asset Whitelisting and Trust Management. IEEE IAS GUCON 2022, 5th IEEE International Conference on Computing, Power, and Communication Technologies.
6. Dutta, H., Bhaumik, P. (2022, November). An Approach to Effectively Manage Access Control, Privacy, and Information Transparency in Hybrid Blockchain for Decentralized IoT Applications, IEEE INDICON 2022
7. Dutta, H., Bhaumik, P. (2023, November). A Blockchain Based Sustainable Digital Insurance Business Parametric Solution Architecture to Protect Realtime QSR Business Interruption Losses. 7th IEEE International Conference CSITSS-2023

4. List of Patents:

None

5. List of Presentations in National / International / Conferences/ Workshops:

1. Dutta H. participated as a speaker at the Jadavpur University ACM Student Chapter workshop on Blockchain and Generative AI; March 2024.
2. Dutta, H., Bhaumik, P. (2023, November). A Blockchain Based Sustainable Digital Insurance Business Parametric Solution Architecture to Protect Realtime QSR Business Interruption Losses. 7th IEEE International Conference CSITSS-2023
3. Dutta, H., Bhaumik, P. (2022, September). A Novel Software Architecture of Self-Managed, Reusable, Interoperable IoT Asset Whitelisting and Trust Management. IEEE IAS GUCON 2022, 5th IEEE International Conference on Computing, Power, and Communication Technologies.
4. Dutta, H., Bhaumik, P. (2022, November). An Approach to Effectively Manage Access Control, Privacy, and Information Transparency in Hybrid Blockchain for Decentralized IoT Applications, IEEE INDICON 2022

PROFORMA – 1

"Statement of Originality"

I Hiren Dutta registered on June 2019, do hereby declare that this thesis entitled "Development of Realtime Data Ingestion and Data Management Software Architecture for Internet of Things Applications" contains literature survey and original research work done by the undersigned candidate as part of Doctoral studies.

All information in this thesis have been obtained and presented in accordance with existing academic rules and ethical conduct. I declare that, as required by these rules and conduct, I have fully cited and referred all materials and results that are not original to this work.

I also declare that I have checked this thesis as per the "Policy on Anti Plagiarism, Jadavpur University, 2019", and the level of similarity as checked by iThenticate software is 4 %.

Signature of Candidate: Hiren Dutta

Date: 8/7/25

Certified by Supervisor(s): Paramita Bhaumik 8/7/2025

(Signature with date, seal)

Associate Professor
Dept. of Information Technology
JADAVPUR UNIVERSITY
Block-LB, Plot-8, Sector-3
Salt Lake, Kolkata-700 106, India

PROFORMA – 2

CERTIFICATE FROM THE SUPERVISOR

This is to certify that the thesis entitled "Development of Realtime Data Ingestion and Data Management Software Architecture for Internet of Things Applications" submitted by Hiren Dutta (Regd. No.: 185/19/E), who got his name registered on 06/06/2019 for the award of Ph. D. (Engineering) degree of Jadavpur University is absolutely based upon his own work under the supervision of myself, and that neither this thesis nor any part of the thesis has been submitted for either any degree/diploma or any other academic award anywhere before.

8/7/2025 Parama Bhaumik

Dr. Parama Bhaumik

Associate Professor
Dept. of Information Technology
JADAVPUR UNIVERSITY
Block-LB, Plot-8, Sector-3
Salt Lake, Kolkata-700 106, India

Associate Professor
Department of information Technology
Jadavpur University
Salt Lake Campus, Sector-3, Kolkata- 700106

To Mother Nature …

Acknowledgements

This thesis is the much-awaited outcome of my hard work and many others who have led to the successful completion of my Doctor of Philosophy in Engineering from Jadavpur University. First, I would like to express my honour towards Mother Nature who is the greatest teacher of all. I am thankful for her blessings and the teachings she offered. I would like to extend my sincere thanks to Dr. Parama Bhaumik, Associate Professor, Department of Information Technology, Jadavpur University for giving me the chance to pursue my doctoral work under her guidance. I would like to thank her for continuous support, encouragement and leading me to the successful completion of my thesis. I would like to express my gratitude to my co-workers, who helped me out with several important contributions. Finally, I would like to thank my parents, family, all my friends, and staff members of Jadavpur University who have lent their blessings and helping hands in pursuing my doctoral work.

Hiren Dutta

Abstract

Although the Internet of Things (IoT) is rapidly connecting an increasing number of devices to the internet, it remains a challenge to handle large amounts of sensitive data in a secure and cost-effective manner. Most current IoT solutions rely on centralized infrastructures, which require high-end servers to handle and transfer data, resulting in significant maintenance costs and trust issues. As a result, developing new approaches to decentralizing IoT is an important research problem. Blockchain technology, the underlying technology of Bitcoin, has emerged as a potential solution for creating a truly decentralized, secure, and trust less environment for IoT. However, the integration of blockchain in the IoT poses significant challenges and issues. For example, while designing blockchain-based solutions, there is a trade-off between auditability and privacy. Blockchains inherently provide a publicly verifiable record of all transactions, which can impede user privacy, especially in IoT applications. Additionally, the computing architecture needs to be improved to incorporate unused resources beyond traditional scaling methods. There are multiple personas who can interact with IoT-enabled systems, and each persona has their own use cases and data governance considerations. Therefore, effective information modelling is required to support multiple personas with the same data access levels and a high degree of data governance. In this thesis, a novel software architecture and practical solution based on blockchain technology have been proposed to address the challenge of managing IoT device interoperability and trust management processes. The proposed architecture leverages core self-managed, whitelisted, and decentralized trust management processes to ensure secure and cost-effective management of large amounts of sensitive data. Furthermore, to address the challenge of effective near real-time information processing and computing architecture, an agent-based microservices software architecture has been proposed, which can handle data surges at minimal cost without compromising performance and other non-functional requirements. In addition, a graph-based information modelling and data governance software architecture has been proposed to support multiple personas with the same data access levels and a high degree of data governance. The thesis introduces a decentralized software architecture that integrates privacy-preserving techniques into the blockchain, balancing auditability and privacy. It is designed to be scalable, efficient, and adaptable for various IoT use cases. Experimental evaluations demonstrate its effectiveness in securely managing large volumes of sensitive data while maintaining user privacy and cost-efficiency. The proposed software architecture concepts have been applied to practical industry use cases, which demonstrate its adaptability and potential for future business growth engine, including a newer business model.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Meaning
ABAC	Attribute-Based Access Control
ACE	Authorization in Constrained Environments
AI	Artificial Intelligence
API	Application Programming Interface
AWS	Amazon Web Services
BAN	Body Area Network
BaaS	Blockchain as a Service
CAGR	Compound Annual Growth Rate
CA	Certificate Authority
CBOR	Concise Binary Object Representation
CDN	Content Delivery Network
CoAP	Constrained Application Protocol
CSV	Comma-Separated Values
DAG	Directed Acyclic Graph
DAPP	Decentralized Application
DCA	Data Collection and Analysis
DeFi	Decentralized Finance
DLT	Distributed Ledger Technology
DNS	Domain Name System
DoS	Denial of Service
DSP	Data Surge Protection
DTLS	Datagram Transport Layer Security
ETL	Extraction, Transformation, and Loading
EVM	Ethereum Virtual Machine
GFS	Google File System

Abbreviation	Meaning
HMACSHA	Hash-based Message Authentication Code Secure Hash Algorithm
HTTPS	Hypertext Transfer Protocol Secure
IBFT2	Istanbul Byzantine Fault Tolerance 2
IDP	Identity Provider
IIoT	Industrial Internet of Things
IPFS	Inter-Planetary File System
JWT	JSON Web Token
JDK	Java Development Kit
JVM	Java Virtual Machine
KPIs	Key Performance Indicators
KYC	Know Your Customer
MAPE	Monitor, Analyse, Plan, and Execute
MCC	Mobile Cloud Computing
MEC	Mobile Edge Computing
MQTT	Message Queuing Telemetry Transport
NFC	Near Field Communication
NFRs	Non-Functional Requirements
OSGI	Open Service Gateway Initiative
P2P	Peer-to-Peer
PBAC	Policy-Based Access Control
PLIM	Product Lifecycle Information Management
PoP	Proof-of-Possession
PoC	Proof of Concept
QoS	Quality of Service
QoD	Quality of Device
QoI	Quality of Information
RBAC	Role-Based Access Control
RDF	Resource Description Framework

Abbreviation	Meaning
REST	Representational State Transfer
ROI	Return on Investment
RPC	Remote Procedure Call
RDBMS	Relational Database Management Systems
SAN	Storage Area Networks
SCADA	Supervisory Control and Data Acquisition
SLA	Service Level Agreement
SOAP	Simple Object Access Protocol
SAGA	Simple API for Grid Applications
SINR	Signal-to-Noise Ratio
UI	User Interface
UGV	Unmanned Ground Vehicle
UAV	Unmanned Aerial Vehicle
VoI	Value of Information

Chapter 1

1. Introduction

1.1 Definition of Internet of Things (IoT)

In recent years, the Internet of Things (IoT) has become a highly dynamic technology that has captured the attention of many. The IoT involves connecting physical objects, devices, and systems through sensors, software, and network connectivity, allowing them to communicate with each other and the internet. It consists of four layers: the perception layer, which collects data from sensors and actuators in the physical world; the network layer, which connects devices and systems for data exchange and communication; the middleware layer, which provides tools and services for managing and processing the data collected from the perception layer; and the application layer, which uses data to generate insights and make decisions that can enhance system performance [1]. The IoT has the potential to revolutionize many industries by providing real-time data and insights that can drive innovation, improve efficiency, and enhance decision-making. In the context of a thesis paper, there are several aspects of the IoT that could be explored. For instance, a thesis paper could delve into the technical aspects of IoT, including communication, security, access control, and data management. Alternatively, the potential applications of IoT in various sectors such as agriculture, transportation, financial services, and telecom could be investigated. Additionally, another angle could be the ethical and social implications of IoT, including concerns about privacy, ethics, and the potential impact on society and employment. The IEEE IoT Initiative defines the Internet of Things (IoT) as a field that brings together multiple technological and social domains [2]. It consists of interconnected devices that gather sensory data and automate localized and large-scale systems comprised of geographically dispersed subsystems. The IoT has found uses in various industrial sectors and has been expanding globally due to the introduction of wearable devices, improved Internet access, and the affordability of embedded computers [3].

The section depicts IoT as a transformative technology, describing its layers, applications across industries, and rapid global growth fuelled by wearable devices and

better internet access. This expansion highlights IoT's potential to reshape industries and everyday life through real-time data and insights. However, like any fast-evolving technology, traditional IoT systems face significant challenges that hinder their widespread adoption and effectiveness. Despite the promise IoT holds, it is essential to address the key obstacles that limit its full potential. The next section will examine these challenges, particularly issues of scalability, standardization, and security that complicate the smooth and reliable implementation of IoT solutions.

1.2 Challenges in Traditional IoT

In recent years, the Internet of Things (IoT) has been expanded at an impressive rate, leading to the transformation of businesses and the enhancement of productivity and connectivity in daily life. However, the effectiveness, scalability, and security of traditional IoT software design are constrained by several problems [4]. *Limited scalability* is one of the main issues with traditional IoT software design. Traditional IoT software architectures are limited in their ability to expand effectively and meet the needs of extensive IoT deployments. The architecture of the system is designed to work with a specific set of devices; when more devices are added, the system becomes less effective. Performance problems, an increase in latency, and decreased reliability may result from this. The absence of standards presents another difficulty. Because the IoT software architecture cannot be standardized, developers are unable to create scalable, interoperable IoT solutions. Due to this lack of standardization, connecting IoT devices from diverse vendors is especially challenging. Because of this, developers must invest more time and money into creating unique solutions for every vendor, which can result in extended project timeframes and higher project expenses. Another major issue with typical IoT software architecture is *security* [3][5]. Traditional IoT software design is prone to cyberattacks due to its weak security features. IoT devices are typically easy targets for hackers, who can then use them to launch additional network attacks once they have them. Data leaks, network outages, and income loss may result from this. Privacy, security, and control are major concerns when deploying IoT devices. Serious privacy concerns are raised by the inherent nature of interconnected devices that acquire and exchange vast quantities of personal and sensitive data. Frequently, users are unaware of the data being collected and how it is being used, which can lead to

misconduct or unauthorized access. In addition, weak authentication mechanisms, unpatched vulnerabilities, and inadequate encryption protocols can compromise user privacy by exposing sensitive data. Additionally, difficulty is, traditional IoT software architecture's *reliance on specific closed loop cloud services*, sometimes operations are bounded by specific cloud vendor tied up with the IoT device manufacturer with proprietary protocols. Traditional IoT software architecture usually makes use of cloud-based services for data processing and storage. This reliance on cloud services may cause latency to increase and reliability to decrease. Costs could go up as a result, as could performance and customer satisfaction. Traditional IoT software design frequently has limitations when it comes to its ability to provide *interoperability* between systems and devices made by different suppliers. With the rapid proliferation of IoT technologies across various industries, the lack of standardized protocols and communication frameworks has hindered seamless integration and collaboration between different devices and platforms. The absence of interoperability not only hampers efficient data sharing and device coordination but also limits the scalability and potential of IoT ecosystems. IoT firmware interoperability is a significant challenge that arises due to the diversity of devices, manufacturers, and software platforms within the IoT ecosystem. Firmware serves as the embedded software that controls and manages the functionality of IoT devices. However, compatibility issues can arise when devices from different manufacturers use proprietary firmware that may not be easily compatible with other devices or platforms. This lack of interoperability hinders seamless communication, data exchange, and collaboration between devices, limiting the full potential of IoT deployments. Another potential issue is the absence of *data management standardization*. Because IoT devices usually use different data formats, it is challenging to integrate them into a cohesive system. Data quality issues may occur. IoT data often exhibits characteristics such as heterogeneity, inconsistency, incompleteness, and noise, which impact its quality and reliability. Ensuring data quality is crucial for making informed decisions and deriving accurate insights. It requires robust data governance practices, data cleansing techniques, and quality assurance measures. Additionally, effective data management strategies, including data integration, storage, and processing, are necessary to handle the high influx of data and ensure its accessibility, reliability, and security. IoT systems with *centralized control* face serious issues that demand attention as well. One of the main issues has to do with a single point of failure. Relying on a central control infrastructure or server provides a

vulnerability because any interruption or compromise might bring down the entire Internet of Things network. In essential applications like healthcare or manufacturing, where downtime or failures could endanger lives and infrastructure, this dependency could have serious repercussions. The infrastructure may be unable to handle the growing volume of data and commands from an expanding number of IoT devices, which is another drawback of centralized control. Delays, inefficiencies, and impaired real-time decision-making may follow from this. In addition, centralization poses privacy and security issues because a single authority that receives sensitive data becomes a major target for security breaches or unauthorized access. Decentralized designs, distributed decision-making, and the implementation of strong security measures can all assist to reduce these problems and promote a more resilient and secure IoT ecosystem. *Information transparency* is a significant challenge in IoT deployments, stemming from the complexity of data flow and diverse formats within interconnected devices. Users often lack visibility into how their data is collected, used, and shared, leading to concerns about data ownership and potential misuse. Furthermore, transparency around data privacy and security practices is crucial to building trust, as users need to understand how their personal information is protected and who has access to it. Addressing these challenges requires standardized decentralized access-controlled mutable information storage, clear privacy policies, and robust security measures to ensure transparency and empower users with a better understanding of their data in IoT ecosystems.

1.3 Decentralized Ledger Technology (DLT) and Blockchain

Distributed ledger technology (DLT) is a type of innovative technology that allows for the secure and decentralized storage and transfer of data, assets, and value. DLT is often used interchangeably with the term "blockchain," but there are several different types of DLT, including permissionless (open) and permissioned (closed) networks, as well as variations like directed acyclic graphs (DAGs) [18]. DLT is often seen as a key component of the "Web 3.0" or "decentralized web" movement, which aims to create a

more open, transparent, and democratic internet that is less reliant on centralized platforms and intermediaries. While DLT is commonly known for its role in enabling cryptocurrencies like Bitcoin and Ethereum, its potential applications go beyond that. The concept of DLT originated with the development of Bitcoin in 2009, which introduced the concept of a trust less, decentralized network powered by a blockchain. A blockchain is a specific type of distributed ledger that employs encryption to store data securely and transparently in blocks, which are then linked together in a chain. Each block in the chain contains a record of transactions or data, and once added to the chain, it cannot be altered or deleted. This ensures that the blockchain is an immutable and tamper-proof database that can be used for various purposes. The decentralized nature of DLT is one of its most significant advantages. Unlike traditional systems that rely on intermediaries like banks or governments to manage transactions, DLT enables direct peer-to-peer transactions among network participants. This results in *reduced costs and improved efficiency* since intermediaries are no longer necessary. Another significant advantage of DLT is its *transparency*. Since the ledger is distributed among all network participants, everyone has access to the same information. This makes it easier to track and verify transactions, thereby increasing accountability and reducing the potential for fraud or corruption. DLT can also improve security by using cryptographic techniques to *encrypt* data and verify transactions. The decentralization and distribution of the ledger among all network participants make it more challenging for hackers or malicious actors to compromise the network. Furthermore, the *immutability* of the blockchain ensures that once a transaction is added to the ledger, it cannot be altered or deleted, providing an additional layer of security. DLT has the potential to be applied to various fields beyond cryptocurrencies, such as supply chain management, voting systems, and healthcare data management. For instance, in supply chain management, DLT can be used to monitor and verify the movement of goods, increasing transparency and reducing the potential for fraud. In voting systems, DLT can enhance the security and transparency of the financial process, minimizing the risk of fraud or interference. In insurance data management, DLT can securely store and share policy rules data, enhancing privacy and security. However, DLT also has its limitations. One of its most significant limitations is *scalability*. As the number of participants on the network grows, so does the size of the blockchain, making it more challenging and time-consuming to process transactions. Additionally, the *energy consumption* needed to maintain the blockchain can be substantial, with some estimates

suggesting that the Bitcoin network alone uses as much energy as a small country. DLT also faces challenges with regards to regulation and adoption. As a novel and evolving technology, there is still a *lack of regulatory* clarity around its use, primarily in areas like cryptocurrencies. Furthermore, there may be resistance to adoption from traditional institutions and stakeholders who may lose power or revenue from the disruption of the status quo. Distributed ledger technology has the potential to revolutionize the way data and assets are stored, transferred, and managed. Its decentralized, transparent, and secure nature offers numerous advantages over traditional systems, but it also presents challenges concerning scalability, regulation, and adoption. As DLT continues to evolve, it is vital to carefully consider its potential applications and limitations, as well as the ethical and social implications of its use.

1.4 Internet of Things and Blockchain

The combination of Internet of Things (IoT) and blockchain technology holds immense promise for businesses and society. The IoT refers to a network of interconnected devices that share data and can automate tasks, while blockchain technology provides secure and transparent data storage. A more powerful and secure network of devices can be created by leveraging both technologies. The primary benefit of integrating IoT and blockchain is increased security, as the distributed ledger technology makes it difficult for malicious actors to compromise the network. Additionally, the transparent and tamper-proof nature of blockchain technology increases accountability and trust in the network. Another advantage is the potential for new business models and revenue streams, as blockchain-enabled smart contracts can automate peer-to-peer transactions and reduce the need for intermediaries, which can lower costs and increase efficiency. Using blockchain in IoT can also enable new business models and revenue streams. For example, blockchain-enabled smart contracts can automate the process of transferring ownership or value between IoT devices, creating new opportunities for peer-to-peer transactions and reducing the need for intermediaries. This can help reduce costs and increase efficiency in various industries. One of the most significant potential applications of IoT and blockchain technology is in supply chain management, where IoT sensors can track product movement and blockchain can store and share data about

the products, increasing transparency and reducing fraud. However, there are challenges to this integration, including scalability and the complexity of integrating IoT devices with blockchain technology. As the number of IoT devices on the network grows, so does the size of the blockchain, which can lead to slower transaction processing. Additionally, energy consumption can be significant, and there may be technical barriers to integrating the two technologies. The combination of IoT and blockchain technology can revolutionize industries and society, but challenges must be addressed to realize its full potential. By increasing security, transparency, and efficiency, the integration of these technologies can pave the way for a more secure and trusted digital future.

1.5 Motivation

Motivation drives research on IoT interoperability, security, trust, centralized system administration, and information transparency. The rapidly growing Internet of Things requires in-depth study and innovative solutions. Advancements in the field are sought, with the goal of establishing a more efficient, safe, and transparent IoT ecosystem through ongoing research efforts. IoT interoperability is driven by the recognition that its actual potential rests in connecting and integrating diverse devices and systems. Resolving interoperability difficulties allows devices to communicate, share data, and collaborate. This allows transformative applications in healthcare, transportation, manufacturing, and agriculture. Interoperable communication, frameworks, and architectures are designed to enable dynamic and scalable IoT ecosystems. Security and trust management are essential Internet of Things research areas. As devices connect, IoT systems become more complicated and vulnerable. Unauthorized access, data breaches, and security breaches threaten individuals, companies, and infrastructures. New methods, authentication systems, and software designs must be developed to protect data in IoT environments. My study aims to establish comprehensive trust establishment framework that instil confidence, minimize threats, and protect sensitive data to promote IoT adoption. Centralized IoT systems are studied because they can introduce single points of failure, scalability constraints, and performance constraints. Edge processing and decentralized decision-making may increase IoT deployment

efficiency, robustness, and fault tolerance. Decentralization spreads computing resources and reduces central authority, improving IoT network performance and adaptability. My study develops scalable, decentralized system integration that improves cooperation, responsiveness, and IoT device integration. Information transparency is also actively researched. As data is considered the lifeblood of the Internet of Things, ensuring visibility and control over its collection, usage, and sharing is regarded as essential. Transparency is seen as a key factor in addressing concerns related to data ownership, privacy, and potential misuse. This research seeks transparency, standardization, and user-centric privacy controls for data exchange. User empowerment and explicit data governance frameworks promote trust, privacy, and responsible data practices in IoT environments. In this field, researchers can perform cutting-edge research and make a lasting impact. This can motivate students and researchers to improve the field and leave a lasting impression. Researching these technologies' ethical and social effects is another draw. As IoT and decentralized systems become increasingly common, I must address their ethical and social impacts. Researching these difficulties allows individuals to propose innovative solutions. IoT and decentralized software architecture research is highly motivating and gives us the chance to change the world. Experts who can find creative answers to real-world problems are in demand as the sector evolves. This subject allows researchers to work on interdisciplinary projects, build open-source software, and address ethical and social issues associated with these technologies. It also allows for world-changing work in academia and industry. Individuals can help advance cutting-edge technology and society.

1.6 Scope of The Thesis

1.6.1 Design of Identity and Trust Management Solution Architecture for Internet of Things (IoT)

This section focuses on developing a robust architecture for identity and trust management in IoT systems. It covers core components such as unique identity creation, cryptographic security, authentication (e.g., biometric, 2FA), and role-based access control mechanisms (RBAC, ABAC, PBAC). The architecture aims to address scalability and device heterogeneity by integrating blockchain-based decentralized

identity systems [8] and AI-driven anomaly detection [6][7] to enhance trust, privacy, and system resilience.

1.6.2 Design of Data Ingestion Solution Architecture for Internet of Things (IoT)

This section presents a scalable and layered architecture for real-time data ingestion and processing in IoT environments. It addresses the roles of the physical, network, and cloud layers, emphasizing low latency, edge computing, and secure data transmission [16][17]. Additional focus is placed on interoperability, data governance, and security frameworks using encryption, intrusion detection, and policy controls to ensure integrity, performance, and regulatory compliance in large-scale IoT systems.

1.6.3 Design of Information Representation Modelling and Management Architecture for Internet of Things (IoT)

This section outlines a conceptual and architectural model for information representation and management in IoT. It discusses the development of flexible data models, unified data integration, and efficient data storage and analytics, including the use of AI/ML algorithms and visualization tools [15]. The design also emphasizes data security, privacy, and stakeholder engagement to ensure alignment with organizational goals and regulatory standards.

1.6.4 Design of Decentralized Internet of Things (D-IoT) Architecture

This section proposes a decentralized architecture for IoT that enhances system scalability, security, and resilience. It integrates peer-to-peer networking protocols [9][10][11][12][14], blockchain for secure transaction records, and distributed computing frameworks like Apache Spark and Hadoop [13]. The use of edge computing further supports low-latency processing and data localization. This architecture is positioned as a robust alternative to centralized models, with careful consideration of governance, privacy, and deployment challenges.

This thesis presents a unified architectural approach to building secure, scalable, and intelligent Internet of Things (IoT) systems through the integration of four interrelated components. First, *identity and trust management* establish the foundation for secure interactions by ensuring robust authentication, authorization, and trust among heterogeneous IoT entities. Building on this, the *data ingestion architecture* provides a scalable framework for real-time data collection, transmission, and processing across

physical, network, and cloud layers, supported by edge computing and strong data governance. The *information representation and management architecture* then organize and contextualizes the ingested data using semantic modelling, integration tools, and analytics to derive actionable insights. Finally, the *decentralized IoT architecture* consolidates these layers into a resilient and autonomous system using blockchain, peer-to-peer networking, and distributed computing, eliminating centralized bottlenecks while enhancing privacy, scalability, and system robustness. Collectively, these four components define a comprehensive, end-to-end IoT solution architecture capable of addressing the core challenges of security, interoperability, data management, and decentralization.

1.7 Objectives of the Thesis

The objective of this thesis is to design and propose a comprehensive, scalable, and secure architectural framework for the Internet of Things (IoT) that effectively addresses the core challenges of identity and trust management, real-time data ingestion, semantic information modelling, and system decentralization. As IoT ecosystems grow in complexity and scale, traditional centralized models struggle to ensure secure, interoperable, and efficient operations across billions of heterogeneous devices. To overcome these limitations, the thesis aims to (i) develop a robust identity and trust management system that enables secure authentication, authorization, and trust evaluation among diverse entities; (ii) construct a high-performance data ingestion architecture capable of handling massive real-time data flows with low latency and high reliability; (iii) establish a flexible information modelling and management architecture that ensures semantic consistency, data integration, and effective analytics; and (iv) implement a resilient decentralized IoT architecture leveraging blockchain, peer-to-peer protocols, and edge computing to enhance privacy, scalability, and fault tolerance. By integrating these four layers into a unified solution, the thesis seeks to enable the development of next-generation IoT systems that are not only secure and interoperable but also adaptive to the demands of emerging applications in smart cities, healthcare, industry, and beyond.

1.8 Organization of The Thesis

I organize the chapters in a logical order to ensure the coherence and progression of ideas in my thesis. By doing this, readers can easily follow the research progression and understand my arguments, as each section builds upon the previous one, reinforcing the overall thesis. This approach not only enhances readability but also aids in presenting a compelling argument by allowing readers to trace the development of my research without confusion. Furthermore, it helps me to maintain a clear focus throughout the writing process, systematically addressing each aspect of my research to ensure that nothing is overlooked. This structured method is crucial for creating a thesis that is well-organized, coherent, and persuasive, making it easier for readers to follow my research, comprehend my arguments, and appreciate the organization of my work.

Chapter Title	Description
Literature Survey on Internet of Things Software Architecture	In this Chapter 2, a concise overview of the prominent studies conducted to date in the domain of software and systems architecture for the Internet of Things is provided. The survey is categorized into distinct areas of research that have been explored.
IoT Identity, Whitelisting and Decentralized Trust management Software Architecture	In this Chapter 3, a concise overview is presented of a novel software architecture that provides self-managed, reusable, and interoperable asset whitelisting and trust management features for IoT. This architecture proves to be highly advantageous, particularly for large-scale IoT networks, where managing asset auto whitelisting and trust can become a difficult and heterogeneous task. The proposed architecture's self-managed, reusable, and interoperable approach helps to improve the security and reliability of IoT networks while simultaneously reducing operational overhead and complexity.
IoT Privacy, Information	This study in Chapter 4 addresses issues related to privacy, access control, permissions, and transparency in information

Transparency and Access Management Software Architecture	transmission, all while maintaining decentralization. My approach and proof of concept help build trust in a decentralized solution architecture.
Computing and Data Ingestion Software Architecture for Managing Unexpected Surges in Data Volume	The proposed mechanism in the Chapter 5 aims to provide real-time control of dynamic data in IoT applications through an auto-rebalancing strategy. The mechanism provides an effective and efficient solution for real-time dynamic data control in IoT applications, which can lead to improved performance, reduced downtime, and increased productivity.
Data Modelling, Management and Data Governance Software Architecture for Internet of Things (IoT) Applications	Graph based data governance and information modelling has been proposed in Chapter 6 to integrate real time data. This approach involves using graph structures to represent data and relationships, which enables the creation of a flexible and extensible data model. By using this model, data governance and information management can be improved, allowing real-time data to be integrated and analysed more effectively.
Distributed Ledger Technology (DLT) based Software Architecture for IoT Industrial Applications	Chapter 7 introduces a decentralized IoT software architecture and application approach grounded in the principles of decentralized networks. In this model, each device can communicate directly with other devices without relying on a central server. As part of this research, I have developed a series of exploratory applications based on use cases from various industry domains. These applications demonstrate how this decentralized concept can spur industry growth and enhance the operations of existing businesses.

Table 1. 1 Organization of Thesis

Chapter 2

2. Literature Survey on Internet of Things Software Architecture

Over time, research has increasingly focused on managing all IoT interactions and systems due to the exponential growth of Internet of Things (IoT) entities. These IoT devices come in a wide range of sizes, processing speeds, storage capacities, and application scopes because of their numerous dimensions. These modern smart technologies can gather, processing, and making decisions in real time without involving any humans. For a variety of industrial applications, a single reference communication architecture is not very suitable. This survey study focuses on comprehensively researching previous work and mapping it into the reference layered IoT architecture that will be developed. This architecture can be balanced depending on the application's requirements and how it will be used. Additionally, this section discusses potential problems with IoT architecture, communication, and data management. Companies having oil pipelines, water pipelines, and electrical wire pipes distributed over millions of kilometres; occasionally, these pipelines reach deserts and the ocean, two environments in which it is practically impossible to perform routine periodic maintenance. When a healthcare system is dealing with a pandemic and conducting data analysis on millions of people, it is challenging to keep records of each individual patient. Drilling devices can penetrate the earth to depths that are hundreds of feet below the surface. IoT-based data analysis is the only method that can successfully track and take preventative action for the maintenance of turbine blades that are swinging hundreds of feet above sea level or aeroplane engines that are running at altitudes that are thousands of feet above sea level. Both situations occur at extremely high altitudes. Processing of data in real time and taking preventative action are both necessary components for some of these use cases. Some of them might be willing to put up with a delayed decision, which would then be followed by a scheduled routine of preventative maintenance. The design, the stacking of numerous levels, and the cross-cutting challenges that arise when attempting to manage IoT connectivity,

storage, management, security, interoperability, and other nonfunctional requirements are the components that are shared by all use cases (NFRs). To carry out a methodical analysis of end-to-end Internet of Things design and administration, the sections of this survey report have been divided into four important sub-sections. These sub-sections are focused on i) IoT Identity management, whitelisting, and authorization of IoT elements. ii) The architecture of edge computing solutions for the Internet of Things, which focuses on administration, processing, and communication at the edge. iii) Information management architecture for the Internet of Things with a primary emphasis on data management, data structure modelling, and storage to facilitate effective Internet of Things communication iv) A decentralized industrial Internet of Things communication architecture, which focuses on the communication architecture of potential industrial use cases in the future.

2.1 Survey on IoT Identity, Whitelisting and Decentralized Trust management Software Architecture

Identity management in these heterogeneous systems is becoming one of the most important areas for regulatory authorities to be concerned about as the number of IoT devices increases exponentially [19], ranging from sensor devices to gateways to multiprocessor IoT devices. Identity management is a method and technology combination that provides secure access to the appropriate degree of information and resources while safeguarding device individual or group profiles. Identity management in the IoT era must be able to recognize gadgets, sensors, and monitors and control their access to sensitive and non-sensitive data. The interactions of IoT devices can be categorized as those between a human and a device, a device and another device, or a device and a service or system. Identity protection for all these types of device interactions must be possible with identity and access management for IoT devices. The IoT IDM also needs to consider several dimensions. The fundamental elements of it are the exponentially increasing number of devices, low cost, limited resource devices,

standardization of IoT devices, and mobility of the devices. It's time for academics to provide solutions to device identity management issues since IoT devices are being implemented in a variety of fields, including public health, smart grids, smart transportation, waste management, smart homes, smart cities, agriculture, and energy management, among others. Many IoT identity management systems have been proposed in recent years. A safe authorization and authentication system was proposed by Timothy Claeys, Franck Rousseau, Bernard Tourancheau, and Hannes Tschofenig to address the primary identity management problems. They suggested ACE (Authorization in Constrained Environments) [20] device authentication using OAuth 2.0 tokens. Instead of using a http-based approach, it employs CoAP (Constrained Application Protocol) and a CBOR (Concise Binary Object Representation) encoded message. Requests made by the device to access any server resources are forwarded to the identity provider. Using the various oAuth2.0 grant type specifications, the device must authenticate itself against the Identity provider. When the device has been authenticated, the IDP (Identity Provider) releases the token so that it can access the resources. CoAP messages are safeguarded by ACE using secure DTLS and object security. Additionally, it introduces the Proof-of-Possession (PoP) token, a symmetric or asymmetric key pair produced at random by IDP. Andri Warda, Mahendra Data, and Adhitya Bhawiyuga Similar token-based concepts were put out by [21] and use JWT-based token exchange for MQTT-based [22] communication in IoT devices. Publisher, subscriber, MQTT broker, and token authentication server are the four essential parts of the proposed design. To obtain a token for either a fresh token generation or an expired token generation and save it locally, the publisher or subscriber must first provide its login and password to the authentication server. The HMACSHA 256 technique is then used to certify self-contained messages sent to the JWT (JSON Web Toolkit) server. A cloud-based approach is suggested by Sivaramon et al. [23] to safeguard IoT devices with security flaws by dynamically controlling firewall rules for accessing the private network. The number of Device authentication and Application layer Identity management of various IoT applications, such as smart homes, near field communication and RFID, wireless sensor networks, etc., was examined by Mohammed El-hajj, Ahmad Fadlallah, Maroun Chamoun, and Ahmed Serhrouchni [24]. Most of the time, these are reciprocal authentication protocols that rely on oAuth 2.0-based authentication, PKI, hardware signatures on devices, digital signatures, and key-based authentication. While these protocols and authentication systems continue to

be the backbone of the Internet of Things, next-generation IoT solutions shouldn't rely on centralised trust and single points of failure because they could be vulnerable to hacker attacks. Later in 2019 Rafael Martnez-Peláez and others [25] contested the concept and demonstrated the vulnerability of these authentications. To improve security, they suggested a new version that considered the login, mutual authentication, and key agreement phases. To demonstrate the user and server's engagement, they also offer verification of the connection attempt sub-phase. The revised plan meets the security demands while having reduced execution costs. The user registration process, cloud server registration process, login process, and authentication procedure make up the phases of this system. Through a secure link, the control server (CS) registers the user. Give the user smart cards as a response from CS. Through a secure link, Cloud Server is also registered, and Cloud Server receives access parameters from CS. When a user wishes to access a resource during the login process, the user submits a login request to the cloud server, which is then routed to the control server (CS) for user verification. In the Authorization step, the Cloud Server exchanges a session key with the user to provide access to various resources following mutual agreement between the User, Cloud Server, and CS. SungJin Yu, KiSung Park, and YoungHo Park later in 2019 shown in their proposed work [26] that a parameter stored in a smart card might extract the user identity even if it was claimed by the original article. This work included mechanisms for user registration, login and authentication, and the phase where passwords are changed. To implement safe mutual 3-factor authentication in an IoT cloud context, they used BAN (Burrows-Abadi-Needham) logic. Joon Young Lee, et.al [27] suggested a three-factor authentication mechanism for a multi-gateway Internet of Things environment based on a similar idea. They implement the notion at the gateway level [Figure 2.1], which manages smart buildings, smart homes, smart offices, etc.

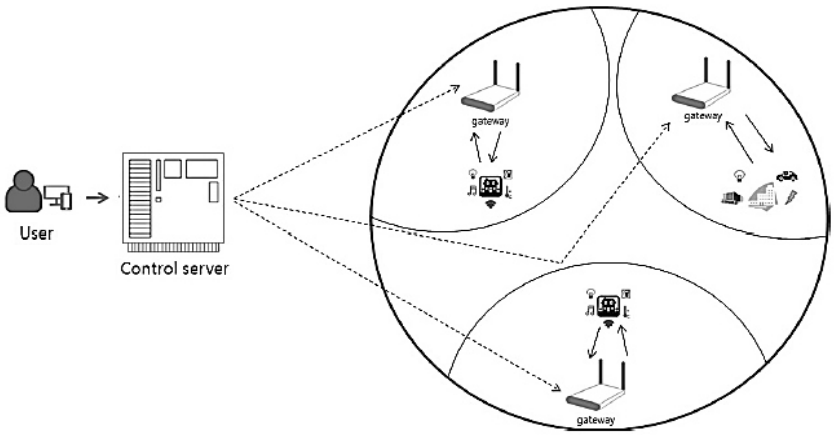


Figure 2. 1 Three-factor authentication protocol for multi-gateway IoT environment proposed in [27]

[24][25][26][27] The authentication component of IoT authentication solutions is driven by centralized control or a control center. The proposed algorithms closely resemble oAuth 2.0's fundamental idea. Relying on the central party is one of these methods' biggest drawbacks. The entire system will be at risk if the control center is compromised. Second, from a maintenance standpoint, managing a distinct entity that is used just for user/device authentication is an additional burden. Recent advancements in blockchain technology have led to the development of a few device identity management schemes that take decentralized trust management in a new direction. Below are a few screenshots of these schemes. A creative use of the blockchain protocol was put out by Diego M. Mendez Mena and Baijian Yang [28] to secure the edge of the home network and IoT devices. The purpose of this study is to describe a blockchain-based gateway that acts as a gatekeeper to separate legitimate actors from fraudulent ones when they attempt to access resources from a private network. The gatekeeper will decide whether to permit traffic through it using the data supplied on a defined smart contract, which can only be amended by pre-established users or machines. Whitelist rules are generated for that device and fed into it once the information has been authenticated. When the device requests access again in the future, the blockchain network-based gatekeeper confirms the device's identification. A magnificent blockchain-based multilayer architecture for IoT authentication was presented by Mohamed Tahar Hammi, Badis Hammi, Patrick Bellot, and Ahmed Serhrouchni [29]. The execution steps are shown in Figure 2.2. The related items in phase (A) can come from different categories (medical, industry, environment, etc.). The initialization step

is represented by phase (B), during which the Group Master selects a group identifier (groupID). Each group member's item has the master's signature on it. The bubble is produced at the blockchain level during phase (C), which occurs when the group is created. The Master sends a transaction including both his own unique identifier and the unique identifier of the group he wants to form. The blockchain verifies the groupID and the objectID of the Master are both unique. The bubble is produced if the transaction is legitimate. The ability to create a bubble is unrestricted on the public blockchain. Phase (D) involves the Followers sending transactions one at a time to be connected to their respective bubbles. The smart contract examines the validity of the Follower's ticket using the public key of the bubble's Master after first confirming the uniqueness of the Follower's identification (objectID) at the blockchain level. The object cannot be connected to the bubble if one of the requirements is not met. After the follower completes the initial transaction in the association stage, further verification is not needed, and message exchange can begin immediately.

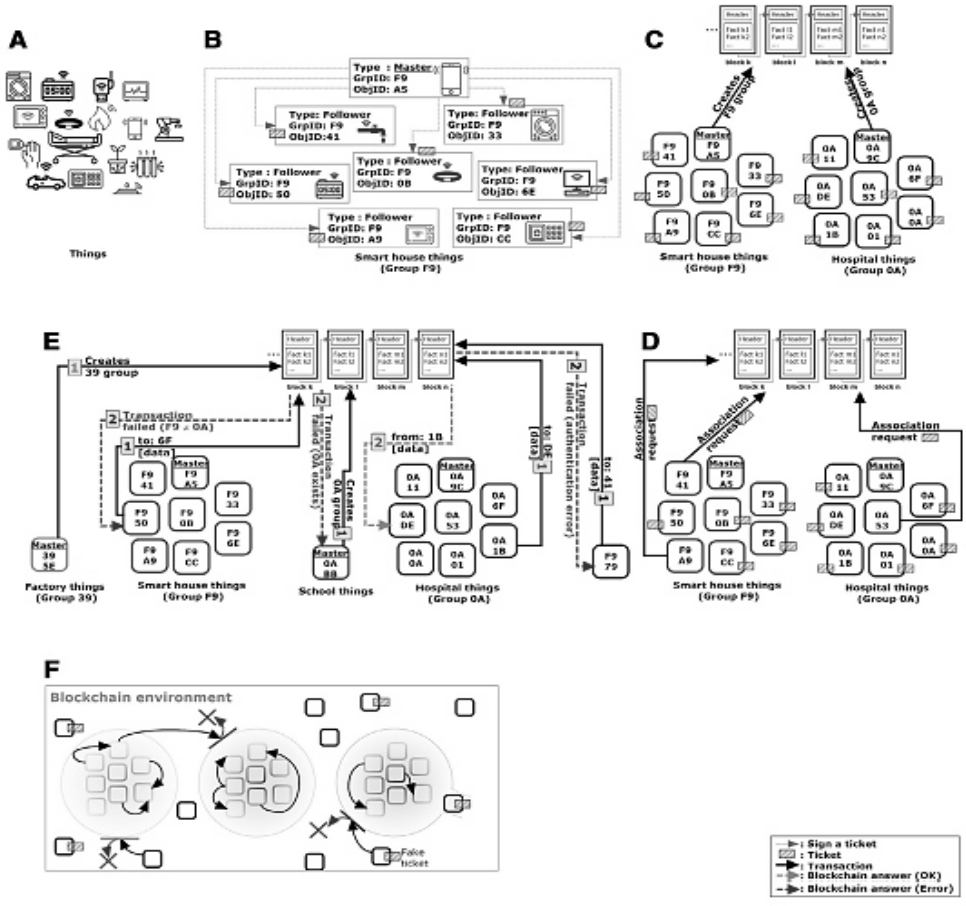


Figure 2. 2 Bubble of trust mechanism proposed in [29]

Samuel Marchal and others presented the autonomous distribute system (AuDI) as a wonderful method of discovering and classifying different IoT device types [30]. In contrast to the direct human-interacted registration process, he presented a method for passively fingerprinting the periodic communication traffic of IoT devices in his research. It doesn't require any prior device type information or label training data to effectively identify an IoT device in any mode of operation.

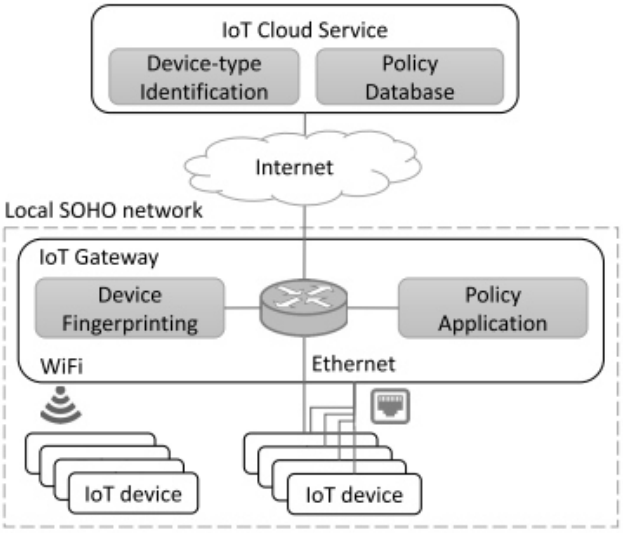


Figure 2. 3 AuDI system model [30]

By using smart contracts to deploy for manufacturers and device owners, Atomoni [31] has made another admirable contribution to blockchain-based identity management. Every time Manufacturer created a new item, it had to be whitelisted by a blockchain record. The phases of a. registration, b. activation, and c. validation are the services that blockchain makes available to the public. A new manufacturer member and its blockchain address are written to the blockchain during manufacturer setup under the control of a smart contract during the registration phase. When a manufacturer adds new devices to the whitelist via the interface, it also controls the posting of new device IDs to the blockchain. Recently, Yuichi Hanada, Luke Hsiao, and Philip Levis [32] proposed using blockchain technology to enable machine-to-machine communication and the development of mutual trust. They suggested using M2M communications to automate gas refuelling. This blockchain-based idea suggests automated gasoline purchases between a car and a gas station when a vehicle needs to refuel its gas. This solution relies on implicit trust management between a vehicle and a gas station and assumes that a group of vehicle entities and a group of gas stations are participating

entities of the same blockchain. It also relies on the consensus algorithm of the underlying blockchain network and pays little attention to identity management of participating entities during initial registration and during interactions. Nita, S.L., and Mihailescu, M.I. has proposed a solution introduces an elliptic curve-based authentication mechanism tailored for IoT devices [107]. Employing a blockchain network ensures the preservation of key blockchain properties, including immutability, decentralization, and heightened security during query authentication. The system architecture encompasses crucial elements such as trusted authority, owner, data users, IoT devices, blockchain network, and storage server. Implementation involves smart contracts deployed by the data owner, enabling IoT devices to locate the appropriate node within the blockchain network for query authentication. Bilinear pairing is employed in the authentication process, with the time complexity contingent on the hash function and bilinear pairing. Comparative analysis establishes the proposed mechanism's security against common attacks targeting IoT devices. While the mechanism demonstrates strong performance and security, practical challenges and unaddressed implementation considerations may exist. Moreover, exploring the real-world scalability and interoperability with existing IoT systems and standards remains a potential avenue for further investigation and consideration. Yousefnezhad, N. et al., proposed security architecture caters to the Internet of Things (IoT) and product lifecycle information management (PLIM) [108], focusing on impeding unauthorized access and enforcing access restrictions based on user roles. It seamlessly integrates and orchestrates the IoT ecosystem, adeptly handling the unique security needs of products and clients across various smart city scenarios and product lifecycle stages. The architecture emphasizes security in both user and device domains, prioritizing authentication and access control for users and robust device identification. Offering a unified interface for product-related data, it promotes software component reuse, facilitating rapid prototyping and innovative data utilization. However, the default access control module's specific permission model and interface may lack compatibility with existing tools. While device-side security is largely automated and requires no device alterations, the architecture might not comprehensively fulfil the security demands of physical products and their virtual counterparts throughout the entire lifecycle.

Vasudev Dehalwar et. al., has proposed a solution for the challenges in the distributed environment for energy management using blockchain technology include the lack of

approved standards in device identification and authentication, leading to interoperability and scalability issues [109]. Additionally, heavy-duty transaction processing for authentication, consensus, and smart contracts may require high-energy systems and lead to delays in adding new blocks to the core block. The proposed solution involves the use of blockchain technology with a focus on cyber security sensitivity and energy data privacy and security within electrical energy networks that incorporate IoT devices for cleaner energy technologies. This approach aims to address the challenges and contribute to the intelligent operation of distributed energy networks. The proposed solution has limitations related to the lack of approved standards in device identification and authentication, which may lead to interoperability and scalability issues. Additionally, the heavy-duty transaction processing for authentication, consensus, and smart contracts may require high-energy systems and lead to delays in adding new blocks to the core block. These limitations may impact the efficiency and scalability of the blockchain-based trust management and authentication system in the smart grid.

2.2 Survey on IoT Privacy, Information Transparency and Access Management Software Architecture

The Internet of Things (IoT) landscape presents a double-edged sword. While it unlocks a treasure trove of opportunities for innovation and efficiency, it also exposes a vast attack surface riddled with vulnerabilities. Hewlett Packard's sobering statistic highlights the gravity of the situation: a staggering 70% of IoT devices are susceptible to attacks due to weaknesses in password security, key encryption, and granular user access rights [124]. Fortunately, researchers are actively exploring solutions to fortify IoT security and privacy. One promising approach gaining significant traction is blockchain technology. Marko Arac et al. proposed a blockchain-based method that integrates security and privacy functionalities for IoT devices [124]. Their approach hinges on a security interface designed specifically for IoT devices, coupled with an IP mapping technique. While the core idea utilizes blockchain to manage trust within a closed, decentralized network [125], the proposed mechanism operates in a centrally

managed environment [126]. This raises concerns about the scalability and feasibility of achieving consensus and establishing robust working principles in a distributed, multi-party context. Further research by G. Shi et al. delves deeper into the design strategies for blockchain-based IoT security [127]. Shantanu Pal et al. offer a comprehensive analysis of the challenges, benefits, and limitations associated with leveraging blockchain for access control in IoT environments [128]. Their work sheds light on recent trends and the burgeoning need for blockchain solutions to address privacy concerns in auto-controlled IoT information sharing, event triggers, and access control. They emphasize the importance of addressing implementation complexities through distinct design and deployment architectures. V. Agarwal et al. propose a side-chain-based architecture for secure IoT data transfer [129]. This approach utilizes detachable sidechains, where most transactions occur offline, connecting to the main chain through sidecar smart contracts. While this design decentralizes immutable transactions on the blockchain for enhanced security, the sidechain's offline layer shoulders the burden of performing the most resource-intensive tasks. Additionally, ensuring seamless merging across numerous sidecars with distinct routes maintained by various stakeholders presents a logistical challenge. Furthermore, any transaction recorded on the main net loses its privacy veil. To address latency and hierarchy issues, M. Ma et al. suggest a fog computing-assisted, blockchain-based distributed key management architecture [130]. This method employs a user access control and protection algorithm, but requires offline execution. Q. He et al. propose storing student credit information on a private blockchain to prevent tampering [131]. However, this approach necessitates maintaining a separate private network, essentially storing non-private data on a private blockchain, thereby undermining the core tenets of decentralization and privacy inherent to blockchain technology. Industrial IoT integration with blockchain is explored by Shubham Joshi et al. [132]. Their research primarily focuses on outlining the general architecture and potential of blockchain in industrial IoT, acknowledging limitations concerning security, privacy, and access control. A clear differentiation between permissioned and permissionless blockchains in the context of IoT systems is presented by L. Peng et al. [133]. While transparency and decentralization are hallmarks of blockchain, they also pose challenges in securing user privacy, highlighting the critical need for a practical and implementable approach to privacy preservation within blockchain-based IoT systems. K. Gai et al. present a blockchain-based internet edge paradigm that leverages the Ethereum network to offer

a privacy model [134]. Their proposed BIoE model aims to efficiently utilize blockchain techniques in task allocation while adhering to three key design objectives: edge-based IoT job allocation, privacy preservation, and tamper resistance. Smart contracts facilitate communication between processes executed by edge nodes. While the blockchain immutably stores task records and state changes, all Ethereum nodes have the ability to track transactions and smart contracts execute them. This persistent visibility of transactions on the blockchain network raises privacy concerns. Y. Lu et al. propose a blockchain-based architecture for IoT data sharing [135]. Their approach involves leveraging machine learning to generate a model that executes intended functions once data is recorded on the blockchain. Since data is neither retained on the chain nor within the model, it safeguards data knowledge and privacy. However, this intricate approach utilizes the blockchain primarily as a data storage unit, potentially compromising on-chain data security, as anyone with access to the mainnet can view chain data blocks.

The integration of blockchain technology within the IoT ecosystem presents a promising avenue for enhancing security and privacy. However, significant research efforts are still required to address challenges related to scalability, privacy preservation within decentralized networks, and the practical implementation of complex architectures. A multifaceted approach that combines the strengths of blockchain with robust security practices, user-centric design principles, and a clear legal framework is essential to unlock the full potential of a secure, transparent, and user-centric IoT future.

2.3 Survey on IoT Data Ingestion Software Architecture

It is extremely difficult for solution architects to build a single reference solution architecture for IoT device provisioning and data ingestion given the expanding number of IoT devices, which range from small sensors to limited resource-owned gateways. At the base level, the sensors gather the data. The next step is to configure this data in a useful way. It's called provisioning. The provisioning of IoT devices consists of two main parts. To monitor something, a device must first be configured, and then sensors or devices must upload data to a gateway or server. Device-provisioning interfaces can be integrated with the gateway, controller, or even directly at sensor nodes, depending

on how the IoT interacts with the platform. The installed firmware on gateways and controllers is responsible for managing device provisioning and data ingestion to the server. Different solution architectures may be used depending on the nature of the use case and the requirements at the edge. It is now highly fascinating to research the tools, technology, architecture, patterns, and related difficulties connected with IoT data intake [33] [34] due to the development of cloud-based platform services, fast data ingestion pipelines, and dedicated databases. It is high time for researchers to suggest various solution strategies to address issues that have arisen because of the nature of interactions, network connectivity, the development of novel communication protocols, and the increasing speed of data input. P.P. Ray detailed the key IoT architectural elements, protocols, and cloud solutions, as well as application domains and domain specific IoT architectures, in his survey study report titled A Survey of IoT Architecture in 2016 [35]. Device, communication, services, management, security, and applications are the major functional building blocks of the Internet of Things (IoT). These building blocks are described as being dynamic and self-adapting, self-configuring, interoperable communication protocols, having a unique identity, and integrating with information networks. Later, in early 2017, Dong Wang, Sooyong Lee, and others proposed a system for provisioning IIoT devices with zero human intervention. [36]. When an Internet of Things (IoT) device is connected to a network, the process begins with an AP scan. The device establishes a connection to the server for authentication based on the appropriate AP. The device gets provisioned on the server after authentication. The suggested approach calculates each AP's signal-to-noise ratio (SINR) value and selects the one with the highest SINR value to increase connection stability for subsequent communication. The new approach to selecting the best network communication channel for communication is demonstrated in this research. In their research article [37] from 2019, Tosiron Adegbija, Roman Lysecky, and Vinu Vijay Kumar described the essential elements of edge computing. They demonstrated how processing may be done even closer to the data source on the fog devices. The integration of the application system and resource limitations make it difficult to move the computation closer to the data source. The functionalities of rightly provisioned IoT edge computing include communication, control (actuation), sense (through various sensors), and compute. Keeping in mind the trade-off between energy consumption and performance, the author has examined the difficulties in properly provisioning for compute tasks, which are divided into data collection, data processing, information and

data storage, and data transmission. The difficulties of content offloading from an edge system to a remote cloud environment are addressed by Pavel Mach and team in 2017 [38]. The study discusses the compromise between MCC and MEC (mobile edge computing) (mobile cloud computing). To achieve high QoS, operations that need low latency, low jitter, hard real-time limits, and hard real-time context event processing should be carried out close to the data source (Quality of Service). Decentralized IoT design, which overcomes NFR (non-functional requirement) concerns around interoperability, scalability, and adaptability, was proposed by Jozef Mocnej et al. in 2018 [39]. Current IoT environments frequently exhibit poor performance from centralized structures. The author has suggested a decentralized architecture that can enhance the above NFRs by monitoring the platform's performance during runtime, guaranteeing the quality of output, and optimizing the use of available resources. Quality of Device (QoD), Quality of Service (QoS), and Quality of Information (QoI) measures are performed to monitor the IoT platform in real-time. To achieve the required output quality, the value of information (VoI) is retrieved by correlating the functions of relevance, integrity, timeliness, and understandability.

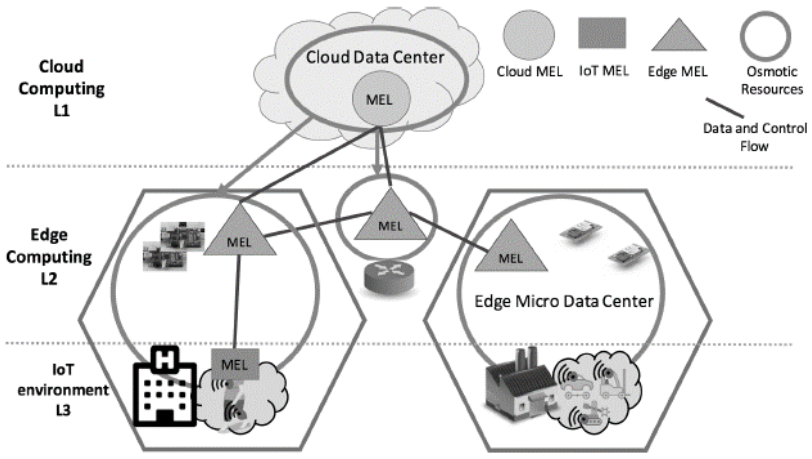


Figure 2. 4 MEL deployment architecture in [40]

A unique architecture was developed by Lorenzo Carnevale et al. in 2019 [41] to enable osmotic computing for distributed multi agent systems in BAN (Body Area Network) applications. The research paper's proposed Osmotic Agent architecture and logical flow are shown below.

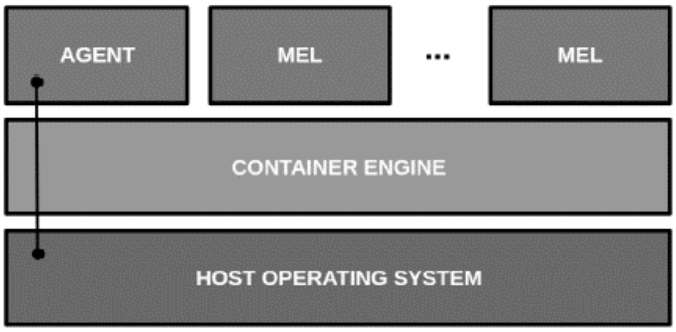


Figure 2. 5 Osmotic computing high level architecture in [41]

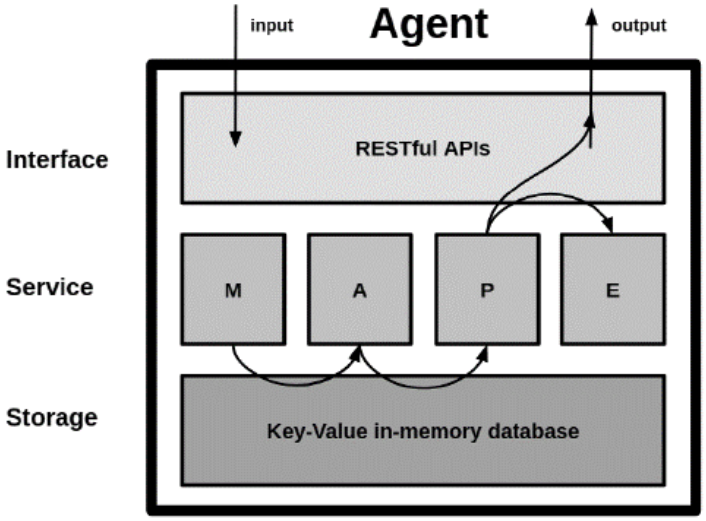


Figure 2. 6 Osmotic computing flow diagram in [41]

RESTful APIs [22] are included in the interface layer so that desired functionalities can be used. This facilitates the development of MEL and its movement between the cloud, edge, and IoT. As a message broker and in-memory database, the storage layer supports MAPE functionality and the pub-sub interface (monitor, analyse, plan, and execute). the service layer, which houses the microservices list and uses an orchestration mechanism to get MAPE functionality. Although dynamic MEL deployment using osmotic computing can aid in dynamic resource allocation inside MEL graphs, the osmotic agent must continuously monitor all deployment conditions along the chain. This could be expensive in a resource-constrained context since it necessitates a constant CPU processing cycle, even when the data transmission and communication networks are reliable. The key architectural pattern [42] [43] for achieving application scalability and modularity through functional deconstruction at the edge layer is the microservice. Since each domain-driven service has its own boundary, storage

(polyglot persistence), strong cohesion, and loose coupling, a modular and scalable application is considerably simpler to create and maintain. Lightweight communication, freely deployable software, minimizing centralized service management, and independent development approaches and technologies are the main objectives of IoT regarding microservice architecture. In 2016 [44], Bjorn Butzin et al. presented a microservice architecture based on SOA for an IoT environment. The hexagonal architecture parts leveraging the microservice technique for IoT were also suggested by the authors. With the containerized micro service solution method, applications can overcome limitations and dependencies with the underlying operating system and hardware device drivers [45]. A magnificent graph-based approach to express the load interdependence among microservices was proposed [46] by R. Yu et al. in 2019. The authors proposed a simple, fully polynomial-time technique for balancing QoS-aware IoT microservices. Point-to-point connectivity and greater complexity compared to monolithic design are the main problems with microservices-based architecture. It is difficult to work with many databases and simultaneous transactions in a distributed system environment. Additional complexity will be brought on by the implementation of cross-cutting issues at the edge and microservice chassis [47]. Reactive IoT is a model of programming [48] where messages are sent asynchronously and non-blocking between services and subsystems. Workflow definitions might be static. The runtime, on the other hand, decides how to schedule those activities based on the amount of core and processor resources available. According to a 2019 [49] proposal by Haidong LV, Xiaolong Ge, and others, an IoT Application Platform built mostly on monolithic architecture and using thread approaches to handle mass concurrent IOT devices with high performance requirements. Actor- and Akka-based solutions were developed to address this problem and get around the bottlenecks in monolithic and thread-based applications. With its high-level abstraction, Akka facilitates the development of distributed, asynchronous, high-performance real-time systems by doing away with low-level ideas like threads, deadlocks, and exclusions. The actor model, utilizing concurrency and fault tolerance, realizes it. Each actor corresponds to a sensor or IoT device. The IoT devices use messages to communicate data to their mapping agent over the Akka-MQTT connector. The message is received asynchronously by the actor, who processes it without blocking. The suggested reactive design for the IoT platform is shown below.

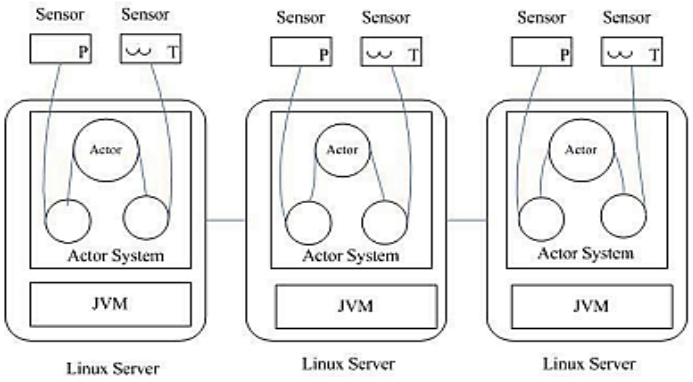


Figure 2. 7 Cluster architecture of IoT platform proposed in [49]

Cleber Jorge Lira de Santana et al. suggested an OSGI (Open Service Gateway Initiative) based reactive micro service based IoT architectural platform in 2019 [50]. The reactive APIs are defined using Vert.x. The platform architecture and the core of reactive microservice interactions are described here.

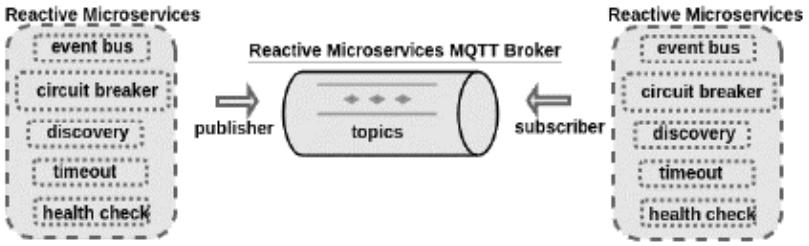


Figure 2. 8 Reactive microservice core and their interactions proposed in [50]

As service binding occurs dynamically at runtime through an internal service registry lookup, OSGI-based IoT containers offer runtime flexibility. Since the underlying operating principle of OSGI-based containers is so closely coupled with the OGI kernel, portability is a significant difficulty. Any external jar (Java Archive file) must first be transformed into a functional bundle before it can be transferred into the kernel. The entire application bundle stack is substantially heavier to deploy due to the larger kernel itself. The production environment is made more difficult by the complexities of bundle resolution at runtime. Because of this, OSGI has the potential to introduce runtime application modularity; nevertheless, it has not yet gained widespread adoption in production environments for application development. Kiourtis, A. et al. (2023) [110] propose a method for Data Ingestion and Prioritization in the context of IoT, particularly focusing on heterogeneous IoT medical devices. Their process involves

integrating these devices with healthcare platforms, transferring data to preferred Cloud Storage services, and prioritizing significant data segments for relevant stakeholders. Although the study concentrates on the healthcare sector, further research is necessary to generalize its applicability to other domains. Naghib, A. et al. (2023) [111] conduct a comprehensive comparison of big data mechanisms within IoT, covering various aspects such as processes, architectures/frameworks, quality attributes, and analytics types. While their review consolidates existing knowledge, it refrains from introducing novel mechanisms. Huang X, et al. (2021) [112] develop and implement a system utilizing a data lake architecture for managing IoT data. However, the paper lacks in-depth details regarding scalability and performance across diverse workloads. Gkonis P et al. (2023) [113] propose a solution that focuses on optimizing network design for IoT devices, leveraging edge servers for data processing, and addressing challenges related to protocol design, security, and resource allocation. This strategy, termed the IoT-edge-cloud continuum, anticipates future networks like 6G by exploring recent advancements and underscoring deployment challenges and potential limitations. Limitations primarily revolve around overcoming challenges associated with protocol design, security, scalability, and resource allocation within the IoT-edge-cloud continuum. Additionally, the integration of advanced technologies, such as those anticipated in 6G networks, may introduce further complexities that require attention. Bixio L et al. (2020) [114] introduce an adaptive microservices architecture tailored for IoT platforms, enabling real-time stream processing at both edge and cloud levels. This solution abstracts underlying complexities and offers flexibility through service definition, rule-based queries, and combinator languages adaptable to various platforms. However, the proposed solution's limitation lies in the constrained expressiveness of stream processing rules, which favours a more flexible deployment model. This limitation stems from predefined and manageable templates for streaming rules, enabling dynamic allocation and composition but constraining the range of rule definition possibilities.

2.4 Survey on IoT Data Modelling, Management and Data Governance Software Architecture

Edge devices generate enormous amounts of data in important systems like manufacturing, healthcare, large-scale supply chains, transportation, and related verticals. Modern edge devices, firmware, and gateways may filter and analyse a small amount of data before transmitting it over the network, thanks to recent advancements in high-speed, low-footprint device capabilities. However, most data are still offloaded into the server due to resource-intensive processing over time governed by middleware platforms [51]. The backend storage and processing system is put under additional strain when statistics are generated over a long period of time to be used in machine learning-based decision-making systems. Because of this, it's crucial to describe and store IoT data effectively so that data intake and data querying may be completed quickly and affordably. In 2019 [52], Mayank Patel et al. from the distributed databases group suggested potential uses of raw data processing for streaming data applications. It controls data in unstructured files to query the application. The first developed raw data processing system to be proposed in 2012 was called NODB. In 2018 [53], Holm Smidt et al. suggested an architecture for modelling, connecting, and controlling IoT devices that considers the dynamic nature of IoT data management. The tracking, management, and exponential scalability of IoT devices are all addressed by this graph-based approach. The foundation of graph databases is graph theory. The IoT device's nodes and edges depict it, along with its connectivity to other node parts. By utilizing a web-oriented application framework, this solution addressed a smart grid application and achieved concurrent, multi-user support through the Web, uniform access to heterogeneous embedded devices, and acceptable performance, to mention a few. The integration layer provides telemetry and control and is made up of several smart devices. The routing control and graph database make up the service layer. The view and application for user interaction are provided by the presentation layer. The same architecture is shown below. The focus of implementation is placed on the service layer, which supports the graph backbone.

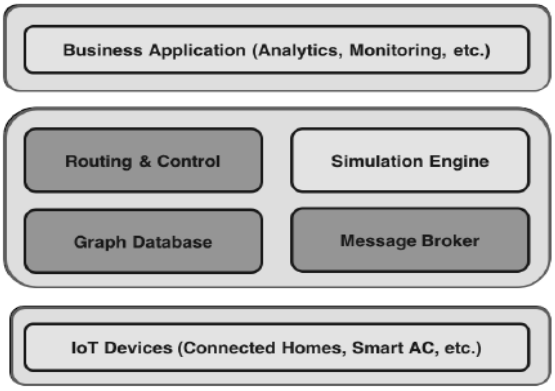


Figure 2. 9 Layered software architecture [53]

Based on the functional domain below figure depicts the illustration of 3 nodes connected by three relationships.

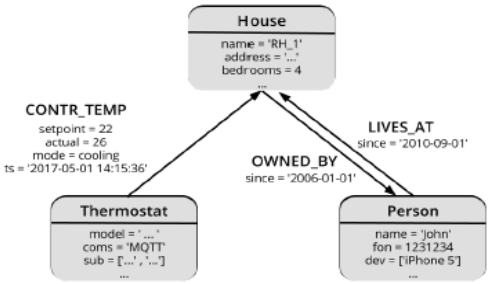


Figure 2. 10 Property graph illustrations [53]

The standardisation of this method is its main drawback. Using various models, different verticals represent their own collection of uniquely vertical use cases. The sharing of cross-vertical IoT models will make it difficult to integrate controls and data. Smart cities, transportation planning, and autonomous vehicles are just a few cross-vertical use cases that are suffering from a lack of uniformity in modelling, information sharing, dynamicity, and endpoint security. Scalability issues with data intake from industrial equipment sensors are a constant problem. The rate of data ingestion is quick, the number of parameters that must be ingested is greater, the analysis time cycle is relatively brief, and time series are designed to prompt action or alarm. The most important factor in establishing an end-to-end channel is the choice of data intake platform, services, products, and architecture. A comparison of various IIoT time series-based database technologies has been suggested by Sergio Di Martino et al. in 2019 [54] to make a judgement regarding their adaptation. The top three databases chosen by the research team to store time series data are Influx DB [55], Mongo DB, and Cassandra. A system called Castor for industrial data intake, context stores, and model stores was proposed by Bei Chen et al. in 2019 [56]. The workflow of the Castor framework loads the time series data, transforms it, trains the model, and then stores the model after scoring it in the model store. Castor uses RabbitMQ as its messaging fabric and Apache OpenWhisk as its architectural framework. Parallel execution is

started by the model scheduler to score and retrain the model. The incoming data flow is handled by the RabbitMQ message bus using a variety of data handlers, including timeseries, models, and external interfaces. Applications are kept on the IBM cloud.

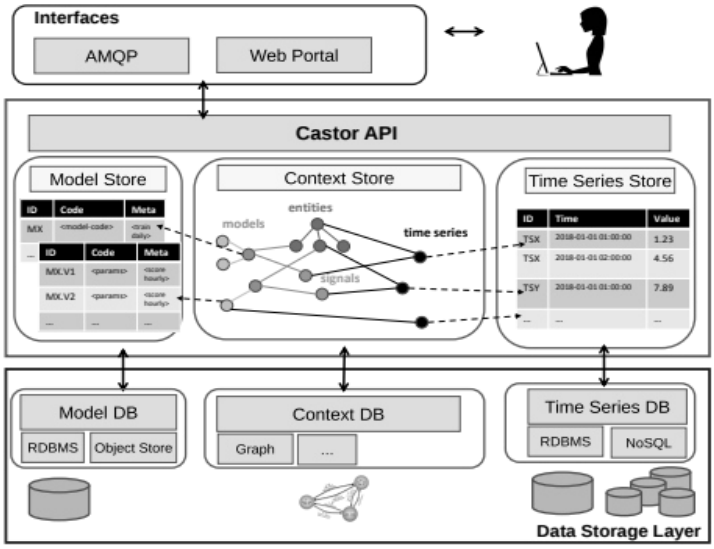


Figure 2. 11 Castor time series and model data management component diagram [56]

To deal with problems related to huge volumes of structured time series data, Eugene Siow et al. introduced TritanDb in 2018 [57] under the Apache 2.0 license. In this study's analysis of the structure of publicly available IoT data, most of the time series data are structured, flat, wide, and numerical. On top of providing the best storage, TritanDb provided an RDF (Resource Descriptor Framework)-based data architecture for maximum interoperability. The semantic web is established using the graph-based query language SPARQL, which is comparable to SQL. To address the shortcomings of the above solution, researchers Harshit Gupta, Zhuangdi Xu, and colleagues proposed the DataFrog platform in 2019 [58]. In this extension, the researchers proposed a geo-distributed data management platform at the edge of the network to meet the need for smart services at the IoT edge. A message-driven data ingestion architecture was presented by Michael Nolan et al. in 2019 [59] to transmit data points to databases (HBase) from the edge with reliability. Researchers suggested message-oriented middleware based on Kafka that sends data through an Apache Strom cluster using a pub/sub architecture [60]. Based on the suggested congestion control technique, Apache Storm disbursed the processing burden, notifying the source of data ingestion to limit the rate in case congestion exceeded the threshold at the sink.

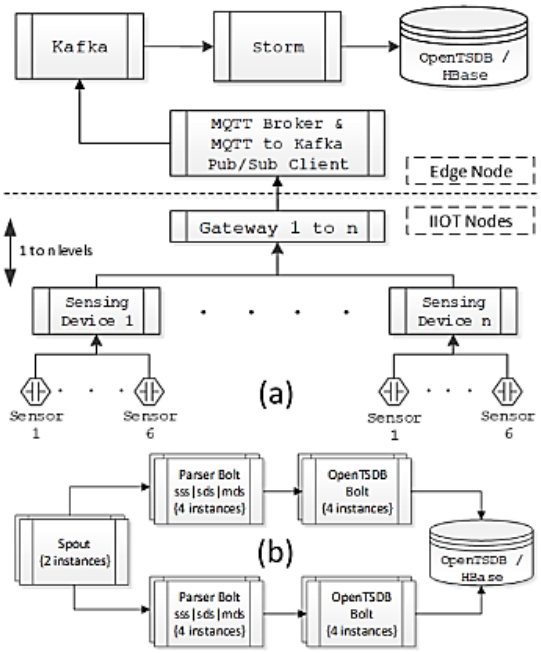


Figure 2. 12 a) Configuration b) Storm topology in [59]

Data gathering architecture from the manufacturing shop floor and push into the AWS cloud for data storage and predictive analytics have been proposed by Wenjin Yu, et al. in 2019 [61]. Older, proprietary manufacturing systems typically provide data elements through OPA DA (e.g., SCADA, DCA unit). The OPC DA data element is transformed into a human readable OPC UA data element by the OPC UA adaptor proxy. After first human readable transformation, OPC Collector (OPC Client) collects additional OPC UA data and pushes it to the cloud.

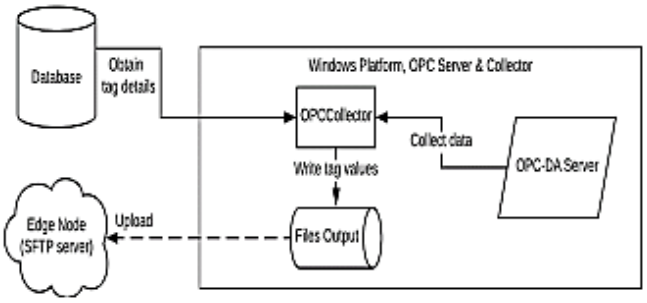


Figure 2. 13 Diagram of the big data ingestion framework in [61]

2.5 Survey on DLT based Software Architecture for IoT Applications

The Internet of Things (IoT) field known as autonomous IoT (A-IoT) is one in which a single IoT thing or a collection of IoT things carry out specific individual tasks to accomplish a group objective devoid of a centrally controlled controller. The intelligent objects could interact with their surroundings or other things around them to communicate autonomously and update their state in a way that advances their goal. Because this set of things operates in a confined environment without centralized control, the study of their communication architecture is crucial. Autonomous Internet of Things: Small groups perform and accomplish local goals to fulfil the broader aim. By employing statistics, machine learning, and optimization algorithms, autonomous IoT actively handles information and decisions on behalf of consumers in the early stages. The extent to which autonomous systems will be designed to support daily tasks will depend on whether users are prepared to delegate those duties to IoT systems. Most A-IoT application architectures enable two different types. In the first scenario, when things are given decision-making authority, environment state data is communicated via a centralized server. In the second scenario, there is no centralized state management, and all state information is transferred point to point or through the environment. The two different architectural facets that must be adhered to define the foundation of such an A-IoT architecture are algorithm-based and heuristics-based. Aidan Fuller et al. highlighted in their 2019 paper "Digital Twin: Enabling Technology, Challenges, and Open Research" [62] how in the industry 4.0 age. A-IoT architecture and the concept of a "digital twin" allow a computerized model of a physical object to operate on its own and make advanced decisions. To stabilize the condition of KPIs, which identify system anomalies, data can flow in either direction from physical to virtual or vice versa. To govern a real-time physical system, a virtual system uses operational parameters produced by a physics-based algorithm. This paper attempts to address the main research issues surrounding digital twins. An automated surface mount technology-based digital twin solution for energy optimization was put up by Neha Karanjkar et al. in 2018 [63]. The line is equipped with associated sensors

(proximity, vibration, energy, etc.) to record the key operating characteristics for calculating the energy usage and machine-wise activity. Data has been gathered, and insights have been drawn from it to suggest a buffering-based solution to increase the production line's energy efficiency and its effects based on a digital twin. For complex assets like industrial motors and turbines, Gómez Berbis et al. suggested a framework to store digital twin information in a knowledge graph in 2019 [64]. It has been argued that enterprise knowledge graphs (EKG) or simple knowledge graphs (KGs) are necessary to support digital twins because semantic technologies can be used to formalize the domain of the digital twin and make it easy to integrate with other semantic domains. A methodical approach to structuring the creation of the digital twin and its data-driven service value was offered by Dominique Heller et al. in 2019 [65]. The group has put forth the bare minimum requirements for a digital twin idea that includes the services and product lifecycle. The following elements provide the bare minimum needs for a digital twin concept: connectivity, defined data structures, sensors that detect a status, and a user interface. At the SoS level, one potential improvement is the throughput time, which can be increased with the use of digital twin simulations. A specified data structure is the third condition that must be met. This indicates that the type of data that is gathered and the processing methods are both well stated. The user interface, often known as visualization and operation, is the fourth point. Over the course of the full product life cycle, the digital twin may and should offer value. During the design phase, a virtual prototype can be created. Simulations can be used to optimize production using a virtual prototype. Predictive maintenance services can be offered as value-added services using the information gathered. The suggested architecture is shown below.

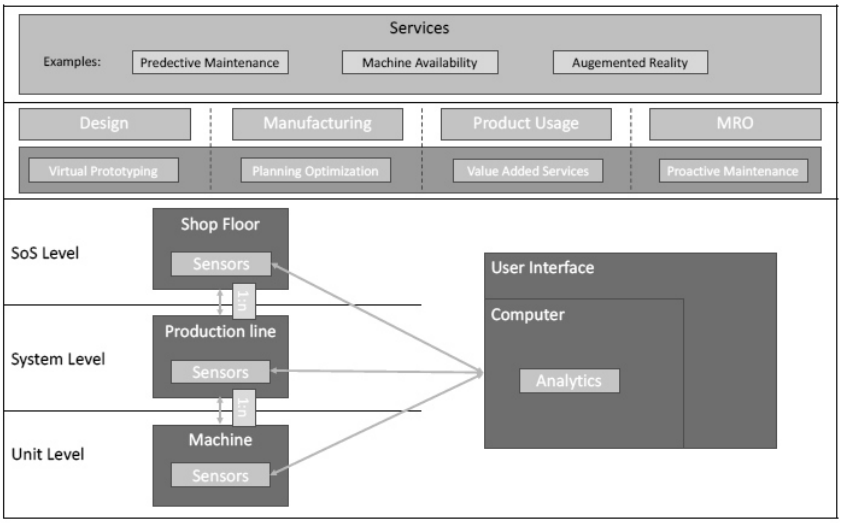


Figure 2. 14 Layered architecture [65]

In 2019, Vinicius Souza et al. proposed [69] a manufacturing vertical IoT-based digital twin solution architecture concept [66] [67] [68]. In his proposal, Souza identified three key structural elements: the physical twin, the manufacturing processes, and the industrial communication protocols. Internal Server, the computer that operates the Digital Twin and simulations, and IIoT Gateway, the gateway offering cross-communication between the physical twin and Internal Server through IIoT devices and wired connections, provide connections inside and outside the business. The suggested architecture is shown below.

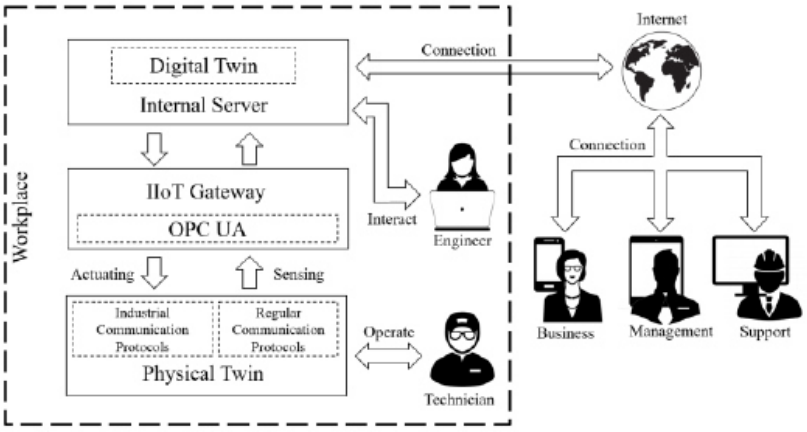


Figure 2. 15 Concept diagram of the Digital Twin architecture [69]

Amilcare In order to create a drone camera-based surveillance system using a lightweight MQTT-based communication protocol, Francesco Santamaria et al. have suggested [70] a decentralized machine-to-machine (M2M) communication

architecture. In the solution, edge and mist computing are used to create a scalable architecture.

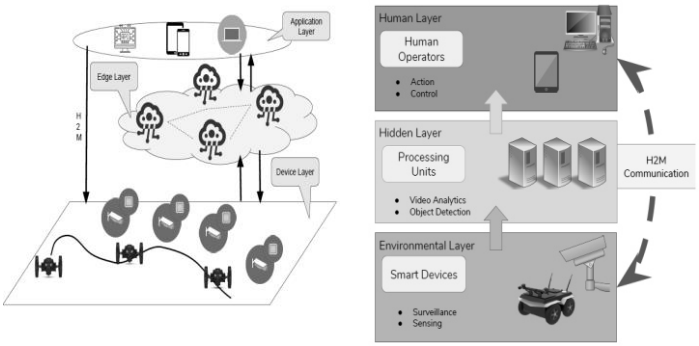


Figure 2. 16 Layered decentralized reference architecture

In a study on Swarm Robotics that Giandomenico Spezzano proposed in 2019 [71], the author developed the NFR criterion and certain fundamental capabilities for swarm robots (IoT components) to interact with the system and environment. In 2015 [72], Laisa C. P. Costa et al. proposed a very pertinent research study on swarm architecture, which is the fundamental building block for network construction and execution using cloud computing and cyber-physical systems. By putting the architecture in the cloud, scalable and flexible systems can arise and work together to accomplish a shared objective. A distributed framework that combines the Universal Data Plane, Data Plane, and Control Plane is called "SwarmOS." The central distributed storage system of the Universal Data Plane To achieve the necessary performance requirements, the data plane controls the runtime execution of the different services that make up the swarm. The control plane controls resources, SLAs, and service level agreements (QOS).

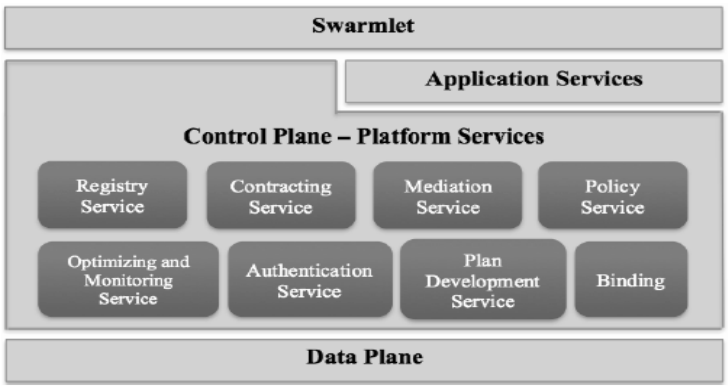


Figure 2. 17 Control plane architecture

Khac-Hoai an architectural paradigm for the future generation of intelligent transportation systems (ITS) was put up by M K, Priyan et al. in 2018 [73] and focuses on the dynamic decision-making of smart connected vehicles based on swarm intelligence (Ant Colony Optimization). The following problems with the intelligent vehicle transport system were addressed by this method: I must select the best architecture for communication between linked vehicles to share and coordinate information. ii) An intelligent algorithm that enables IoT-enabled vehicles to make wise decisions. iii) Using various scenarios to model and simulate the transportation system Laisa an architecture for an automated personal assistant that meets the challenges of interfacing with IoT was proposed by Costa, Laisa et al. in 2019 [74]. It incorporates NLP and systematic integration of personal data from many sources on top of the Swarm platform [72]. The architecture is made up of three basic parts, the main one being the communication agent, which handles user interaction and natural language processing. Natural language processing (NLP) is used in the Swarm Assistant's current iteration for both speech and text. This agent offers a cross-platform mobile web interface as well as a backend that translates verbal intents into actual Swarm network orders. Personal database: to keep track of one's location, interests, preferences, and relationships to build an ontology; crawler: to gather all pertinent data from various sources. The suggested architecture is shown below.

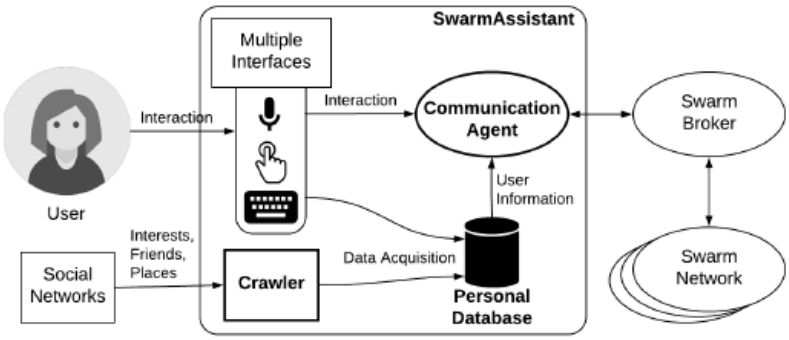


Figure 2. 18 Architecture of swarm assistant in [74]

In their research, Kang, Seongju and colleagues have suggested a MQTT-based context-aware autonomous system to offer autonomous service for MQTT-based devices [75]. The suggested system implements an asynchronous request/response strategy using the MQTT pub/sub messaging pattern to achieve service discovery. By

utilizing semantic web technologies to recognize the context, the gateway may connect with devices on its own. The rule execution time is the system's main drawback. As there are more devices, there will be more rules, which means that as there are more devices, more rules must be executed at once, raising maintainability and slowing down performance over time. This strategy, though, will be effective for a fixed or smaller number of devices. An autonomous communication architecture [76] has been developed by Abenezer Girma and others to be used in disaster-affected areas to help first responders find victims and sources of hazards by observing the environment. A framework was proposed by the researcher to make it simple for first responders working in a disaster region to collaborate with unnamed aerial vehicles (UAVs) and unnamed ground vehicles (UGVs). Easy interactions are made possible using an effective MQTT [22]-based M2M communication protocol. First responders can oversee the operation from a distance thanks to a cloud-based remote-control station (RCS). The system's centralized management and the devices' inability to communicate with one another to create a clustered zone are its limitations. The entire system's processing power is transferred to the cloud, which may cause a delay for systems that require hard real-time computing.

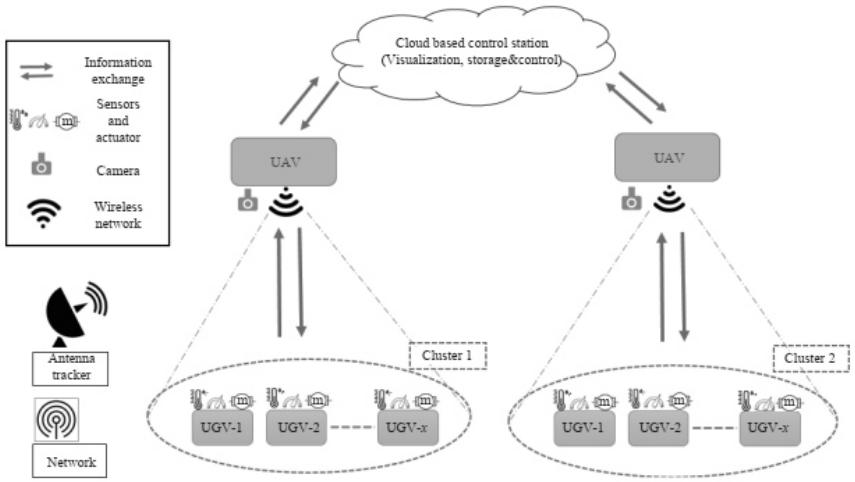


Figure 2. 19 System architecture, IoT based autonomous system [76]

Recent research relevant to industry has explored topics such as decentralized finance, sustainability, and decentralized Industrial Internet of Things (IIoT) aspects. Mingyu et al. (2023) [115] introduce a pioneering privacy-preserving parametric insurance framework that relies on succinct zero-knowledge proofs (zk-SNARKs). Their objective is to streamline underwriting and claim processes on a blockchain while

safeguarding user privacy. However, the inherent openness of blockchain platforms and decentralized access control issues may potentially compromise user privacy. In a similar vein, a paper published in 2021 [116] delves into the utilization of blockchain and smart contracts within the insurance sector. It explores avenues for decentralization and RegTech solutions, seeking to enhance efficiency and regulatory compliance. Matthias Nadler et al. (2022) [117] propose a comprehensive decentralized insurance protocol based on smart contracts for decentralized finance (DeFi). Their protocol addresses gaps present in existing DeFi insurance models, striving to bolster security and reliability within decentralized financial ecosystems. Shoufeng Cao et al. (2023) [118] put forward a blockchain-enabled architectural framework designed to ensure trustworthy communication regarding the sustainability attributes of food products. By harnessing blockchain-based traceability, consumers gain access to verifiable evidence, thereby enhancing supply chain transparency and reliability. This research offers valuable insights into sustainability communication empowered by blockchain technology, benefiting both academic research and industry practitioners, especially within the context of Industry 4.0. Luyun Zhao et al. (2024) [119] highlight the critical role of effective recharge facility planning in sustaining E-bike sharing programs, with battery-swapping technology showing promise. However, the efficiency of recharging swapped batteries remains an underexplored aspect. Their study introduces multi-decentralized swapping (M-DS) to enhance recharging efficiency and proposes a modelling framework for optimizing facility planning. Despite demonstrating significant efficiency gains, widespread adoption of M-DS hinges on overcoming barriers related to infrastructure implementation and user acceptance. Heeß, P. et al. (2024) [120] present a solution aimed at verifying sustainability claims within global supply chains by introducing Digital Product Passports. These passports facilitate data sharing among stakeholders while ensuring privacy and addressing concerns regarding data disclosure. However, a limitation arises from the necessity for stakeholders to willingly share relevant data, which may pose challenges in terms of privacy and data sovereignty.

2.6 Conclusions and Future Research Directions

2.6.1 IoT Identity, Whitelisting and Decentralized Trust management Software Architecture

Since its inception, device authentication has mostly relied on cryptographic algorithms, and self-authentication occurs frequently inside a small group or inner circle of devices. Once a device has been validated, the organisation generally believes it to be reliable. In client-service-based communication systems via the internet, digital signatures provide an additional degree of authentication. The production and verification of digital signatures take a sizable amount of time, which is a drawback of employing them in an IoT setting. Therefore, communication speed will decrease with frequent message exchanges. The private key must also be kept in a secure manner. If the user must regularly change the private key, further complexity may develop. In that situation, identifying outdated messages could be a major problem. A smart card-based next-generation authentication architecture has been offered as a solution to these problems. Based on user and server registration, the control server handles the user authentication portion. In exchange for access to the server's resources, the user and the control server mutually agree to share the session key. The concept is being increasingly standardised, and a single sign-on architecture based on oAuth 1.0 and eventually oAuth 2.0 is suggested for cloud-based dispersed environments. Every resource access in this case needs to be verified; alternatively, the user or device must verify their identity with the identity provider using their username and password. A token is given to the user or device to access the resources for a short time based on the grant type. The main difficulty with this strategy is that the IDP (Identity Provider)/Control server is the single point of failure. All resources won't be available if IDP fails. IoT devices with little computing power are unable to handle the complexity of token management and communication. Further JWT-based authentication is suggested to reduce communication complexity, but this approach is not well suited for Internet of Things (IoT) devices due to their sophisticated algorithm execution and encryption logic. Centralised trust management is a problem in every situation because if the control

server or IDP is compromised, the corporate systems are at risk. With the development of blockchain technology, a decentralized trust management solution has been put forth to address this identity management problem. However, the whitelisting component is still centralised in all the recent papers that offer blockchain-based identity management. Auto-whitelisting and identity management solution architecture are key areas of research that are difficult to address in place of the manual whitelisting method.

2.6.2 IoT Privacy, Information Transparency and Access Management Software Architecture

The rapidly evolving Internet of Things (IoT) landscape offers immense potential for innovation and efficiency but also brings significant challenges in privacy, information transparency, and access management. Addressing these complexities requires a robust software architecture that can protect sensitive data, build trust, and ensure authorized access. Managing the vast and varied data generated by IoT devices is a major challenge, as this data is often personal and needs to be safeguarded against unauthorized access and misuse. Privacy-enhancing technologies like differential privacy and homomorphic encryption offer promising solutions. Differential privacy protects individual data by introducing controlled noise, allowing for meaningful analysis without compromising privacy. Homomorphic encryption enables computations on encrypted data, maintaining confidentiality throughout the process.

Information transparency is crucial for establishing trust among IoT stakeholders. Blockchain technology, with its immutable and transparent ledger, can enhance accountability by recording data provenance and access history. However, scalability and energy consumption issues need to be addressed for broader adoption. Access management in IoT environments requires nuanced approaches that consider different devices, users, and contexts. Traditional models like role-based access control (RBAC) and attribute-based access control (ABAC) are being evaluated for their effectiveness. Context-aware access control, which adjusts permissions based on real-time factors, and intelligent access control systems using machine learning are gaining traction. Integrating IoT devices into critical infrastructure demands stringent security measures, including secure boot, firmware updates, and intrusion detection systems. The concept of zero-trust security, which continuously verifies user and device identities, is also

becoming important. Microservices and service-oriented architectures (SOA) are being considered for IoT systems to offer flexibility, scalability, and resilience. However, interoperability and data consistency must be managed carefully. A holistic approach combining technical measures with legal and ethical frameworks is essential to address IoT privacy, information transparency, and access management. Privacy by design principles should be embedded in the development lifecycle, and robust data protection regulations enforced. Public awareness and education about IoT risks are crucial for building trust and promoting responsible usage. Ongoing research includes developing privacy-preserving machine learning algorithms for IoT data analysis, exploring blockchain-based platforms for secure and transparent data sharing, and designing context-aware access control systems. As the IoT landscape evolves, innovative software architectures will be key to realizing its full potential while safeguarding individual rights and societal interests. A well-defined trust model is fundamental for IoT security, outlining relationships between entities and their roles. Federated learning, a distributed machine learning technique, is promising for privacy-preserving data analysis, minimizing data leakage. Cryptographic techniques like digital signatures and hash functions are vital for ensuring data integrity and authenticity. Edge computing, which processes data closer to IoT devices, offers privacy and performance benefits by reducing the amount of sensitive information transmitted to the cloud. Privacy-preserving data sharing techniques like secure multi-party computation (SMPC) and differential privacy can enhance privacy guarantees. Standardized data formats and communication protocols are essential for interoperability and secure data exchange. User-centric IoT systems are crucial for fostering trust, requiring user-friendly interfaces, clear privacy policies, and transparent data handling practices. Continuous monitoring, vulnerability assessments, and staying updated on security best practices are essential for maintaining a robust defence against evolving IoT threats.

Developing a secure, transparent, and user-centric IoT ecosystem requires a multifaceted approach. An IoT future that benefits society while preserving individual rights and freedoms can be built through the leveraging of cutting-edge technologies, rigorous research, and collaboration among academia, industry, and policymakers.

2.6.3 IoT Data Ingestion Software Architecture

IoT edge processing has been limited to local environments (HMI) since the advent of PLC and SCADA systems. Each vendor had their own proprietary software for data collection and processing, along with a finite amount of storage. Service-Oriented Architecture (SOA) made it possible for IOT manufacturers to push sensor data directly to servers using modified SOA communication protocols. Numerous smart middleware and processing engines, such as Edge Firmware and Smart Gateways, are now used between sensors and servers because of advancements in sensory technology and software. Product vendors are encouraged to develop a wide range of IoT products and services across industry verticals because of the standardisation of the communication protocol stack, including OPC-DA, OPA-UA, and Pub-Sub for SCADA communication, MQTT for Pub-Sub communication, and CoAP for communication over HTTP. Because of the limitations of REST/SOAP-based SOA architectures, containerized (Docker/Kubernetes, etc.) microservice-based architectures with smaller deployment footprints that can be mass-deployed at the edge and controlled from a server or cloud were adopted. IoT management and security capabilities are added through the development of cloud based IoT services. Point-to-point microservice design is challenged by multiple thread processing, and reactive microservice-based IoT communications are being proposed. There are still several issues to be resolved before a fully reliable solution architecture can be developed to fulfil the demand for future IoT edge processing space, though. How are IOT devices going to handle backpressure with guaranteed real-time message delivery given their limited capacity, power, and resources? To achieve the needed QoS without any problems, how will IoT devices handle offline storage capabilities, dynamic pipeline filtering, and use the store and forward principle? It is always preferable for edge processing to handle difficult real-time processing tasks. Due to the increase in linked devices, processing the same piece at the server at the same performance level may occasionally be difficult. Is it possible to dynamically shift some of these processing tasks to the edge without compromising the QoS of edge processing? One of the solutions could be osmotic computing. However, more research on agent deployment and execution, as well as deployment solution architecture for osmotic computation at the edge, is required.

2.6.4 IoT Data Modelling, Management and Data Governance Software Architecture

Storage and processing are becoming increasingly difficult due to the exponential increase in the number of sensors. Systems that can currently handle terabytes of data will soon need to handle petabytes of data. Time series, document-based, graph, and relational database systems are examples of well-known classics that swiftly become obsolete. Architecturally, there may be a few approaches to improving read-write performance by removing impurities and unused processing power and optimizing every step along the route, from data ingestion through data storage. The current processing and storage needs may be extended and enhanced within current capabilities by data ingestion systems and patterns like SAGA and CQRS, event sourcing, osmotic computing architecture, and change data capture.

2.6.5 DLT based Software Architecture for IoT Applications

Using classical machine learning methods that are based on physics as well as optimization algorithms, a decentralized IoT actively handles data and makes decisions on behalf of consumers. However, it is crucial to maintain the ability for users to make decisions accurately and easily regarding their overall needs and comfort. In the Industrial IoT, digital twins are being used to forecast failure, as well as automatic decision-making and predictive maintenance, but the architecture and data communication systems still need to advance across the levels. Decentralized IoT communication architecture offers significant advantages and promises to shape the future of connected devices. By distributing communication and decision-making capabilities across the network, this architecture enhances scalability, resilience, and security. Through peer-to-peer communication and consensus protocols, it reduces reliance on centralized servers and minimizes single points of failure, ensuring greater reliability and availability of IoT services. Furthermore, the decentralized nature of this architecture promotes data privacy and ownership, empowering users to have more control over their personal information. However, challenges such as interoperability, resource constraints, and governance models need to be addressed for widespread

adoption. Despite these hurdles, the decentralized IoT communication architecture holds immense potential in enabling seamless connectivity, fostering innovation, and revolutionizing industries. Designing and architecting the underlying platform for data communication, storage, and delivery layers is the primary area to explore soon. It is crucial to explore novel approaches and collaborations to unlock the full benefits of this transformative architecture and create a connected world that is efficient, secure, and user centric.

Chapter 3

3. IoT Identity, Whitelisting and Decentralized Trust management Software Architecture

The Internet of Things (IoT) has revolutionized the way human being interacts with the world around us, encompassing a vast network of interconnected devices, sensors, and applications. As this IoT ecosystem continues to expand, ensuring the security and privacy of its participants becomes increasingly critical. The seamless integration of diverse devices and services demands a robust framework that can effectively manage identities, enforce whitelisting mechanisms, and establish a distributed trust management architecture. This thesis chapter embarks on a comprehensive exploration of IoT Identity, Whitelisting, and Distributed Trust Management Architecture, aiming to tackle the complex architectural challenges associated with IoT security. By delving into the intricacies of identity management, understanding the significance of whitelisting in access control, and harnessing the power of distributed trust, this research endeavours to pave the way for a secure and trustworthy IoT landscape. Through a combination of theoretical analysis and practical experimentation, this study seeks to contribute innovative solutions and actionable insights that can fortify the foundation of IoT systems and safeguard the privacy and integrity of IoT-enabled applications. As an era of unprecedented connectivity and technological advancement is entered, the knowledge gained from this research is expected to play a pivotal role in the shaping of IoT security's future and in the facilitation of the seamless coexistence of interconnected devices in a safe and reliable manner.

3.1 Trust, Identity, and Whitelisting in IoT: Challenges and Industrial Application Roadblocks

The rapid proliferation of IoT devices has driven technological progress but also introduced critical challenges in identity, whitelisting, and trust management. Ensuring each device has a unique and secure identity is increasingly complex, exposing systems to risks like identity spoofing and unauthorized access. Whitelisting trusted devices becomes difficult to scale, requiring constant updates and monitoring, and poses significant risks if compromised. Trust management is further complicated by the diversity of devices, manufacturers, and communication protocols, leading to inconsistent security capabilities and increased vulnerability. Addressing these issues demands collaboration among manufacturers, service providers, and policymakers. Solutions include robust identity frameworks, blockchain-based device authentication, standardized protocols, and continuous monitoring to strengthen IoT security and support its sustainable growth.

In industrial and consumer sectors, IoT significantly reduces human effort in routine tasks; however, the rapid proliferation of IoT devices across various domains has created a pressing need for decentralized, autonomous device management. Centralized systems are increasingly inadequate, especially in competitive markets where multiple service providers deploy proprietary infrastructures, such as in the television and Internet broadcasting industry. Here, each provider maintains its own network components, leading to high capital and operational costs. Consumers also face recurring expenses for setup, activation, and device management, particularly when switching providers, often resulting in device redundancy and e-waste. The lack of interoperability and reuse across vendor-specific IoT systems not only burdens consumers but also leads to inefficient resource utilization and environmental concerns. This thesis addresses these challenges by proposing an architecture that promotes interoperability, decentralization, and sustainable IoT infrastructure management. To encapsulate the associated difficulties:

1. Transparent Device Lifecycle: IoT devices should have an end-to-end transparent lifecycle to track usage, understand depreciation, and enhance security through monitored and authorized access.
2. Secure, Distributed Whitelisting: The device whitelisting process must be transparent and decentralized to eliminate vulnerabilities and prevent unauthorized access, ensuring system resilience.
3. Automated Vulnerability Mitigation: The system should autonomously detect vulnerable devices and apply preventive actions (e.g., patching, isolation) to maintain network security and performance.
4. Interoperable Information Exchange: Device data exchange should be standardized and interoperable across service providers to maximize reusability, minimize e-waste, and support scalable, vendor-neutral ecosystems.

In brief, Identity management in IoT involves processes and technologies that ensure secure, appropriate access to information while protecting device profiles. As IoT manufacturing and services expand, many organizations offer closed-loop firmware ecosystems that bundle devices, software, and services. However, these proprietary ecosystems lead to **vendor lock-in**, where consumers face recurring service costs and limited access to their data. Each provider uses its own software architecture and data pipeline, hindering interoperability and contributing to growing IoT device waste with minimal recyclability. Additionally, data transparency remains a major issue, as control over user data is typically retained by the ecosystem providers.

3.3 System Design, Solution Architecture, and Implementation

Consider the scenario where multiple service providers offer identical services, each maintaining its distinct ecosystem for firmware-based identity administration, resource authorization, data administration, and data access. When a specific consumer desires to reuse the same firmware for similar services, the subsequent design will elucidate how open software architecture and design principles can effectively address

interoperability challenges. At the heart of this solution lies the incorporation of blockchain technology, which ensures the highest level of data transparency. To facilitate this interoperability, Providers and Consumers, represented by IoT firmware, engage with an open distributed network, paying a nominal fee for access. Within this open network, they partake in the exchange, management, and transaction of information as an integral part of service enablement and management. This interaction allows IoT firmware to seamlessly connect with a widely dispersed network through four fundamental operations.

1. Registration – In this initial phase, IoT firmware attempts to establish a direct connection with the network using a token generated from the device's identifier, code, and timestamp. A provider-specific one-time code can be generated via magnetic strip or digitally via the application. When the registration request is submitted, the distributed network end will initiate an automated validation process to corroborate the call's authenticity (Smart Contract execution). Once affirmed, IoT firmware will be correctly registered, and the service token will be shared with the invoker for subsequent communications.

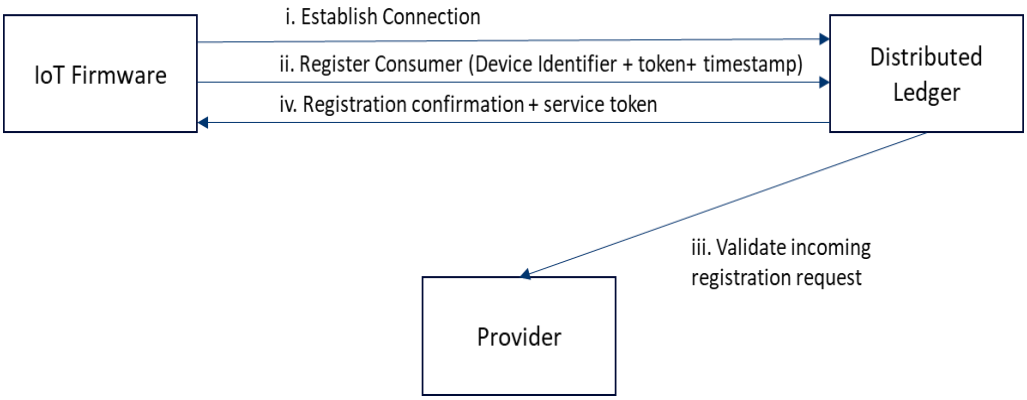


Figure 3. 1 Asset registration operations

2. Authorization - In subsequent service requests, the IoT firmware must transmit the service token to the distributed network. This process involves validation by an intelligent contract, which examines both the incoming service token and assesses the Trust Score. The Trust Score serves as a lightweight scoring mechanism, executed upon the conclusion of a smart contract, aimed at enhancing or diminishing the trustworthiness of the IoT firmware. The Trust

Score starts at a baseline of 0 when a new registration request is made. Over time, as a sequence of stringent, zero-tolerance authorization processes is successfully executed, the trustworthiness of the IoT firmware undergoes a gradual improvement. This enhancement is contingent upon various factors, including the number of successful registrations, logins, logouts, and the utilization of infrastructure, all accomplished without triggering any concerning red flags. Specifically, for each secure login that does not exhibit vulnerabilities, the trust score receives a steady increase of 1. Conversely, any sign of suspicious activity can lead to a significant reduction in the trust score, potentially by as much as 10 points. In instances where there is a persistent pattern of suspicious behaviour and the trust score approaches a critical low of 0, it becomes imperative to act. This action may involve initiating a re-registration process or generating new service tokens as a means of addressing the deteriorating trustworthiness of the IoT firmware.

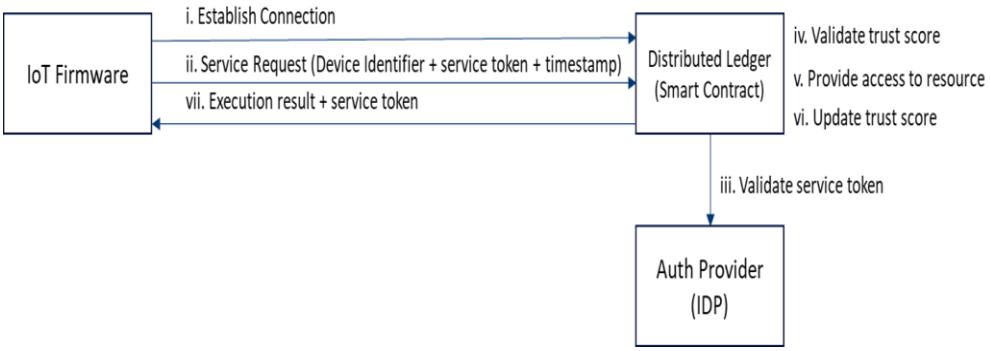


Figure 3. 2 Asset Authorization process flow

3. Transaction - Once authorized, IoT firmware can perform multiple authorized operations (like accessing user profile access, data and consumable services access) by executing smart contracts. For each operation and execution of the distributed ledger, caller firmware for smart contracts must pay a network charge (like gas price for public Ethereum network). Each transaction initiated is essentially a new version to the underlying immutable distributed ledger. IBFT2 consensus protocol has been configured within the blockchain network. IBFT2 (Istanbul Byzantine Fault Tolerance 2) is an evolution of its predecessor, IBFT, and is specifically engineered to address the challenges of Byzantine fault tolerance within a consortium blockchain context. This consensus mechanism

relies on a fixed set of validators, typically predefined and trusted participants, to reach agreement on the order and validity of transactions in the blockchain. Unlike some other consensus algorithms, IBFT2 does not require resource-intensive proof-of-work computations, making it highly energy-efficient. It also offers near-instant finality, meaning that once a block is added to the chain, it is virtually immutable, providing a high level of security. IBFT2 is a popular choice for enterprise-grade blockchain applications where performance, scalability, and trust among network participants are paramount considerations. With each response of a successful transaction, a new service token is generated and attached to the response for use in the subsequent operation. Service token is generated by smart contract based on the successful authentication. The lifespan of the service token resides within the authenticated session / request (configurable) of the IoT entity.

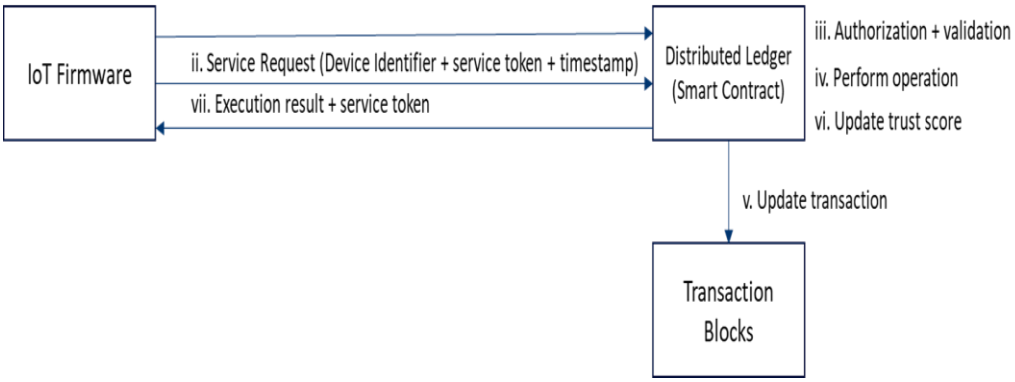


Figure 3. 3 Asset Transaction process flow

4. Deregistration – Deregistration remains the essential process through which IoT firmware can be disengaged from its current owner. This process encompasses three distinct scenarios. Firstly, it occurs when a user intends to switch from one service provider to another, necessitating the disassociation of the IoT firmware. Secondly, deregistration becomes necessary when the IoT firmware needs replacement due to critical issues, at which point the owner can transfer control to the service provider. In the course of the deregistration process, the IoT firmware is obligated to transmit the token alongside the authorization request. The smart contract then executes the deregistration procedures, and in doing so, updates the distributed ledger by resetting all prior variables, including the trust

score and the existing provider ID. Moreover, if the IoT firmware is voluntarily surrendered, the proprietor's information undergoes a reset as well. This comprehensive process ensures the seamless transition of control and the appropriate management of the IoT ecosystem.

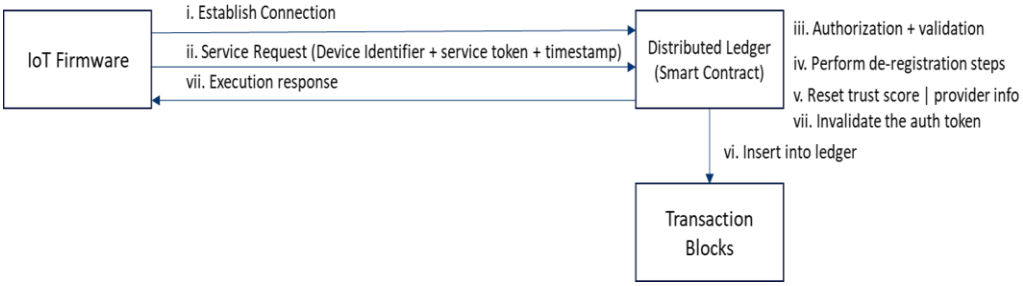


Figure 3. 4 Asset De-registration process flow

Below is the solution architecture to implement the solution of proposed operations.

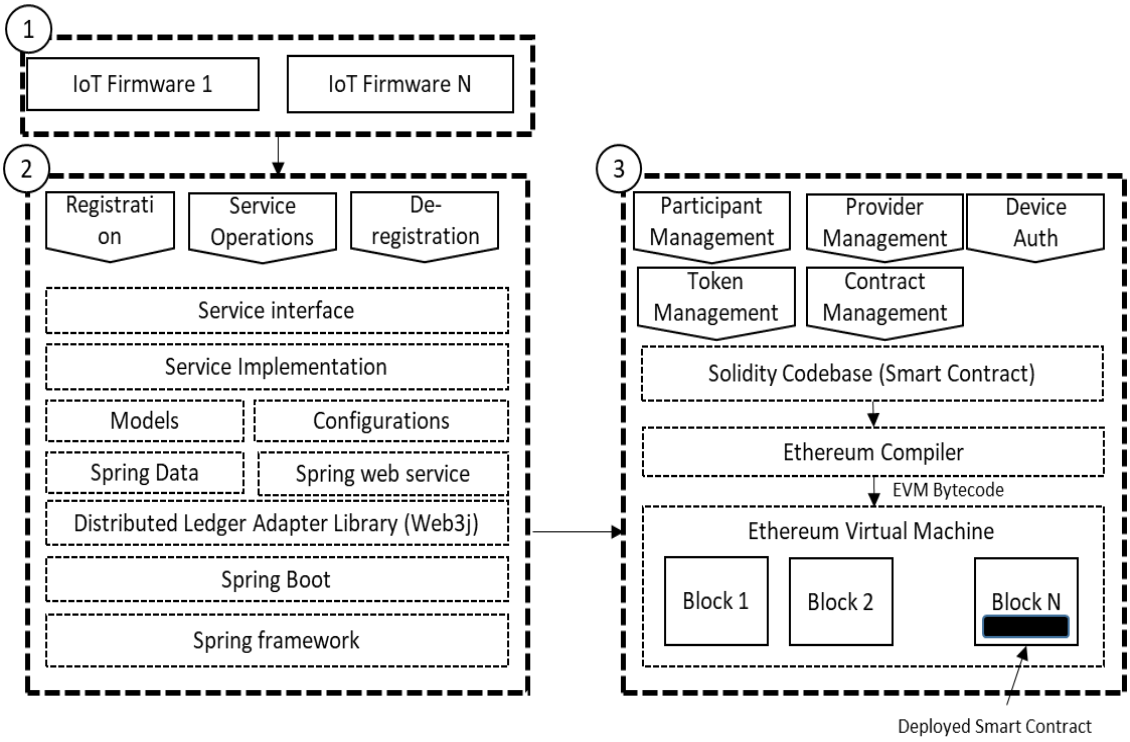


Figure 3. 5 Blockchain integrated Solution Architecture

All functionalities, services, and components supporting this architecture were developed independently. The service layer is built using the Spring Framework and Spring Boot, leveraging open-source libraries provided by the Spring.io community to implement the necessary server-side logic and components. Additionally, the smart contracts, token management, and decentralized communication mechanisms are

implemented on top of Ethereum public blockchain client libraries, enabling secure and transparent interactions within a decentralised network.

1. Typically, the IoT firmware layer within the IoT device initiates service operations. All service operations continue to be REST calls. The microservice layer typically exposes service operations.
2. Microservice layer is developed according to the fundamental principles of 12 factor App [77] and the microservice architectural pattern. It follows the domain-driven design meticulously to identify many domains associated with business functions within a constrained context. In my experiment, I identified a singular domain with four distinct business functions, most notably the set of operations. Microservice carries out all primary validations and serves as the middleware between the distributed ledger and the effective devices.
3. Smart contract execution is supported by the distributed ledger, which is the public blockchain network (Ethereum). The contract is deployed to the blockchain network and preserved in the transaction block. The distributed contract layer executes all contract operations and validations correctly. Upon successful execution of a smart contract, the miner typically generates a new transaction hash that is eventually recorded into the blockchain's immutable transaction block.

The proposed architecture is highly scalable because the backend distributed ledger is the public ledger, which stores transaction blocks on tens of thousands of constantly expanding nodes. The Microservice layer is deployable in a cloud container with dynamic scaling capabilities. A fault-tolerant well with built-in security at its foundation. If the information being exchanged is altered in any way, the transaction will be denied at the blockchain level. Token-based resource authorization grants the appropriate IoT object access to the service layer and the network. Cloud has built-in disaster recovery and 24x7 availability for microservice layer.

To prove the proposed concept, a physically distributed ledger based on the Ethereum blockchain is contemplated. Smart contracts are implemented with the solidity programming language, which is the de facto standard for implementing smart contracts on the Ethereum blockchain. For development implementation, the Ganache environment is utilized. Ganache is an easy-to-use Ethereum blockchain simulator for

constructing, testing, and executing smart contracts as well as inspecting the transactional state prior to executing the same on the public Ethereum blockchain. Through the truffle console, the truffle suit is used to construct and manage the smart contract project structure. The truffle configuration specified in the project enables the development space to connect to the ganache environment to deploy the contract to the blockchain. A Java spring boot-based application is utilized for the client stratum. This proof of concept is defined as a smart contract (AssetManagementContract) that is executed for each blockchain system interaction.

```
contract AssetManagementContract {
  address public owner;

  struct Asset {
    string name;
    string description;
    address currentOwner;
    uint256 trustScore;
    bool isRegistered; // Added to track asset registration status
  }

  mapping(uint256 => Asset) public assets;
  uint256 public totalAssets;

  event AssetAdded(uint256 assetId, string name, string description, address currentOwner);
  event AssetTransferred(uint256 assetId, address previousOwner, address newOwner);
  event TrustScoreChanged(uint256 assetId, uint256 newTrustScore);
  event ServiceTokenGenerated(uint256 assetId, string serviceToken);
  event AssetRegistered(uint256 assetId); // New event for asset registration
  event AssetDeregistered(uint256 assetId); // New event for asset deregistration

  modifier onlyOwner() {
    require(msg.sender == owner, "Only the contract owner can perform this operation");
    _;
  }

  constructor() {
    owner = msg.sender;
```

```
    }

    function addAsset(string memory name, string memory description) public onlyOwner {
        uint256 assetId = totalAssets++;
        assets[assetId] = Asset(name, description, msg.sender, 0, true); // Set isRegistered to true
        emit AssetAdded(assetId, name, description, msg.sender);
    }

    function transferAsset(uint256 assetId, address newOwner) public {
        Asset storage asset = assets[assetId];
        require(asset.currentOwner == msg.sender, "You are not the current owner of this
asset");

        asset.currentOwner = newOwner;
        emit AssetTransferred(assetId, msg.sender, newOwner);
    }

    function executeService(uint256 assetId) public {
        Asset storage asset = assets[assetId];
        require(asset.currentOwner == msg.sender, "You are not the current owner of this
asset");

        // Add your service execution logic here

        // Increment the trust score
        incrementTrustScore(assetId);
    }

    function incrementTrustScore(uint256 assetId) internal {
        Asset storage asset = assets[assetId];
        asset.trustScore++;
        emit TrustScoreChanged(assetId, asset.trustScore);
    }

    function decrementTrustScore(uint256 assetId) internal {
        Asset storage asset = assets[assetId];
        require(asset.trustScore > 0, "Trust score cannot be negative");
```

```
        asset.trustScore--;
        emit TrustScoreChanged(assetId, asset.trustScore);
    }

    function generateServiceToken(uint256 assetId) public returns (string memory) {
        // Generate a random alphanumeric service token (you can use your preferred method)
        string memory serviceToken = generateRandomAlphanumericToken();

        emit ServiceTokenGenerated(assetId, serviceToken);
        return serviceToken;
    }

    function registerAsset(uint256 assetId) public onlyOwner {
        Asset storage asset = assets[assetId];
        require(!asset.isRegistered, "Asset is already registered");
        asset.isRegistered = true;
        emit AssetRegistered(assetId);
    }

    function deRegisterAsset(uint256 assetId) public onlyOwner {
        Asset storage asset = assets[assetId];
        require(asset.isRegistered, "Asset is not registered");
        asset.isRegistered = false;
        emit AssetDeregistered(assetId);
    }

    // Helper function to generate a random alphanumeric token
    function generateRandomAlphanumericToken() internal pure returns (string memory) {
        uint256 tokenLength = 12; // You can adjust the length as needed
        bytes memory tokenBytes = new bytes(tokenLength);
        string memory charset =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
        uint256 charsetLength = bytes(charset).length;

        for (uint256 i = 0; i < tokenLength; i++) {
            uint256 randIndex = uint256(keccak256(abi.encodePacked(block.timestamp,
block.difficulty, i))) % charsetLength;
```

```
            tokenBytes[i] = bytes(charset)[randIndex];
        }
        return string(tokenBytes);
    }

    function getAsset(uint256 assetId) public view returns (string memory, string memory,
address, uint256, bool) {
        Asset storage asset = assets[assetId];
        return (asset.name, asset.description, asset.currentOwner, asset.trustScore,
asset.isRegistered);
    }
}
```

The simplified template for an AssetManagementContract that includes basic functionality for adding, transferring, querying, performing transactions, updating trust score and generating service tokens of an asset.

Domain Object Model - To represent IoT domain object at the client layer, an asset object is created and assigned a unique serial number, serial number, asset name, trust score, asset proprietor, and cost. The trust rating is determined by the number of successful invocations. The value begins with zero. Each effective invocation increased the trust score by 1. In the event of a failed attempt or vulnerability, it decreased by 10. The range of Trust scores is from zero to one. Create a Participant object to interact with the blockchain. The participant in the system may be the proprietor (device) or the intended future owner (device). The attributes user key, secret, username, classification, and address define the participant. Each participant should have a pocketbook from which to conduct transactions. During the creation of a new participant (device), a pocketbook is assigned that remains with the participant throughout its lifetime. An identity provider provides the user key and secret, with which an access token is generated and validated during execution. The ownership object is defined with the attributes asset id, owner id, and owner address to maintain asset ownership within the system. Throughout the lifecycle of the execution asset, participants update their possession.

Operations - The registerAsset operation is properly registering a new asset into the system. The smart contract confirms that the asset is registered only once with the unique one-time code by the same provider. Once deregistered new provider can

register the asset again with the unique code and serial number. Previous information about the same asset can only be audited into the blockchain. This typically brings the asset interoperability into the entire value chain of the system. The addParticipant includes a new participant (device) into the system. A smart contract recognizes the uniqueness of the participant for that provider. The newOwner operation transfers the asset from the existing participant to the new participant. There could be two possibilities. An asset can be transferred from provider participant to consumer participant for the first time. An interoperable asset can be transferred among consumer participants to get the maximum reusability from the asset. The smart contract also validates that only the owner can be able to transfer the asset and self-transfer of the asset is prohibited. Participant (device) authentication [78] [79] is addressed by authenticateParticinat operation. For the new registration, the key and secret are confirmed and assigned an access token for the following communications. For successive calls, the smart contract executes the operation to validate the token ensuring participant (device) authenticity. In this proof of concept, token generation and validation mechanisms have been implemented within the smart contract.

Consumer Channel - The consumer channel is implemented in a thin lightweight Java layer [80]. This distinct layer can be properly executed on any IoT firmware device having minimum JVM support. Web3j Library is used to compile the solidity smart contract to java smart contract stub. All the operations invoked on the stubs layer are submitted to the Ethereum virtual machine over an HTTPS channel.

Token Implementation - TradeCoin is created to interact with EVM based on ERC20 specifications. This coin represents a medium of exchange in the Ethereum system of a real-world tangible object i.e., the exchange of IoT devices between participants. Through the token exchange smart contract gets executed, and network miners get paid for hashing out the transaction. During register and transfer operation within a smart contract actual asset gets transferred based on the exchange of trade coins between two exchange participants.

The snapshot of execution results based on the Local blockchain network (Ganache) and the test blockchain network (Ropsten) is given below.

Total Number of assets	100 unit	
Operation	Local Network (Ganache)	Ropsten Network
Register asset	180 ms	15467 ms
Transfer Asset (randomly)	230 ms	19173 ms
De-register Asset (randomly)	205 ms	17934 ms
Gas used for each operation	40000 gas units	40000 gas units
Gas Limit	6721975 gas units	6721975 gas units
Number of generated access tokens	175 unit	175 unit

Table 3. 1 Operations execution results

Contrasting rates of block creation between Ganache, a local blockchain network with auto-mining capabilities, and Ropsten, which encounters a process comparable to that of the public Ethereum blockchain. Ganache's auto-mining feature is remarkably effective at generating new transactional blocks, concluding the task almost instantaneously. The Ropsten network, on the other hand, requires more time for the same operation, although its efficacy is comparable to that of the public Ethereum blockchain. The Public Ethereum blockchain project team remains committed to enhancing the overall efficacy and mining efficiency of the blockchain network. However, the current emphasis of their experimentation is on exploring and proposing solutions for challenges, as opposed to focusing merely on efficiency or throughput enhancements. This deliberate focus on solution-oriented approaches is influenced by the ongoing upgrade of Ethereum 2.0, the underlying technology's backbone. This upcoming iteration of Ethereum will introduce a plethora of innovative features that will transform the blockchain landscape. Considering the impending changes brought about by Ethereum 2.0, the project team focuses its efforts on addressing particular

aspects. The objective is to align their solutions with the forthcoming innovations, which will substantially improve the network's capabilities and overall performance.

3.4 Conclusion and Future Directions

In this study, a novel blockchain-based software architecture is introduced, aiming to foster collaboration among interested service provider participants. The primary objective is to establish a network where a common consensus can be achieved by these participants to create an interoperable and highly reusable ecosystem. By doing so, the pressing issue of electronic waste can be tackled, and a decentralized, auto-trust management, interoperable solution tailored for Internet of Things (IoT) applications is presented. In meticulous detail, the essential operations and process flow design of the proposed architecture are described. To validate the concepts, a proof-of-concept implementation is conducted, both on a local environment and a test blockchain network. Throughout the experimentation phase, it is found that remarkably high average transaction times for each successful operation are observed when using the public Ethereum blockchain or similar test blockchain networks. This delay is primarily attributed to the time-consuming hash mining process and block creation in these test networks. However, future improvements in performance are expected, particularly with the advent of Ethereum 2.0. This highly anticipated upgrade in the Ethereum ecosystem promises to significantly enhance the overall efficiency of the blockchain. The core focus of the software architecture and experiment is to establish a foundation for a reusable, interoperable, and self-managed IoT asset whitelisting process and its corresponding operations. Furthermore, future enhancements are envisioned by defining standard communication semantics and attributes to support improved consensus, data isolation, and privacy, thereby boosting efficiency even further. For added security, the integration of different encryption methods at the transport and application levels is proposed, complementing the existing architecture and platform infrastructure. Through the experimentation, the concept and core foundational software architecture are firmly established, paving the way for an exciting future roadmap. Enthusiasm is expressed about the potential of this solution to revolutionize the IoT landscape, bringing the community closer to a more sustainable and interconnected future.

This chapter establishes the foundational security mechanisms required to uniquely identify devices, manage trust relationships, and enforce secure communication within a decentralized IoT ecosystem. Building on this foundation, the following chapter on IoT Privacy, Information Transparency, and Access Management Software Architecture extends the discussion by focusing on how these identified and trusted devices handle data responsibly ensuring privacy, enabling transparent data flows, and implementing fine-grained access control. Together, these chapters present a cohesive framework that addresses both the "who" (identity and trust) and the "how" (privacy and access) of secure and ethical IoT system design.

Chapter 4

4. IoT Privacy, Information Transparency and Access Management Software Architecture

The emergence of the Internet of Things (IoT) has ushered in an unprecedented level of daily convenience and efficacy. However, this technological evolution has also generated significant privacy and data security concerns. As IoT devices and applications accumulate vast quantities of personal and sensitive data, it becomes imperative to ensure robust privacy protection, information transparency, and access management. This thesis chapter explores in depth IoT Privacy, Information Transparency, and Access Control Management in an effort to address the complex architecture challenges of protecting user data while maintaining seamless functionality. This research aims to set the groundwork for a more secure and privacy-centric IoT landscape by investigating the complexities of privacy-preserving techniques, analysing the significance of transparent data handling practices, and developing effective access control mechanisms. This study attempts to provide innovative solutions and actionable insights that enable individuals to retain control over their data and foster trust in IoT technologies by combining theoretical analysis with practical experiments. As the Internet of Things (IoT) continues to reshape the world, this research will play a crucial role in moulding the future of data protection and privacy in this interconnected and digital era. The business world is moving toward a network of decentralized and self-managed supply chains, which is being driven by the development of use cases. Information interchange is being decentralized throughout the financial, manufacturing, and shipping industries, as well as the automotive, food, and real estate supply chains. The data collected by IoT devices is sent to a decentralized network, which allows transactions to be completed without the need for a centralized authority. This architecture has several issues that need to be resolved, including centralized trust management, interoperability, reusability, privacy, and access control for the exchange of IoT data. It's possible that the banking, finance,

and insurance businesses need decentralized group formation and information visibility constraints to fulfil their missions. In previous work (Chapter 3), I pushed for decentralized trust management, as well as reusability and interoperability. This study solves concerns relating to privacy and access control, as well as permission and transparency in information transmission, without compromising decentralization. My method and the proof of concept both contribute to the establishment of confidence in a decentralized solution architecture.

4.1 Challenges of IoT Privacy, Information Transparency, and Access Control

The internet of things (IoT) ecosystem is constituted of entities and devices that produce enormous amounts of processed and intermediate data, both of which are crucial for security and safety. Due to frequently evolving sustainable business models that demand novel, reasonably priced solutions for privacy, access control, and transparency, a new ecosystem of enterprises is forming. On the other hand, the number of IoT devices is growing significantly at an exponential rate. Centralized access and privacy management over semi-private networks continue to be extremely burdensome in terms of upkeep, the absence of open standards, resource limitations, and widespread acceptance. In addition to raising maintenance costs, the system's single point of failure is the system itself. Because each ecosystem player has a configuration for centrally managing their own devices, reusability and interoperability are equally challenging to maintain. Every other solution available now manages device access control by establishing policy rules [82] and characteristics [81]. Various players within the ecosystem may have different interpretations of this set of laws. The crucial network's entire value chain will collapse if the central system is seriously affected. To better understand the scenario, let's use an example from DTH and the media supply business [83]. In the industry, there are numerous DTH providers. A set-top box with a unique serial number and an individual identification number is often assigned when an existing subscriber joins a DTH provider. Currently, each service provider keeps a set-top box specific to a certain vendor and its individual customer base. In addition to a

group of suppliers who provide the set-top boxes, the provider also has an independent sales channel for acquiring the customer base. This consolidated corporate ecosystem presents numerous difficulties for operations management and centralized data management. Due to the upfront commitment of setup costs, users have problems because they are unable to transfer providers frequently in cases where providers have a monopolistic nature (interoperability issue). Active users might not be aware of how the supplier manages their personal information, such as KYC data and specific bank details. Users are not very aware of what would happen to the data stored with the prior provider even if they switched to a different provider (privacy issue). The old setup, including the set-top box, router, and cables, naturally becomes e-waste when a user switches service providers, even if it has the potential to be recycled because, within the same non-monopolistic DTH business, typical user configuration, devices, and setup are identical (reusability issue). I have previously presented a public blockchain-based approach for the automated device-managed trust management, interoperability, and reusability of IoT assets. The logical extension of these fundamental problems reveals that privacy on a public blockchain faces difficulties. The system normally needs to be decentralized access-controlled data transparency with the provider ecosystem after building a vital ecosystem with a group of top providers on a public blockchain. Users may have the ability to smoothly move between providers. Every DTH provider, however, inevitably has its own unique ecosystem of supply chain participants, including set-top-box makers, user service providers, warehouse managers, customer care support, etc. Information about the cross-provider supply chain ecosystem network need not be known by DTH providers. Additionally, a restricted group of networks allows participation from service providers, top suppliers, users, and/or IoT devices (set-top boxes) from the public network. Even one of the top suppliers might participate in numerous supply chain ecosystems (Access Control issue).

The following are the potential challenges in the context of the blockchain-based supply chain business:

- Without compromising the characteristics of a truly distributed network, service provider companies must maintain the confidentiality of supply chain information that is held by participating entities.

- Participating entities and enterprises must operate within the public network for the service to be fundamentally distributed and easily accessible to all participants.
- The organization must be able to offer participants in the public network the appropriate level of access control and information privacy.
- Promoting access control and privacy on open public networks as opposed to closed private or semiprivate networks to promote consistency, open standardization, and adaptability.

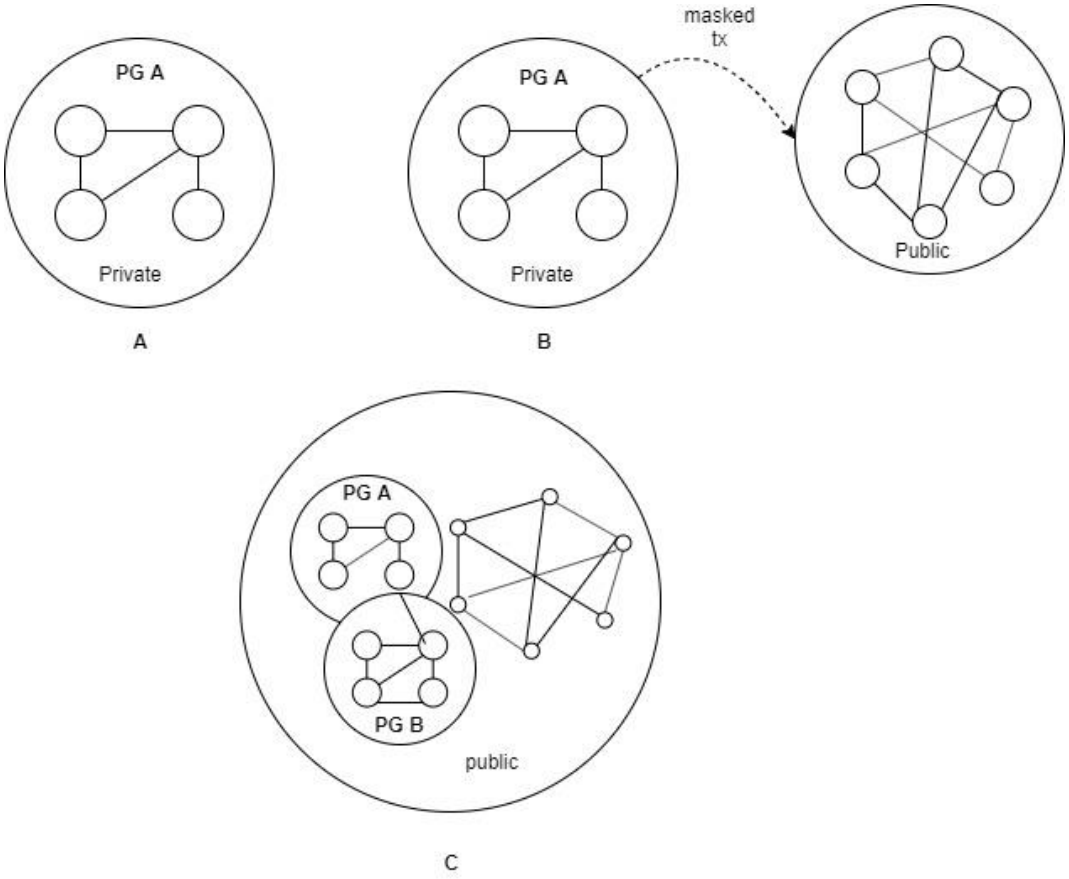


Figure 4. 1 A. Private Blockchain network B. Semi-private Blockchain network C. Hybrid Blockchain network

The majority of current distributed blockchain network privacy and access control strategies are based on private blockchains. The private blockchain network, which is developed and operated within organizational borders, is shown in Figure 4.1-A. The distribution network, privacy, communication protocol, and user and node access are completely under the control of the organization. Private blockchain implementation

differs from company to company, which hinders interoperability and lessens transparency because this method compromises integrity. Researchers have suggested a semi-private paradigm, which is shown in Figure 4.1-B, to address the transparency concerns. The resulting masked transactions in this scheme are sent onto the open blockchain, where everyone can see them. Since the majority of core blockchain operations are carried out within organizations and the resulting data is posted to the public blockchain, the technique does not adhere to real distributed blockchain principles. Semiprivate networks can occasionally be constructed without adhering to the accepted distributed ledger definitions and technology. This considers network regulation and traceability issues. Flexible privacy group building continues to be crucial in a decentralized setting where few individuals require full network access while others may require restricted access to the data. A strong solution architecture is required to enable the platform at the center to address potential challenges related to privacy, security, and access control in the decentralized environment. As all information on transactions and balances is available to the public, V. Chang et al. [84] emphasized the security and privacy leakage issue as one of the major difficulties with blockchain-based transactions.

A trust model is to be created to the fullest extent possible without compromising or restricting the capabilities of the public distributed ledger shown in Figure 4.1-C. An open standard design is followed by the system, and it is built on top of a public blockchain, resulting in extreme interoperability, sustainability, and the necessary level of information transparency being offered to all parties involved. A hybrid blockchain with the appropriate level of privacy and access control within the public blockchain network can be created by developing soft privacy groups and built-in distributed transaction managers for widespread adoption. Permissioned blockchain networks are automated by the system through the utilisation of smart contracts.

This research contributes a novel, privacy-preserving, decentralized architecture for IoT-enabled business ecosystems using a public Ethereum-compatible blockchain (Hyperledger Besu) as the execution layer. Addressing the limitations of private and hybrid blockchain models, such as centralization, poor scalability, and lack of data transparency, I designed a public, permissioned blockchain network architecture that supports granular access control, private transactions, and seamless interoperability among distributed actors like device manufacturers, service providers, regulators, and

end users. A key innovation is the integration of smart contract-based permissioning for both nodes and accounts, allowing secure onboarding, role-based access enforcement, and dynamic suspension of compromised actors. The system introduces Privacy Groups and Private Transaction Managers (PTMs) to enable transaction visibility strictly among intended parties while preserving ledger immutability and traceability. Unlike traditional off-chain privacy approaches that compromise decentralization, this work implements on-chain privacy on a public network using encrypted hash distribution and privacy group IDs, ensuring secure transaction execution without leaking sensitive data. A Proof of Concept (PoC) demonstrates end-to-end deployment using three Besu nodes with Tessera-based PTMs and showcases practical configurations for smart contract deployment, privacy group creation, and secure service invocation. The framework is further validated through an IoT use case involving DTH service providers and devices, demonstrating dynamic trust boundaries, auditable access for regulators, and seamless service provider switching without data leakage. This contribution stands out in its ability to combine enterprise-grade privacy and access control with the openness and resilience of public blockchain networks, laying a robust foundation for scalable, transparent, and secure decentralized IoT ecosystems.

4.2 System Design and Decentralized Solution Architecture

Several research activities have been implemented to safeguard the solution design of the blockchain network. A distributed ledger functions by mining a transaction hash, adding it to an immutable chain, and replicating the chain across thousands of blockchain nodes to prevent an intruder from interfering with each node. Increasing use cases necessitate that participant users' access to blockchain-written information be obscured or secured. Private and hybrid blockchains employ several modest solutions to the same issue [85] [86]. Private, hybrid, and isolated blockchain networks are incapable of reconciling use case requirements with security, privacy, and access control. The creation of numerous private blockchains is not a viable long-term solution because it resists mainstream adoption. Certain private blockchain networks are

centralised, which violates the distributed ledger principle [87]. Any user can join the open network and use the service, but blockchain data privacy and user access control is maintained automatically. On an open public network, a system design with protected privacy, access control, and data transparency should provide businesses with the most flexibility to adopt it. In my example use case, various DTH service providers partake in their ecosystem. Leading providers collaborate with device manufacturers to produce DTH set-top boxes that adhere to open standards, so that when a user switches providers, only minor adjustments are required for integration. When a user transfers from one prominent DTH provider to another, the system should restrict access to previous transaction information. If a central regulator or authority desires to conduct an audit, all DTH box transactions are immediately accessible. The blockchain network architecture illustrating the design considerations is shown below.

Active blockchain network participants are,

- Manufacturer - One manufacturer can build DTH boxes for multiple providers. They can have separate terms, contracts, and suppliers.
- DTH Service Provider - Provides service to User. Let's assume one user has one DTH set-top box for simplicity. A single user may have many set-top boxes. Distinct providers can own different set-top boxes.
- DTH Set top Box - This IoT asset may execute security checks, token exchange, auto registration, and correct authorisation.
- Regulator Authority - Can audit any transaction, regardless of service provider.
- User maintains DTH box (IoT entity). In auto transactions, the IoT entity handles the user's active account.

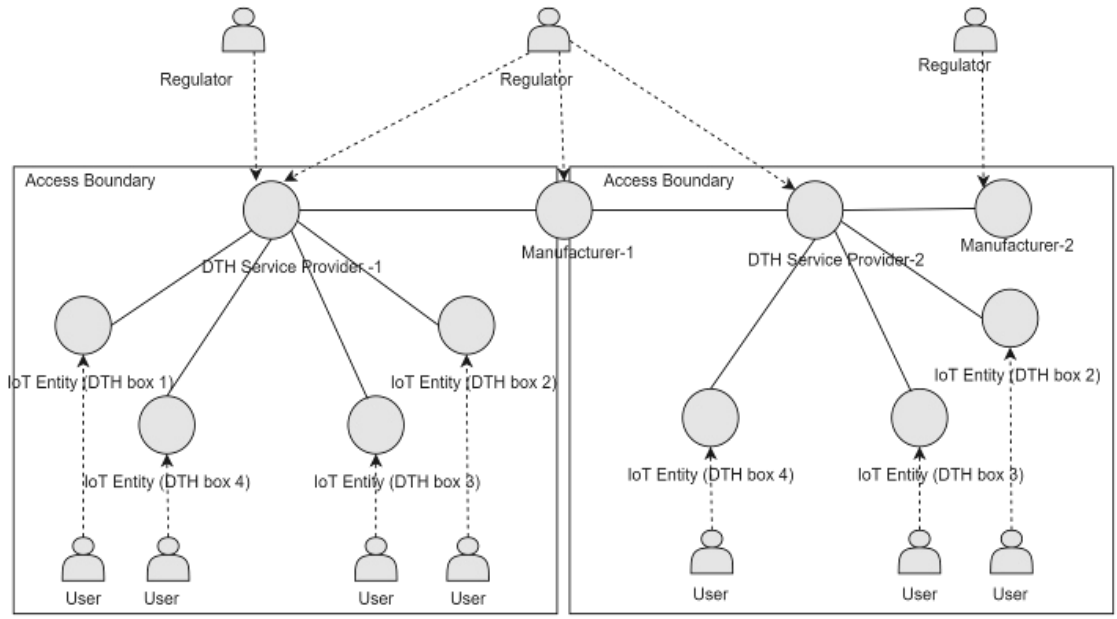


Figure 4. 2 On Chain participants

A smart contract regulates the participation of a new participant. The participant in question can join the network without requiring central approval. The onboarding smart contract is activated when a new participant enters the network. The permissioning smart contract on the blockchain must whitelist all participant wallets that will conduct transactions. Once accepted via the decentralized induction smart contract, the participant is introduced to the network and can conduct transactions after their account has been whitelisted. Now, transaction demarcation boundaries will be determined by the actions of the participants. This activates the intended access control level for each participant. Account permissions and node permissions are distinct permissions.

Node Permissioning – Any node can join the public blockchain with minimum setup. Permissioned blockchain nodes are required by the blockchain's smart contract to safeguard it from unlawful players. Blockchain only allows known players. This lets the governance system control node connections. In Figure 4.3, regulators can view transactions from all nodes. If a DTH box and its users are on the same blockchain, they can access the network.

Account Permissioning – This allows the blockchain node to enforce the identity requirement and decide which account can complete the transaction. If a permissioned account malfunctions, further network transactions can be suspended. In this scenario,

when an IoT entity wanted to malfunction and violated transaction restrictions in the on-chain permissioning smart contract, the account from which the IoT entity was transacting was suspended, isolating the entity from the network.

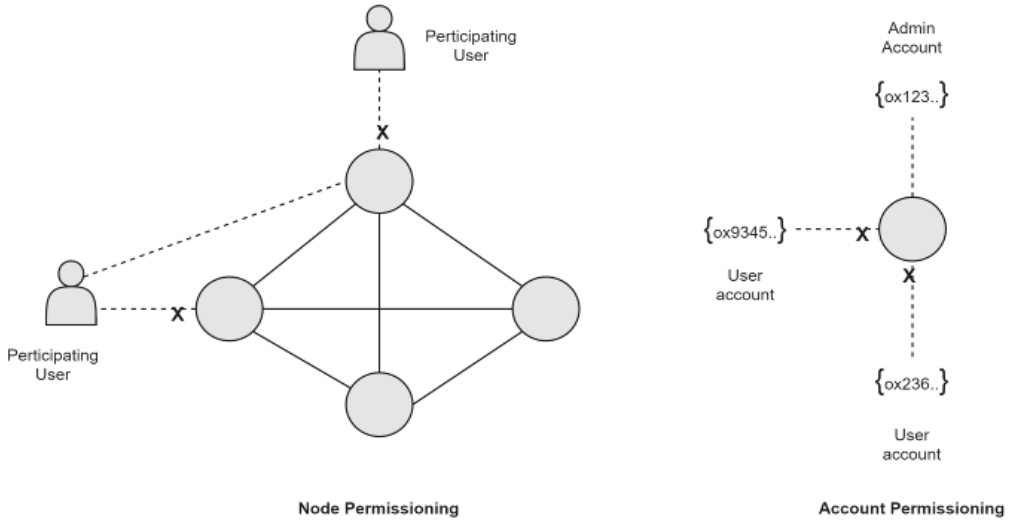


Figure 4. 3 Node and Account permission

On-chain privacy is crucial. Most blockchain privacy designs focus on private networks outside the main public network. This holistic approach allows private transactions to be maintained outside of the public network, breaking the distributed ledger notion. In this approach, I propose privacy-enabled transactions on the public mainnet, where only the intended participants can see the transaction hash. Below is a diagram that illustrates privacy groups and private transactions.

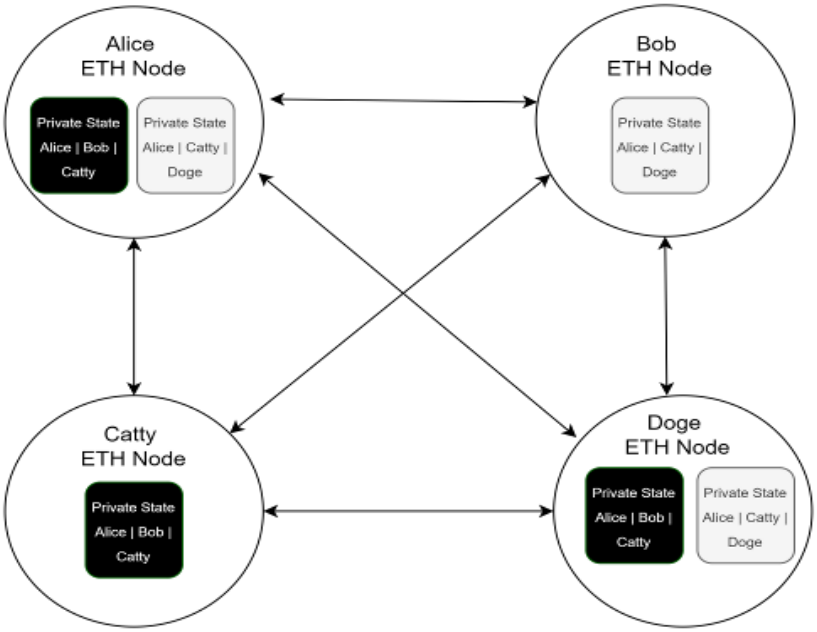


Figure 4. 4 Privacy groups and related transactions

Privacy Group – All nodes in a privacy group can communicate with each other. All smart contracts written into a privacy group can be accessed quickly within that group. Any publicly implemented smart contracts are freely accessible. Cross-group smart contract access is allowed if multiple privacy groups are established. Public smart contracts can't access private smart contracts in privacy groups. A node can belong to numerous privacy groups, and operations are controlled per group. Figure 4.4 defines two privacy groups. Alice, Bob, and Catty formed the group ABC, and Alice formed another with Catty and Doge (say, privacy group Id ACD). The privacy group Id is an alias. By using the privacy group Id, the system can submit a transaction to the group's nodes. Alice initializes a contract between Alice, Bob, and Catty to transmit a transaction, so only they may access it. Doge can't see transaction information. Alice initializes the transaction contract in a privacy group with Catty and Doge. Catty can't witness any transactions between them.

Private Transaction Manager (PTM) - In practical implementation, privacy groups and transaction privacy can be achieved using a private transaction manager running in every assured privacy capable node (Say, Alice, Bob, Catty, and Doge). The privacy transaction manager provides an API to connect with other privacy transaction managers and privacy enabled Ethereum clients. It develops a P2P self-discovery

network to find similar PTMs. It produces and maintains private/public key pairs for private communication.

Private Transaction Mechanism – Below diagram shows transaction flow in a privacy-enabled network.

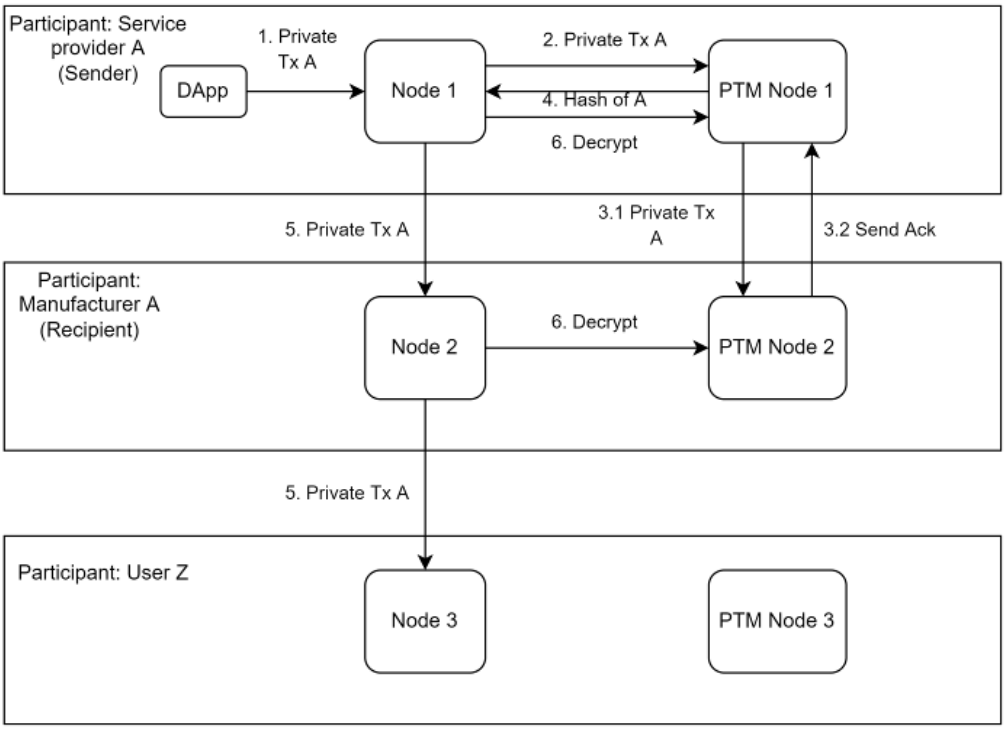


Figure 4. 5 Private transaction mechanism on a public network

Node 1, Node 2, and Node 3 are privacy-enabled nodes within the public and visible blockchain to all in Figure 5. Each node contains a Private Transaction Manager (PTM) that is running inside each privacy-enabled node. Service Provider A (SP-A), Manufacturer A (M-A), and User Z (U-Z) are participants in this network. Let's consider that Service Provider A (SP-A) initiates a transaction (Say, Order transaction.) with Manufacturer A (M-A) privately.

Following is the process flow-

1. SP-A sends a private transaction to Node 1 within the privacy group.
2. Node 1 serializes the private transaction and sends it to PTM 1 with participant details.
3. PTM 1 encrypts the data and distributes it among transaction participants.
4. PTM 1 returns the hash of encrypted data to Node 1

5. The private transaction data is replaced with a hash, and after signing, the hash is distributed in the network by marking the transaction as private.
6. All nodes can follow the transaction. However, only transaction participants can receive the decrypted data from the private transaction manager and execute the transaction.

In this flow Node 3, will only be able to recognize the encrypted hash but cannot decrypt the raw data.

With the aforementioned group of components and privacy flow understanding below the overall system integration architecture to enable the same for a business application.

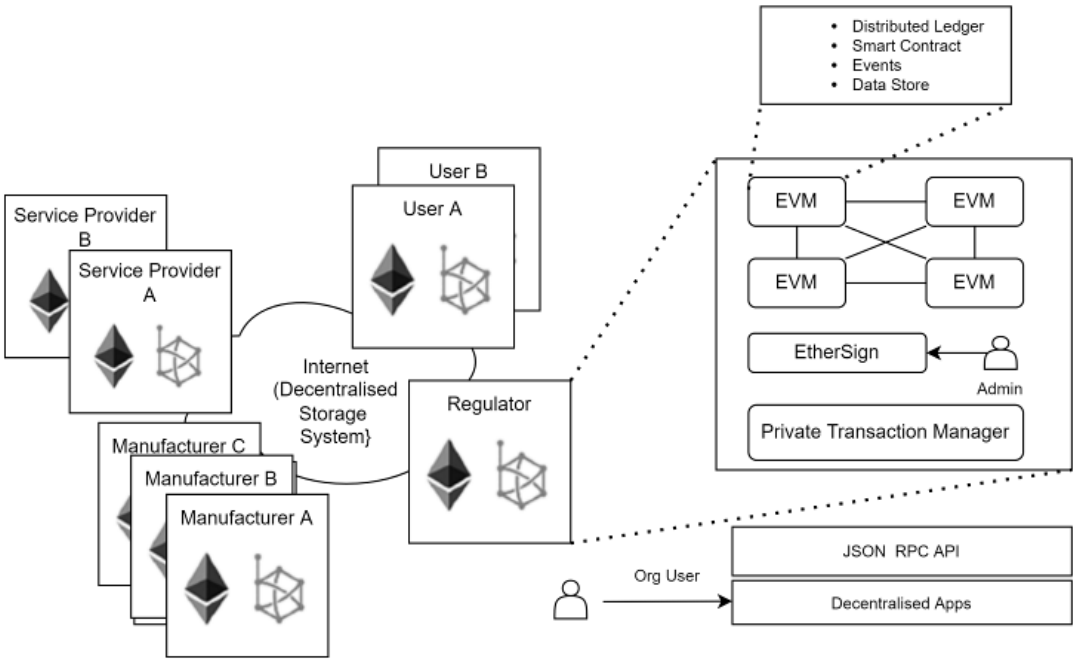


Figure 4. 6 Blockchain Network Architecture

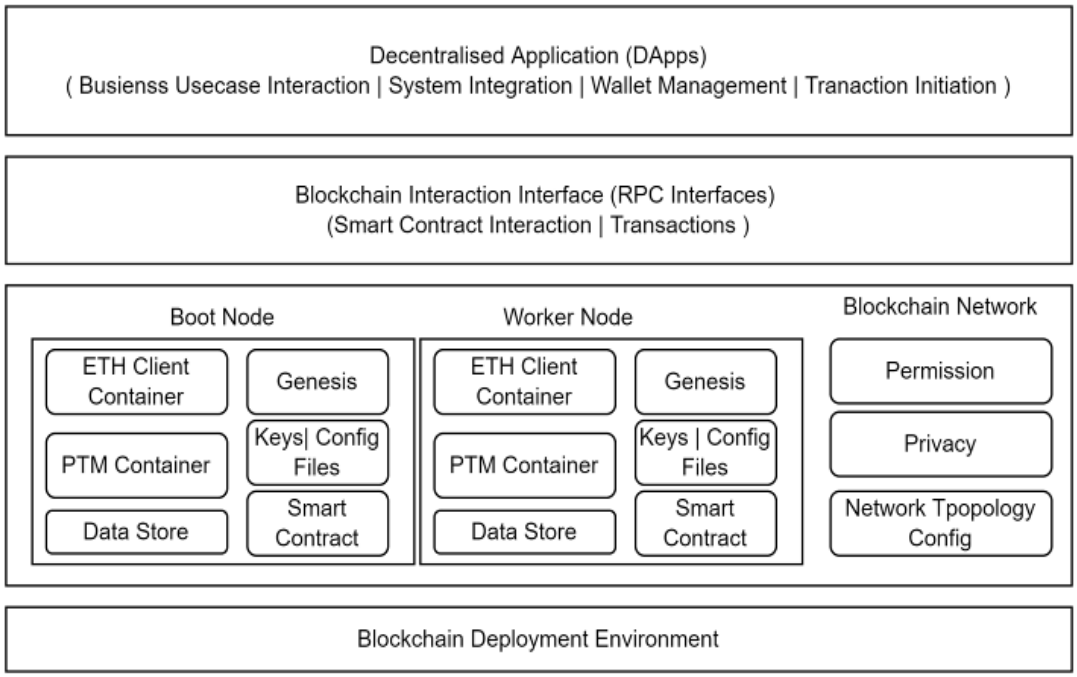


Figure 4. 7 Logical System Component Architecture

Figures 4.6 and 4.7 depict the overall architecture of the system as well as how the system components are coupled to fulfil the requirements for privacy imposed by decentralized business applications. The Boot Node is obligated to find all the system peers. When a worker node is introduced to the network, the boot node could automatically add it to the network. Every privacy-enabled node contains a PTM node, which is responsible for carrying out private transactions and producing an encrypted hash. RPC APIs are made available to the Ethereum blockchain client so that it can communicate with the network. These application programming interfaces (APIs) are used by decentralized applications (DApps) to perform commercial use cases. To be demonstrated, privacy and permissions are being focused. An in-depth description of my tests and the typical protocols that are followed to validate the core idea will be provided in the next paragraph.

This Proof of concept created three Hyperledger Besu nodes. One Node is a boot node, and the others are worker nodes. The consensus algorithm is PoW EthHash [88]. Below are system specifications and initialization instructions for the execution environment.

Operating System	Ubuntu Linux

Memory	16 GB
Disk space	64 GB
System	AWS t2.xlarge
Number of execution core	4 vCPU

Table 4. 1 Execution environment specification

Step 1: Setting up the Tessera environment –

Define tessera.conf file to keep all configurations in one place and execute to start the tessera node. The tessera is typically running in all three Besu nodes which will be defined in the next step.

```
> tessera$../../tessera/bin/tessera -keygen -filename nodeKey
> tessera$../../tessera/bin/tessera -configfile tessera.conf
```

Step 2: Three Node Besu setup

BootNode initialization. Initialize workers. Once worker nodes are functioning, peers are automatically discovered. All nodes are online and accepting client queries.

```
> besu --data-path=data --genesis-file=../genesis.json --permissions-accounts-contract-enabled --
permissions-accounts-contract-address "0x0000000000000000000000000000000000008888" --
permissions-nodes-contract-enabled  --permissions-nodes-contract-address
"0x0000000000000000000000000000000000009999" --permissions-nodes-contract-version=2 --rpc-http-
enabled --rpc-http-cors-origins="*" --rpc-http-api=ADMIN,ETH,NET,EEA,PRIV,PERM,IBFT,WEB3 --host-
allowlist="*" --privacy-enabled --privacy-url=http://127.0.0.1:9102 --privacy-public-key-
file=tessera/nodeKey.pub --min-gas-price=0 --miner-enabled --miner-
coinbase=fe3b557e8fb62b89f4916b721be55ceb828dbd73 --privacy-flexible-groups-enabled="true"
```

Worker node- Execution commands are the same for both nodes except for port details.

```
> besu --data-path=data --genesis-file=../genesis.json --
bootnodes=enode://853a2112d47517a791ad3c60515ae8b82bcb0a2f8ea305d5110378433c38e60d093d103e00eb5142
72b9ccf1b7de9114529e5ffad124cf3d337ccf364ed6ba0b@127.0.0.1:30303 --permissions-accounts-contract-enabled --
permissions-accounts-contract-address "0x0000000000000000000000000000000000008888" --permissions-nodes-
contract-enabled  --permissions-nodes-contract-address "0x0000000000000000000000000000000000009999" --
permissions-nodes-contract-version=2 --rpc-http-enabled --rpc-http-cors-origins="*" --rpc-http-
api=ADMIN,ETH,NET,EEA,PRIV,PERM,IBFT,WEB3 --host-allowlist="*" --p2p-port=30304 --rpc-http-port=8546 --
privacy-enabled --privacy-url=http://127.0.0.1:9202 --privacy-public-key-file=tessera/nodeKey.pub --min-gas-price=0 --
miner-enabled --miner-coinbase=fe3b557e8fb62b89f4916b721be55ceb828dbd73 --privacy-flexible-groups-
enabled="true"
```

```
pool-7-thread-3 | INFO  | BlockMiner | Produced #16,356 / 0 tx / 0 om / 0 (0.0%) gas /
(0x8fda741cf1391e49a057b91cc54861305908ee9207a0a40bf17e5dc02b85632d) in 0.022s
EthScheduler-Timer-0 | DEBUG | FullSyncTargetManager | Caught up to best peer: 16355,
Peers: 2
```

Step 3: Setting up the decentralized application to interact positively with the Besu nodes for deploying and managing smart contracts and transactions. Permissioning smart contract is deployed properly in Besu node through the decentralized app that is getting triggered whenever Admin wants to whitelist a new node or account within on-chain privacy Besu Network. Below are the deployed smart contract details on Besu Network.

```
Replacing 'NodeRules'
   ----------------------
   > transaction hash:
0x2a8ca7e6e5dacd1b0b5ace5636eaae9e9f5b2c396cd8ca810eab4b552398e4b4
   > Blocks: 0          Seconds: 0
   > contract address:    0xE03Ef2490316bfF9808d936eEe70f23896F07548
   > block number:        28674
   > block timestamp:     1653975937
   > account:             0xFE3B557E8Fb62b89F4916B721be55cEb828dBd73
   > balance:             57543
   > gas used:            2632495 (0x282b2f)
   > gas price:           0 gwei
   > value sent:          0 ETH
   > total cost:          0 ETH


   > Rules deployed with NodeIngress.address = 0x0000000000000000000000000000000000009999
   > and storageAddress = 0x6023FF0A8203ea32E737819B301D1672Dd2ECBE0
   > Rules.address 0xE03Ef2490316bfF9808d936eEe70f23896F07548
   >>> Set storage owner to Rules.address 0xE03Ef2490316bfF9808d936eEe70f23896F07548
   > Updated NodeIngress contract with NodeRules address =
0xE03Ef2490316bfF9808d936eEe70f23896F07548
```

```
Replacing 'AccountRules'
   ------------------------
   > transaction hash:
0x3f5bbb1eb1b87dba7b1018ab202868976d258b729b325f7b16b44e6dd3c34ee2
   > Blocks: 0          Seconds: 0
   > contract address:    0x6aA8b700cD034Ab4B897B59447f268b33B8cF699
   > block number:        28686
   > block timestamp:     1653975949
   > account:             0xFE3B557E8Fb62b89F4916B721be55cEb828dBd73
   > balance:             57567
   > gas used:            1817279 (0x1bbabf)
   > gas price:           0 gwei
   > value sent:          0 ETH
   > total cost:          0 ETH


   > Rules deployed with AccountIngress.address =
0x0000000000000000000000000000000000008888
   > and storageAddress = 0x7eF84473a4E772fB6aDfA1B0C6728A3dbf268Dd7
   >>> Set storage owner to Rules.address 0x6aA8b700cD034Ab4B897B59447f268b33B8cF699
   > Adding Initial Allowlisted Accounts ...fe3b557e8fb62b89f4916b721be55ceb828dbd73
   > Initial Allowlisted Accounts added: 0x627306090abab3a6e1400e9345bc60c78a8bef57
   > Updated AccountIngress contract with Rules address =
0x6aA8b700cD034Ab4B897B59447f268b33B8cF699
```

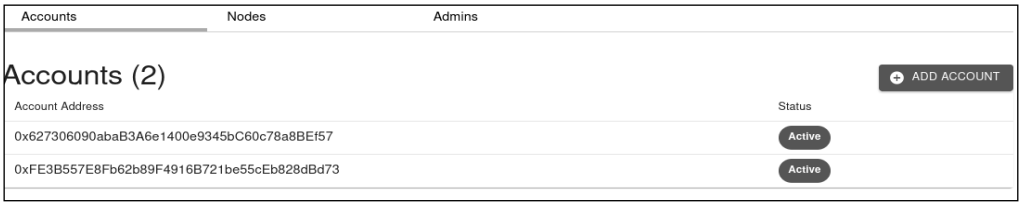
Step 4: Privacy group creation and enablement

Privacy group Id is randomly generated and assigned to a set of participant accounts. Any private transaction within the privacy group is visible to the qualified participants.

```
// Connecting Besu network and Creating privacy groupId

BesuNetwork besuNetwork = new BesuNetwork();
besuNetwork.privacyGroupId = Base64String.wrap(generateRandomBytes(32));
besuNetwork.rootNode = rootBesuNode;
PollingPrivateTransactionReceiptProcessor processor = new PollingPrivateTransactionReceiptProcessor(
rootBesuNode.getBesuNode(), 15000, 20);


// Creating onChain privacy group

String txHash = rootBesuNode.getBesuNode()
.privOnChainCreatePrivacyGroup(besuNetwork.getPrivacyGroupId(),
rootBesuNode.getCredential(),
rootBesuNode.getEnclaveKey(),
participantEnclaveKeyList)
.send()
.getTransactionHash();

//Creating onchain private transaction manager

BesuNode workingNode = besuNetwork.getRootNode();
PrivateTransactionManager tx_Node1 = new PrivateTransactionManager(workingNode.getBesuNode(),
workingNode.getCredential(), processor, BesuNode.CHAIN_ID, node1.getEnclaveKey(),
besuNetwork.getPrivacyGroupId(), Restriction.RESTRICTED);
```

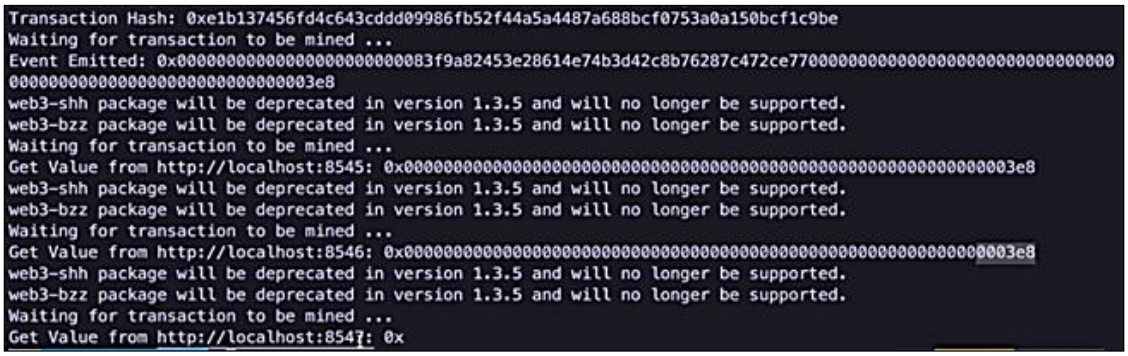
In the above code snippets, Base64String privacy group id is typically created and, on the chain, the privacy group is created with participant lists. The participant list contains a list of enclave keys of participants (public keys are generated in the tessera nodes.).

When Besu Network on-chain permissioning contracts (Node Rule contract and Account Rule contract) are installed and the decentralized application is operating, Admin can log in using its whitelisted Admin account. Admin configures app nodes and accounts. Only trusted nodes and accounts can participate. The web application can deactivate problematic nodes or accounts without affecting the network.



The admin account is always active and cannot be removed.

When a privacy-enabled transaction is submitted, all participating members can be able to see the transaction, however not participating members within the group cannot be able to see the transaction details. In the below diagram, as Node1 (Port: 8545) and Node2 (Port: 8546) are participating nodes, they both can verify the same transactions however Node 3 (Port: 8547) is unable to typically see the transaction.



The below flow diagram depicts the transaction flow in privacy enabled network based on the above execution.

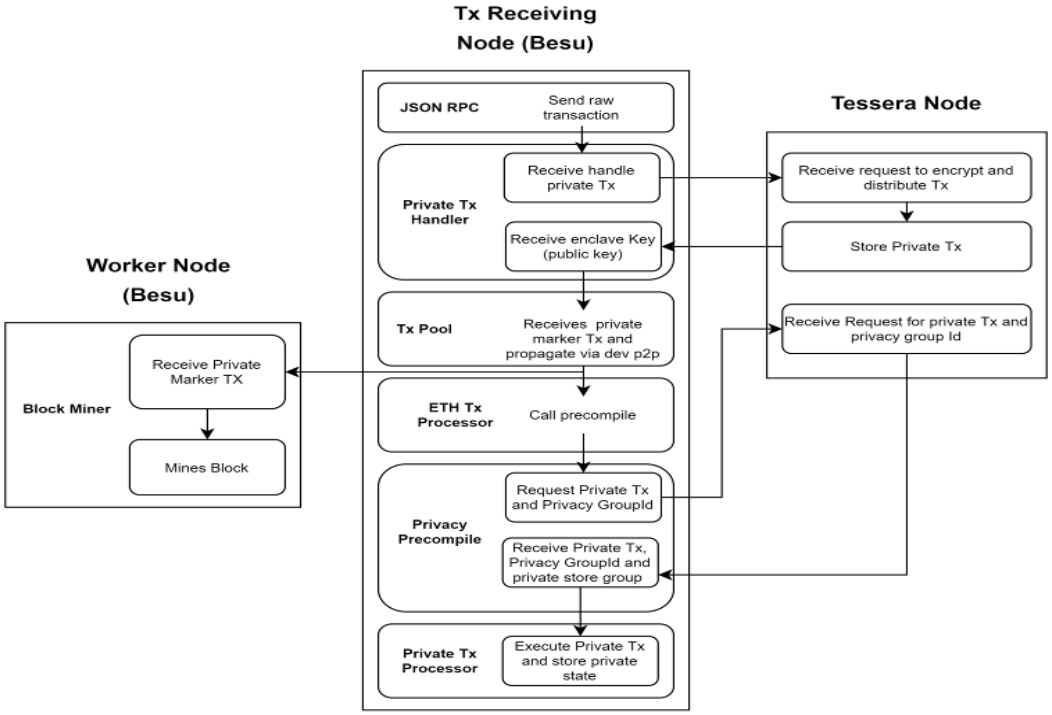


Figure 4. 8 Privacy transaction execution

In above Figure 4.8, the raw transaction is submitted with the privacyGroup Id, privateForm, which uniquely specifies the sender and restriction, to let the network know the transaction is restricted to intended participants only.

4.3 Conclusion and Future Directions

Using an example from an Internet of Things supply chain, the importance of having adequate privacy protections, permissions, and access management in public blockchain networks was stressed. The most recent research papers on privacy control mechanisms and architecture on private blockchains were analysed, and the results showed that public network restrictions linked to centralised control, semi-private transaction handling, closed network data architecture, closed group transaction access control, and mass adoption are all acceptable. It was recommended that access control and privacy features for the public network be implemented using privacy groups. The Ethereum (Besu) node's private transaction manager makes it possible for users to take part in several privacy groups while still using the same network. In order to prove the fundamental concept, the solution established a software infrastructure in the cloud that could run 3 + 3 Besu and Tessera nodes and carry out private transactions. The results

of the experiment show that transactions are frequently initiated by decentralized applications and that these transactions might be made visible to a subset of nodes based on the privacy group and access control settings. Smart contracts with on-chain permissioning and access to permission nodes and accounts. A network administrator can regulate vulnerabilities and threats by adding nodes and accounts to the network, which may then be revoked dynamically via the web. This architecture needs to be proven in several verticals using a broader software infrastructure, with each virtual machine instance running a different Besu and Tessera node. Only then will it be possible to establish a distributed ledger that is permissioned and able to protect users' privacy.

This chapter, cantered on IoT Privacy, Information Transparency, and Access Management Software Architecture, establishes the foundational framework for secure, permissioned, and transparent data handling across decentralized IoT environments. This architectural groundwork is not just critical in its own right it directly enables and strengthens the capabilities explored in the subsequent chapter on IoT Data Ingestion.

As data begins to flow at scale with increasing volume, velocity, and variety the principles defined in this chapter become indispensable. The use of privacy groups, permissioned smart contracts, and decentralized access management mechanisms ensures that privacy boundaries are preserved and access remains tightly controlled, even under high-throughput ingestion conditions. This seamless transition from privacy architecture to data ingestion highlights a cohesive and scalable design, where each layer builds upon the previous to maintain data integrity, confidentiality, and authorized usage. It ensures that the system is not only secure and privacy-respecting but also robust enough to meet the real-world demands of modern IoT deployments.

.

Chapter 5

5. Data Ingestion Software Architecture for Managing Unexpected Surges in Data Volume

In today's rapidly evolving landscape of social media and e-commerce, the importance of fast data processing and near real-time analytics cannot be overstated. The success of businesses in these domains' hinges on their ability to effectively harness data from a variety of sources, including structured, semi-structured, and unstructured data, in real-time or near real-time. Extracting valuable insights from this data torrent is not only a competitive advantage but also drives strategic decision-making processes. The temporal nature of data in this context is remarkably short-lived. Consider the scenario of a user exploring an e-commerce retailer's website for a specific product. At this very moment, providing recommendations for related products holds immense value for cross-selling. Furthermore, gathering and analysing user reviews, tweets, social media posts, and data from Internet of Things (IoT) devices becomes pivotal in understanding the prevailing sentiment in the market regarding product adoption and perception. A technology in enabling this ecosystem is the microservices-based architecture. This architectural approach compartmentalizes functionalities into discrete, loosely coupled services. This not only facilitates agility in development and deployment but also supports the scalability and manageability required for real-time data processing. With the internet's unprecedented growth over the past decade, coupled with a burgeoning user base, the Platform as a Service (PaaS) model is emerging as a prime choice for deploying applications. Its benefits encompass streamlined management, reduced overhead, and optimal resource allocation. PaaS solutions are poised to become the de facto standard, given the prevailing trends. The cloud's expanding influence has ushered in a paradigm shift in how industries perceive and leverage technology solutions. A cost-effective and highly scalable application deployment platform is the need of the hour, as businesses seek to capitalize on the cloud's advantages. Online retailers, particularly, seek robust solutions to seamlessly accommodate sudden surges in data

during special events like festive seasons. Ensuring uninterrupted service and a responsive user experience in such scenarios is imperative.

In response to these challenges and opportunities, innovative agent-based architecture is proposed. This architecture is designed to facilitate auto scaling [89] of applications, whether they are deployed on-premises or within a PaaS environment. The central tenet of this approach is the ability to sense data surges in real-time and dynamically adjust resource utilization. This adaptability ensures that performance and other non-functional requirements are maintained while minimizing infrastructure and service costs associated with cloud-based platforms. To substantiate the concept, a real-world application has been implemented focused on Twitter user sentiment analysis. This microservices-based application serves as a proof of concept, demonstrating continuous data flow into the system. This continuous data influx triggers the need for auto scaling, prompting the application to dynamically adjust its resources based on the incoming data surge. The framework laid out provides guidelines for this dynamic and adaptive auto scaling process, thus establishing a blueprint for efficient utilization of resources. In conclusion, the convergence of social media, e-commerce, and real-time data analytics necessitates the adoption of cutting-edge technologies and strategies. Fast data processing, near real-time analytics, microservices-based architectures, and cloud-based platforms are the pillars supporting this ecosystem. The proposed agent-based auto scaling architecture presents a solution to the challenges posed by sudden data surges, ensuring a cost-effective, scalable, and responsive platform that aligns with the evolving dynamics of the digital era.

5.1 IoT Data Tsunamis: Meeting the Challenges of Unexpected Data Surges

In 2023, Flipkart faced a major outage during its annual 'Big Billion Days' sale, leading to the suspension of its grocery delivery services due to an overwhelming surge in user traffic beginning October 8th. Similar issues occurred in October 2014, when Flipkart acknowledged that server crashes were caused by poor queuing mechanisms and

performance bottlenecks [104][105][106]. Such incidents reflect a broader challenge across digital platforms, especially during peak demand events or breaking news, where latency increases due to surging data. Over the past five years, the non-linear growth of internet-connected devices has dramatically raised user expectations for seamless and rapid digital interactions. To meet these demands, modern systems increasingly rely on cloud-based solutions, including load balancing algorithms, Content Delivery Networks (CDNs), and elastic cloud infrastructure, which dynamically distribute traffic, reduce latency, and allow real-time scaling. Traditional vertical or horizontal scaling often fails to respond efficiently to unpredictable data spikes. Advanced strategies now involve automated resource scaling, but determining optimal VM instance counts remains complex, requiring predictive analytics. Excessive provisioning leads to increased costs, while under-provisioning risks service disruption. Therefore, architectures capable of real-time data analysis and adaptive scaling are critical. A microservices-based approach [90][91], used in this research to process live Twitter data streams for sentiment analysis, allows independent scaling of components. Integrated with an agent-based data surge analyser [92], the system dynamically adjusts compute resources in response to load variations, ensuring uninterrupted performance and efficient cloud resource utilization—even during sudden traffic surges. To summarize, the primary challenges to address in the context of unforeseen IoT data surges are as follows,

- **Rising User Expectations**: Increasing connectivity and device usage have raised user expectations for uninterrupted and frictionless digital experiences, placing pressure on load balancing mechanisms.
- **Performance Challenges**: Sluggish-loading e-commerce websites during festive events and latency in accessing data during high-activity periods on social media highlight the need to address performance issues.
- **Real-Time Load Management**: An agent-based data surge analyser collaborates with the infrastructure to maintain load balancing and take appropriate measures during unforeseen data spikes, preserving consistent user experiences.
- **Optimization Challenges**: Balancing setup costs and operational expenses in load balancing strategies becomes complex in the face of nonlinear data volume growth, necessitating the use of historical data analysis and adaptable architectures.

- **Resource Allocation Flexibility**: Cloud-based strategies provide the flexibility to adjust VM instances, CPU capacity, and memory allocation as needed, no cloud provider-imposed resource restrictions, ensuring optimal resource utilization.
- **Resource Constraints**: The challenge lies in accurately determining the right number of VM instances in a cloud environment, striking a balance between over-provisioning, which escalates operational costs, and under-provisioning, which could lead to performance issues.

This research presents a dynamic, scalable, and real-time data ingestion architecture designed to effectively manage unexpected data surges in IoT environments. Built on a hybrid of Lambda Architecture and Apache Storm's Spout and Bolt topology, the system enables both real-time and batch processing through a modular, microservices-based framework. A key contribution is the implementation of a Rebalancer Agent, which continuously monitors queue metrics using principles from Little's Law to dynamically scale processing units based on incoming data rates, thereby ensuring consistent performance and fault tolerance during traffic spikes. The architecture extends into cloud PaaS environments, offering adaptive memory management and SLA-compliant scaling strategies that prevent resource overutilization or failure. Technologies such as Spring Boot, MongoDB, Weka, and JMX API were integrated to create a robust proof of concept, validated through simulated high-load scenarios using real-time Twitter streams. The system demonstrates how intelligent rebalancing and service orchestration can maintain system integrity, responsiveness, and reliability in the face of unpredictable and high-volume data streams, offering a forward-looking solution for scalable, cloud-native IoT applications.

5.2 Empowering IoT Data Surge Resilience: System Design, Solution Architecture, and Execution

In an era where the Internet of Things is reshaping industries and generating unprecedented volumes of data, this resource delves into the critical aspects of system design, solution architecture, and execution to ensure that organizations can harness the

full potential of IoT data while maintaining the integrity and responsiveness of their systems. From scalable architectures to fault-tolerant designs, this examination equips professionals with the knowledge and tools necessary to navigate the complexities of IoT data surges successfully.

5.2.1 Lambda Architecture

The solution topology is based on the Lambda Architecture at its core. Lambda architecture is a sophisticated data processing framework that has become instrumental in managing the complexities of fast data ingestion and analysis. This architecture is designed to handle the dual challenges of batch and real-time data processing simultaneously, providing a comprehensive solution for organizations dealing with diverse data streams. The Lambda architecture is structured into three layers: the batch layer, the speed layer, and the serving layer. The batch layer manages the processing of historical data in large, incremental batches, ensuring accuracy and completeness. On the other hand, the speed layer handles real-time data in small, rapid increments, delivering timely insights. The serving layer combines results from both layers to provide a unified and up-to-date view of the data. This architecture excels in maintaining fault tolerance and scalability, which are crucial for managing the vast volumes of data generated in today's digital landscape.

Lambda architecture plays a crucial role in managing IoT data surges by providing a scalable and reliable framework for processing and analysing large volumes of data generated by IoT entities in both real-time and batch modes. Here's how Lambda architecture helps address the challenges posed by IoT data surges:

- **Real-time Data Processing (Speed Layer):** IoT devices often produce data in real-time or with very low latency. The speed layer of Lambda architecture handles this data by ingesting and processing it quickly. This allows organizations to react immediately to critical events and derive real-time insights from IoT data, such as monitoring equipment status, tracking vehicle locations, or responding to security breaches promptly.
- **Batch Data Processing (Batch Layer):** IoT devices also generate historical data, which may need to be analysed for trends, anomalies, or compliance reasons. The batch layer efficiently manages and processes large volumes of historical IoT data

in incremental batches. This ensures that organizations can perform in-depth analyses, such as predictive maintenance, long-term performance evaluations, and compliance audits.

- **Serving Layer Integration:** The serving layer in Lambda architecture harmonizes and combines results from both the speed and batch layers, creating a unified view of the data. This integration allows users to access a complete and up-to-date dataset that includes real-time insights as well as historical context. In the context of IoT data surges, this means that organizations can make informed decisions based on a holistic understanding of their data.
- **Scalability:** IoT data surges can be unpredictable and vary greatly in volume. Lambda architecture is inherently scalable, allowing organizations to add more resources to accommodate spikes in data traffic as needed. This scalability ensures that the system remains responsive even during periods of intense data generation, preventing bottlenecks and delays.
- **Fault Tolerance:** IoT systems require high levels of reliability, as downtime can have significant consequences. Lambda architecture is designed with fault tolerance in mind, ensuring that data processing continues even in the presence of failures. This resilience is critical for maintaining data integrity and system availability during IoT data surges.
- **Adaptability:** IoT applications and use cases can evolve rapidly. The flexibility of Lambda architecture enables organizations to adapt their data processing pipelines to changing requirements without overhauling their entire infrastructure. This adaptability ensures that IoT systems can continue to meet the needs of the organization as it grows and evolves.

Lambda architecture is a powerful framework for managing IoT data surges by providing a comprehensive solution for real-time and batch data processing. It ensures that organizations can harness the full potential of IoT data, making timely decisions, extracting valuable insights, and maintaining system reliability even in the face of unpredictable data surges.

5.2.2 Spout and Bolt Topology

The Spout and Bolt topology constitutes a cornerstone in the realm of distributed stream processing systems, notably exemplified by Apache Storm. This architectural paradigm is instrumental in addressing the pressing challenges posed by the real-time processing of vast and continuous data streams, which have become increasingly ubiquitous in today's data-driven landscape.

Spouts, the initial point of contact in this topology, play the role of data ingress. They serve as the conduits for fetching data from an array of sources, which can encompass anything from sensor readings and social media posts to financial transactions and log files. Spouts have the essential function of ingesting this raw data and subsequently emitting it into the processing pipeline as discrete units called "tuples." This continuous and seamless data ingestion ensures an uninterrupted flow of information into the system.

Bolts, conversely, are the core processing units within this framework. Each Bolt assumes a specific function or task in the data processing workflow. They are responsible for carrying out various operations on the incoming tuples, which can range from basic filtering and aggregation to more complex transformations and enrichments. Bolts apply custom logic and rules to manipulate the data as required by the application's objectives. Crucially, Bolts can also emit new tuples as their output, and this output can be directed to other Bolts or back to the Spouts, resulting in a dynamic and interconnected data processing pipeline.

The elegance of the Spout and Bolt topology lies in its adaptability, scalability, and fault tolerance. Spouts can be scaled horizontally, enabling the system to efficiently manage high-velocity data streams by distributing the load across multiple instances. Bolts can be added, removed, or adjusted with ease to accommodate changing processing requirements or evolving business needs. Furthermore, the architecture is inherently fault-tolerant, ensuring that data is not lost even in the event of a Bolt failure. The system orchestrates automatic tuple reprocessing to guarantee data integrity and system resilience.

In essence, the Spout and Bolt topology serves as the bedrock for the development of robust, scalable, and agile real-time stream processing applications. It empowers organizations to extract actionable insights, make informed decisions, and trigger responsive actions from streaming data. This dynamic framework stands at the forefront of modern data processing, facilitating the transformation of raw data into valuable

intelligence, all in real-time. As the data landscape continues to evolve, the Spout and Bolt topology remains an indispensable tool for leveraging the potential of streaming data effectively.

The Spout and Bolt architectural topology bring notable benefits in the management of IoT data surges, effectively tackling the issues arising from the substantial upsurge in data emanating from IoT devices. Let's explore how this architecture efficiently handles IoT data surges:

- **Scalability**: IoT data surges can be sudden and unpredictable. Spout and Bolt architecture allows for horizontal scalability, meaning you can easily add more Spout and Bolt instances to handle the increased data volume. This scalability ensures that your system can cope with surges in data without performance degradation.
- **Real-time Processing (Bolts):** IoT often requires real-time or near-real-time processing, especially in applications like monitoring, predictive maintenance, or emergency response. Bolts in this architecture are well-suited for real-time data processing tasks. They can filter, transform, or aggregate incoming IoT data on the fly, allowing organizations to respond promptly to critical events.
- **Batch Processing (Batch Bolts):** While real-time processing is essential for immediate insights, historical data analysis is equally crucial in IoT applications. Batch Bolts in the Spout and Bolt architecture can efficiently handle large volumes of historical data in incremental batches. This capability enables organizations to conduct in-depth analyses, identify long-term trends, and ensure compliance with regulations.
- **Adaptability:** IoT applications evolve over time, and the data they generate can change in volume and structure. Spout and Bolt architecture's adaptability allows you to modify or add new Bolts to accommodate changing data processing needs. This flexibility ensures that your system remains relevant and effective as your IoT ecosystem grows.
- **Fault Tolerance:** IoT systems need to be highly reliable. Any downtime or data loss can have severe consequences. The Spout and Bolt architecture ensures fault tolerance by reprocessing data in case of failures, preventing data loss and maintaining system integrity, even during data surges.

- **Complex Event Processing (CEP):** IoT data often involves complex event processing, such as pattern recognition or anomaly detection. Bolts can be customized to perform these tasks, helping organizations identify critical events and take immediate actions in response to IoT data surges.

The Spout and Bolt architecture provides a robust framework for managing and controlling IoT data surges. It offers scalability, real-time processing capabilities, adaptability, and fault tolerance, making it a valuable tool for organizations dealing with the challenges posed by the ever-expanding world of IoT data. By leveraging this architecture, organizations can harness the full potential of IoT data while maintaining system responsiveness and reliability, even in the face of surges in data volume.

5.2.3 Solution Architecture

The Basic solution topology architecture is as follows.

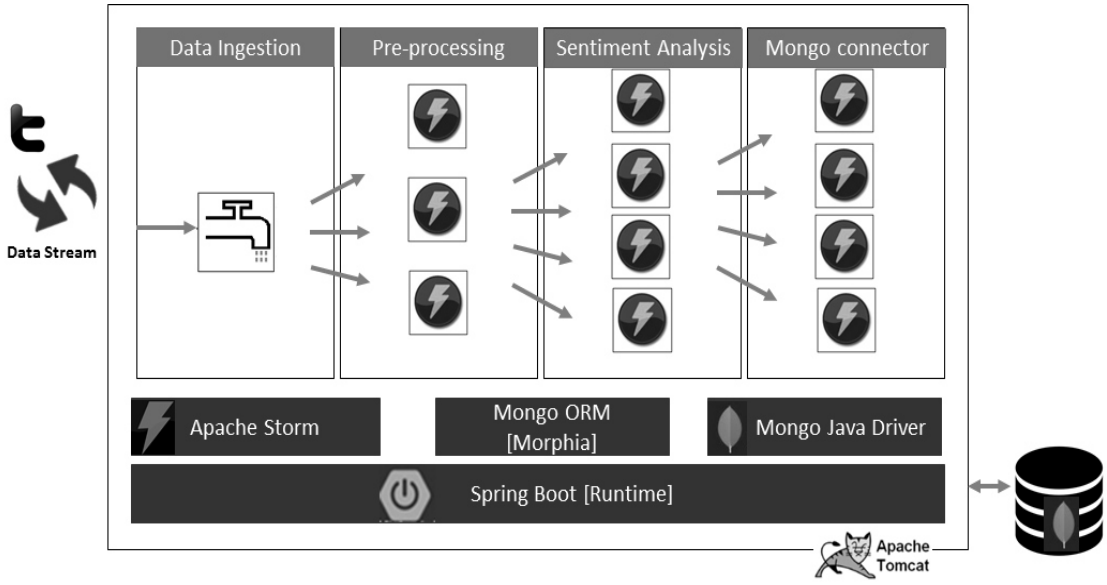


Figure 5. 1 Basic solution topology

Within the Lambda architectural framework, a dynamic system has been devised for the ingestion of live Twitter streams via a crucial component known as the Spout. The Spout, intricately configured with a designated data source, establishes a logical connection to the Twitter application, from which it periodically retrieves new tweets at predefined intervals. Spouts are capable of being categorized as either reliable or unreliable, with the former possessing the capability to retransmit a tuple in the event

of its failure to undergo processing by the Storm system. In contrast, the latter promptly discards emitted tuples upon their emission. It is noteworthy that Spouts exhibit versatility, allowing them to concurrently emit multiple data streams.

Complementing the Spout are the Bolts, which function as the processing powerhouses within the topology. Bolts demonstrate remarkable versatility, with the capacity to execute a wide spectrum of tasks, encompassing filtering, function execution, aggregation, database interactions, and even complex operations such as dataset joins. Bolts are also adept at implementing straightforward stream transformations, although intricate transformations frequently necessitate multiple steps, each executed by distinct Bolts. Within the topology, Bolts are thoughtfully partitioned into distinct logical phases: Preprocessing, Sentiment Analysis, and Mongo Connector. The Preprocessing phase assumes responsibility for a range of tasks, including the elimination of non-English content, the filtration of extraneous words, and the removal of alphanumeric characters. Subsequently, the Sentiment Analysis phase engages algorithms in the classification of English tweets based on their sentiment, complete with a confidence rating. Finally, the Mongo Connector Bolts are configured to establish an interface with a MongoDB NoSQL database, streamlining the process of storing classified tweets. he adaptability of the number of Bolt instances within each phase is contingent upon variables such as the incoming data flow rate and the time required for processing at each stage. Although this topology capably manages constant or minimally fluctuating data rates, it does face challenges related to scalability during runtime. The continuous flow of data remains uninterrupted, and making substantive alterations to the topology necessitates the temporary suspension of the application—a practical impracticality within real-world scenarios. Furthermore, the initial definition of the topology is grounded in historical data, thereby rendering it susceptible to breakdowns when confronted with sudden surges in data flow. In response to the challenge posed by unanticipated data surges, the imperative of real-time monitoring of incoming streams becomes apparent. The system should proactively scale up by rebalancing the topology during runtime, thereby ensuring its resilience in the face of unpredicted inundations of data.

Here is the suggested architectural framework to address this problem statement.

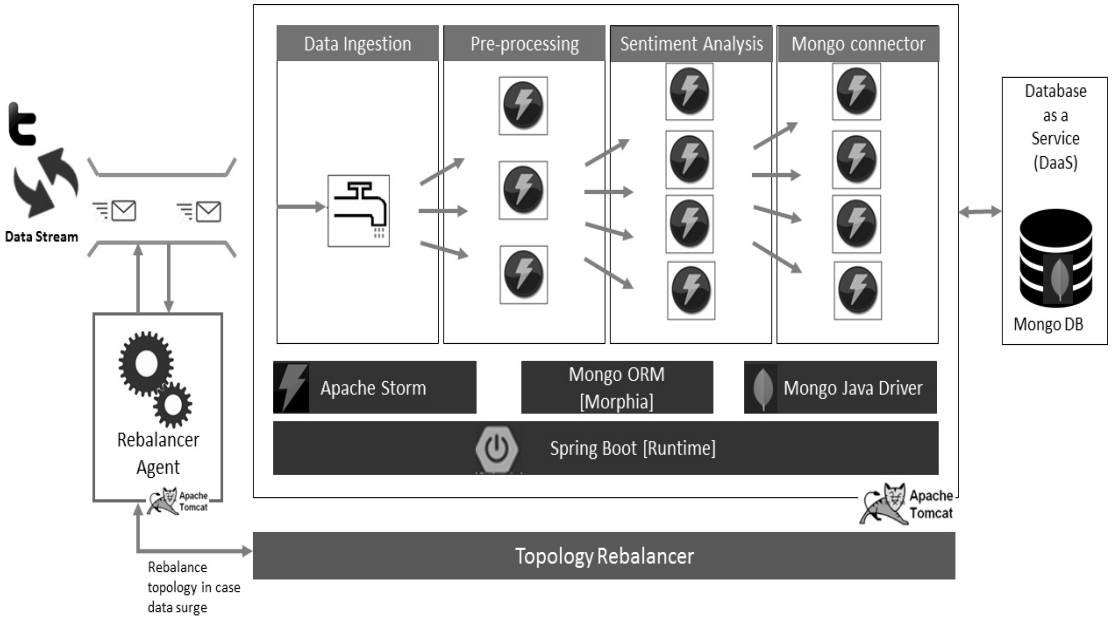


Figure 5. 2 Solution topology with Rebalancer

An enhancement to the previous solution architecture involves the introduction of a constrained-capacity queue positioned between the Twitter data source and the network topology. To oversee this queue and manage data surges effectively, a Rebalancer Agent (RA) has been implemented. This Rebalancer Agent continually monitors the flow of data within the Data Surge Protection (DSP) queue, employing principles akin to Little's Law [93] to gauge the data movement dynamics. It promptly triggers a rebalancing action when the queue length approaches a preconfigured threshold level. This threshold level is dynamically determined, considering the available queue storage and real-time queue elements. The following steps outline the execution process for defining the rebalancing strategy in response to unexpected data surges.

QueueSize = N

RebalanceCounter = 0

Threshold = 80% of N

1. Get incoming message rate (Iin)
2. Get outgoing message rate (Iout)
3. Compute average number of messages in queue

   AvgQueueLength = |Iout - Iin|

4. If AvgQueueLength >= Threshold

```
    Execute RebalanceTopology
    IncrementCounter = function (JMX parameters)
    RebalanceCounter = (RebalanceCounter + IncrementCounter)
  Else
    DecrementCounter = Function(JMX parameters)
    RebalanceCounter = (RebalanceCounter + DecrementCounter)
    Continue
```

At the outset, the queue size is established as 'N'. The average rate at which messages are ingested into the DSP (Data Surge Protection) queue is denoted by 'Iin', while the rate at which messages are consumed from the DSP queue is represented by 'Iout'. It's important to note that 'Iin' is essentially dictated by the behaviour of external data sources, making it beyond system control. However, control can be exercised over the consumption side, represented by 'Iout'. The core of this rebalancing algorithm is a periodic scan of the DSP queue to monitor its length. I initially set a threshold at 80 percent of the total queue size. When the average queue length surpasses this threshold, the rebalancing algorithm is triggered. Its primary objective is to determine the number of instances that should be increased to effectively handle sudden data surges, ensuring robust protection. It's worth emphasizing that the algorithm isn't solely about scaling up. It can be extended to take adaptive measures, including instance reduction, when the incoming data rate decreases. This proactive approach serves two vital purposes: it provides an extra layer of precaution against potential data surges and contributes to the overall stability of the rebalancing process. In essence, this algorithm serves as a dynamic safeguard, optimizing resource allocation to maintain the system's responsiveness and integrity in the face of fluctuating data flows.

5.2.3.1 Microservice Architecture-based Topology Definition

A novel method for addressing the complexities of safeguarding against dynamic data surges involves the structured delineation of micro-components. Microservices architecture [94] [95] has been instrumental in defining component boundaries based on functional decomposition. Within this architecture, it becomes apparent that the rate at which data is ingested into the Spout, processed by the pre-processing engine,

sentiment analyser engine, and eventually connected to the data connector introduces a delay in processing each tweet. The advantage of this architecture is made evident when it is considered that the pre-processing engine and analyser engine are implemented as microservices within the Java Virtual Machine (JVM), allowing them to be dynamically scaled to achieve processing synchronicity.

In the context of this topology, each Bolt executes a specific functionality, thus aligning with the microservices philosophy. Each functional boundary is clearly defined as microservices, complete with the appropriate interfaces for interaction. The interplay between functions across boundaries is governed by well-defined message contracts, ensuring seamless communication. Initially, the topology begins with a single instance of the Spout and Bolts at each layer within a single JVM. The pivotal moment occurs when the rebalancing algorithm evaluates the incoming data rate via the DSP queue. It dynamically determines how many instances are needed at each processing layer based on this assessment. Consequently, the rebalancing act comes into play, creating the requisite number of microservice instances in an operational state to effectively safeguard against sudden data surges. This dynamic rebalancing is made possible by the microservices and micro-component-based architectural definition, underlining the adaptability and scalability inherent in this approach.

5.2.3.2 Cloud PaaS Based Architecture

Extending the rebalancing architecture into a cloud Platform as a Service (PaaS) environment offers significant advantages. In the current landscape of deploying applications in cloud PaaS, developers typically need to specify the number of instances required for each service, which is crucial for scaling the application effectively, whether it involves scaling up or down. Additionally, developers need to make determinations regarding memory allocation and the instances necessary as part of capacity planning in the cloud. However, the consideration of memory and space requirements has considerably diminished over recent years in cloud PaaS environments. Nowadays, it often translates into low, static, flat costs, thanks to the offerings of many cloud PaaS providers. The primary challenge in capacity planning then centres around precisely defining the number of instances needed for each component to consistently meet the desired service level agreements (SLAs) on a 24 x

7 basis. In cases where an application encompasses numerous services, keeping track of their usage and implementing responsive actions becomes an intricate task. The analogy of not being able to change the wheels of a running train comes to mind. Developers must rely on usage analysis, often facilitated by Web API gateways or log data, to make informed decisions about the optimal number of instances required for all components. This process has a cascading effect on downstream components, affecting their processing capabilities.

Hence, accurately foreseeing and managing all components to maintain SLAs at the desired level becomes a challenging endeavour. This complexity underscores the importance of leveraging the rebalancing architecture, particularly in the context of cloud PaaS, as it offers a dynamic and responsive solution to manage these intricate capacity planning issues effectively.

Contemporary cloud applications face a notable challenge concerning runtime scalability, particularly in terms of memory utilization. Typically, in the configuration of each application, a minimum memory threshold is defined, often set at 1.3 times the available memory capacity during peak load periods. This configuration is a precautionary measure to mitigate the risk of encountering OutOfMemory runtime errors in the event of an abrupt spike in memory demand. However, this approach introduces a level of uncertainty, particularly in real-time IoT applications, where ensuring 99% application uptime is critical. If the available memory limits are exceeded, it can result in the application coming to a halt, potentially causing recurring disruptions for applications running within the same PaaS Org space.

Another facet of this issue pertains to the internal memory consumption within the application. To pre-empt the possibility of a sudden application memory shortage, developers often initialize the heap size to 1.3 times the required memory. However, in practice, much of this additional memory often remains unutilized, effectively translating into wastage for which usage costs are incurred by PaaS providers. Therefore, a robust architecture should possess the scalability to address out-of-memory concerns triggered by a rapid surge in processing demands. Simultaneously, it should fully optimize the use of available internal application memory, ensuring that the allocated heap is maximally utilized before resorting to requesting additional memory at runtime.

The architecture outlined above readily extends to cloud environments, offering a viable solution to both challenges.

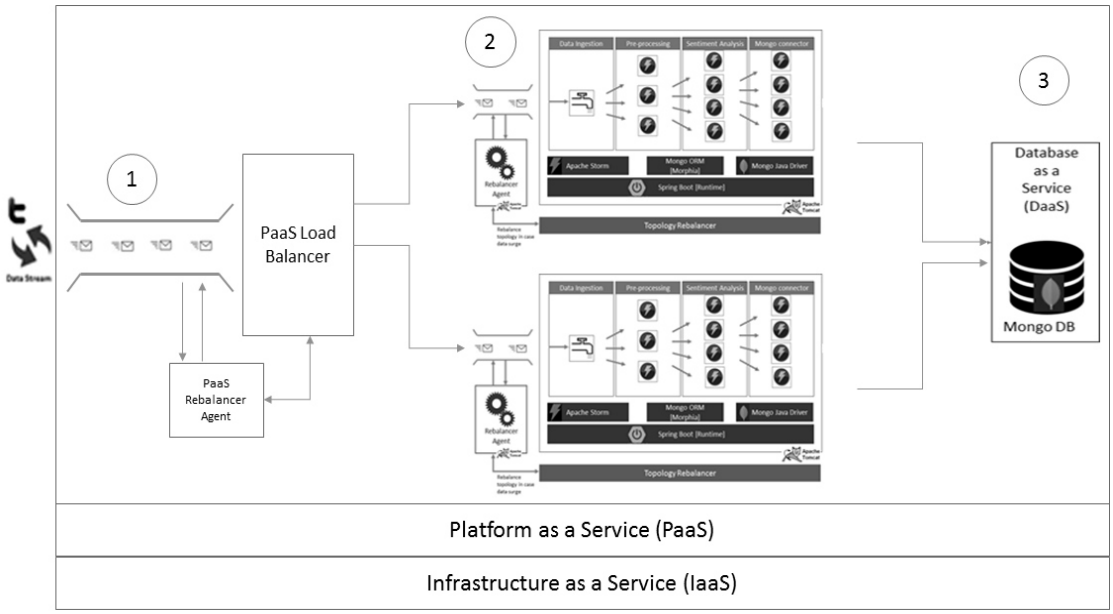


Figure 5. 3 Rebalancing topology architecture in PaaS

In the architectural framework outlined above, the key steps involved in handling real-time data surges can be delineated:

Step 1: The real-time data flows into an input load balancer queue. Here, an Adaptive PaaS Rebalance Agent closely monitors this input queue to make informed decisions about the number of application instances required. The PaaS load balancer plays a pivotal role in this step, determining the precise count of application instances needed in response to a sudden surge in data within the input queue. It then efficiently distributes incoming messages across the existing instances.

Step 2: Each application operates within its own application heap. Inside the heap of each application, the DSP (Data Surge Protection) and queue rebalancer take on the responsibility of deciding how many bolt instances are necessary to meet the demand for processing incoming messages. This step optimizes the allocation of processing resources within each application instance.

Step 3: Every application instance is equipped with its own set of data persistence bolt units, facilitating connections to database service instances in the cloud. These units

enable the seamless storage of data into the document store, ensuring the persistence of crucial information.

In essence, Step 1 harmonizes with Step 2 to ascertain the number of application instances needed in alignment with the processing unit requirements determined by Step 2. The overarching objective of this architecture prioritizes the scaling of application processing power by increasing heap sizes over scaling the number of application instances, particularly when such scaling is deemed necessary at runtime. This approach provides cloud architects with the utmost flexibility to design microservice-based applications and determine the appropriate number of instances required to provide maximum protection against unexpected data surges. It's an approach that fosters efficiency and adaptability in the face of dynamic data demands.

5.2.4 Solution Implementation

5.2.4.1 Technology Key Components

A microservices-based architecture rooted in Java has been embraced to establish the core of the real-time data ingestion framework. The infrastructure, based on Apache Storm, a robust real-time data ingestion engine, has MongoDB serving as the backend document storage. To ensure flexibility and mitigate the risk of vendor lock-in, the server-side framework has been implemented using the Pivotal Spring stack. The process model is activated through a Spring Boot-based microservices architecture. The incorporation of a sentiment analyser microservice into the system has been achieved by leveraging the Weka machine learning library. Additionally, the implementation of the rebalancer topology is carried out using the Java JMX API, which periodically monitors the DSP queue to guarantee smooth operation. While application development and deployment are facilitated by the Spring framework, the foundation for flexible spout and bolt-based framework is provided by the Apache Storm library. This architecture is meticulously designed around the concept of segregating responsibilities, fostering modularity and scalability. Presented below is a tabular representation outlining the primary software and technologies constituting the backbone of the system:

Software	Version	Usage
JDK7	1.7.72	JDK and JVM installation
Mondo DB	2.6.5	Act as a backend document-based data storage.
Pivotal Spring Ecosystem	2.6.0	Spring boot-based spring ecosystem to build micro service-based application
Apache Tomcat	8.0.32	Embedded Web server for micro service application deployment.
Twitter4j	3.0.3	Twitter java API to ingest live twitter data into application
Morphia	1.0.0	Used for Process modelling.
Weka	3.7.10	Collection of machine learning algorithms for data mining tasks.
Unirest-java	1.4.5	REST client implementation library.
Activemq Broker	5.12.3	DSP Queue implementation library.
Apache Storm	0.9.0.1	Define spout and bolt topology as application backbone.

Table 5. 1 Software employed and their functions

5.2.4.2 Operational Infrastructure

The infrastructure for this application development incorporates a Spring Boot-based Apache Storm real-time data ingestion framework, seamlessly integrated with MongoDB to offer comprehensive support for the software infrastructure layer. Custom-designed microservices are deployed within an embedded Tomcat server, ensuring efficient and reliable execution. The table below provides a detailed breakdown of the server configuration for the real-time data ingestion engine:

System Configuration	Details
Data Ingestion Framework	Spring Boot-based Apache Storm
Backend Data Storage	MongoDB
Deployment Environment	Embedded Tomcat Server
Custom Microservices Deployment	Integrated for efficient execution
Operating System	Windows 10
Memory	8 GB
Disk space	1TB
Processor	Intel family i3 generation
Number of execution Core	4

Table 5. 2 Server configuration

This combination of technologies forms a robust execution environment that underpins the functionality of the real-time data ingestion engine. Here are the definitions for the standalone application packages and their interactions.

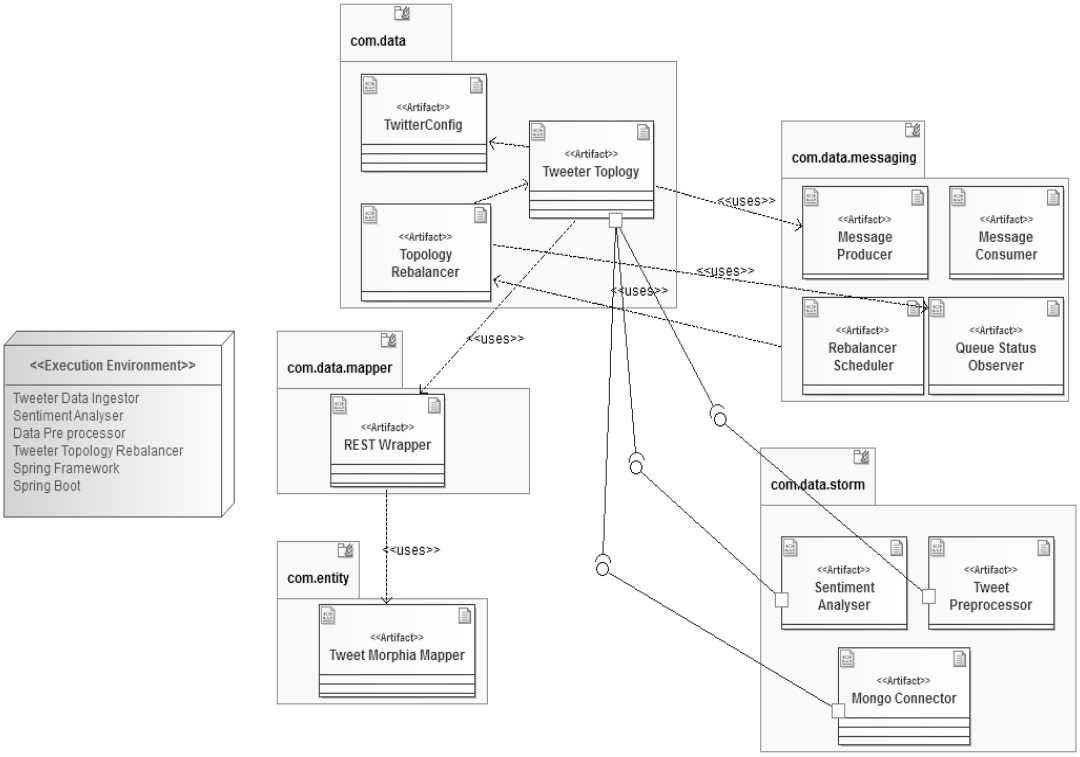


Figure 5. 4 Application package definition and execution environment

5.2.4.3 Execution of the Proposed Concept

To emulate real-time data surge scenarios, I instantiate multiple instances of the Twitter spout, which serves as the data ingestion source. I deliberately set a low scheduler time window and maintain a short data surge protection queue length to ensure the rapid ingestion of a substantial volume of messages within a brief time frame, thereby simulating the data surge phenomenon. In response to a sudden surge in data volume, the rebalancer engine dynamically reconfigures the number of bolt instances (processing units) to restore system stability and effectively control the surge.

The rebalancer engine employs asynchronous data collection via the JMX API, periodically gathering key parameters from the Data Surge Protection (DSP) queue. These parameters include average enqueue time, consumer count, queue size, and average dequeue time. They are collected as part of each scheduled job execution, contributing to the execution of the rebalancer algorithm. When the average number of messages in the queue surpasses a predefined threshold, an event is triggered, prompting the rebalancer to adjust the topology by scaling the processing units to the desired count. The performance results, based on the configured execution parameters, are presented below to provide insights into the system's behaviour under various conditions.

Parameters:

- The threshold for the number of messages in the queue is set at 8000, equivalent to 80% of N.
- The rebalancer scheduler is configured to run every 10 seconds.
- The maximum number of processing units is capped at 5, initially starting with 1.
- To mimic a data surge within the system, I incrementally instantiate 3 instances of the data ingested component.

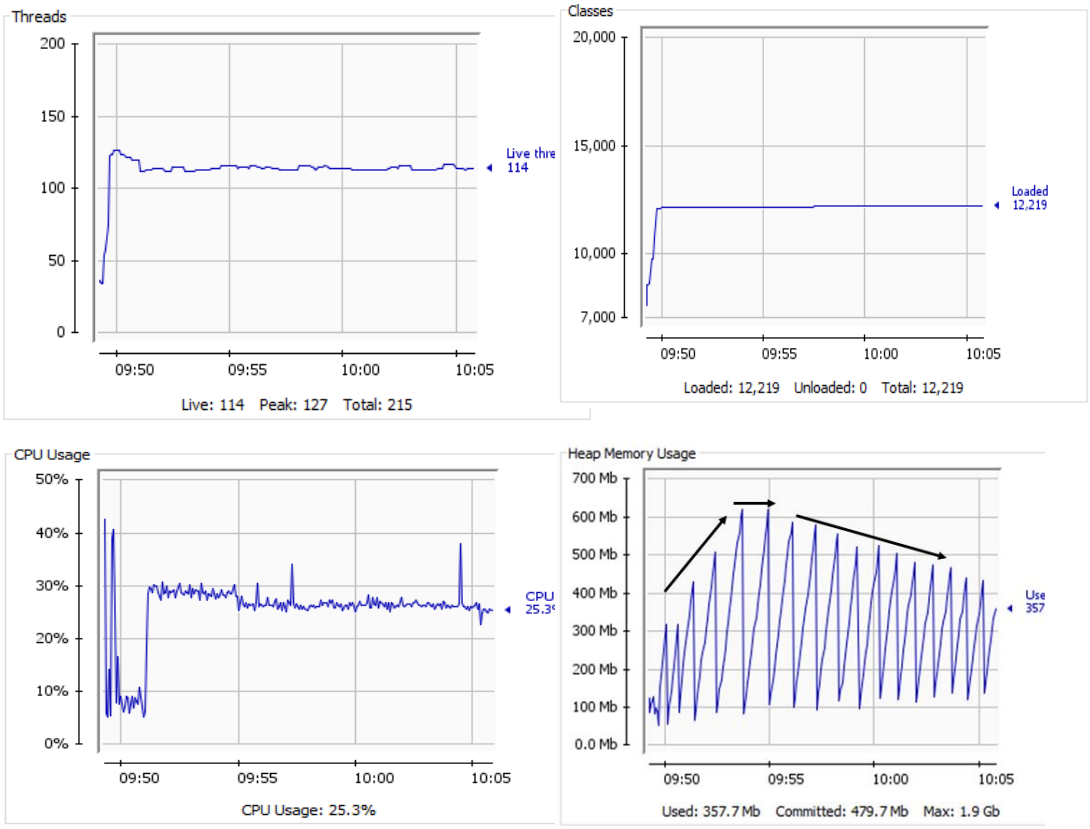


Figure 5. 5 JVM memory and CPU usage

The heap usage graph provides valuable insights into the relationship between memory consumption and the influx of messages, particularly during data surges. It's evident that as the number of messages increases (indicative of a data surge), memory consumption also rises. However, this surge in memory usage is temporary and stabilizes as the data ingestion rate increases over a relatively short period.

Upon the initial execution of the scheduler job, the rebalancer algorithm promptly detects the abrupt data surge within the system. In response, the rebalancer dynamically reconfigures the topology by increasing the number of processing units within the same JVM process. This strategic adjustment enables the system to process a higher volume of messages, resulting in a gradual reduction in memory consumption. Once the data surge stabilizes, the rebalancer gradually reduces the number of processing units to optimize resource utilization.

The memory usage graph displays a sawtooth pattern, indicative of effective garbage collection and the absence of memory leaks during the rebalancing process. Meanwhile, the CPU usage graph exhibits occasional spikes in the later stages, corresponding to the execution of rebalancing operations. However, apart from these instances, the system

demonstrates remarkably stable CPU usage throughout. Additionally, the number of execution threads within the system and the rate of classes loaded remain relatively constant, underscoring the system's stability and consistent performance.

5.2.4.4 PaaS Cost Implications

PaaS platform costs can fluctuate based on the extent of services used and the level of usage. Typically, PaaS is structured around a subscription-based, pay-as-you-go pricing model, offering businesses the flexibility to access and scale services according to their evolving needs. There are two prevalent pricing models within PaaS offerings: pay by the hour or pay by the month. For instance, an Amazon EC2 instance featuring 16 CPUs, 64 GB of memory, and on-demand scaling, without any additional services, can incur a monthly cost of approximately $630.

Now, let's consider a scenario where 8 application instances are deployed, each with a maximum heap size of around 2 GB. These instances collectively consume all available memory on the EC2 instance. Traditionally, if an application needs to scale further, it necessitates the purchase of additional memory or a higher-tier processing unit, even though there might be unused memory within the heap of each application. This over-allocation of memory is a precautionary measure against data surges.

However, by implementing the architecture described above, there's no need to set the maximum heap size upfront. Instead, with minimal allocation, the dynamic behaviour of the rebalancing algorithm autonomously determines when and how to scale memory processing units. In the long term, this approach is poised to significantly reduce PaaS resource consumption costs. Moreover, for PaaS providers, an effective rebalancing strategy can enhance infrastructure resource management at data canters, potentially leading to reduced PaaS infrastructure costs. This dual benefit ensures cost optimization for both businesses and service providers within the PaaS ecosystem.

5.2.5 From Vulnerability to Resilience: A Case Study on Data Real-Time Surge Protection Solution

Amidst dynamic festive seasons and promotional events, a prominent US-based banking and financial institution encountered a substantial operational challenge as a surge in credit card transactions ensued. The institution, utilizing an adept cloud-based microservices architecture, initially grappled with the intricacies of accommodating a sudden and significant uptick in transactional activity. Despite the system's inherent intelligence, it faced constraints in agility, struggling to adapt swiftly to rapid transactional escalations.

In response to this challenge, a dynamic solution was devised, cantered around the Lambda architecture. The architecture incorporated the Spout and Bolts mechanism for real-time data management within the transactional infrastructure. However, the Bolts, functioning as the operational hub of the system, encountered difficulties in efficiently scaling during periods of heightened transactional throughput. This inefficiency stemmed from the system's inability to make substantial adjustments without necessitating a temporary suspension of applications, thereby impeding real-world practicality. Furthermore, its reliance on historical data rendered it susceptible to unexpected transactional surges.

To address these issues, based on the proposed architecture and solution strategy, a proactive approach involving real-time data monitoring and rapid system expansion was implemented. This involved introducing a specialized queue, termed a "constrained-capacity queue," and a sophisticated monitoring entity known as the "Rebalancer Agent." Positioned between the data ingestion component and the processing system, the Rebalancer Agent meticulously tracked the queue, dynamically adapting to its full potential during peak demand without significant increments in microservices instances. This strategic enhancement ensured the system's ability to handle substantial data loads without compromising performance efficiency.

Expanding on this solution, the rebalancing architecture seamlessly transitioned into a cloud Platform as a Service (PaaS) environment. This extension successfully addressed challenges associated with specifying instances, memory allocation, and capacity planning. The resultant dynamic and responsive nature of the solution, coupled with the

cost-effectiveness inherent in cloud PaaS, emerged as a critical factor in upholding Service Level Agreements (SLAs) consistently.

The case study delved into contemporary challenges within runtime scalability and memory utilization in cloud applications. The refined architecture showcased resilience, effectively managing concerns related to minimum memory thresholds during peak operational periods and optimizing internal memory consumption. The dynamic and responsive characteristics of the rebalancing architecture within cloud PaaS environments were identified as pivotal in providing a holistic solution to intricate capacity planning challenges, ensuring a seamless and disruption-free experience during high-volume credit card seasons.

5.3 Conclusion and Future Directions

In this comprehensive study, an exhaustive analysis has been conducted regarding the advantages and challenges associated with real-time, unpredictable data streams originating from IoT data sources. To tackle the dynamic nature of these data surges, an innovative approach rooted in microservices architectural principles has been introduced. This pioneering microservices-based application architecture is further enhanced through the incorporation of a data surge protection mechanism, which enables the dynamic rebalancing of application instances. Consequently, the system's ability to seamlessly adapt to sudden increases in incoming data is assured. Furthermore, a thorough exploration has been undertaken to investigate the extensibility of this rebalancing strategy as it transitions from on-premises environments to Platform as a Service (PaaS) settings. This migration to the cloud environment not only facilitates the establishment of a more flexible and scalable infrastructure but also holds the potential to significantly curtail platform consumption costs, a paramount concern for enterprises operating in the cloud.

Looking forward, a vision is articulated for the future extension of this strategy into an adaptive rebalancing approach. This progression entails the seamless integration of the strategy with existing load balancing frameworks within PaaS environments, positioning it as a versatile and generic component. This adaptability enables it to be seamlessly assimilated into diverse PaaS architectures. By perpetually adapting to evolving data dynamics and fine-tuning resource utilization, this adaptive rebalancing

strategy emerges as a promising avenue for enhancing the efficiency and cost-effectiveness of data-intensive applications in both contemporary and forthcoming PaaS ecosystems.

Building upon the capabilities developed in this chapter, the subsequent chapter delves into the critical layers of Data Modelling, Management, and Governance for IoT applications. While robust ingestion mechanisms ensure that high-velocity data streams are efficiently captured and buffered, it is equally essential to structure, contextualize, and govern this data to extract meaningful insights and ensure compliance. The transition from ingestion to management highlights a natural progression, from handling the volume and velocity of IoT data to addressing its veracity, value, and lifecycle governance. This continuity ensures that once data is ingested securely and reliably, it is also organized, standardized, and made actionable within a well-governed framework tailored for large-scale IoT ecosystems.

Chapter 6

6. Data Modelling, Management and Data Governance Software Architecture for Internet of Things (IoT) Applications

A groundbreaking technological revolution is represented by the Internet of Things (IoT), which is reshaping the way interactions with the world around us are conducted. At its core, IoT revolves around three fundamental pillars: information modelling, data management, and governance architecture, which collectively form the backbone of this transformative ecosystem. Information modelling entails the intricate process of defining how data from various IoT devices and sensors are structured and represented. It involves creating standardized data models and schemas that enable seamless communication and interoperability among heterogeneous devices, ensuring that data flows smoothly across the IoT landscape.

Data management is the next critical component, as the sheer volume of data generated by countless IoT endpoints demands sophisticated strategies for storage, processing, and analysis. Effective data management not only ensures the efficient use of resources but also empowers organizations and individuals to derive meaningful insights from this wealth of information. It's the cornerstone upon which actionable intelligence, predictive analytics, and data-driven decision-making are built.

Lastly, Information Architecture, within the context of IoT, emerges as a strategic imperative for achieving better data governance. It serves as the foundational framework that not only structures and organizes the immense volumes of IoT-generated data but also orchestrates its lifecycle, accessibility, and security. In this intricate ecosystem, Information Architecture becomes the linchpin that ensures that IoT data is harnessed efficiently, adheres to compliance requirements, and empowers organizations to derive meaningful insights. This introduction explores the pivotal role that Information Architecture plays in enhancing data governance within the IoT realm, shedding light on how it enables organizations to navigate the complexities of IoT data while safeguarding its integrity and maximizing its value.

In essence, the triumvirate of information modelling, data management, and data governance architecture defines the IoT's capability to revolutionize industries, improve efficiency, and enhance the quality of life. As IoT continues to proliferate and evolve, these three pillars will remain at the forefront, shaping the future of the interconnected world.

Data governance stands as a foundational pillar within any robust data management program, operating in tandem with the imperative facet of data quality. In the context of an industrial Data Lake, a vast reservoir that rapidly ingests copious amounts of unstructured data from various source systems, the challenges become even more pronounced. This deluge of data streams in at high velocity, while multiple channels are utilized for querying and transforming data from the Data Lake. Traditional data governance systems, built around the principles of structured data management, face a formidable challenge when confronted with the 3Vs of big data: volume, velocity, and variety. In the contemporary landscape, establishing governance protocols for semi-structured or unstructured data within an Industrial Data Lake poses a genuine conundrum for enterprises. It encompasses the intricacies of querying, creating, maintaining, and securely storing this data. Simultaneously, diverse stakeholders, including Business, IT, and Policy teams, each bring their unique perspectives and requirements to the table. They seek to visualize the same data through different lenses, conducting analyses, imposing constraints, and devising efficient workflow mechanisms for policy makers' approvals. Within the context of this intricate scenario, the chapter introduces a forward-looking approach cantered around a property graph-based governance architecture and an accompanying process model. This innovative strategy is designed to provide organizations with the tools they need to achieve real-time control over their data, enabling enhanced governance, visualization, management, and querying of the unstructured data residing within the vast expanse of the Industrial Data Lake. This solution architecture is tailored to address the complexities arising from the continuous influx of various data types in today's data-abundant landscape. By adopting this novel approach, organizations can effectively overcome the challenges posed by the data deluge. This, in turn, empowers them to bolster their data governance practices and, critically, facilitates more insightful and informed decision-making processes.

In essence, this chapter offers a forward-thinking solution that not only keeps pace with the evolving data ecosystem but also positions organizations to harness the full potential of their data assets for improved operational efficiency and strategic advantage.

6.1 The Essence of Near Real-Time IoT Data Governance Architecture

The contemporary landscape marked by the era of near real-time Big Data [96] and stringent enterprise regulatory mandates is driving substantial transformations in how different industrial sectors handle and leverage the diverse array of data they accumulate and retain. The conventional infrastructure, comprising Relational Database Management Systems (RDBMS), Enterprise Data Warehouses (EDW), and Storage Area Networks (SAN), which organizations currently rely on to establish isolated data environments, is proving too inflexible to accommodate the escalating demands for extensive storage, high-performance analytics, and the processing of a broader spectrum of data types.

Attempting to force-fit this legacy architecture into the dynamic requirements of modern enterprises is both costly and fraught with risk. What enterprises require today is a NoSQL-based Data Lake, capable of aggregating disparate data types emanating from the multifarious systems that span their operations. The Data Lake boasts immense data storage capacity and a robust processing engine, offering the ability to house virtually any kind of data while facilitating concurrent access. The process of data ingestion is pivotal, representing the mechanism through which real-time data is seamlessly integrated into the Data Lake from external source systems. The Data Lake operates by swiftly loading data in its raw form, obviating the need for months of development work often associated with relational EDWs and Extract, Transform, Load (ETL) processes. This agility, bestowed by Data Lake services, is proving invaluable for enterprises grappling with cost pressures, adapting to evolving customer needs, and meeting the growing demands for transparency.

However, the high volume of data transmission across various source systems necessitates the implementation of a robust high-performance data governance architecture to scrutinize and effectively govern and manage incoming and outgoing

data within the Data Lake. The focus of this study is to explore the efficacy of a Graph-based architecture as a solution for high-performance data governance within the Data Lake, especially in the context of real-time data ingestion.

Additionally, it explores how different stakeholders, including Business, IT, and Policy teams, can collaborate to view, monitor, configure, and apply governance metadata policies tailored to specific business needs, ensuring that these policies are effectively queried and enforced within the Data Lake environment. This approach aims to provide a comprehensive solution that addresses the evolving data management challenges faced by enterprises while maximizing the value of real-time data in the modern digital landscape.

6.1.1 Goal of Real-time Data Governance

In pursuit of the objectives set forth by real-time or near real-time data governance, there are several critical goals that need to be accomplished:

- **Data Discovery and Profiling for Unearthing Hidden Data Quality Issues:** The primary aim is to facilitate extensive data discovery and profiling initiatives, a pivotal objective within the realm of real-time or near real-time data governance. This endeavour involves delving deep into the intricate landscape of an organization's extensive data repositories, regardless of their dispersed locations. The goal is to reveal any latent or concealed data quality issues that might be lurking within these vast data troves. Identifying and bringing these issues to the forefront assumes critical importance, as it lays the foundation for their subsequent rectification, ensuring that data quality remains at the heart of the organization's data management efforts. This proactive approach not only enhances the reliability of data but also pre-empts potential challenges that could compromise data integrity and usability down the line.
- **Data Lineage and Proactive Data Quality Monitoring:** Another crucial objective is to set up a strong system for tracking where data comes from and where it goes, along with actively keeping an eye on the quality of this data. This effort is all about carefully following the path of data within a company's entire data system. The goal is to make sure that any problems with the quality of data are spotted and addressed

promptly, not just once but consistently over time. This way, organizations can ensure that their data meets the expected standards and stays trustworthy, ultimately enhancing the reliability of their valuable data resources.

- **Effective Data Management for Authoritative Views:** Attaining a good understanding of data from different angles, such as those of Business, IT, and Policy roles, stands as a key objective in my research. This requires the deployment of effective data management strategies that ensure data is in sync across various parts of the organization, each with its own responsibilities and viewpoints. The outcome of this effort is the creation of a unified and precise depiction of data. This cohesive representation is instrumental in facilitating well-informed decision-making processes. In essence, my research seeks to establish a common ground where everyone in the organization sees the same data in a way that makes sense for their specific tasks, enabling more effective and informed choices to be made.
- **Metadata Management for Enhanced Visibility and Change Management:** Finally, proficiently managing metadata emerges as a crucial necessity in furnishing the necessary visibility and resources for navigating alterations in data integration with precision and efficiency. Metadata stands as a pivotal cornerstone for grasping the intricacies of data, including its structures, relationships, and alterations over time. Proficiently handling metadata empowers organizations to stay agile amidst shifting data environments, guaranteeing that data retains its accuracy, relevance, and alignment with the overarching business goals. In essence, it acts as a compass that guides organizations through the dynamic data terrain, helping them maintain data quality and suitability in the face of evolving needs.

These goals form the cornerstone of real-time or near real-time data governance, allowing organizations to uncover and rectify data quality issues, maintain data integrity, and establish authoritative data views that cater to the diverse needs of Business, IT, and Policy stakeholders. Effective metadata management further ensures adaptability in the face of changing data environments, ultimately fostering a data-driven culture that promotes informed decision-making and sustainable growth.

6.1.2 Scope of Real-time Data Governance

The scope of real-time data governance encompasses several key facets, each aimed at ensuring the quality and compliance of data in a dynamic and fast-paced environment:

- Real-time Data Quality Assurance: The foremost objective is to guarantee data quality right now of its capture, leaving no room for errors or inconsistencies to creep in.
- Immediate Data Standard Adherence: Real-time governance ensures that data aligns with corporate data standards instantly, minimizing any deviations from established norms.
- Prompt Error Feedback: It allows for swift error detection and correction as data enters or exits the system, preventing erroneous data from propagating.
- Defective Data Mitigation: The goal is to eliminate the spread of flawed data, preventing it from proliferating throughout the organization's data ecosystem.
- Efficiency in Maintenance: By addressing data defects early in the process, real-time governance reduces the overall burden of system maintenance, streamlining operations and reducing costs.

The conventional contingency model, which is elaborated upon in reference [97], lays out a governance framework that hinges on several critical factors such as performance strategy, organizational structure, and decision-making style. Initially crafted with the aim of addressing the requirements of traditional data quality management (DQM), this model has historically proven effective in scenarios where data processes are relatively stable and data volumes are more manageable. However, when the focus is shifted to the dynamic and rapidly evolving landscape of real-time big data governance, the limitations of this traditional framework are increasingly revealed. This is primarily due to the fact that a new set of challenges epitomized by the well-known three Vs: volume, velocity, and variety is confronted in real-time data governance. Firstly, the sheer volume of data being generated and processed in real-time scenarios is found to far exceed that typically encountered in traditional data governance contexts. This immense data influx necessitates a level of scalability and speed that traditional models struggle to accommodate. Secondly, the velocity at which data is generated, ingested, and acted upon in real-time environments is unprecedented. Data in motion demands

immediate attention and decision-making, leaving little room for the more deliberative processes that traditional models are accustomed to. Lastly, the variety of data types and sources that real-time big data governance encounters is remarkably diverse. This includes structured and unstructured data, streaming data, and data from a multitude of sources. Adapting to such variety is a considerable departure from the structured and consistent data sources typically managed under traditional DQM practices. In essence, while the traditional contingency model has proven effective, it is ill-suited to address the unique challenges presented by real-time big data governance. In this fast-paced, data-driven era, a new approach is required, one that can seamlessly handle the tremendous volumes, rapid velocities, and diverse varieties of data that characterize real-time environments.

In reference [98], a comprehensive exploration unfolds, shedding light on the fundamental distinctions that set big data governance apart from the more conventional realm of traditional information governance. This insightful examination underscores the necessity for a fundamentally different approach when dealing with the complexities of big data. First and foremost, big data governance requires the establishment of a robust analytic architecture. Unlike traditional information governance, where data may have been primarily used for reporting and historical analysis, big data governance necessitates an infrastructure that can handle the massive influx of data and perform real-time or near-real-time analytics. This involves the deployment of advanced data processing technologies, such as distributed computing frameworks and machine learning algorithms, to glean meaningful insights from the deluge of data. Next-generation IT processes play a pivotal role in this context. In contrast to the more linear and structured processes of traditional governance, big data governance demands agility and adaptability. IT teams must be equipped with the skills and tools to manage and process data at unprecedented scales while responding swiftly to changing data landscapes. This might involve the integration of DevOps practices, automation, and continuous monitoring to ensure data quality and security in real-time environments. Additionally, adaptable system architecture emerges as a critical component. Big data governance systems must be able to accommodate a wide variety of data types, sources, and formats, including structured, semi-structured, and unstructured data. They should also be scalable to cope with fluctuating data volumes and capable of seamlessly integrating with new data sources as they emerge. This adaptability is in stark contrast to the more rigid architectures often associated with

traditional information governance. Importantly, the overarching goal of big data governance is not merely automation but effective analysis and actionable insights. While automation certainly plays a role in streamlining processes, the primary focus is on deriving value from data by uncovering trends, patterns, and correlations that can inform decision-making and drive innovation.

This underscores that big data governance represents a departure from traditional information governance in terms of its technological requirements, process dynamics, and architectural demands. To succeed in the IoT fast and big data era, organizations must embrace robust analytic capabilities, next-generation IT processes, and adaptable system architectures that enable effective data analysis and insights generation, ultimately leveraging the full potential of their data assets.

This research is greatly influenced by the strategies elucidated in reference [99], which accentuate the critical importance of two key elements in the evolving data landscape: the burgeoning influx of unstructured data and the need for fostering collaboration among a wide spectrum of stakeholders. These parameters are at the forefront of the investigation and are pivotal in shaping my approach to real-time big data governance. The exponential growth of unstructured data represents a formidable challenge and opportunity. In the modern data ecosystem, structured data, which neatly fits into relational databases, coexists with a vast volume of unstructured data, such as text documents, images, videos, and social media content. Taming this unstructured data deluge requires innovative approaches, and one such approach central to my study is the utilization of graph-based data modelling, storage, and processing. Graph-based techniques, rooted in the concept of interconnected data nodes, are exceptionally well-suited for handling unstructured data. They excel in capturing relationships, patterns, and contextual information within data, which is often essential for extracting meaningful insights from unstructured sources. By leveraging graph-based methodologies, I am aiming to harness the latent value hidden within unstructured data, contributing to more comprehensive and accurate data governance practices. Furthermore, effective real-time big data governance necessitates active collaboration among diverse stakeholders, including business units, IT teams, and policy makers. These stakeholders bring different perspectives and priorities to the table, and harmonizing their interests is vital for successful data governance. I draw inspiration from contemporary strategies that emphasize collaborative data governance approaches, aiming to create a framework that enables efficient communication,

coordination, and decision-making among these stakeholders. In essence, this research endeavours to navigate the unique challenges posed by real-time big data governance by integrating innovative graph-based data modelling techniques to handle unstructured data effectively. Additionally, it underscores the significance of collaborative governance strategies to ensure that diverse stakeholders are engaged in the process of managing data in a rapidly evolving landscape. Ultimately, my goal is to glean insights from these contemporary strategies to establish a robust framework for effective data management and governance in the dynamic world of real-time big data.

6.2 Industrial DataLake and Data Ingestion

An industrial data lake plays a pivotal role in the modern industrial landscape, serving as the cornerstone for the accumulation and management of an expansive array of data types generated within industrial settings. This data reservoir seamlessly integrates structured and unstructured data originating from an intricate web of sources, including precision sensors, complex machinery, bustling production lines, and interconnected enterprise systems. However, the true magic lies in the meticulous data ingestion process, an integral component of this data architecture. This multifaceted procedure follows a systematic approach encompassing three crucial stages: extraction, transformation, and loading—commonly referred to as ETL. During the extraction phase, data is meticulously harvested from an array of source systems, capturing a rich tapestry of information that ranges from operational metrics to equipment diagnostics. These raw data sets then undergo a series of transformative processes, where they are refined, standardized, and enriched to ensure consistency, quality, and compatibility with the data lake's unified schema. Once the data is harmonized, it is efficiently loaded into the data lake, where it assumes its role as an asset accessible for a multitude of purposes. This well-structured repository empowers industrial organizations with the capability to harness its data for a myriad of industrial applications. Real-time insights can be gleaned from this reservoir, allowing for swift decision-making, proactive maintenance, and streamlined operations. Furthermore, the data lake serves as the bedrock for the application of advanced analytics and machine learning techniques, enabling predictive maintenance models, process optimization algorithms, and other

data-driven solutions to flourish. This, in turn, propels industrial operations into the realm of efficiency, quality enhancement, and innovation, ultimately shaping the future of industry through the power of data.

6.2.1 Unlocking the Potential: Why Data Lake Governance is a Necessity

In today's rapidly evolving data landscape, the sheer dynamism of data is undeniable. A constantly connected world is being experienced, in which myriad systems are engaged in continuous communication, enabling the relentless transfer of vast volumes of data around the clock. The advent of the big data era has ushered in a transformative epoch, characterized by the relentless storage, transmission, and daily analysis of staggering amounts of information. This data deluge is further compounded by the incessant influx of data from multifarious sources, including the ever-expanding realms of social media, web crawling, and crowdsourcing. While this influx fuels innovation and insights, it simultaneously introduces a multifaceted layer of uncertainty, casting a spotlight on the paramount issue of data quality within the industry. In this epoch of big and fast data, the conventional paradigm of relying on manual, rule-based data quality checks or traditional automated engines has proven itself to be inadequate. The primary culprits are the formidable challenges posed by the three Vs of big data: volume, velocity, and variety. As data continues to accumulate at an exponential rate, the task of ensuring its integrity, accuracy, and security evolves into an intricate puzzle that demands innovative solutions.

The paramount concern in this multifaceted landscape is data security, especially within the context of a heterogeneous data communication environment. With data traversing diverse networks and systems, safeguarding sensitive information has assumed monumental proportions. Moreover, the paradigm shift from in-house data management to cloud-based solutions, although replete with advantages, ushers in a fresh set of complexities. While cloud providers offer substantial benefits, they share the responsibility of providing optimal security with organizations themselves. The true challenge lies in crafting security measures that are finely attuned to the unique

attributes of each industry, organization, project, business, and geographical location. These attributes are far from static; they morph and evolve over time, adding further layers of complexity to the already intricate security landscape.

Considering these multifaceted challenges, there emerges an undeniable and compelling need for an effective, automated, and agile data governance engine. Such an engine must extend beyond merely safeguarding data; it should be capable of adapting to the ever-evolving data environment. Scalability, upgradability, and user-friendliness are the cornerstones of this vital solution, ensuring that organizations can navigate the tumultuous seas of data with optimal ease and agility. In essence, as data continues its ceaseless surge and diversification, the clarion call for a dynamic, responsive, and unwavering data governance solution resonates ever more emphatically. This solution stands as the linchpin of data-driven success in the modern industrial landscape, a testament to human innovation amidst an era of unparalleled data transformation.

The following points effectively encapsulate the context regarding the importance of industrial data governance.

- Data is constantly evolving and dynamic in today's interconnected world and vast volumes of data are transferred continuously across diverse systems.
- The era of big and fast data involves the storage, transmission, and analysis of staggering amounts of information.
- Data quality becomes uncertain due to the influx of data from various sources, including social media and web crawling.
- Manual and rule-based data quality checks are inadequate to handle the challenges posed by the three Vs of big data: volume, velocity, and variety.
- Data security is a paramount concern in a heterogeneous data communication environment.
- The shift to cloud-based solutions adds complexity to data security responsibilities.
- Security measures need to be tailored to the unique attributes of industries, organizations, projects, businesses, and geographical locations, which are dynamic.
- An effective data governance engine is needed to address these multifaceted challenges.
- Such an engine must go beyond data security to adapt to the evolving data environment.

- Scalability, upgradability, and user-friendliness are essential for data governance solutions.
- Data governance is the linchpin for success in the modern industrial landscape amidst the data transformation era.

6.2.2 Elevating Data Governance with Graph-Based Solutions

NoSQL databases are meticulously engineered to excel in handling data that shares similarities in terms of both data size and data quality. They provide a diverse array of data models, including key-value stores and document-based databases, each finely tuned to cater to specific dimensions of data management. These databases leverage the power of compound aggregate values to effectively address challenges associated with scaling out and accommodating the immense volumes of data that define the contemporary data landscape. It's important to note that different types of NoSQL databases are purpose-built to tackle challenges within the realm of data management, offering tailored solutions to meet diverse needs.

One of the enduring challenges encountered in the field of data governance is the meticulous upkeep of governance metadata. Additionally, the need to present the same dataset in diverse formats to meet the varied requirements of different stakeholders is a common predicament. This is precisely where property graph databases step in to provide invaluable assistance. Rather than opting for the performance-degrading approach of data denormalization, property graph databases employ a strategy of attribute normalization into nodes and edges. This normalization process simplifies a range of critical operations, including data movement, filtering, projection, and aggregation, rendering the entire data governance process significantly more efficient and adaptable to evolving needs. Property graph databases distinguish themselves with their remarkable prowess in real-time query capabilities. They shine in processing extensive volumes of raw data, seamlessly integrating with technologies like MapReduce in Hadoop and real-time event processing systems such as Apache Storm, Apache Spark, or Esper. Notably, these databases do not merely excel at managing raw data; they are also adept at projecting computation results into a graph-based structure.

This feature facilitates in-depth analysis and the extraction of profound insights from the data, transcending traditional data storage and management paradigms.

The genuine strength of graph databases becomes most apparent when tackling complex data queries. Commencing from a central node and traversing through a labyrinth of potential connections, these databases uncover intricate and often unexpected relationships concealed within datasets. In a world where the principles of data governance and the extraction of meaningful insights hold paramount importance, property graph databases play an indispensable role in elevating data management, analysis, and visualization to new heights. Their ability to unravel the profound stories hidden within the vast sea of data is nothing short of transformative in today's data-driven landscape.

6.2.3 Graph Based Solution Architecture

It becomes evident that the architectural foundation of a real-time data governance model, supported by property graphs, represents a comprehensive and dynamic solution meticulously crafted to tackle the intricate challenges that define the landscape of modern data management. Within this model, the data ecosystem is marked by its perpetual evolution, characterized by the continuous inflow of real-time data streams originating from a diverse array of sources. These sources span the entire data spectrum, encompassing unstructured, semi-structured, and structured data, each distinguished by its unique attributes and arriving with variances in both timing and velocity that demand keen attention. To deftly navigate this deluge of data, the architecture introduces specialized data consumer components known as "Spouts," a term well-recognized in the context of technologies like Apache Storm. These Spouts shoulder the pivotal responsibility of seamlessly ingesting incoming data with alacrity, swiftly ushering it into a meticulously organized data processor queue. What sets this architecture apart is its innate adaptability; here, a single consumer can effortlessly engage with multiple data processors. This inherent flexibility ensures the system's capacity to gracefully accommodate the gamut of data velocities, all while upholding efficiency and data integrity as non-negotiable principles. At the very core of this architectural framework lie the data processors, often referred to as "Bolts" within the realm of technologies like

Apache Storm. These Bolts take center stage in the data journey, bearing the critical responsibility of perusing the queued messages. Their distinguishing feature lies in their remarkable ability to perform data transformation with precision and finesse. These data processors meticulously reshape incoming data into a canonical, standardized format—an indispensable step that guarantees data consistency and enhances its overall usability. Once the data undergoes this transformation, it flows seamlessly to the data manager component, maintaining a consistent data velocity throughout this intricate journey.

The data manager component assumes the role of the linchpin in the data governance process, orchestrating vital functions such as data validation and the rigorous enforcement of predefined policies. To accomplish this, the component leverages a high-speed, non-blocking graph database—a cutting-edge innovation adept at meticulously validating incoming data. This capability is especially pivotal, considering the inherently unstructured nature of the data in question. The validation process involves a meticulous examination of data against the backdrop of established policies and rules, ensuring unwavering compliance with the organization's governing framework. Data that successfully navigates this stringent validation process is securely archived within the data lake, where it stands ready for subsequent analysis and application.

When it comes to data lake storage, this model leans on robust, high-capacity unstructured data storage engines such as HDFS (Hadoop Distributed File System), GFS (Google File System), or analogous solutions custom-crafted to shoulder the weight of monumental unstructured data volumes. These storage systems form the bedrock upon which the data lake is meticulously constructed, promising unfettered data accessibility, impeccable organization, and unfailing scalability, thereby laying the groundwork for future data retrieval and analytical pursuits. Furthermore, this architecture extends its reach to encompass a critical facet of data governance—facilitating interactions with third-party applications. When external applications seek access to the treasure trove of data housed within the data lake, a meticulously choreographed process unfolds. Requests and responses undergo rigorous scrutiny via the very data governance mechanisms that govern internal data flows. This painstaking review ensures that data access remains in exact alignment with established business policies and regulatory constraints, serving as an unwavering guardian of the organization's data assets' integrity and security.

In summary, the property graph-based real-time data governance model emerges as a robust, adaptable, and dynamic framework meticulously designed to tackle the relentless surge of data emanating from a multitude of sources. Beyond its proficiency in data validation and standardization, it serves as a secure, scalable bastion within the data lake. In an era where data governance and compliance reign supreme as non-negotiable imperatives, this model stands as a testament to the ever-evolving data management landscape. It seamlessly melds innovation with practicality, emerging as a solution tailor-made to confront the multifaceted challenges of the contemporary data-driven world. It shines as an exemplar of excellence in the realm of data management, a statement of professional prowess in the field.

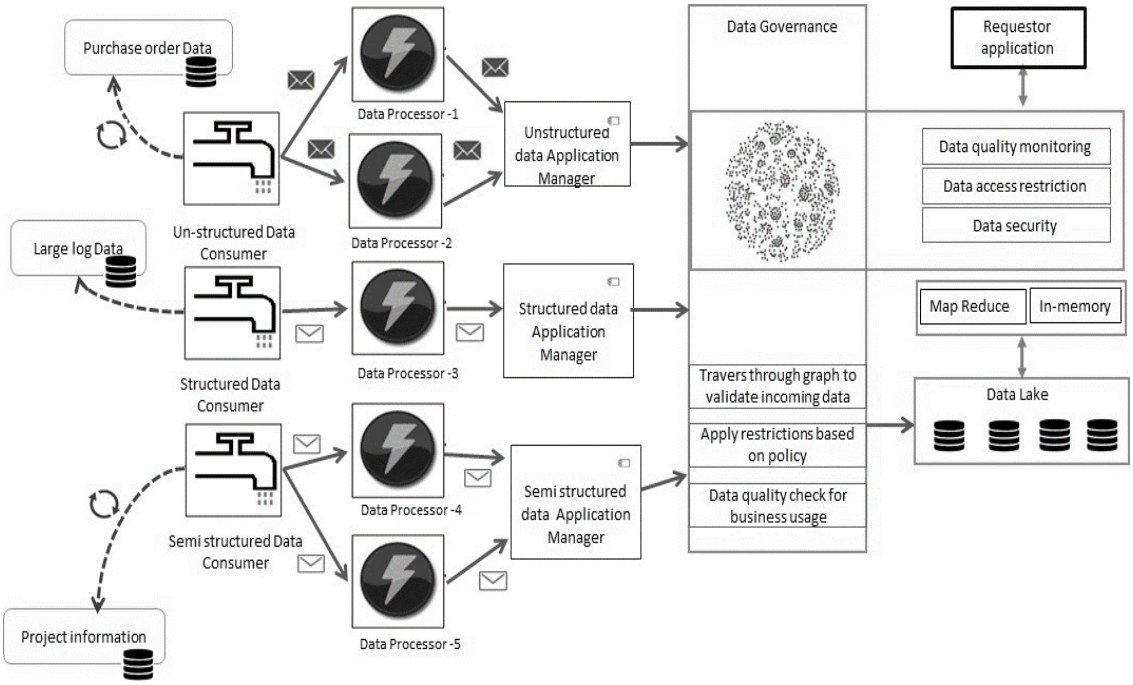


Figure 6. 1 Real time data governance architecture

6.2.4 Process Model Definition

The creation of a process model stands as a pivotal step in the journey of encapsulating all the essential elements of data governance within the confines of a graph structure and its intricate relationships. The process model serves as the bedrock upon which the organized representation of knowledge is meticulously constructed. It empowers users

with the ability to forge, navigate, share, and critically assess the knowledge model with precision and clarity. In essence, the process model assumes the role of a blueprint for graph data governance, enabling a profound analysis of fundamental structural attributes, the complexities inherent to the graph, the existence of cycles within it, and the access patterns governing graph data, including nodes and their associated attributes. At its core, the process model acts as a unifying force, bringing together the multifaceted facets of data governance into a coherent and comprehensible structure. This structure not only provides a visual representation but also serves as a powerful tool for decision-makers, data stewards, and other stakeholders to gain deep insights into the intricacies of data governance. With the process model as their guide, users can systematically define and understand the roles, responsibilities, and relationships that underpin effective data governance. Furthermore, the process model acts as a dynamic repository of knowledge, allowing for real-time updates and refinements to the data governance framework as it evolves to meet the ever-changing needs of the organization. It becomes a living document that reflects the current state of data governance, while also serving as a compass for charting the future course of data management and stewardship.

In practical terms, the process model fosters transparency and accountability by clearly delineating the steps, procedures, and workflows that govern data governance practices. It offers a visual representation of the flow of data, highlighting key touchpoints, decision nodes, and critical processes that ensure data quality, compliance, and security. This transparency not only aids in understanding the intricacies of data governance but also facilitates collaboration among teams and departments, fostering a shared commitment to data excellence. Moreover, the process model becomes a valuable tool for identifying potential bottlenecks, inefficiencies, and areas for improvement within the data governance framework. By analysing the model, organizations can pinpoint areas where data governance may be faltering or where additional resources and measures are required to enhance its effectiveness. This data-driven approach to governance ensures that decisions are rooted in empirical insights, enabling organizations to allocate resources judiciously and optimize their data governance efforts.

Creation of a process model is a fundamental step in the quest for effective data governance. It serves as a visual, dynamic, and comprehensive representation of the knowledge and practices that underpin data governance, offering a roadmap for

navigating the complexities of data management. As organizations strive to harness the full potential of their data assets, the process model emerges as an indispensable tool, guiding them towards a future where data governance is not just a necessity but a strategic advantage in the data-driven landscape.

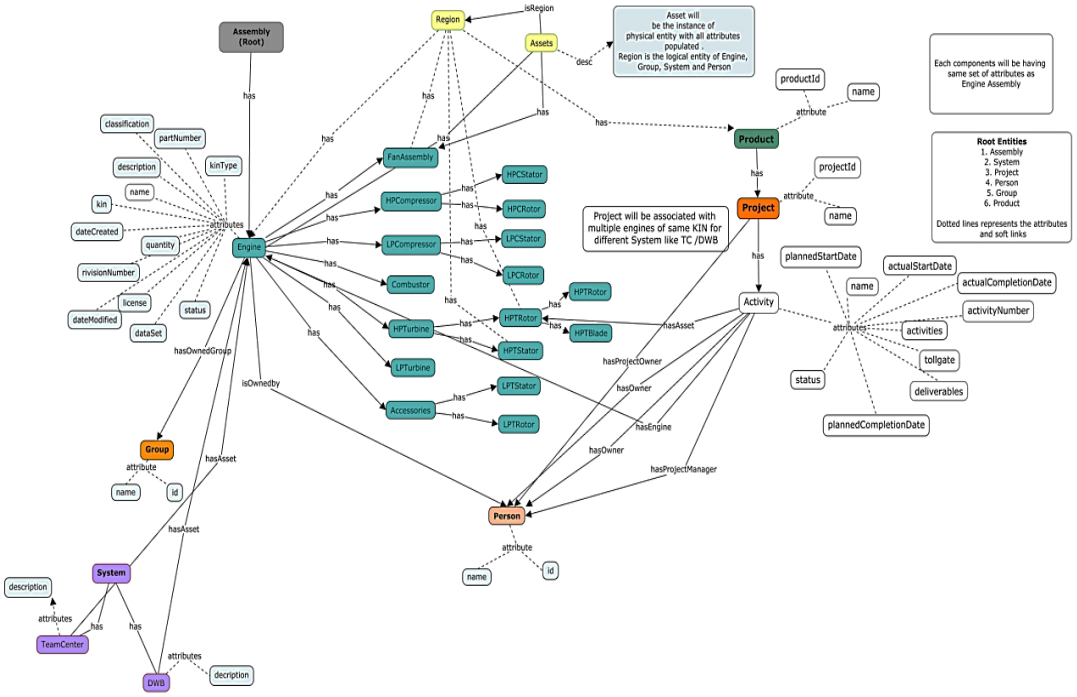


Figure 6. 2 Graph based data governance process map

The graph-based process model provides an invaluable framework for enabling advanced data wrangling, offering a rich visual analytic perspective that caters to diverse stakeholders and entities within an organization. It thrives on its capacity to present the same graph data in multiple ways, serving the distinct needs and interests of various stakeholders and entities involved in data governance. At its core, this approach recognizes the intricate interplay between different facets of the organization, weaving a tapestry of connections that link projects to business objectives and systems to specific projects.

The architecture of the Process model establishes a comprehensive graph-based ecosystem aimed at encapsulating an organization's essence across People, Process, Devices, and Technology dimensions. Within this model (depicted in Figure 6.2), individuals (People) are intricately linked to Projects, which in turn are associated with a series of activities constituting a specific Product (Process). Conversely, Systems are

characterized by clusters of devices, wherein a stream of discrete and continuous data is assimilated (via IIoT), thus steering overall operations, while digital twins optimize operational parameters. Notably, an individual (People) can assume ownership of such a device, or even oversee a collective managing industrial devices and systems. This software architecture, represented graphically, introduces an unparalleled level of flexibility, enabling information visualization from a 360-degree perspective, offering practically boundless insights.

From the perspective of the technical persona, such as the IT team, the ability to visualize systems that are intricately linked to both inbound and outbound data injection is of paramount importance. This visualization equips IT professionals with a holistic understanding of the data flow within the organization's ecosystem. It allows them to identify potential bottlenecks, monitor data quality, and ensure the seamless operation of data pipelines. By leveraging the graph-based process model, IT teams gain a comprehensive view of how data moves within the organization, enabling them to make informed decisions regarding system optimization, data security, and compliance.

For members of the business team, the graph-based process model extends its utility by offering a versatile platform for exploring the same system information from a project-centric perspective. This flexibility is particularly beneficial when different representations of data are required to align with various business objectives and requirements. Business team members can seamlessly transition between different project views, gaining insights tailored to their specific needs. This dynamic approach empowers business stakeholders to make data-driven decisions that directly impact project outcomes, enhancing efficiency and strategic planning.

Policy makers, on the other hand, find immense value in the graph-based process model's capacity to manipulate and interact with policy rules as an integral part of project compliances. With the ability to visualize and customize policy rules within the context of specific projects, policy makers can ensure that data governance practices align precisely with regulatory requirements and organizational standards. This level of granular control enables policy makers to adapt data governance policies to evolving compliance landscapes, ensuring that the organization remains resilient and responsive to changing regulatory demands.

In essence, the graph-based process model serves as a unifying platform that bridges the gap between technical, business, and policy-oriented perspectives within the realm of data governance. It fosters collaboration and synergy among diverse stakeholders,

enabling each group to leverage the same graph data while tailoring their views and interactions to suit their unique objectives. This flexibility, adaptability, and transparency are foundational to an organization's ability to navigate the complex landscape of data governance successfully. Ultimately, the graph-based process model emerges as an indispensable tool in the arsenal of modern data-driven organizations, facilitating informed decision-making, enhancing efficiency, and ensuring compliance in an increasingly data-centric world.

6.2.5 Technology Considerations and Execution Environment

A Java-based framework has been employed to establish the foundation for the creation, loading, and querying of the property graph database, serving as the backbone of the real-time data governance process. A standalone infrastructure has been configured, relying on the Titan graph database processing engine with Cassandra employed as the columnar backend storage solution. The deployment of the front-end visualization application for graph metadata management has been executed on Apache Tomcat, where user interface components have been constructed using D3.js and JQuery to facilitate user interactions. The server-side framework, designed to mitigate vendor locking concerns associated with the underlying graph database, has been diligently implemented using the Tinkerpop blueprint stack. Additionally, the realization of the process model has been achieved through the utilization of Tinkerpop Frames, contributing to the flexibility and adaptability of the system. The following table presents a tabular summary of the software and technologies that have been leveraged in this implementation:

Software	Version	Usage
JDK7	1.7.72	JDK and JVM installation
Apache Cassandra	2.1.2	Act as a backend graph data storage.

Rexster-Server	2.6.0	Graph server with embedded titan exposes graph over REST protocol, Web gremlin console.
Apache Tomcat	8.x	Web server for UI development, Java app development.
Blueprint-Tinkerpop	2.6.0	Blueprint specification for graph database., Frames definition
IHMC CMap tool	6.0.0	Used for Process modelling.

Table 6. 1 Used software and their usage

The Rexster-Titan graph processing engine has been seamlessly integrated with Apache Cassandra, offering an out-of-the-box solution that brings REST API support to the forefront. This strategic integration empowers users with a versatile and efficient means of interacting with the graph database, facilitating data retrieval and manipulation through RESTful endpoints. Moreover, custom-tailored services have been meticulously developed and subsequently deployed within the Tomcat server environment. This deployment serves as a pivotal component in the overall architecture, providing the necessary infrastructure for executing specialized functionalities and handling requests from various stakeholders.

The following table presents a concise overview of the server configuration for the graph metadata execution engine, shedding light on the key components that constitute this robust system:

Server Configuration	Description
Rexster-Titan Graph Processing Engine	A seamlessly integrated solution that combines the power of Rexster and Titan, facilitating REST API support and efficient graph data processing.
Apache Cassandra	A columnar storage backend that complements the graph processing engine, ensuring optimal data storage and retrieval capabilities.

Custom-Implemented Services	These services, thoughtfully crafted to align with specific organizational needs, are deployed within the Tomcat server. They serve as the bridge between user interactions and the graph metadata execution engine, enabling custom functionalities and facilitating data access and manipulation.
Tomcat Server	The deployment platform for the custom services, Tomcat plays a pivotal role in ensuring the availability and accessibility of these services.

Table 6. 2 Server configuration and descriptions

This configuration represents a harmonious fusion of cutting-edge technologies and custom-built services, working in unison to empower the organization with a robust, scalable, and responsive graph metadata execution engine. Through this integration, the organization is poised to harness the full potential of its graph database, facilitating efficient data governance and analysis in a dynamic and data-centric environment.

Below is the architecture of a dedicated standalone graph server designed for the management of graph metadata and the execution of query operations.

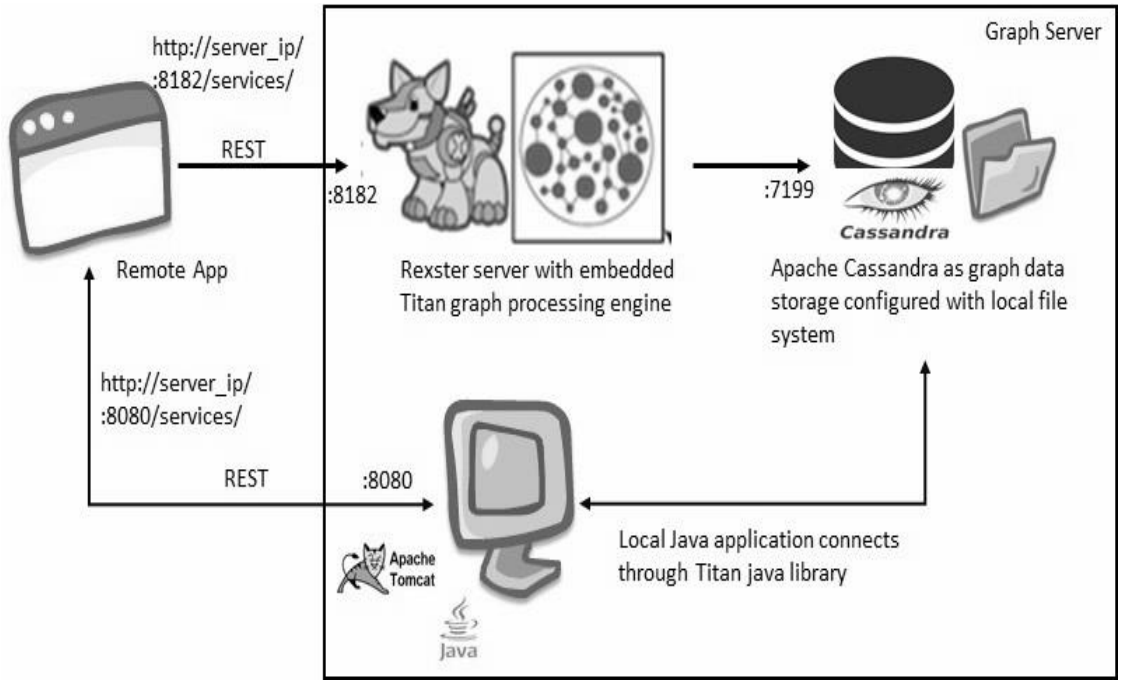


Figure 6. 3 Graph server architecture

Below is the architectural representation of the graph processing model designed to facilitate the loading, updating, porting, and maintenance of governance metadata within the graph.

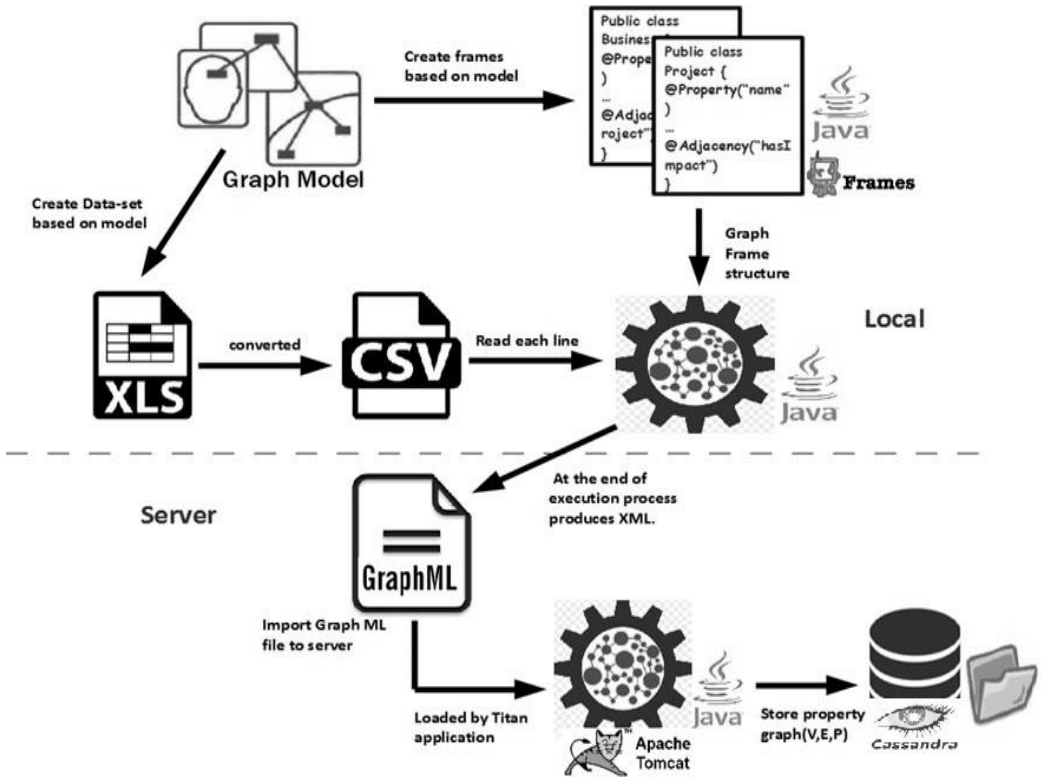


Figure 6. 4 Graph loading process

The process model establishes the interface through which incoming data structures communicate with TinkerPop Frames. Governance metadata information can be seamlessly introduced into the system via two primary methods: directly through the user interface or by importing spreadsheet files in CSV format. To facilitate this data ingestion, a localized Java processing engine meticulously scans the CSV metadata files, undertaking a transformation process that converts the flat, tabular data into object-oriented frames. Subsequently, these frames, now enriched with a structured data model, undergo a conversion process into the portable GraphML format [100][101], recognized as an open standard across various graph processing engines. This format ensures compatibility and interoperability, enabling any graph processing engine to efficiently load the data into the designated graph server. The adoption of GraphML as the data exchange format underscores the system's commitment to flexibility and its ability to work seamlessly with diverse graph processing environments.

The ultimate destination for the graph data resides within the robust columnar storage backend powered by Apache Cassandra. This choice of backend storage guarantees the durability, scalability, and performance required to support the governance metadata and streamline graph data operations. This configuration serves as the bedrock of the system's data management capabilities, ensuring the secure and efficient storage of critical information within the organization's infrastructure.

6.2.6 Demonstration of the Proposed Concept

The prototype dataset comprises essential metadata information for various businesses. In my initial experiment, I focused on the data from five distinct businesses. To effectively harness this data, I employed a localized custom processing engine, which adeptly transformed the metadata into a property graph model. The resulting graph structure exhibits a remarkable balance, with each node consistently carrying an average of three attributes, enriching the dataset with valuable contextual information. As part of the data preparation process, I conducted automatic indexing on critical components, including vertices, edge identifiers, and business names. This meticulous indexing procedure enhances the efficiency of data retrieval and query operations, streamlining the overall data management process. To assess the performance of the system, I conducted comprehensive tests, comparing the execution outcomes between single-threaded and multi-threaded approaches. The table below provides a detailed overview of the performance results, shedding light on the system's responsiveness and efficiency under varying execution conditions. This performance evaluation serves as a vital benchmark, offering valuable insights into the system's capability to handle data processing and query tasks, thereby informing future optimizations and scalability considerations.

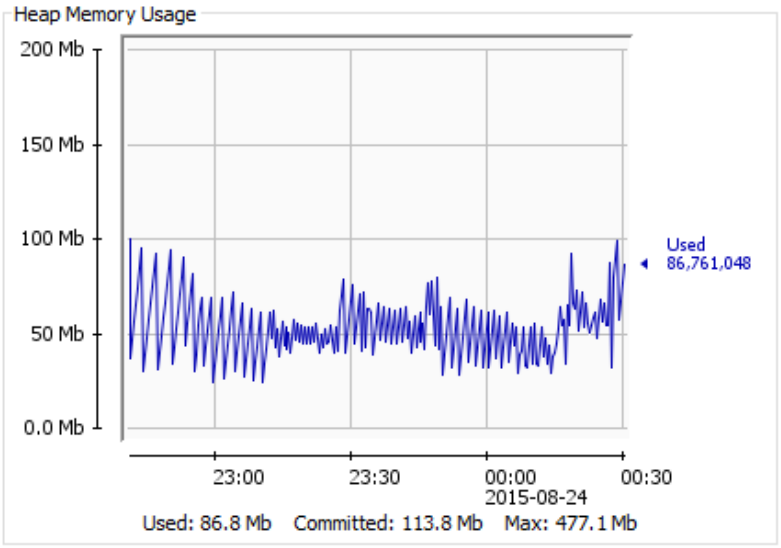
Number of Nodes – 384, Number of edges – 493

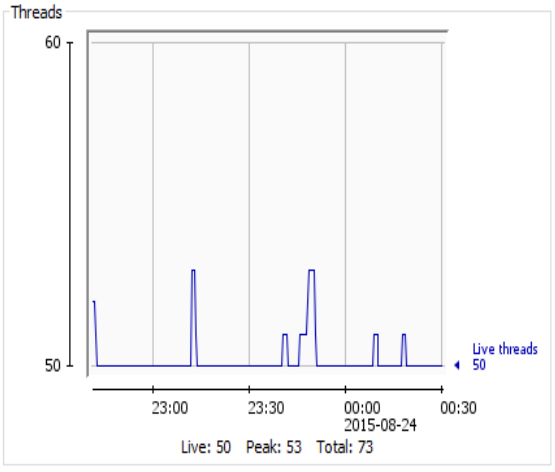
Load test – simultaneous 10 threads, Strategy – Simple

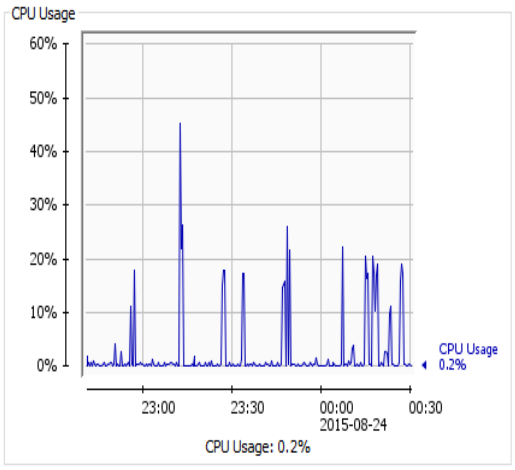
Test step	Single request (ms)	Load test average (ms)
Query Vertices	1455	1726

Query Edges	1257	1895
Query Vertices Id	9	12.76

Table 6. 3 Execution results







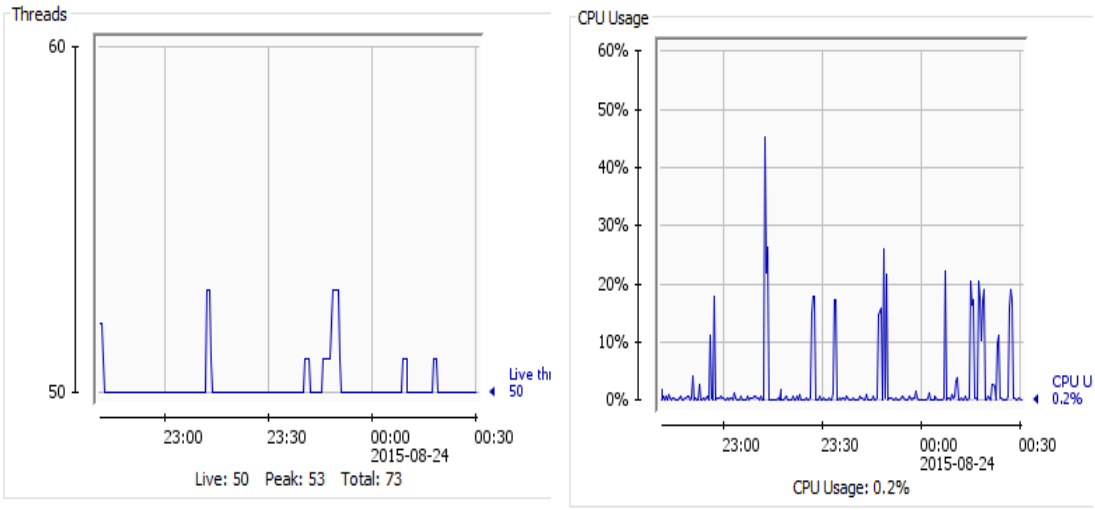


Figure 6. 5 JVM memory and CPU usage

It was observed that an increase in the number of nodes did not significantly affect the execution time, highlighting the crucial role played by indexes in this context. The actual node data resides within the backend storage, while the indexes are loaded into memory once the system is initialized. These indexes serve as a rapid access mechanism, allowing the system to retrieve actual graph information from storage with remarkable efficiency. The system's performance remains consistently high, provided that the available memory can accommodate the entire index structure within the RAM. Experimentation involving the gradual augmentation of nodes and edges revealed that processing time exhibited minimal variance. In practical terms, the search time operates at an impressive O(1), offering a substantial advantage over relational databases where search time experiences a linear increase in response to growing data volumes. Furthermore, the monitoring of heap usage, represented by a sawtooth graph pattern, confirmed the absence of memory leakage during object loading, with garbage collection processes functioning efficiently. CPU usage exhibited stability throughout the experimentation, with its impact diminishing when the implementation is deployed within a cloud infrastructure. In a cloud-based environment, processing power is often considered virtually limitless, further underscoring the system's robust performance characteristics.

In summary, my experimentation with the prototype dataset showcases the successful transformation of business metadata into a well-structured property graph model. This model serves as a valuable foundation for data-driven insights and analysis. The

subsequent performance evaluation, comparing single-threaded and multi-threaded execution, provides critical information about system efficiency and scalability, paving the way for further enhancements in data management and query capabilities.

6.2.7 Case Study: Transforming Data Governance in a Leading US Aviation Parts Manufacturing Business

In the highly competitive landscape of aviation parts manufacturing, AeroTech Parts Inc. (For security purposes, the name has been altered to that of a fictional organization), a major player in the industry, has embarked on a transformative journey to revolutionize its data governance practices. Acknowledging the critical role of real-time data management in the aviation sector, AeroTech Parts has implemented an innovative data governance model to navigate the complexities of handling diverse, real-time data streams.

Business Overview:

AeroTech Parts Inc. stands at the forefront of supplying critical components to major aerospace companies, navigating a vast network of suppliers and customers. The aviation industry's stringent regulations and the imperative for seamless operations make efficient data management a cornerstone for success.

Near Real-time Data Governance Architecture:

AeroTech Parts has embraced a state-of-the-art real-time data governance model founded on a property graph structure. This architecture introduces specialized components such as "Spouts" for data ingestion and "Bolts" for data transformation, ensuring adaptability to the varying velocities of incoming data streams. The architecture's capacity to seamlessly ingest and process data positions AeroTech Parts to efficiently manage the dynamic data spectrum, including unstructured, semi-structured, and structured data.

Data Manager Component:

At the heart of the architecture lies the data manager component, leveraging a high-speed, non-blocking graph database for rigorous data validation against predefined policies. This capability is particularly crucial given the unstructured nature of aviation data. Successful validation results in the secure archiving of compliant data within a robust data lake, supported by high-capacity unstructured data storage engine.

Third-Party Interactions:

The real-time data governance architecture extends its capabilities to facilitate interactions with third-party applications. When external applications seek access to the treasure trove of data housed within the data lake, a meticulously orchestrated process ensures that data access aligns precisely with established business policies and regulatory constraints, safeguarding the integrity and security of AeroTech Parts' data assets.

Process Model Definition:

To encapsulate all essential elements of data governance within a graph structure, AeroTech Parts has defined a comprehensive process model. Serving as a blueprint for graph data governance, this model empowers users to forge, navigate, share, and critically assess the knowledge model with precision and clarity.

Perspectives from Different Personas:

The implementation of the graph-based process model caters to various personas within AeroTech Parts Inc. The **Technical Persona**, represented by the IT team, gains a holistic understanding of data flow, enabling the identification of bottlenecks and ensuring the seamless operation of data pipelines. The **Business Persona**, comprising members of the business team, utilizes the model as a versatile platform for exploring system information from a project-centric perspective, making informed, data-driven decisions tailored to specific needs. Meanwhile, the **Policy Makers** leverage the model's capacity to manipulate and interact with policy rules, ensuring precise alignment with regulatory requirements and organizational standards.

In conclusion, AeroTech Parts Inc.'s adoption of a property graph-based real-time data governance model, coupled with a dynamic process model, positions the company as a trailblazer in effective data management within the aviation parts manufacturing sector.

This adaptable, transparent, and collaborative approach proves indispensable for informed decision-making, efficiency enhancement, and compliance in an industry increasingly reliant on data. AeroTech Parts Inc. stands as a beacon of excellence, seamlessly integrating innovation and practicality in the realm of data governance within the dynamic landscape of aviation manufacturing.

6.3 Conclusion and Future Directions

In this study, the issues concerning real-time data governance within the context of unstructured data were thoroughly analysed, leading to the proposal of a graph-based architecture meticulously designed to ensure the effective governance of data, regardless of whether it is inbounded to or outbound from the data lake. The framework introduced in this work has the capability to accommodate a diverse range of stakeholders, and the mechanisms were elucidated, allowing these stakeholders to interact with the system seamlessly. This interaction grants them access to a shared repository of metadata while affording them the flexibility to perceive and manipulate this information from various vantage points, each tailored to their specific roles and responsibilities.

To validate the practicality and utility of my proposed architecture, a comprehensive experiment was conducted, involving the implementation of the graph model and the subsequent testing of the application utilizing real-time datasets. This empirical phase of my work served as a critical milestone, providing tangible evidence of the architecture's efficacy and its capacity to function optimally in real-world scenarios.

Furthermore, this research opens the door to future extensions, particularly in the realm of distributed graph processing. This envisioned expansion seeks to address the inherent challenges of scalability and performance by leveraging a distributed environment. Within this distributed framework, I anticipate the storage and querying of governance graph metadata at scale, facilitating robust data governance practices. A promising avenue for realizing this vision is the utilization of Faunus, a Hadoop-based graph analytics engine specifically designed to analyse graphs within a multi-machine cluster. Faunus stands out for its ability to process and analyse graphs of virtually infinite size,

all while adhering to functional and MapReduce computing paradigms, rendering it a compelling candidate for the next phase of my research.

This chapter establishes the foundational framework for structuring, integrating, and securing the massive and heterogeneous data generated by IoT systems. It emphasizes the importance of standardized data models, effective data lifecycle management, and robust governance mechanisms to ensure data consistency, quality, and compliance across diverse IoT ecosystems. Building on this foundation, the next chapter on Distributed Ledger Technology (DLT) based Software Architecture for IoT Industrial Applications extends the scope by introducing a decentralized and tamper-proof mechanism for managing trust, provenance, and traceability in multi-stakeholder environments. While the current chapter ensures semantic clarity and policy-driven control over IoT data, the next chapter leverages DLT to ensure immutability, auditability, and distributed consensus, addressing critical industrial requirements such as data integrity, transaction verification, and regulatory transparency. Together, these chapters present a cohesive architecture that integrates strong data governance with decentralized trust enforcement, forming a comprehensive framework for scalable and secure IoT deployments.

This chapter acts as a critical junction that integrates and matures the architectural foundations laid in earlier chapters, each of which is essential for building a robust and scalable Distributed Ledger Technology (DLT)-based Software Architecture tailored for industrial IoT environments. The journey begins with establishing IoT Identity, Whitelisting, and Decentralized Trust Management, which ensures authenticated and trusted device participation within the network. This is fortified by Privacy, Information Transparency, and Access Management, which enforces secure and permissioned interactions across decentralized nodes. The Data Ingestion Architecture adds the necessary resilience to handle unpredictable surges in data volume without compromising performance or data fidelity. Finally, Data Modelling, Management, and Governance bring semantic structure, consistency, and policy-driven control to the vast data generated by IoT systems. Together, these layers create the technical and trust infrastructure required for a DLT-enabled architecture, where industrial IoT systems can benefit from immutability, decentralized consensus, and auditable traceability, hallmarks of next-generation industrial automation and intelligence.

Chapter 7

7. Distributed Ledger Technology (DLT) based Software Architecture for IoT Industrial Applications

Distributed Ledger Technology (DLT) has revolutionized the architecture of Internet of Things (IoT) applications by providing a secure, transparent, and decentralized framework for managing and recording transactions. At the core of this architecture is a decentralized network of nodes, each equipped with a copy of the distributed ledger, ensuring that data is not stored in a centralized location vulnerable to single points of failure or malicious attacks. The DLT-based IoT application architecture employs smart contracts, self-executing contracts with coded rules that automate and enforce the terms of agreements within the network, thereby reducing the need for intermediaries. The use of consensus mechanisms, such as Proof-of-Work or Proof-of-Stake, enhances the security and integrity of the data by requiring network participants to reach a mutual agreement before validating and adding new transactions to the ledger. This architecture promotes data immutability, meaning that once information is recorded on the distributed ledger, it cannot be altered or tampered with, ensuring a trustworthy and auditable record of IoT device interactions. Additionally, DLT facilitates interoperability among diverse IoT devices and platforms by providing a standardized and secure communication protocol. Through the transparent and decentralized nature of DLT, stakeholders can trace the entire lifecycle of data, from its origin to its current state, fostering trust and accountability in the rapidly evolving landscape of IoT applications.

7.1 Scope and Advantage of Blockchain Usage in IoT applications

- **Decentralization and Trust:** Blockchain's introduction of a decentralized approach to IoT applications fundamentally transforms the traditional paradigm by eliminating the reliance on a central authority. In conventional IoT systems, a central entity typically controls and validates transactions, serving as a potential single point of failure and vulnerability to cyber threats. However, with blockchain, this centralized control is replaced by a distributed network of nodes, each maintaining a copy of the ledger. This decentralization not only mitigates the risk of a single point of failure but also significantly enhances trust within the IoT ecosystem. Participants, whether devices or entities, can interact directly with one another in a peer-to-peer fashion, fostering a trust less environment where transactions and data exchanges occur without the need for intermediaries. The decentralized nature of blockchain instils confidence in the integrity and transparency of the IoT ecosystem. Transactions and data generated by IoT devices are recorded in a tamper-resistant manner across all nodes in the network. The cryptographic linkage between blocks ensures that once a block is added to the chain, altering any information within it becomes practically impossible without consensus from majority of the network. This tamper-proof characteristic not only safeguards the authenticity of data but also provides a transparent and auditable trail of all activities. Participants can trace the history of transactions, ensuring a clear record of events without the risk of manipulation or unauthorized alterations. Consequently, this transparency builds a foundation of trust among participants, whether they are end-users, manufacturers, or service providers, fostering a more resilient and reliable IoT ecosystem.
- **Security Enhancement:** The security paradigm that blockchain introduces to IoT applications is rooted in its immutable and cryptographic foundations, providing an unparalleled level of data integrity and protection. The immutability of the blockchain refers to the unalterable nature of recorded data once it has been added to the ledger. This characteristic is achieved through the cryptographic hashing of each transaction, where the output of one block becomes the input for the next. This

chaining of blocks not only creates a chronological and transparent record but also establishes a robust security mechanism. The cryptographic hashes act as unique digital fingerprints for each block, making it exceptionally challenging for malicious actors to tamper with or alter any transaction data within the chain. The cryptographic links between transactions contribute to the overall integrity of the IoT data. Attempting to alter a single block would require recalculating the cryptographic hash for that block and all subsequent blocks, as each hash is intricately connected to the one before it. This process necessitates an immense amount of computational power and time, rendering the data on the blockchain highly resistant to unauthorized modifications. In scenarios where the integrity of data is paramount, such as in healthcare applications where patient records must remain accurate and unaltered, or in critical infrastructure systems where the reliability of operational data is critical for safety and security, the robust security offered by blockchain becomes indispensable. Furthermore, the cryptographic nature of blockchain enhances the confidentiality of sensitive IoT data. Participants in the network can securely engage in transactions knowing that the information is protected by advanced encryption algorithms. This heightened security is particularly crucial in sectors like healthcare, where patient privacy must be upheld, or in critical infrastructure, where unauthorized access to operational data could have severe consequences. In essence, the immutable and cryptographic attributes of blockchain fortify the security posture of IoT applications, instilling confidence in the integrity, confidentiality, and authenticity of the data exchanged within these interconnected systems.

- **Smart Contracts for Automation:** The integration of blockchain into IoT applications brings forth a powerful capability: the facilitation of smart contracts. Smart contracts are self-executing agreements with predefined rules and conditions encoded directly into the blockchain. In the context of IoT, this innovative feature opens the door to a new era of automation and efficiency. Smart contracts operate on a "law code" principle, meaning that once certain conditions are met, the contract automatically executes without the need for intermediaries, enhancing the speed, accuracy, and transparency of transactions. In the realm of supply chain management, smart contracts play a transformative role. Consider a scenario where IoT devices, such as sensors and RFID tags, are integrated into the supply chain network. When goods are shipped from one location to another, these IoT devices

continuously transmit real-time data to the blockchain. Smart contracts, being an integral part of this system, can be programmed to execute specific actions based on the received data. For instance, upon the successful delivery of goods to their destination, the smart contract can automatically trigger the release of payment to the supplier. This automation not only eliminates the need for manual intervention but also reduces the potential for errors and disputes, as the execution of the contract is contingent on the verifiable data provided by the IoT devices. The use of smart contracts in IoT applications extends beyond supply chain management. In sectors like energy, smart contracts can automate billing and payment processes based on the real-time consumption data from IoT-enabled devices. In agriculture, these contracts can be employed to automate irrigation systems based on weather data collected by sensors. This automation not only streamlines operations but also enhances precision and responsiveness in decision-making. Furthermore, the transparency of smart contracts ensures that all stakeholders within the network have visibility into the terms and conditions of the agreement. This transparency fosters trust among participants, as the automated execution of contracts is based on verifiable, tamper-resistant data recorded on the blockchain. In summary, the incorporation of smart contracts into IoT applications revolutionizes operational processes by automating actions, reducing reliance on intermediaries, minimizing errors, and enhancing the overall efficiency and transparency of complex systems.

- **Interoperability:** The potential of blockchain to serve as a standardized and interoperable platform for a myriad of IoT devices represents a pivotal shift in the landscape of connected technologies. The decentralized and open-source nature of blockchain protocols addresses the inherent challenges posed by the heterogeneity of devices, fostering compatibility and seamless communication across diverse IoT ecosystems. Traditionally, interoperability issues have plagued the IoT landscape due to the multitude of devices operating on different communication protocols, standards, and data formats. Blockchain's decentralized architecture facilitates a universal framework where devices can interact and transact without the need for a central authority. Each device in the network possesses a copy of the distributed ledger, ensuring a shared source of truth. This common ledger becomes a standardized reference point, enabling devices with disparate specifications to communicate effectively. The cryptographic principles underpinning blockchain ensure the integrity and security of the shared data, fostering a trustworthy

environment for diverse devices to collaborate. Open-source nature further enhances blockchain's appeal as a unifying platform for IoT. The transparent and collaborative development model allows for the creation of interoperable protocols and standards that can be adopted across the industry. This openness mitigates vendor lock-in and promotes innovation by encouraging developers to contribute to the shared infrastructure. As a result, blockchain becomes a dynamic and evolving solution that adapts to the diverse needs of the IoT landscape. Moreover, the decentralized nature of blockchain eliminates the need for intermediaries, reducing friction in communication between devices. Smart contracts, embedded in the blockchain, facilitate automated and trust less interactions between devices based on predefined conditions. This streamlined communication ensures that devices can seamlessly collaborate, whether in a smart home environment, an industrial setting, or a smart city infrastructure. In overcoming the challenges of heterogeneous IoT ecosystems, blockchain establishes a foundation for the next generation of interconnected devices. It not only provides a standardized and interoperable platform but also instils trust, security, and transparency in the complex web of interactions between diverse IoT devices, paving the way for a more cohesive, efficient, and scalable IoT landscape.

- **Data Integrity and Immutability:** The immutability of data within the blockchain framework stands as a bedrock principle, presenting a revolutionary solution to the challenges of data integrity in IoT applications. Once information is recorded on the blockchain, it attains an unalterable state, impervious to modifications or deletions. In the realm of IoT, where an extensive network of interconnected devices constantly generates data, this immutability feature becomes a linchpin, assuring the reliability and trustworthiness of the information exchanged. This unyielding characteristic of the blockchain is particularly invaluable in scenarios where data accuracy is not just a preference but a critical necessity, such as in regulatory compliance or legal contexts. Regulatory frameworks often mandate the precise and unaltered recording of data to ensure adherence to industry standards and legal requirements. Blockchain's immutability provides an auditable trail of events, offering a transparent and tamper-resistant record of every transaction or piece of information generated by IoT devices. Consider a healthcare IoT scenario where patient records, treatment plans, and medication dispensations are recorded on the

blockchain. The immutability of this data ensures that medical histories remain accurate and unmodifiable, crucial for ensuring the well-being of patients and complying with healthcare regulations. In legal contexts, such as in contracts facilitated by IoT devices, the unchangeable nature of blockchain data guarantees the integrity of agreements and supports the resolution of disputes with a clear and verifiable record. Furthermore, the immutability feature of blockchain technology aligns seamlessly with the principles of transparency and accountability. Participants in the IoT network, whether they are individuals, organizations, or regulatory bodies, can confidently rely on the blockchain's unalterable ledger to verify the authenticity of data. This not only reduces the risk of fraudulent activities but also streamlines processes that require a high degree of accuracy and reliability. In summary, the immutability of data on the blockchain not only safeguards the integrity of information generated by IoT devices but also establishes a foundation of trust and accountability. It serves as a technological guarantee, assuring stakeholders that the recorded data remains unchanged and accurate, even in the most critical and highly regulated environments.

- **Reduced Costs and Efficiency:** The integration of blockchain into IoT applications brings about a transformative paradigm shift by diminishing reliance on intermediaries and introducing streamlined automation, ultimately leading to significant cost reductions. One of the primary mechanisms through which blockchain achieves this is the utilization of smart contracts. These self-executing contracts operate on predefined rules and conditions encoded within the blockchain, effectively automating processes that traditionally required intermediaries for validation and execution. Smart contracts play a pivotal role in reducing the need for intermediaries in various facets of IoT applications. For instance, in supply chain management, where numerous entities are involved in the production and distribution processes, smart contracts can automate payment settlements upon the completion of predefined milestones, eliminating the necessity for financial intermediaries. This not only accelerates transaction times but also slashes associated costs by removing the fees associated with intermediary services. The automation facilitated by smart contracts further translates into substantial time and resource savings. In scenarios like asset tracking, where IoT devices continuously update the blockchain with real-time information on the location and condition of

goods, smart contracts can trigger automated actions, such as rerouting shipments in response to unforeseen events. This automation not only expedites decision-making processes but also reduces the need for human intervention, thereby optimizing resource utilization and minimizing operational delays. Moreover, the transparency inherent in blockchain technology contributes to operational efficiency by mitigating disputes and errors. The decentralized and tamper-resistant nature of the blockchain ledger ensures that all participants within the IoT network have access to a consistent and auditable record of transactions. This transparency not only enhances trust among stakeholders but also acts as a preventative measure against disputes arising from discrepancies in recorded data. In instances where disputes do occur, the transparent and traceable nature of the blockchain enables swift resolution by providing an indisputable record of events. Blockchain's ability to eliminate intermediaries and introduce automation through smart contracts has a profound impact on the cost efficiency of IoT applications. By reducing the need for manual intervention, cutting transaction times, and fostering transparency, blockchain not only streamlines operations but also translates into tangible cost savings, making it a pivotal enabler for the widespread adoption of efficient and cost-effective IoT ecosystems.

- **Enhanced Scalability:** Certain blockchain architectures, specifically Directed Acyclic Graphs (DAGs), present a revolutionary approach to scalability, addressing a critical need in the context of IoT applications characterized by a vast and continuous influx of data from myriad devices. Unlike traditional blockchains with a linear chain structure, DAGs leverage a more intricate network topology, facilitating parallel transaction processing and thereby offering notable advantages in scalability.

- In the realm of IoT, where an ever-expanding network of devices continuously generates and transmits data, scalability is paramount for the sustainable and efficient functioning of the system. DAGs, by design, allow for the simultaneous confirmation of multiple transactions, eliminating the bottleneck effect often observed in linear blockchain structures. This parallel processing capability significantly enhances the overall throughput of the network, ensuring that as the

number of devices and transactions increases, the blockchain system can handle the rising volume without compromising performance. The scalability advantages of DAGs become particularly relevant in scenarios where real-time data processing is crucial, such as in smart cities where connected devices monitor and respond to various urban parameters, or in industrial IoT applications where sensors collect and transmit data from manufacturing processes. In these contexts, the ability to rapidly process a high volume of transactions is not just desirable but essential for ensuring timely decision-making, responsiveness, and overall system efficiency. Moreover, the scalability of DAGs aligns with the dynamic nature of IoT ecosystems. As new devices are added to the network or existing ones increase their data output, the blockchain must adapt to accommodate the growing demands seamlessly. Scalable blockchain solutions, especially those employing DAG architectures, provide the necessary flexibility to meet the evolving needs of IoT applications, ensuring that the network can expand organically without encountering performance bottlenecks. In essence, the scalability advantages offered by blockchain architectures like DAGs play a pivotal role in enhancing the viability and efficiency of IoT applications. By empowering blockchain networks to handle the continuous stream of data from numerous devices simultaneously, scalable solutions contribute to the seamless integration of IoT technologies into various industries, fostering innovation and responsiveness in the face of the ever-expanding landscape of connected devices.

- **Improved Data Ownership and Privacy:** Blockchain, with its emphasis on decentralization and privacy-centric features, empowers users with unprecedented control over their data, especially in the context of the Internet of Things (IoT). This newfound control is exemplified through decentralized identity solutions, a revolutionary concept that grants users the ability to retain ownership of their IoT-generated data while selectively granting permissions. In traditional IoT ecosystems, data generated by devices often resides in centralized databases controlled by service providers or organizations. This centralized model raises concerns about data ownership, security, and privacy. Blockchain disrupts this paradigm by distributing the ownership and control of data back to the users. Decentralized identity solutions on the blockchain enable individuals to have a unique, self-sovereign digital identity. This identity, secured by cryptographic

principles, serves as a gateway for users to manage access to their IoT data autonomously. The selective permission model facilitated by blockchain allows users to decide who can access their data and for what purpose. In healthcare, for instance, patients can have granular control over their medical IoT data, deciding which healthcare providers or researchers can access specific information. This not only aligns with principles of patient privacy but also ensures that sensitive health data is shared responsibly and ethically. This user-centric data ownership model is equally relevant in personal IoT devices, where individuals generate data through wearables, smart home devices, or other personal gadgets. With blockchain, users can selectively share data with trusted applications or services, preserving their privacy while still benefiting from the functionalities offered by the IoT devices. This shift towards decentralized data ownership is pivotal in addressing privacy concerns associated with the increasing ubiquity of IoT devices in daily lives. Furthermore, blockchain's tamper-resistant nature ensures the integrity of the permissions and access controls set by users. Once permissions are recorded on the blockchain, they cannot be altered without the user's explicit consent. This immutability strengthens the security of user data and instils confidence in individuals, knowing that their preferences regarding data access and sharing are maintained with the highest level of integrity. Blockchain's role in empowering users with greater control over their IoT-generated data through decentralized identity solutions aligns with fundamental principles of privacy and data ownership. This has transformative implications across various industries, assuaging concerns related to sensitive data, fostering trust in IoT ecosystems, and establishing a more ethical and user-centric approach to the management of personal information in the digital age.

The scope and advantages of incorporating blockchain into IoT applications extend from enhancing security and trust to enabling automation through smart contracts, ensuring interoperability, improving data integrity, reducing costs, and providing a scalable and privacy-centric framework for the evolving landscape of connected devices.

7.2 Role of Blockchain in Risk Management and Parametric Insurance Business

Blockchain, the decentralized and distributed ledger technology, is set to redefine the landscape of risk management within the insurance industry, ushering in unparalleled levels of transparency, efficiency, and security. The fundamental principles of blockchain – immutability, traceability, and consensus – hold the promise of alleviating longstanding challenges within the sector. Foremost among these benefits is the enhancement of data integrity and trust. By securely storing policy information, claims history, and risk assessments on the blockchain, accessible only to authorized parties, the technology eliminates data silos, minimizes fraud, and fosters trust among insurers, reinsurers, and policyholders. The integration of smart contracts automates key processes, including policy issuance, premium payments, and claims processing, thereby reducing human error, and streamlining workflows. In addition to bolstering data integrity, blockchain revolutionizes risk assessment and pricing. Real-time data sourced from connected devices, sensors, and external channels feeds directly into the blockchain, empowering insurers to assess risks in real-time and personalize premiums based on individual behaviours and changing circumstances. This dynamic approach not only promotes proactive risk mitigation but also ensures fairer pricing structures, aligning more closely with the specific risk profiles of policyholders.

The transformation extends to claims processing, where blockchain's automated verification of claims against policy terms and instant access to relevant data expedite the entire process. This results in reduced administrative costs and shorter settlement times. Smart contracts, a cornerstone of blockchain technology, can even trigger payouts automatically upon the fulfilment of pre-defined conditions, further enhancing efficiency and elevating customer satisfaction.

Furthermore, blockchain facilitates collaborative risk sharing across the insurance ecosystem. The technology enables the creation of consortium-based platforms where insurers can share anonymized data and collaborate on risk pools. This collaborative approach allows for the efficient spreading of liabilities, improving overall solvency in the industry. Such collaboration fosters innovation in product development and risk

management strategies, ultimately benefiting policyholders with a broader array of coverage options and more competitive premium rates.

While acknowledging the challenges, such as regulatory hurdles and scalability concerns, it is undeniable that blockchain has the potential to transform risk management within the insurance sector. Embracing this disruptive technology allows insurers to unlock a future characterized by enhanced transparency, efficiency, and resilience, paving the way for a more secure and prosperous industry for all stakeholders involved.

In the insurance sector, one of the most significant applications is Parametric insurance, a groundbreaking approach that finds its ideal companion in blockchain technology. Departing from traditional claim investigations, Parametric insurance relies on objective data triggers, and blockchain serves as the secure, distributed ledger that facilitates this innovative paradigm. Envision a scenario where flight delays automatically initiate payouts or a drop in crop yields below a predetermined threshold triggers immediate funds release to farmers. This is the futuristic realm of blockchain-powered parametric insurance, where smart contracts, encoded onto a secure ledger, autonomously execute payouts based on predefined parameters sourced from reliable oracles.

The operational process unfolds as follows:

- **Automated Oracles Deliver Data**: Various sources such as sensors, weather stations, and flight trackers feed real-time information into the blockchain through oracles. These oracles act as objective and tamper-proof witnesses, eradicating the subjectivity and potential bias associated with traditional claims adjusters. This ensures fair and prompt settlements.

- **Smart Contracts Execute Payouts**: Policy terms are encoded as self-executing smart contracts, eliminating the need for manual intervention. When predetermined parameters, such as a delayed flight or a specific level of rainfall, are met, the smart contract automatically triggers the payout, transferring funds directly to the insured party's digital wallet. This not only removes friction but also reduces administrative costs and ensures nearly instantaneous financial relief.

- **Democratizing Access and Coverage**: By eliminating human gatekeepers and relying solely on objective data, blockchain-powered parametric insurance broadens access to underserved populations and unconventional risk categories. Smallholder farmers in remote areas can now access weather-based crop insurance, micro-entrepreneurs can safeguard against lost shipments, and gig workers can insure against income fluctuations—all without the burden of cumbersome paperwork or fear of claim denial.

- **Enhanced Transparency and Trust**: The immutable nature of blockchain guarantees transparency and auditability at every stage, from data input to payout execution. This fosters trust among all stakeholders, including insurers, reinsurers, and policyholders. Moreover, the decentralized nature of the ledger makes it resistant to fraud and manipulation, further reinforcing confidence in the system.

- **Paving the Way for Innovation**: As reliance on centralized authorities diminishes, parametric insurance on the blockchain opens avenues for creative product development. Consider index-based insurance tracking economic indicators or parametric coverage for political instability or cyberattacks. The potential for innovation is boundless, fuelled by the limitless capabilities of data and automation.

Despite existing challenges related to regulatory frameworks and technical obstacles, the undeniable synergy between parametric insurance and blockchain is reshaping the insurance landscape. By automating payouts, democratizing access, and fostering trust, this powerful alliance is poised to make the insurance industry faster, fairer, and more inclusive for all.

7.3 Industrial Use Case: Parametric Transport and Logistics Insurance

Parametric insurance contracts, characterized by explicit and easily determinable parameters such as wind speed, weather data, and seismic conditions, have become

integral components in covering disaster risks within the insurance industry. While the current landscape predominantly embraces parametric insurance for disaster-related risks, other sectors like logistics, travel, retail, agriculture, and health insurance are poised for potential disruption as well [102]. Traditional insurance plans, burdened by intricate claim procedures, often lead to prolonged claim resolution times, presenting a stark contrast to the streamlined and autonomous claims processing offered by parametric insurance, where predetermined conditions ensure consensus. The insurance sector finds itself at a crossroads, grappling with a confluence of factors including emerging technology, evolving consumer expectations, and the advent of novel business models. This shifting landscape, coupled with the heightened unpredictability arising from the increased frequency and severity of incidents related to the risk trifecta of climate change, geopolitical unrest, and cyberattacks, has prompted insurers to turn to parametric solutions. Positioned as a flexible and cost-effective alternative, parametric insurance is gaining traction in the market. The industry, valued at $11.7 billion in 2021, is projected to witness substantial growth, reaching $29.3 billion by 2031, with an estimated Compound Annual Growth Rate (CAGR) of 9.9% from 2022 to 2031, as indicated by this comprehensive analysis [103]. This expansion underscores the industry's recognition of the need for adaptive and responsive insurance solutions in the face of an increasingly complex and dynamic risk environment.

The efficient movement of fleets within the transportation and logistics supply chain is a critical physical operation, often occupying a pivotal position in various scenarios. Traditional insurance plans designed for fleet and transportation coverage typically adopt a group policy approach, encompassing all vehicles within a unified policy group. Generally, this insurance provides protection against loss or damage to automobiles resulting from unforeseen events like fire, accidents, theft, and natural disasters. The management of information related to indemnity policies and regulations is centralized, involving various system processes and human-centric tasks before the distribution of claims. The end-to-end processing of a claim is often prolonged due to the need for the insurance company to verify multiple features and criteria.

Typically, fleet insurance covers either a single vehicle or a group of cars, with limitations on addressing commercial losses incurred in the supply chain due to fleet accidents. Even when partial coverage is available, obtaining a separate policy is often necessary to comprehensively protect against commercial losses. For instance, if a vehicle transporting perishable goods experiences a mishap due to an internal factor

outlined in the fleet policy (such as a breakdown), the insurance may cover repair costs. However, if perishable goods sustain damage due to a delivery delay, the responsibility for associated losses falls on the supply chain stakeholders, as these losses are not typically covered.

The existing challenge lies in the scarcity of insurance policies that specifically address such supply chain losses, requiring extensive proof before claims are processed and payments are approved. The intricacies of these scenarios underscore the need for insurance solutions that go beyond conventional fleet coverage, acknowledging the broader implications of delays and losses within the logistics and supply chain framework. The following are the key advantages of parametric insurance over regular policies in the transportation and logistics area.

- **Faster Payout** - Indemnity insurance payouts are subject to a drawn-out verification process. Following the loss adjustment process, a payout is determined, and the disbursement process is started. The payment process is faster with parametric insurance because it is based on an observed and independent event. In general, indemnity insurance takes months to pay out after receiving a claim notification, compared to parametric insurance, which typically pays out in days.

- **Flexibility in Process** - Design of parametric insurance is very flexible and adaptable on a need-basis, unlike indemnity insurance policies. If a fleet's "delay" event is used to represent parametric insurance, compensation can be determined based on how long the delay lasts. Even the type of delay, such as weather, lane-related accidents, traffic congestion, etc., might be parameterized in terms of compensation consideration.

- **Coverage for difficult-to-model losses** - Any monetary loss that the stakeholder has experienced because of the covered event may be covered by the payout from a parametric policy. This allows the insured to allocate funds as they believe is appropriate. Given the high level of uncertainty around prospective losses, parametric insurance is particularly suited for new and developing risks that may not be covered by indemnity policies.

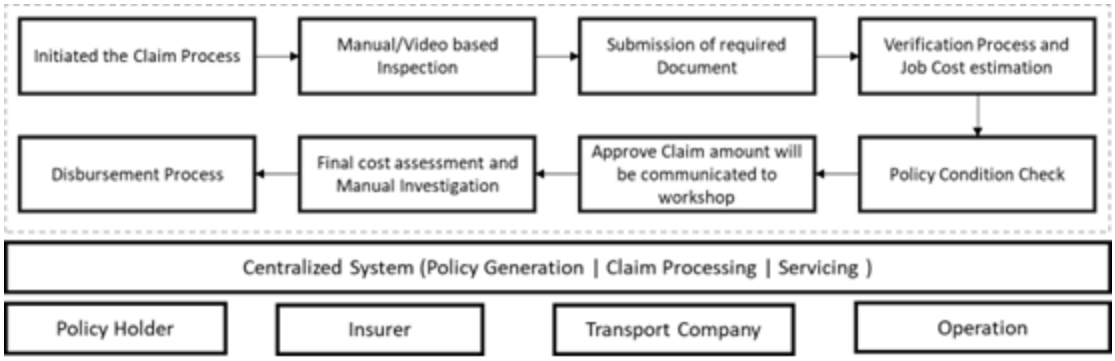


Figure 7. 1 Traditional fleet insurance processing

Figure 7.1 illustrates the conventional claim process for fleet insurance, depicting the sequence of steps involved. In the event of a physical occurrence, such as an accident, affecting the insured fleet, the policyholder initiates the process by submitting a claim to the insurance provider. Subsequent to a thorough examination, both physically and digitally, the claim is officially registered, accompanied by the submission of supporting documentation in both physical and digital formats. The operations team of the insurance company then undertakes a series of verifications before granting approval for the claim. The verification process encompasses various checks and assessments to ensure the validity and accuracy of the claim. These checks may involve scrutinizing the provided documentation, evaluating the details of the incident, and confirming compliance with the policy's stipulated rules and definitions. Only after the operations team completes all necessary verifications does the claim distribution process commence, adhering to the predefined rules and definitions outlined in the insurance policy. This systematic approach ensures a comprehensive and meticulous evaluation of claims, promoting accuracy and adherence to policy guidelines before the disbursement of claims to the policyholder. In summary, the fundamental operational challenges identified within the existing fleet insurance business domain are outlined below:

- Each insurance company centralizes the management of its user base and claim processing data.
- Utilization of opaque procedures that are time-consuming.
- Implementation of a new revenue-generating business plan by the fleet insurance company.

- Brand recognition and claim settlement rates have become pivotal indicators of trust due to a lack of information transparency.
- The need for technological advancements to support contemporary business models.

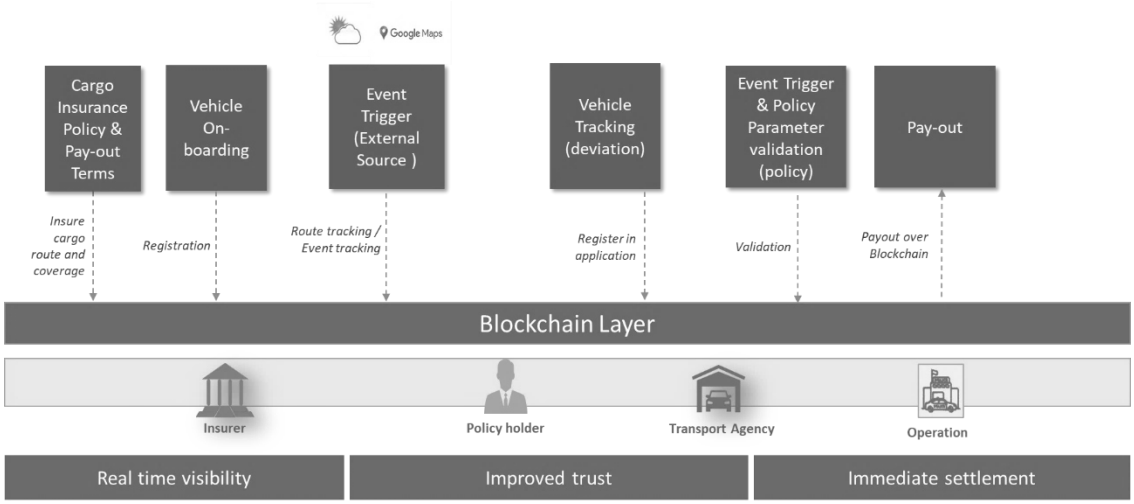


Figure 7. 2 Parametric fleet insurance approach

The methodology illustrated in Figure 7.2 employs smart contracts to address specific events, such as those triggered by traffic incidents, accidents, or adverse weather conditions, which can autonomously lead to the initiation of a claim payment. This innovative solution is devised through the implementation of smart contracts, which are digital contracts rooted in blockchain technology, featuring constraints linked to their execution. Operating on the pre-programmed logic embedded within the smart contract, claims are automatically disbursed in response to the occurrence of predefined events, streamlining and expediting the claims settlement process. This approach leverages the decentralized and transparent nature of blockchain technology to automate and enhance the efficiency of claim payments, ultimately fostering a more responsive and seamless insurance experience. Following are major business and technical benefits achieved out of this solution:

- Elimination of paperwork, as the policy is set up on the blockchain.
- Elimination of human touchpoints from claims processes.
- Intermediaries are removed from the process.
- Immediate payouts for a covered event are enabled.
- Cut down the administration and claim processing costs.

- New business model for revenue generation.
- Win-Win condition for both insurer and insurance organization.
- Working principle is based on trust less highly decentralized model.
- Adaptation of cutting age digital technology stack.
- Security and information transparency at its core.

7.3.1 The Strategic Blueprint: Optimizing Solution Topology

By innovatively addressing coverage demands, insurers can not only reduce operational costs but also enhance transparency in coverage and expedite claims processing. A transformative approach in this regard is the adoption of parametric insurance, which, when coupled with cutting-edge digital technology, has the potential to revolutionize the insurance sector. This paradigm shift relies on an index, often referred to as the delay factor, which gauges the likelihood of trigger events such as malfunctions, accidents, or changes in weather conditions. The parametric insurance model ensures predefined compensation in correlation with the severity of the identified occurrences. The development of the parametric platform solution was guided by key principles aimed at maximizing its effectiveness. These principles include:

- **The solution should be configurable** - the platform and solution should be able to manage a variety of parametric use cases with minimal modifications.
- **The solution should be extensible** - The solution architecture should be designed to be extensible so that any additional use cases, such as onboarding a new supply chain member, extending data transparency to a new participant, or adding an event or index, can be integrated with the existing solution.
- **Insureds' loss prevention should be a primary concern** - To lower the loss factor, insureds should use the risk mitigation and preventive services provided by the insurer.
- **Conditions should be transparent** – The specifications for parametric insurance coverage should be clear enough for average people to comprehend them and should not include any legal or difficult terminology.
- **Risk assessment in real time and an automated process** - Real-time risk assessment should be performed based on index factors for higher ROI. Claim

assessment and processing should be fully automated to reduce the payment cycle if policy criteria are fulfilled.

- **Simple, diverse coverage for the ecosystem** – Simplicity is the core of the solution. It is simpler to price risk and manage operations when the product and coverage criteria are concise. The solution must be effective enough to address a wide range of neglected and underinsured markets. The solution should also address a variety of risks related to the stakeholders involved, including insurance companies, aggregators, policyholders, and technology service providers.

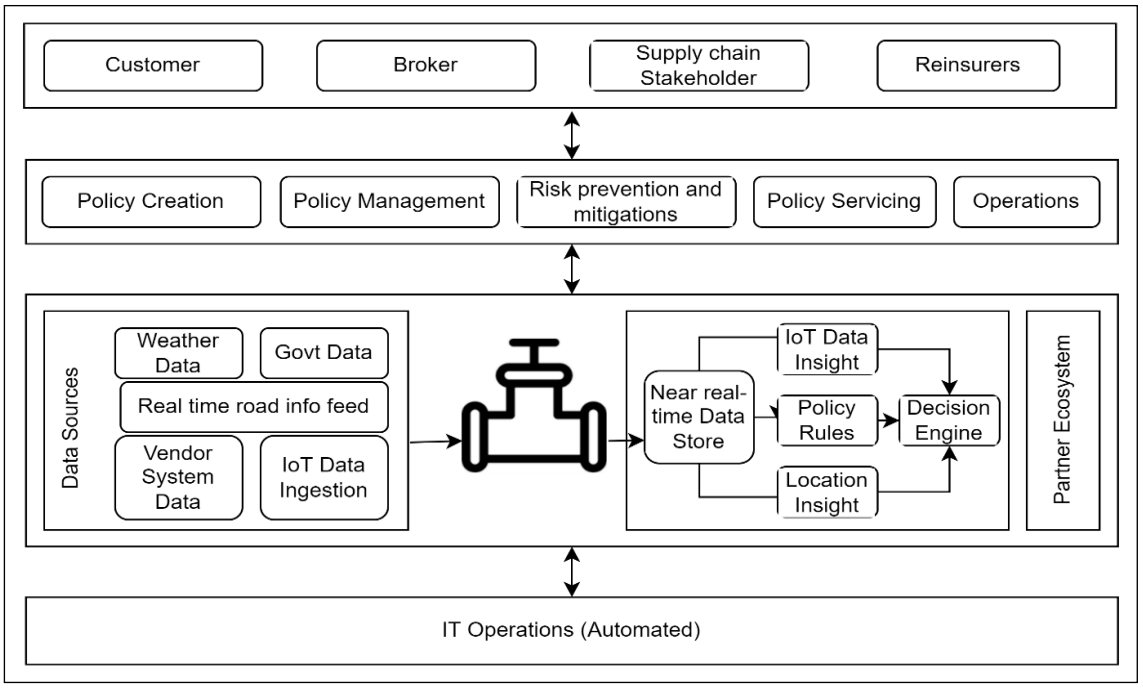


Figure 7. 3 Reference architecture of parametric transport digital insurance platform

Figure 7.3 illustrates the reference architecture of the parametric digital platform, comprising distinct layers that collectively enable the creation, management, and servicing of policies. Layer 1 encompasses actors and stakeholders, interacting with the platform via APIs or user interfaces to engage in various policy-related activities. Digital insurance services, available to Layer 1 users, form Layer 2, laying the foundation for the seamless provision of insurance solutions in the digital realm. The crux of the digital parametric insurance solution lies in Layer 3, where, in the case of parametric transport insurance, real-time data is systematically integrated into the system. The execution of policy rules is facilitated through context event processing and intelligent decision-making systems. The intricate workings of the parametric

digital platform are further supported by the IT operation platform, constituting the IT infrastructure, services, and data storage necessary for the platform's intended functionality. This underlying IT operations layer must exhibit a level of intelligence that goes beyond mere responsiveness—it should be predictive. The predictive capabilities ensure coverage is proactively ensured based on the real-time capture and triggering of events. This predictive nature is crucial for the system to handle multiple events concurrently through an effective triggering mechanism, contributing to the platform's agility and efficiency. In essence, the layered architecture of the parametric digital platform underscores its sophistication, where real-time data, context event processing, intelligent decision-making, and predictive IT operations synergistically converge to deliver a robust and dynamic parametric insurance solution.

7.3.2 Unravelling the Dynamics of System Design

For the purposes of this experiment, a Parametric Transport Insurance solution that automatically processes claims based on external data sources and creates an audit trail of claim processing outputs and sources of data verification in an immutable fashion for the logistics business is used. I used a smart contract-based parametric insurance use case from the transportation industry. The use case deals with the possibility of supply chain business losses brought on by freight transportation caused by delayed arrival or route diversion. Assuming that this is only applicable to trip delay insurance for trucks, the following factors must be considered when generalizing the proof of concept into a production-ready service:

- Add transport companies that can obtain insurance.
- Allow trucking companies to run blockchain nodes but not require it.
- Ensure that information that is confidential to one customer is not accessible to others.
- Allow each customer to choose their own parameters, thresholds, triggers, and payment levels within the parameters set by the product.
- Allow the use of trip-specific parameter values and payments.
- Provide a way for trigger events and parameter values to be set with digital authentication at the time policy triggers and parameters are created.

- Authenticate and provide proof of such authentication when the triggering event happens.

A. Functional Considerations: A trigger, parameter values, and payment trigger rules make up the definition of parametric insurance. Each insured organization may have a separate set of triggers, parameter values, and rules as there may be several insured organizations. This version may be organized by having distinct phases for configuration, definition, and execution.

- Part of Configuration, each insured can have one or more master policies which defines i) the trigger and a way to authenticate the trigger, ii) parameter values and a way to authenticate them, iii) rule(s) that trigger the payments and iv) the payment method.
- Part of Definition, every time a trip is to be insured a policy is instantiated from the master policy. Attributes are i) the master policy and ii) trip specific parameter values with authentication.
- In Execution, a trigger event happens with i) an authentication and ii) trigger specific parameter values with authentication.

B. Scalability: In the pursuit of enhanced oversight and management of claim evaluation and payment processes, newly established insurance entities may opt to deploy nodes on the Blockchain network. The system is meticulously designed to facilitate the seamless addition of nodes, insured companies, and policies over time, accommodating the potential influx of concurrent trips and triggering events—an aspect carefully considered in the architecture design. Leveraging Hyperledger Fabric as the blockchain framework of choice, renowned for its enterprise-grade capabilities, scalability in a production environment is achieved through several strategic measures. Hyperledger Fabric's modular architecture offers horizontal scalability by facilitating the seamless addition of more nodes to the network. Each node can operate on a separate virtual machine, simplifying the scaling process and enhancing network capacity. The Channels system within Hyperledger Fabric further contributes to scalability by enabling the operation of multiple independent networks within the same blockchain infrastructure. This allows for parallel processing of transactions, significantly boosting the overall scalability of the

system. Efficient data management is pivotal to scalability, and Hyperledger Fabric addresses this through its capability to store private data on-chain and manage state data effectively. This approach plays a crucial role in reducing the size of the blockchain, thereby enhancing the network's scalability without compromising on data integrity. The endorsement policies inherent to Hyperledger Fabric represent another key factor in augmenting scalability. These policies offer a flexible approach to transaction validation, enabling nodes to swiftly endorse transactions. This flexibility streamlines the validation process, contributing to increased scalability by expediting transaction endorsement across the network.

The incorporation of Hyperledger Fabric in the blockchain architecture not only aligns with enterprise-grade standards but also strategically employs modular design, channels system, efficient data management, and endorsement policies to ensure scalability in the production environment. This multifaceted approach positions the system to adeptly handle the potential growth and complexity associated with an expanding network of nodes, insured companies, and policies over time.

C. **Blockchain and Smart Contracts:** The following architectural factors are considered as smart contracts run on blockchain nodes.

- Latency and Throughput - Each smart contract execution is a Blockchain transaction, subject to validation and consensus, which imposes restrictions on throughput and latency. This would necessitate running a portion of the claim engine off chain; ideally, all calculations related to a particular trip should run off chain.
- Rule Execution - Complex rules engines cannot be run by smart contracts, hence any necessary rules must either be mapped into the contract code or executed off chain. If rules are mapped within smart contract, every new rule needed by a master policy necessitates the coding and deployment of a new smart contract, which can increase the maintenance required for the claims engine. Therefore, it is advised that rule review take place off chain.

- Triggering - Smart contracts cannot call external services for parameter values or cause events to occur outside of a Blockchain. Furthermore, since everyone can

view every piece of information on the ledger, smart contracts are unable to retain any kind of credentials. As a result, an off-chain service is needed to both gather all parameter values required for the rule execution and to initiate payments once they have been computed.

According to the aforementioned factors, the following claim engine components must be off chain.

- Acquire the trigger and gather the pertinent parameter values.
- Consider the regulations that demand payments.
- Execute the payments.

On the other hand, the tamper-resistance of a blockchain ledger can be used to,

- Keep track of authentication methods.
- Perform authentication (subject to meeting the scalability requirements).
- Keep track of critical events: trigger receipt, parameter values, proof of authenticity, proof of payments, etc.

D. Network System Participants: Both the organization and a chosen subset of insured companies possess the eligibility to function as nodes on the Blockchain network. As numerous insured companies actively participate in the network, granting them access to critical information, the preservation of data privacy emerges as a paramount concern during the meticulous development of the network architecture. Deliberate considerations are made regarding the type of information destined for storage on the ledger, ensuring that the architecture not only fosters seamless collaboration but also upholds the confidentiality and security of sensitive data within the network. This strategic approach is fundamental in establishing a robust and trustworthy blockchain infrastructure that prioritizes the privacy concerns inherent to the diverse entities operating within the network.

E. Security: When deploying and configuring Blockchain nodes on the corporate network, it is imperative to meticulously consider both network security and the safeguarding of enterprise information, especially considering potential interactions with internal systems. This involves a comprehensive evaluation of security

protocols and measures to ensure the seamless integration of Blockchain nodes while upholding the integrity and confidentiality of critical enterprise data. The deployment process should be conducted with a focus on mitigating potential risks and vulnerabilities, aligning with established network security standards, and integrating seamlessly with the existing enterprise infrastructure. By prioritizing these aspects, organizations can fortify the resilience of their Blockchain network, creating a secure and cohesive environment that effectively interacts with internal systems without compromising on data security.

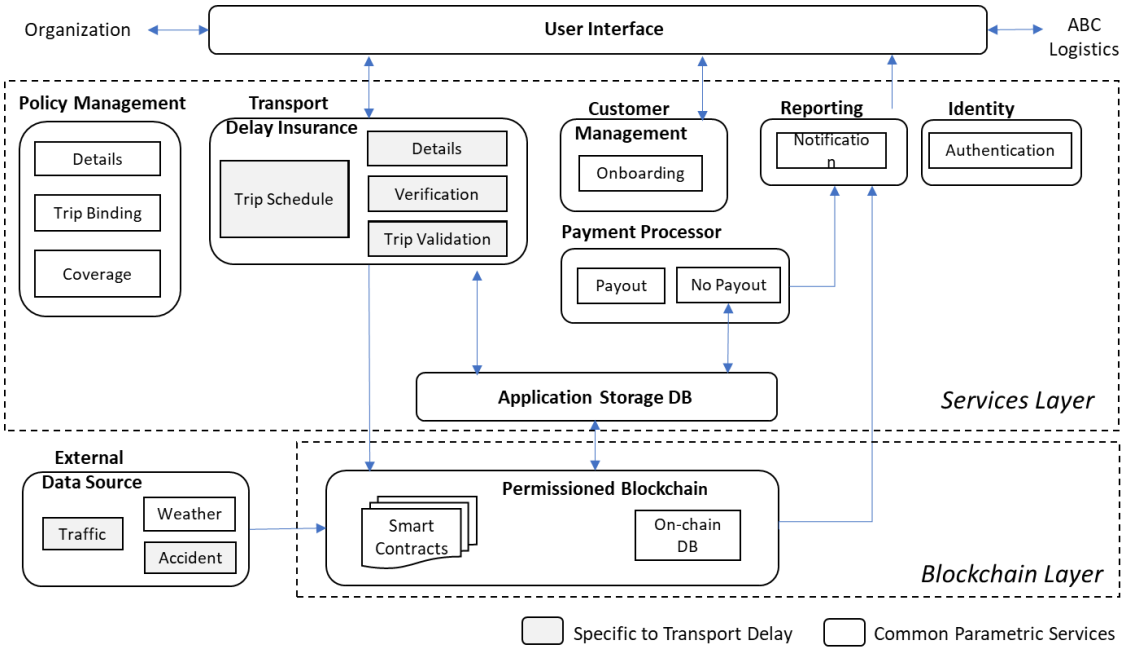


Figure 7. 4 Parametric insurance platform logical architecture

The delineation of a three-layer logical architecture, as depicted in Figure 7.4, introduces external data sources that play a pivotal role in facilitating seamless interactions within the blockchain ecosystem. These external data sources, authorized by participant parties, are classified into three distinct categories:

- Decentralized Oracle Networks: This category encompasses entities such as Chainlink and other decentralized oracle networks, acting as trusted third-party sources that contribute validated data to the blockchain network.
- Data Suppliers: The second category involves renowned data suppliers, including Tom-Tom, AccuWeather, and Google Maps, whose authorized data feeds enrich the blockchain with real-time and accurate information, enhancing the overall reliability of the system.

- Service-Oriented Companies: The third category comprises service-oriented companies like Digifleet and Digital Farming, offering specialized services that contribute to the robust functionality of the blockchain ecosystem.

Within the service layer, a cohesive group of modular components takes charge of managing the intricacies of claims processing, payments, customer interactions, and coverage. These components collectively constitute a dynamic framework that orchestrates the execution of smart contracts at various stages during the off-chain phases, ensuring the seamless flow of information and transactions. Situated above this service layer is the permissioned blockchain layer, where smart contracts are strategically employed to authenticate data as it is placed on-chain. This layer ensures the integrity and reliability of data stored on the blockchain, reinforcing trust among network participants, and validating the information sourced from the external data providers.

In essence, this three-layer logical architecture harmonizes external data sources, modular service components, and permissioned blockchain layers, creating a robust and secure framework that not only leverages trusted third-party data but also efficiently manages and authenticates on-chain processes through the strategic execution of smart contracts.

7.3.3 From Concept to Reality: Steps to Effective Implementation

A. Application Architecture and Solution Steps

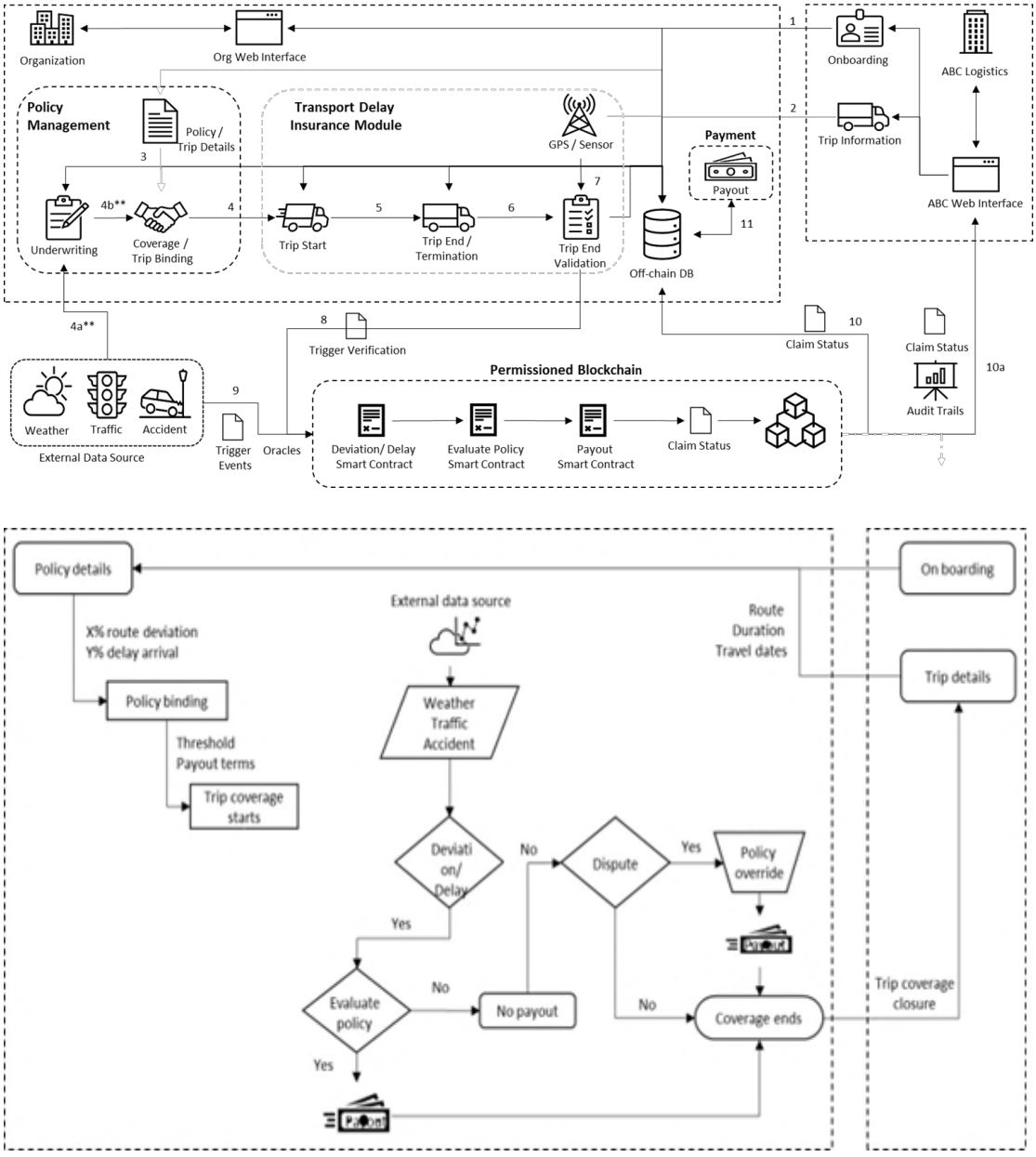


Figure 7. 5 a. Application architecture b. Process flow diagram

Illustrated in Figures 7.5a and 7.5b, the comprehensive solution architecture and process flow for the transport parametric solution pilot are delineated. The envisaged solution leverages Hyperledger Fabric blockchain technology, renowned for its

attributes as an open-source, permission-enabled, immutable distributed ledger technology (DLT) platform. Tailored for business contexts, Hyperledger Fabric offers distinctive advantages in terms of privacy and scalability compared to other prominent distributed ledger or blockchain systems. The permissioned nature of the Fabric platform ensures that users are not anonymous to one another, emphasizing a secure and accountable environment. Certificates will be duly granted to network users and the requisite web application, enabling seamless communication and transaction submissions on the blockchain platform. Web applications interact with blockchain smart contracts through REST APIs provided by the blockchain platform, known as chaincode. These smart contracts, in turn, facilitate the capture of essential information, including onboarding, policy details, threshold values, travel specifics, and more. Utilizing smart contract query capabilities, the web application gains insights into blockchain data, delving into aspects such as payout determinations and claim statuses. Concurrently, application databases, referred to as off-chain databases, remain synchronized with blockchain storage at the organizational level, ensuring coherence in data representation and enabling efficient internal queries.

Data, originating from various logistics businesses, is acquired almost in real-time through the blockchain by the organization. Conversely, data generated within the organization is also promptly transmitted to the blockchain. Simulated trip end parameters are dispatched to the blockchain, invoking a smart contract that scrutinizes potential delays or deviations in the journey. External data sources are then employed by the smart contract to discern whether these deviations result from factors such as traffic, accidents, or weather conditions. Upon validation, the smart contract adjusts the payout status to True and marks the claim state as Approved if the travel delay or deviation aligns with acceptable external circumstances (as outlined in Table 1). Conversely, if the reasons are deemed unacceptable, these attributes are denoted as False and Rejected, respectively. For a more detailed understanding, the following includes pseudo code and a state change diagram elucidating the intricate validation process.

If Trip Delay == delayDuration in events, then

if not routeDeviation then

check for eventTypeCode

if evenTypeCode matches then

Set Payout Status = "Paid"

else

Set Payout Status = "Not Covered"

else

Set Payout Status = "Not Covered"

else

Set Payout Status = "Not Covered"

Set Coverage Status as "Coverage End"

If Trip Status == "Cancelled" then

set Payout = "NA" and

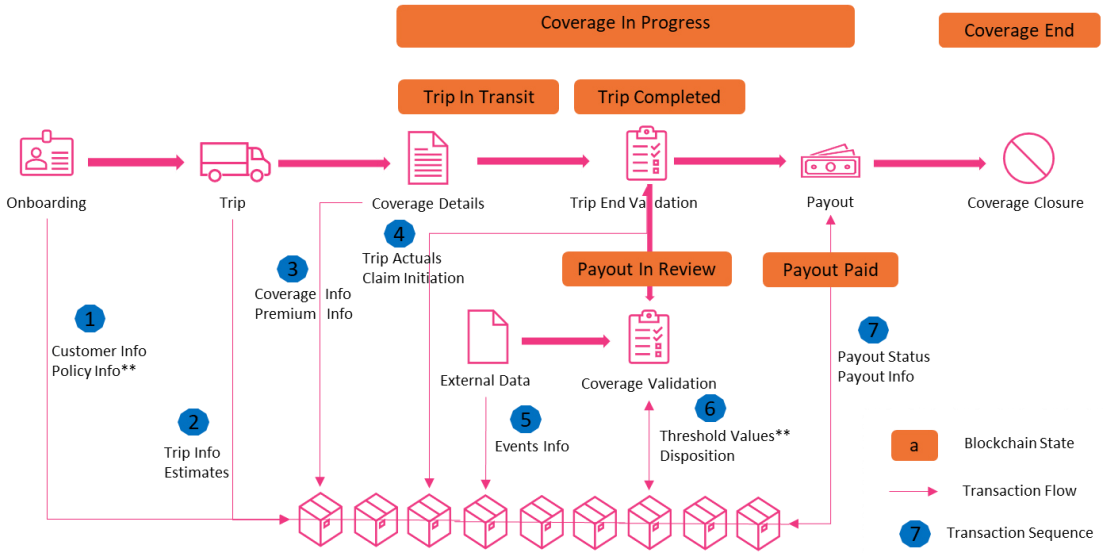
Coverage Status = "Coverage End"

Else

Trip Delay = Trip End Actuals - (Projected End Time + 10% of Projected End Time)

If Trip Delay > 0 then

Populate Payout status as "In Review"



Allowed Categories	0. Unknown 1. Accident 2. Fog 3. Dangerous Conditions 4.Rain 5. Ice 6. am 7. Lane Closed 8. Road Closed 9. Road Works 10. Wind 11. Flooding 12. Broken Down Vehicle

Scenario	Estimated Duration	Projected End Time	Actual Arrival Time (with Buffer)	Delay	Route Deviation	External Events			Payout Status
						Weather	Traffic	Accident	
Vehicle reached on or before the scheduled time at destination	HH	HH	HH:MM	Yes | No | NA	Yes | No | NA	Yes | No | NA	Yes | No | NA	Yes | No | NA	Eligible | Not Eligible
Due to unknown reason vehicle took a different route and was NOT able to reach destination within defined time for the trip	HH	HH	HH:MM	Yes | No | NA	Yes | No | NA	Yes | No | NA	Yes | No | NA	Yes | No | NA	Eligible | Not Eligible
Due to unknown reason vehicle took a different route and was NOT able to	HH	HH	HH:MM	Yes | No | NA	Yes | No | NA	Yes | No | NA	Yes | No | NA	Yes | No | NA	Eligible | Not Eligible

reach destination within defined time for the trip									
There was accident enroute due to which vehicle took a different route and was NOT able to reach destination within defined time for the trip	HH	HH	HH:MM	Yes | No | NA	Yes | No | NA	Yes | No | NA	Yes | No | NA	Yes | No | NA	Eligible | Not Eligible
There was traffic enroute due to accident and vehicle took a different route and was NOT able to reach destination within defined time for the trip	HH	HH	HH:MM	Yes | No | NA	Yes | No | NA	Yes | No | NA	Yes | No | NA	Yes | No | NA	Eligible | Not Eligible
There was heavy rain enroute due vehicle was NOT able to reach destination within defined time for the trip	HH	HH	HH:MM	Yes | No | NA	Yes | No | NA	Yes | No | NA	Yes | No | NA	Yes | No | NA	Eligible | Not Eligible

Table 7. 1 Payout Rule table based on execution scenarios

The smart contract, upon computation of the claim status, will officiate the legal conclusion of the coverage life cycle, duly notifying both the Organization and the relevant Logistics company. This marks a crucial milestone in the streamlined and automated management of claims within the blockchain ecosystem. Both the Organization and individual Logistics companies hold the capability to access real-time status updates pertaining to claims and coverage. This transparency enhances accountability and ensures that all relevant stakeholders are well-informed throughout the entire process. In the future, the Organization's Blockchain nodes will be empowered with the capability to access detailed reports highlighting the number of issues requiring attention, rectification, and resolution. This valuable insight will be

accessible through a variety of application programming interfaces (APIs), providing a comprehensive overview for strategic decision-making and continuous improvement.

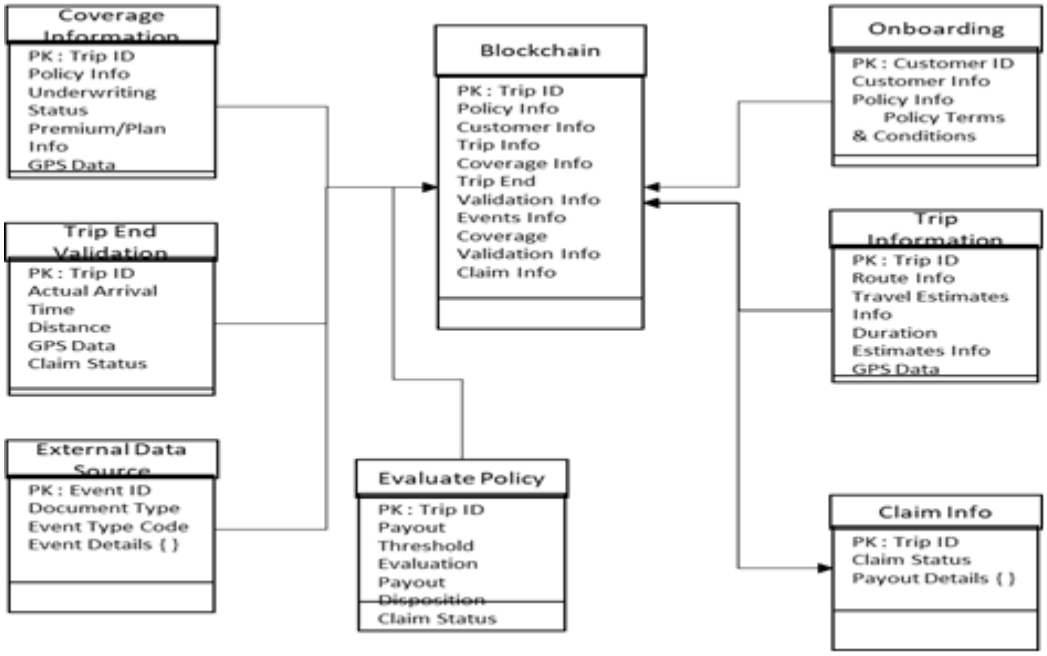


Figure 7. 6 Off-chain database data model

B. Smart Contracts and API Implementation

Following smart contract operations are implemented.

- *Add trip information* - Enter information about a new trip that will be covered by the policy.
- *Update trip info with coverage details* - Based on selected event criteria, trip info is updated.
- *Trip initiation* - Initiate the trip to track the policy coverage.
- *Trip end validation* - Validate the policy rules when the trip ends.
- *Injecting external events* - Based on the predefined time, events are captured from external event sources.
- Coverage validation - Validate the coverage terms.
- *Read Trip information* - Get Road Trip Information and Geocoordinates.
- Retrieve Claim status - Read current claim status once the event occurs.
- *Retrieve payout status* - Read the payout information and status.
- *Block information* - Read transaction data for auditing.

APIs are created for the key implementation functionalities listed below.

- Able to display KPIs for in transit trips, completed trips and payment trips.
- Able to add trip details and able to submit it and these details need to be submitted to database and blockchain.
- Able to see suggested premium for my coverage, total coverage value and parameters covered.
- Able to view trip summary which includes invoice number, premium details etc.
- Grouped trips with trip details, trip status and payment status with visual colours.
- Show complete information of trip, coverage, payment, and coverage parameters.
- Update trip end information and populate trip delay.

C. Technology Considerations and Infrastructure

The proposed strategy calls for a two-node blockchain network, where the first node stands in for the Organization and the second node for the logistics company. The solution that is being suggested is based on a decentralized distributed application (DApp) that makes use of the Hyperledger Fabric Blockchain platform as a blockchain protocol. There will be one peer for each organization. Under the AWS Managed Hyperledger Fabric Blockchain Platform [21], the Proof-of-Concept (PoC) solution will include:

- Single Channel
- Two Organizations
- One peer each
- Single Raft Ordering service with 3 consenters
- State DB using CouchDB
- Fabric CA for certificates and Identities
- The comprehensive technological and deployment architecture for this project, supported by Amazon Web Service (AWS) capabilities and integration, is shown in Figure 7. The main tools, platforms, and services used for this experiment are listed below.
- Platform - AWS BaaS and Linux VM
- Key and Certificate Management – Local storage and Fabric CA generated.
- Application Database – Couch DB

- Container Orchestration- Docker 18.06.1-ce, Docker compose 1.22.0
- Compute and DB Resources –

Org	Organization	Component	Stack	Hosting	vCPU	RAM (GB)	Disk(GB)	OS
DEV	Organization	Peer node 1, CouchDB 1	App	AWS BaaS	2	4	50	AWS Linux
		Blockchain client node	App	AWS EC2	2	4	50	AWS Linux
		Webserver App server	Web	AWS EC2	2	4	50	Ubuntu
		CA1, CA2, Orderer CA, Orderer and Off chain DB	App	AWS BaaS	2	4	50	AWS Linux
	Logistics	Peer node 2, Couch DB	App	AWS BaaS	2	4	50	AWS Linux

Table 7. 2 Resource Definition

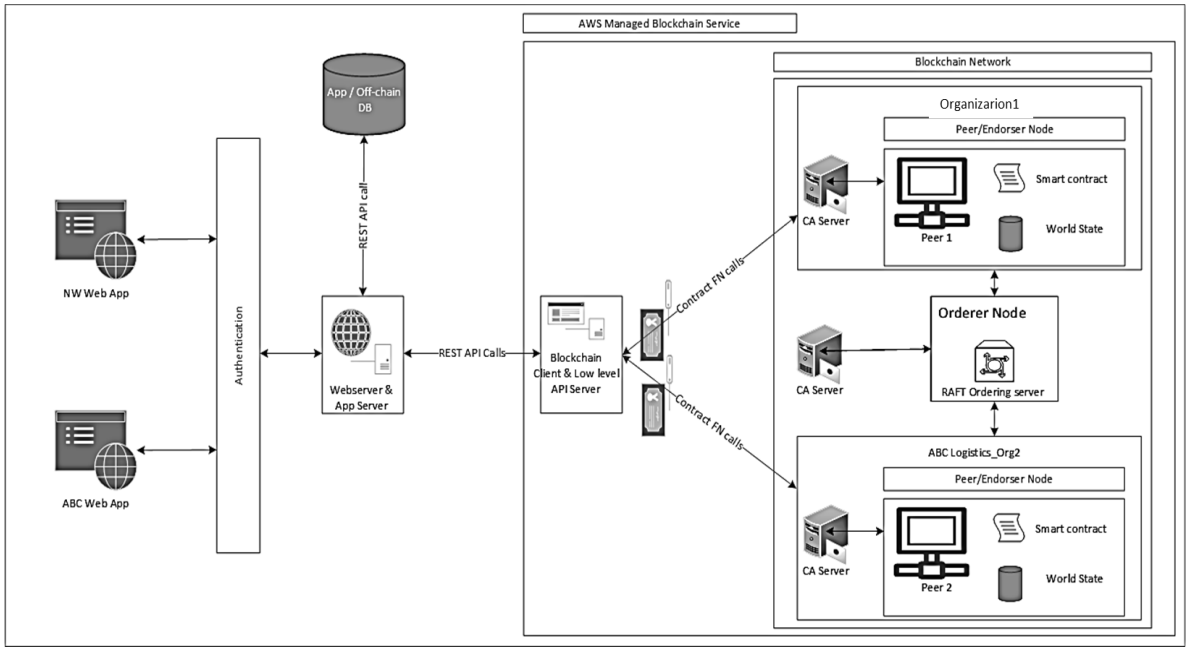


Figure 7. 7 Deployment Architecture in AWS Platform

The implemented solution uses a three-tier architecture. The Figure 8 below explains each of the tech stacks used in this of Concept.

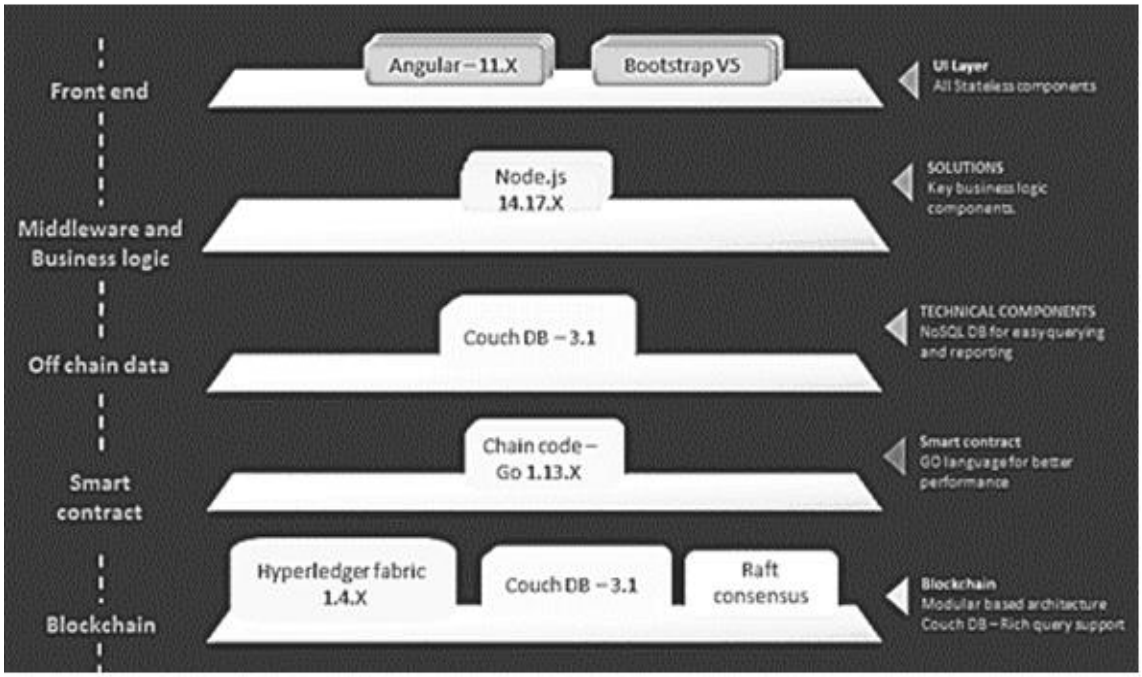


Figure 7. 8 Application technology stack

D. User Interface (DApp) and Information Flow Implementation

Figure 7. 9 Dashboard user Interface. The general snapshots of trip activity, completed journeys, trips in transit, and payments made are shown on the dashboard

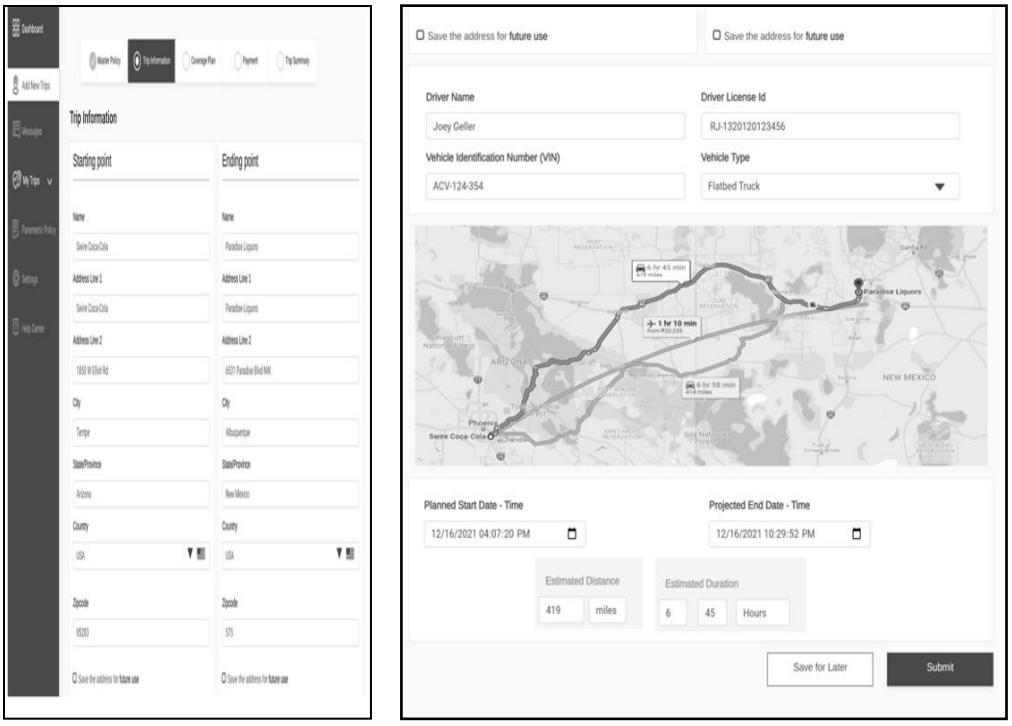


Figure 7. 10 Trip Information with routing info. The interface records the trip details, geo locations, and travel routes for each journey

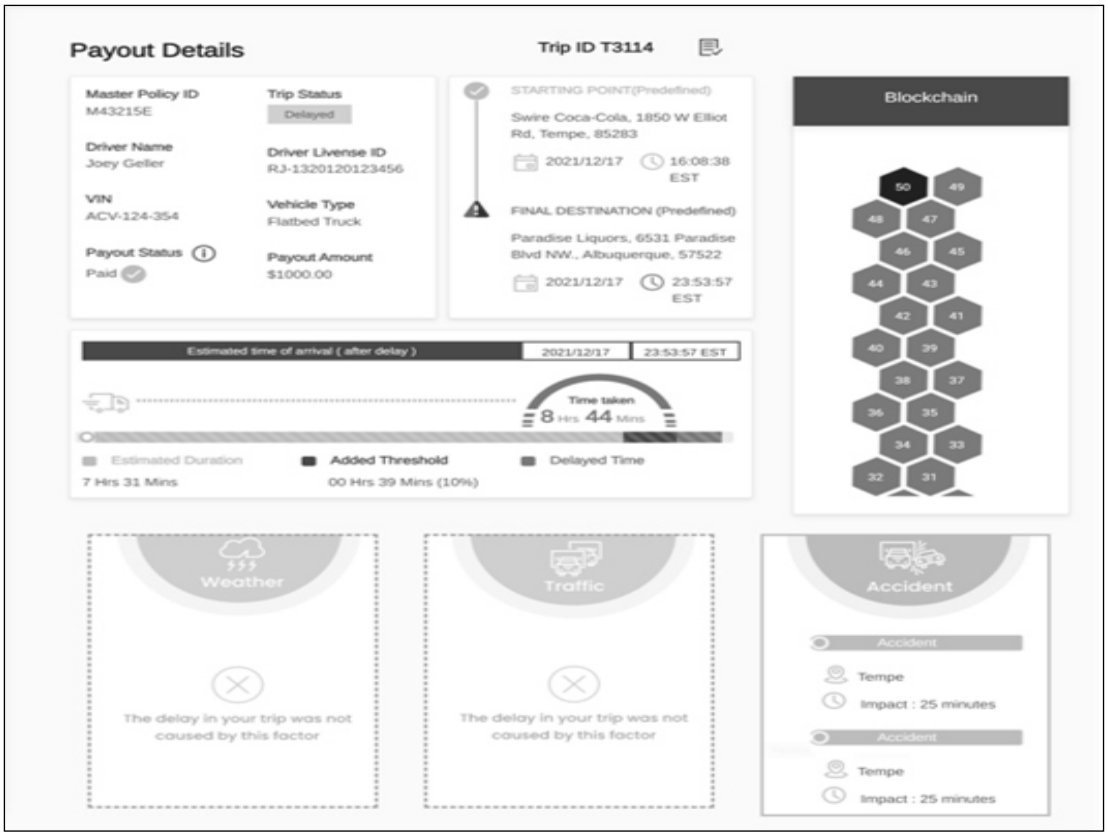


Figure 7. 11 Event Tracking and Claim Processing. Events are monitored throughout the trip in accordance with the policy guidelines for that particular trip. The system will immediately process the claim for the insured if the rule is broken

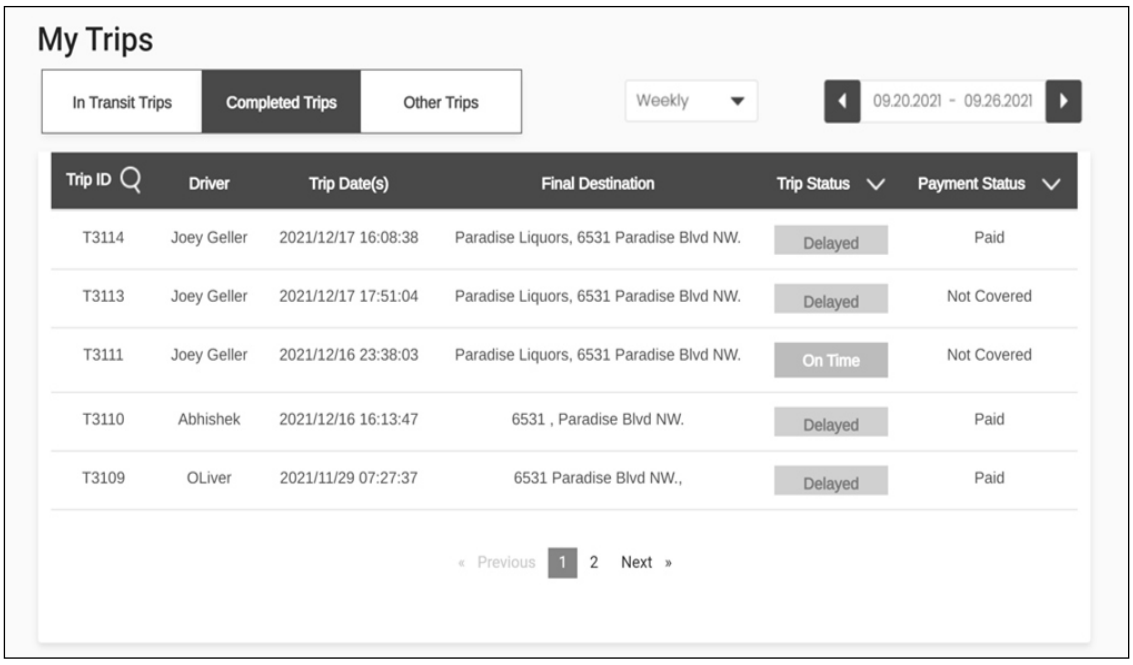


Figure 7. 12 Trip summary details. The list of trips that has been completed with status

7.3.4 Illustration of the Proposed Concept

Upon the completion of a trip, the system application diligently logs critical information, encompassing the trip's initiation, conclusion, and any events that transpired throughout the policy term. External third-party services contribute event data to the application database, which is subsequently fed into the smart contract. The smart contract, employing parameters such as journey start and finish times, registered events, and a predefined buffer period, meticulously validates the received data. If a legitimate travel delay is identified, the event is documented, and the initiation of claim generation is signalled within the blockchain. Following this, the insurer commences the settlement process, adhering to the stipulated terms of the policy, and disburses the agreed-upon amount to the insured offline.

In the context of the proof-of-concept test execution, approximately 300 journeys are concurrently initiated to ensure the precision and reliability of the test results. This multi-journey approach is designed to validate the system under realistic conditions. Each trip's replication involves the random selection of occurrences from a pool, ensuring a diverse range of scenarios. Events are uniformly dispersed across approximately 80% of the simulated route, introducing elements of unpredictability and challenges. The remaining 20% of journeys are deliberately designated as "normal," signifying a lack of obstacles, enabling on-time completion without complications.

The outcome of the simulation aligns consistently with the anticipated results, affirming the effectiveness of the parametric insurance solution under varied conditions. The compilation of test cases, coupled with the mapping of test scenarios against Table 7.3 (outlining payout rule execution based on test scenarios), facilitates a thorough examination and comparison with the expected outcomes. This meticulous testing methodology ensures that the parametric insurance system performs as envisioned, offering reliable and efficient claim processing in real-world scenarios.

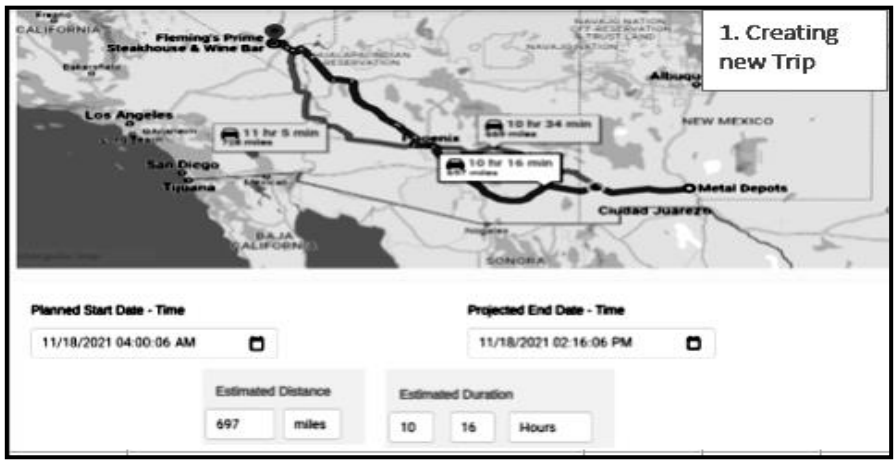
Scenario	Estimated Duration	Projected End Time	Actual Arrival Time (with Buffer)	Delay	Route Deviation	External Events			Payout Status
						Weather	Traffic	Accident	
Vehicle reached on or before the scheduled time at destination	10hrs	11hrs	10:50hrs	No	NA	NA	NA	NA	Not Eligible
Due to unknown reason vehicle took a different route and was NOT able to reach destination within defined time for the trip	10hrs	11hrs	11:50hrs	Yes	No	No	No	No	Not Eligible
Due to unknown reason vehicle took a different route and was NOT able to reach destination within defined time for the trip	10hrs	11hrs	11:50hrs	Yes	Yes	No	No	No	Not Eligible
There was accident enroute due to which vehicle took a different route and was NOT able to reach destination within defined time for the trip	10hrs	11hrs	11:50hrs	Yes	Yes	Yes	No	Yes	Not Eligible
There was traffic enroute due to accident and vehicle took a different route and was NOT able to reach destination within defined time for the trip	10hrs	11hrs	11:50hrs	Yes	No	No	Yes	Yes	Eligible

There was heavy rain enroute due vehicle was NOT able to reach destination within defined time for the trip	10hrs	11hrs	11:50hrs	Yes	No	Yes	No	No	Eligible

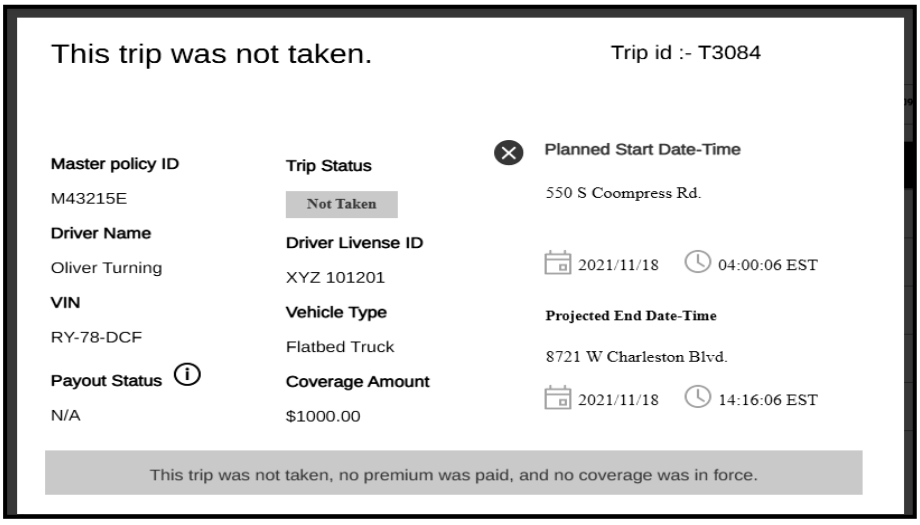
Table 7. 3 Parametric payout rule execution based on test scenarios

Below are the execution steps with user interfaces for Scenario #1

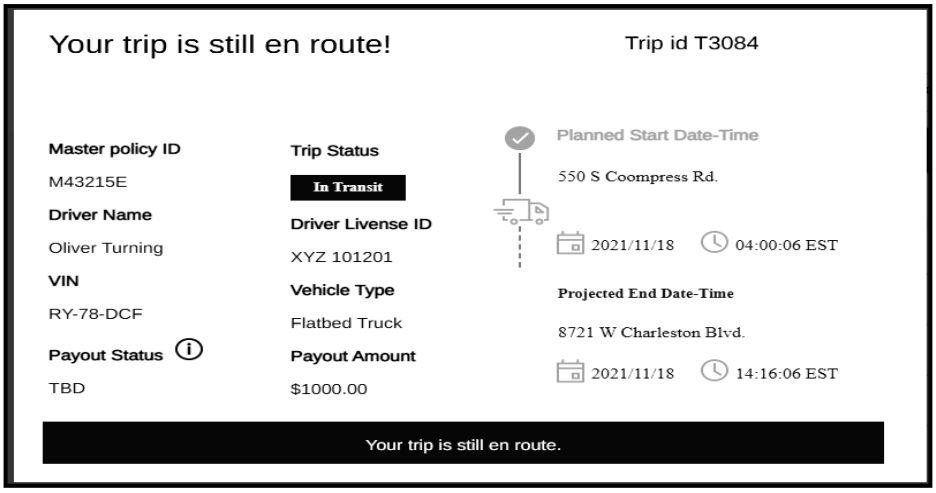
Step 1: A new trip is created in the system.



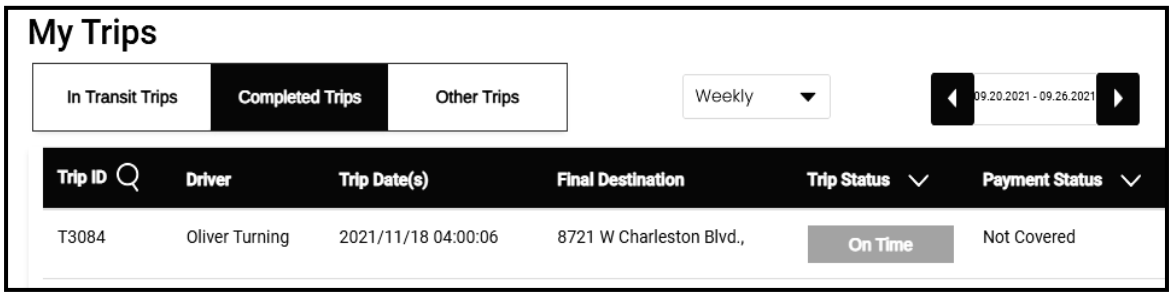
Step 2: The trip is created, premium is not paid and coverage is not taken yet.



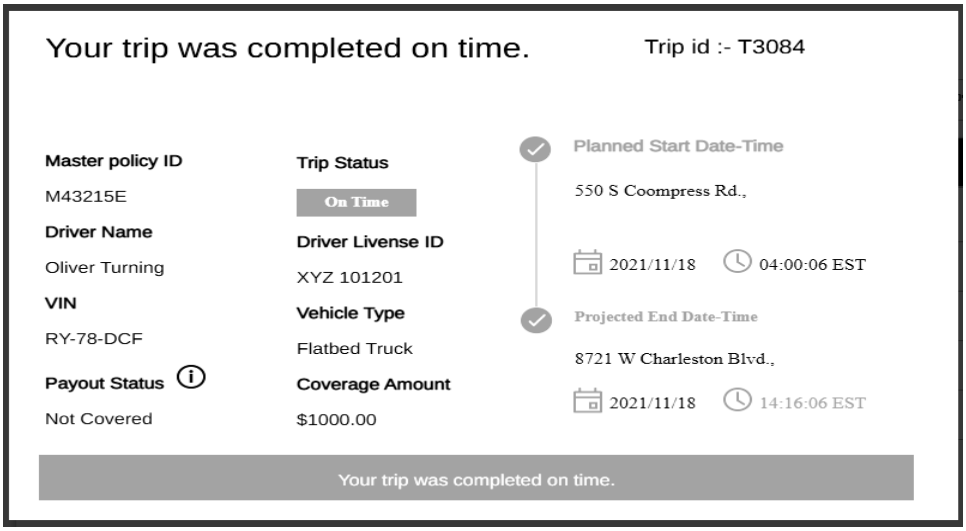
Step 3: Required premium has paid, trip is in en-route, coverage has started.



Step 4: Trip Ended with tripStatus ="OnTime" because trip has ended before scheduled time.



Step 5: Trip Summary, after the trip ends



7.4 Industrial Use Case: Sustainable Risk Modelling of Realtime Quick Service Restaurant (QSR) Business Interruption Losses

Parametric insurance contracts rely on easily determined parameters (e.g., wind speed, weather data, seismic factors) and are used for disaster risks. They offer quicker and autonomous claims processing compared to traditional insurance with complex claim procedures. The insurance industry faces pressure from technological advances, changing consumer expectations, and new business models. Various aspects of the insurance industry's business model and growth strategies are evolving because of the development of digital technologies and fresher market entry chances.

- Coverage is expanding to include risks not previously insured, such as non-catastrophic risks, intangible assets, and novel hazards, in addition to typical catastrophe risks.
- Along with governments and risk aggregators, customer categories are now also including corporations, small and medium-sized organizations, and individuals.
- Products are increasingly being positioned as either a stand-alone, one-of-a-kind solution in cases where there is no existing product or as an integrated part of existing products to fill in any coverage gaps rather than as a less priced alternative to existing products.
- Weather-related parametric triggers were once a common occurrence, but they are now more long-term, unconnected to the weather, and generated in clever ways.
- Weather-related parametric triggers were once a common occurrence, but they are now more long-term, unconnected to the weather, and generated in clever ways.
- Insurance should enhance business sustainability by providing swift and transparent payouts based on predefined triggers, enabling faster recovery from unforeseen events, and reducing operational downtime, thus bolstering resilience and

continuity. This approach aligns with sustainable practices by minimizing financial disruptions and promoting proactive risk management strategies.

Insurance companies are undergoing a comprehensive reassessment and expansion of their parametric product portfolios, introducing innovative policies designed to mitigate risks that fall outside the purview of conventional insurance coverage. The scope of parametric solutions has broadened significantly to encompass a diverse array of risks, ranging from natural disasters and adverse weather conditions to indirect losses that impact businesses. These dynamic solutions now extend their protective umbrella to include non-weather-related hazards such as political instability, supply chain disruptions, cyberattacks, and reputation damage. Furthermore, they play a crucial role in safeguarding against financial losses stemming from a spectrum of events, including but not limited to threat alerts, nearby explosions, or pandemics. In the context of the food and quick-service restaurant supply chain, where the seamless transfer of cargo is a mission-critical operation, traditional cargo insurance traditionally addresses unforeseen events like fires, accidents, theft, and natural disasters. However, it falls short in addressing the intricate web of financial damages that can ripple through the supply chain. Even when a fraction of the risk is sustained at any given point, the existing paradigm necessitates separate insurance coverage. For instance, while fleet insurance may adequately cover vehicle repair costs arising from accidents, it fails to provide compensation for damages incurred due to delivery delays. The responsibility for such losses typically falls on various stakeholders within the supply chain, and these losses often go unreimbursed by insurance without stringent proof of occurrence. This paradigm shift in insurance strategies reflects a nuanced understanding of the multifaceted risks faced by businesses and supply chains, prompting the industry to adopt parametric solutions that offer comprehensive and responsive coverage in the face of evolving challenges.

In response to the unprecedented challenges posed by the COVID-19 pandemic, Quick Service Restaurants (QSRs) are undergoing substantial modifications to adapt their services. The intricacies of the supply chain for QSR services have been significantly influenced by shortages of food grains and other essential items. Simultaneously, QSR franchises are strategically collaborating with food aggregators and delivery services to boost sales demand in the aftermath of the pandemic. In this dynamically shifting landscape, smaller food chains and QSRs are compelled to offset losses resulting from

sluggish demand. These losses are multifaceted, encompassing challenges such as diminished foot traffic and supply chain disruptions triggered by adverse weather conditions and external factors.

Consider QuickBite, a hypothetical Quick Service Restaurant owned by Ms. White, situated in a tourist hotspot on the outskirts of the city. Relying heavily on tourist sales and offering home delivery through IT-enabled services, QuickBite operates in a market that was valued at $10.3 billion in 2018, with a projected growth of 8.2% CAGR to reach $16.6 billion by 2024 (TheExpressWire, April 2022). Confronting sales fluctuations, especially in the post-COVID-19 era, Ms. White's conventional policy provides coverage for natural disasters but grapples with the intricacies of supply chain instability. This predicament has given rise to challenges such as abrupt declines in foot traffic, escalated production costs, and delayed deliveries of perishable goods, culminating in substantial business losses and brand damage. Considering these pressing issues, Ms. White is actively seeking financial security for QuickBite to fortify her restaurant against the uncertainties of the evolving business landscape.

The standard protocol for initiating an insurance claim for cargo is depicted in Figure 1. In the event of any physical damage to the insured fleet, such as an accident, the policyholder is required to initiate a claim with the insurance provider. After the completion of both physical and digital verification processes, the claim is officially registered, and all supporting documentation is transmitted in both physical and digital formats. Prior to authorizing a claim payout, the insurance company's operations team undertakes a comprehensive series of checks and balances. If the requisite verifications are successfully executed, the claim distribution process is initiated in accordance with the terms and conditions stipulated in the insurance policy.

Outlined succinctly below are the predominant operational challenges observed in the contemporary cargo insurance sector:

- Each insurance company centrally manages customer databases and information related to claims processing.
- The reliance on intricate and opaque procedures.
- Cargo insurance providers are introducing innovative revenue-generation strategies.

- Due to a lack of information transparency, brand recognition and the speed of claims settlement emerge as pivotal indicators of trust.
- The imperative need for technology development to support contemporary business models.

My strategy for devising a solution, revolves around leveraging smart contracts to encompass specific occurrences (e.g., traffic, accidents, weather, or low footfall) that can automatically trigger a claim payment based on indices derived from reliable sources. This methodology ensures coverage for various specific incidents, including those stemming from traffic issues, accidents, adverse weather conditions, or low footfall. The solution has been intricately constructed through the implementation of smart contracts, which are digital contracts built on blockchain technology and come with predefined constraints governing their execution. Smart contracts, serving as the digital backbone, automatically facilitate claim compensations when predetermined events occur, guided by the underlying logic embedded in the smart contract. The adoption of this solution yields a plethora of notable economic and technical advantages:

- Paperwork is eradicated through the utilization of blockchain for policy storage.
- Human intervention in the claim's processing workflow is eliminated.
- Intermediaries are removed from the process, streamlining efficiency.
- Instantaneous payouts become feasible for covered events, enhancing overall responsiveness.
- Administration and claims processing costs witness a substantial reduction.
- A revamped revenue model is introduced, proving beneficial for businesses.
- Mutual benefits accrue for both insurers and organizations involved.
- The model operates on a trust less and highly decentralized foundation.
- Advanced digital technology is harnessed for seamless implementation.
- Paramount emphasis is placed on information security and transparency.

7.4.1 Strategic Solution Topology and System Design

Insurance companies can revolutionize their operations, elevate coverage transparency, and accelerate the claim settlement process through innovative methodologies. One such groundbreaking approach is the incorporation of parametric insurance, a strategy that harnesses advanced digital technologies to usher in transformative changes to the industry. Operating on an index, such as average footfall, parametric insurance gauges the likelihood of trigger events (e.g., malfunctions, accidents, weather changes, or pandemic outbreaks) and provides predetermined compensation proportional to the severity of the event. The foundational principles guiding the design of the parametric platform solution are elucidated below:

- The solution's architecture should prioritize scalability, allowing seamless integration of new use cases, such as incorporating additional events or indices or onboarding new participants in the supply chain.
- Flexibility in the platform and solution is essential to accommodate diverse parametric use cases, ensuring adaptability to the unique requirements of different scenarios.
- The paramount focus should be on providing robust loss protection for insured parties, encouraging the utilization of risk management tools and preventive measures to mitigate potential losses.
- The requirements for parametric insurance coverage must be communicated in clear and straightforward language, devoid of legalese or jargon that might pose challenges for the average individual to comprehend.
- Real-time assessment of risk based on index components should be a continuous process to maximize return on investment. Claim review and processing should be fully automated upon satisfaction of policy requirements, minimizing the time taken for claims settlement.
- The solution should maintain simplicity, ensuring clear product and coverage requirements for straightforward risk pricing and efficient operations management. It should comprehensively address overlooked and underfunded markets while considering the diverse risks associated with involved stakeholders, including insurance companies, aggregators, policyholders, and technology service providers.

This experiment utilizes a Parametric food carrier and QSR supply chain insurance solution, automating claims processing based on external data sources and creating an immutable audit trail for QSR businesses. It demonstrates a real industrial use case for smart contracts in parametric insurance, focusing on potential supply chain losses for small quick-service restaurants due to decreased foot traffic during extreme weather. When transitioning this proof of concept into a production-ready service, it is important to consider the following:

- Include QSRs offering customer insurance.
- Allow large QSR chains to run blockchain nodes optionally.
- Ensure customer confidentiality.
- Let customers choose parameters, thresholds, and payments within limits.
- Support specific and time-based parameter values.
- Enable digital authentication for configuring policy triggers and parameters.
- Authenticate users upon event triggers for confirmation.

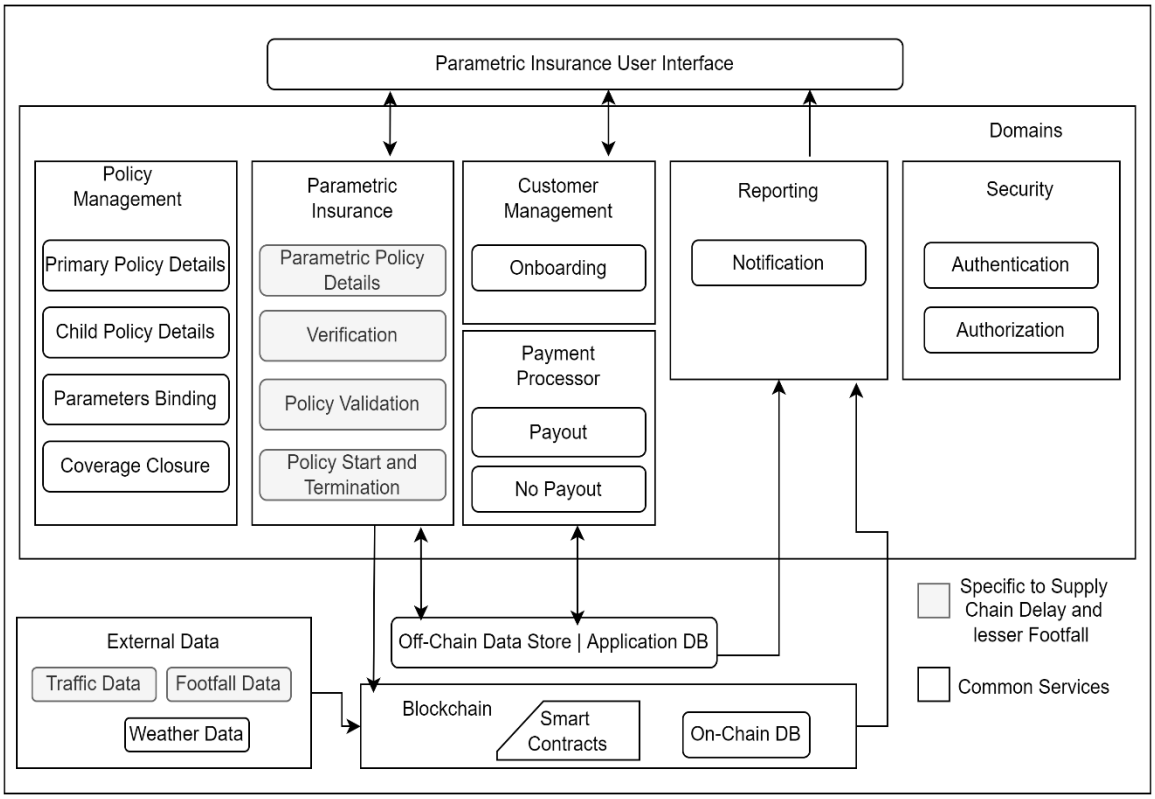


Figure 7. 13 Parametric insurance solution logical architecture

Within Figure 7.13, I unveil a comprehensive and intricately designed three-layered logical architecture that envelops all essential domain components of the parametric insurance solution. At the zenith of this architectural framework is the top layer, housing a role-based integrated application meticulously crafted for process control, thereby ensuring airtight control over authorized access to the system. The subsequent domain layer, positioned just beneath the top tier, serves as the repository for an array of business capabilities thoughtfully extended to users. As I delve deeper, the service layer emerges as the operational nucleus, orchestrating the management of critical elements such as claims, payments, customers, and coverage through a sophisticated interplay of modular components. The bedrock of this architectural marvel lies in the utilization of reliable external and third-party data sources, encompassing decentralized Oracle networks, data from esteemed providers like TomTom, AccuWeather, and Google Maps, along with pertinent tourist statistics from government sources tailored to specific locations. During off-chain processes, providers trigger smart contracts at strategic junctures, facilitating seamless execution. Meanwhile, the permissioned blockchain layer assumes a pivotal role by leveraging smart contracts for data authentication during the incorporation of information into the blockchain, thereby fortifying the system's integrity and ensuring a secure and immutable ledger. This meticulously structured logical architecture not only encapsulates the entirety of the parametric insurance solution but also underscores its resilience, efficiency, and adaptability in handling diverse operational aspects.

7.4.2 Implementation

A. Application Architecture and Solution Steps

The Hyperledger Fabric blockchain technology, as elucidated in Figure 7.14, presents a distinctive and effective solution, setting it apart from other blockchain systems by functioning as an open-source, permission-enabled, and immutable distributed ledger platform tailored for corporate applications. Renowned for its emphasis on privacy and scalability, Hyperledger Fabric introduces a structured ecosystem comprising participating entities like Organizations and network participants, a shared ledger, chain code applications, and the ordering service node, all orchestrated through channels.

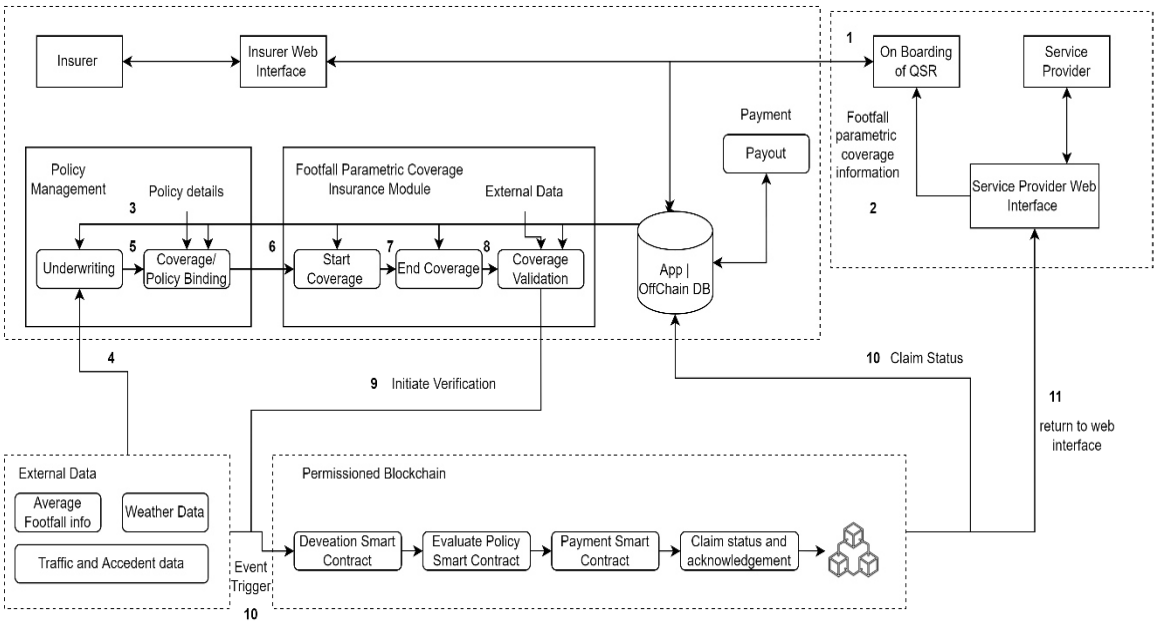


Figure 7. 14 QSR parametric application architecture

Transactions within the network are confined to specific channels, mandating authentication, and permissions for involved parties. In this intricate framework, each channel peer obtains identity credentials from a Membership Service Provider (MSP) for secure authentication within the channel. A distinctive aspect of Hyperledger Fabric lies in its permissioned nature, ensuring that participants are not anonymous, and certificates are issued to network user peers and relevant web applications. These certificates enable seamless interactions with blockchain smart contracts, also known as chain code, through REST APIs. Smart contracts play a pivotal role in furnishing essential data, ranging from onboarding information to policy details, thresholds, and footfall data, facilitating web applications' functionality. Through these contracts, web applications can query blockchain data for payout determination and claim status. Synchronization between organizational application databases, termed off-chain databases, and blockchain storage guarantees consistent data copies, facilitating real-time data retrieval from diverse Quick Service Restaurant (QSR) enterprises and nearly instantaneous data updates within the organization. Simultaneously, simulated footfall data and site-related parameters are transmitted to the blockchain throughout the policy period, enabling smart contracts to identify deviations. Upon detecting deviations, the smart contract scrutinizes reliable data sources to discern the cause, such as a pandemic, severe weather conditions, or roadblocks. Legitimate explanations for decreased foot

traffic based on external events prompt the smart contract to set the payout status to True and the claim state to Approved. Contrarily, if the deviation lacks justifiable grounds, these attributes are marked as false and subsequently rejected. For an exhaustive understanding of the validation process, please refer to the accompanying state change diagram and pseudo code provided below.

Assumptions – Reliable statistics on the average number of tourists per location are available. This can be checked using an official digital system that tracks the number of tourists who have been granted authorization to travel through a particular location.

Footfall Parametric Coverage –

Within policy Date Range

if foottraffic < average touristtraffic in event then

if not an event, then // Lesser foot traffic due to other reason

validate event Type Code and terms parameters.

if validation == true and eventTypeCode is within the list of matching eventType

Set Payment Status = "Paid"

else

Set Payment Status = "Not Covered"

else

Set Payment Status = "Not Covered"

else

Set Payment Status = "Not Covered"

Set Coverage Status as "Coverage End"

Supply Delay Coverage –

Assumptions – Reliable third-party sources are available for data on roadblocks, accidents, and traffic in close to real time.

if SupplyDelay >= delay timeframe in events then

if not path deviation, then.

validate event Type Code and terms parameters.

if validation == true and eventTypeCode is within the list of matching eventType

Set Payment Status = "Paid"

else

Set Payment Status = "Not Covered"

else

Set Payment Status = "Not Covered"

else

Set Payment Status = "Not Covered"

Set Coverage Status as "Coverage End"

if supplyDelay == "Cancelled" then // Raw material supply is cancelled

set Payment = "NA" and

Coverage Status = "Coverage End"

else

supply deviation = 10% of Projected supply end Time

supply delay = actual supply endtime - supply deviation

If supply delay > 0 then

Set Payment Status = "In Review"

The flowchart below shows the process.

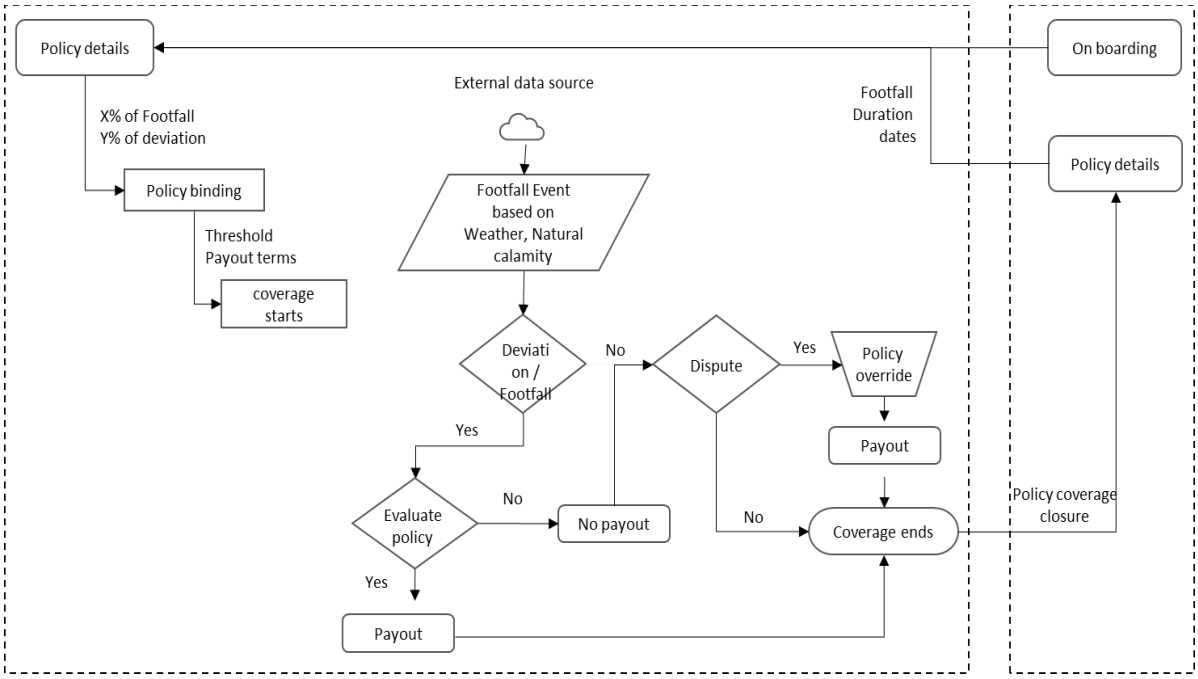


Figure 7. 15 Policy coverage flow diagram

The permitted set of events for the proof of concept, which is simulated using third-party data providers, is listed below.

Allowed Categories	0. Unknown 1. Accident 2. Fog 3. Dangerous Conditions 4. Rain 5. Ice 6. am 7. Lane Closed 8. Road Closed 9.Road Works 10. Wind 11. Flooding 12. Broken Down Vehicle 13. Average tourist footfall

The following are tables of payout rules based on different execution scenarios. These tables are used by smart contracts while making payment decisions.

Tourist Footfall Deviation - Scenario	Estimated Start Date Time	Projected End Date Time	Actual Arrival Date Time (with Buffer)	Footfall Deviation	External Events			Payout Status
					Weather	Roadblock | Pandemic | Accident	High Tourist Footfall	
There are more visitors than usual at a certain site.	DD:MM:YYYY HH:MM	DD:MM:YYYY HH:MM	DD:MM:YYYY HH:MM	No	Yes | No | NA	Yes | No | NA	Yes	Not Eligible
Due to an unforeseen factor, tourism declines	DD:MM:YYYY HH:MM	DD:MM:YYYY HH:MM	DD:MM:YYYY HH:MM	Yes	No	No	No	Not Eligible
Due to a known factor, there are fewer visitors, however this is not covered by the conditions of the policy.	DD:MM:YYYY HH:MM	DD:MM:YYYY HH:MM	DD:MM:YYYY HH:MM	Yes	No	No	No	Not Eligible
Due to known weather-related issues, there are less tourists.	DD:MM:YYYY HH:MM	DD:MM:YYYY HH:MM	DD:MM:YYYY HH:MM	Yes	Yes	No | NA	No	Eligible
Natural disasters (roadblocks) and/or accidents	DD:MM:YYYY HH:MM	DD:MM:YYYY HH:MM	DD:MM:YYYY HH:MM	Yes	Yes | No | NA	Yes	No	Eligible

cause a long-term decline in tourist foot traffic.								
Travel restrictions are in place due of pandemic limitations.	DD:MM:YYYY HH:MM	DD:MM:YYYY HH:MM	DD:MM:YYYY HH:MM	Yes	Yes | No| NA	Yes	No	Eligible

Supply Delay - Scenario	Estimated Duration	Projected End Time	Actual Arrival (with Buffer)	Supply Delay	Route Deviation	External Events			Payout Status
						Weather	Traffic	Accident	
Fleet arrived at the location at or before the specified time.	HH	HH	HH:MM	No	Yes | No | NA	Yes | No | NA	Yes | No | NA	Yes | No | NA	Not Eligible
Due to an unforeseen circumstance, the vehicle took a different route and was unable to arrive at the destination within the allotted travel time.	HH	HH	HH:MM	Yes	Yes | No | NA	Yes | No | NA	Yes | No | NA	Yes | No | NA	Not Eligible
Fleet took an alternative path for a known cause, not arriving at the location in the allotted amount of time.	HH	HH	HH:MM	Yes	Yes | No	Yes | No	Yes | No	Yes | No	Eligible
Due to an accident in route, the fleet took a different route and was unable to arrive at the destination within the allotted amount of time.	HH	HH	HH:MM	Yes	Yes | No | NA	Yes | No | NA	Yes | No | NA	Yes	Eligible
Due to congestion caused by the accident, the fleet took a different route and was unable to	HH	HH	HH:MM	Yes	Yes | No | NA	Yes | No | NA	Yes	Yes | No | NA	Eligible

arrive at the destination in the allotted amount of time.									
Fleet was unable to arrive at destination in the allotted amount of time because of heavy rain on the road.	HH	HH	HH:MM	Yes	Yes | No | NA	Yes	Yes | No | NA	Yes | No | NA	Eligible

Table 7. 4 Rule execution tables

Following the evaluation of the claim status and subsequent notification to both the insurance provider and stakeholders, the smart contract effectively wraps up the entire policy life cycle. Stakeholders are afforded access to comprehensive information encompassing claims and coverage details, inclusive of reports delineating issues that demand attention and resolution. This accessibility is facilitated through a variety of Application Programming Interfaces (APIs), ensuring seamless communication and data retrieval. Crucially, all transactional event data is securely stored on the blockchain, contributing to the platform's transparency and immutability. This meticulous recording of events not only serves as an auditable record but also bolsters the reliability and accountability of the insurance processes.

The implementation of the blockchain is facilitated through Hyperledger Fabric. When a quick-service restaurant seeks a parametric insurance policy from an insurance provider, the process commences with a transaction request initiated through the insurance portal client interface. The client application transmits this request to the endorsement peer. Endorser peers meticulously authenticate the authority and transaction particulars, subsequently executing the chain code, and furnish responses indicating approval or rejection. Transactions that receive approval are forwarded to the ordering service by the client application. The ordering service's peer then disseminates the transaction to the nodes of network participants, triggering the update of their respective local ledgers and culminating in the finalization of the new transactions. This decentralized and permissioned blockchain structure ensures secure, transparent, and efficient handling of parametric insurance processes for quick-service restaurants.

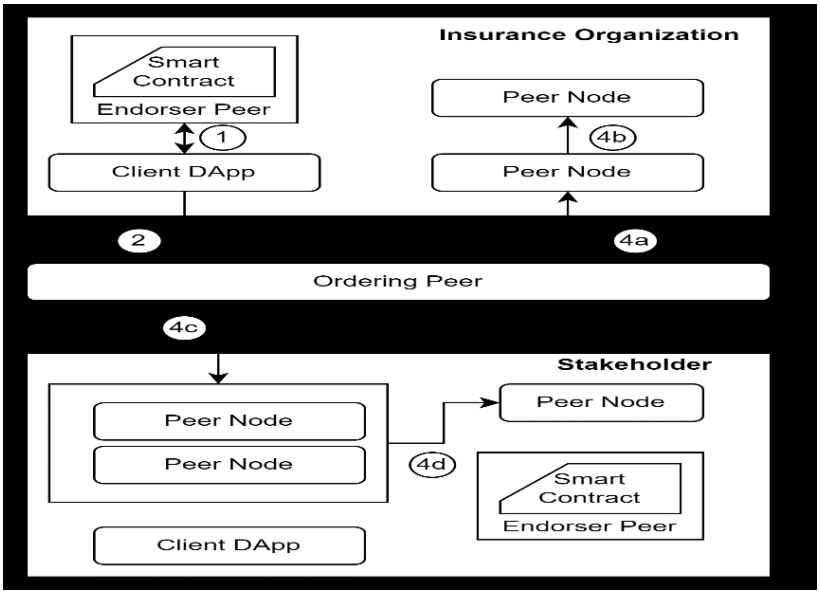


Figure 7. 16 Blockchain process flow

B. Smart Contact (Chain Code) and Application Programming Interface Implementation

In the implementation of supply delay scenarios, a range of smart contract operations are executed to cover key aspects. These operations encompass adding trip information, dynamically updating trip details according to event parameters, initiating trips, ensuring compliance validation, capturing external events, confirming coverage terms, retrieving claim and payout status, and conducting audits on transaction data. To facilitate these functionalities, a suite of APIs has been developed. These APIs support various operations such as providing key performance indicators (KPIs) for trip statuses, allowing the addition and transmission of travel information to both off-chain and blockchain databases, presenting premium data, summarizing trips, managing trip groups, and facilitating updates for trip end details and delays. In the context of tourist footfall parametric coverage, the smart contract operations include the management of footfall coverage details, dynamic updates to parameters, acquisition of external footfall data, validation of coverage terms, and retrieval of claim and payout status based on predefined rules. Dedicated APIs for footfall coverage play a crucial role, enabling the creation of policies, presenting status indicators, and empowering users to view premiums based on selected parameters. This comprehensive suite of smart contract

operations and APIs ensures a robust and efficient implementation of parametric coverage for both supply delay and tourist footfall scenarios.

C. Technology stack and Infrastructure used

For the proof-of-concept implementation, a single-node, single-peer blockchain network is employed, with the node serving the role of the insurance provider. The proposed methodology hinges on the utilization of a decentralized application (DApp) that leverages the Hyperledger Fabric Blockchain platform as its underlying blockchain protocol. The Proof-of-Concept (PoC) solution is intricately configured with the AWS Managed Hyperledger Fabric Blockchain Platform, featuring a singular channel, a solitary organization with one peer, a lone ordering service, and an off-chain database utilizing key database platforms. The services and technologies incorporated in this experimental setup encompass Linux VM and AWS BaaS (Blockchain as a Service), local storage, Fabric CA (Certificate Authority) for creating a certification infrastructure, CouchDB for the application database, and Docker for containerization. This comprehensive setup ensures the robust execution of the proof of concept, providing a tangible demonstration of the proposed approach within the specified blockchain framework.

7.4.3 Demonstration of the Proposed Concept

Conducting the proof-of-concept test involves the simultaneous simulation of approximately 100 fleet trips, specifically focusing on supply delay scenarios. The execution randomly selects scenarios from a meticulously prepared pool, encompassing 80% of trips with various events and 20% representing standard routes devoid of impediments. For QSR footfall coverage, the simulation spans around fifty scenarios, reflecting typical tourist numbers for the designated location. Each iteration of execution selects a scenario within the specified scope, enabling a thorough validation of the parametric insurance solution architecture, smart contract execution, and blockchain transactions. It is crucial to underscore that the proof of concept intentionally omits performance evaluations or benchmarking. The outcomes of the simulation align seamlessly with the anticipated results, effectively cross-referencing

test cases and scenarios against pre-established rule tables and predefined payout scenarios.

Coverage	Number of Execution	Hit	No Hit	Simulation Execution Time (Avg)
Supply delay coverage	100	80	20	<= 5 sec
QSR footfall delay coverage	50	40	10	<= 3 sec

Table 7. 5 Test execution results

7.5 Conclusion

Parametric solutions represent a distinctive form of insurance that hinges on mitigating the risk associated with a predefined event occurring. In contrast to compensating actual losses incurred, this innovative model centres around proactively covering the risk of the event itself. Operating as a catalyst for growth in the fiercely competitive insurance landscape, parametric insurance creates a mutually beneficial scenario for both insurers and the insured. With a primary focus on swift payout and addressing losses challenging to model, the parametric solution confronts issues inherent in centralized claim processing systems, intricacies of existing processes, and losses within supply chains that are challenging to quantify. Each study delves into a transport parametric use case and Quick Service Restaurant (QSR) parametric use case aiming to construct and execute a comprehensive platform solution rooted in decentralized immutable distributed ledger technology. This encompasses system services, APIs, event triggering, claim initiation, and near real-time auto claim processing, establishing an immutable audit trail for logistics businesses and organizations. The solution seamlessly processes claims based on external data sources, showcasing a low-level system design, technological integrations, and state modifications across various trip delay criteria. The end-to-end application incorporates a sequence of user interfaces managing trips, claim processing, and payouts. The existing architecture, blockchain network, and infrastructure hold the potential to facilitate network expansion by integrating diverse logistical providers with minimal modifications.

7.6 Industrial Use Case: Harmonizing Efficiency and Regulation in Decentralized Capital Markets

Delays within existing capital markets are largely due to the redundant processes inherent in multi-tiered record-keeping systems and the lack of transparency in asset registries maintained by Central Securities Depositories (CSDs). Blockchain technology addresses these inefficiencies by enabling decentralization, which allows participants to directly interact, verify, and exchange value without relying on intermediaries. One of the key features of blockchain is its ability to digitize credentials into decentralized identities. These digital credentials can be used to authenticate counterparties in transactions within decentralized marketplaces. By handling tokenized assets and verifying counterparties using these credentials, blockchain eliminates the need for traditional intermediaries, such as brokers and clearinghouses. This streamlining effect speeds up transaction execution by ensuring asset ownership is verified prior to trading, reducing the time and complexity associated with trade settlements. Decentralized asset registries on blockchain platforms enhance visibility into asset ownership and status, which helps eliminate the duplication of transaction recording. This improvement in transparency and data accuracy significantly boosts market efficiency by reducing the time and resources spent on reconciliation processes. However, despite these advantages, certain roles within current market structures remain essential. Custodians and exchanges play critical roles in trade matching, asset safekeeping, and ensuring compliance with regulations related to anti-money laundering (AML), sanctions, and fraud prevention.

In this implemented industrial solution, I propose a novel approach to integrate the functions of exchanges and custodians into decentralized capital markets. This integration aims to maintain the necessary compliance with regulatory requirements while leveraging the efficiency benefits of blockchain technology. By incorporating these roles into a decentralized framework, market efficiency can be enhanced, asset security can be ensured, and regulatory compliance can be maintained. This approach

balances the benefits of disintermediation with the need for oversight and safe handling of assets, thereby offering a more streamlined and efficient capital market structure.

7.6.1 Decentralized Finance (DeFi) Security Token Exchange: Current Processes, Inefficiencies, and Challenges

Decentralization empowers individuals by giving them direct control over their assets and information, eliminating the need for intermediaries. Blockchain technology, though still in its nascent stages, is foundational in establishing public trust. It digitizes credentials into decentralized identities, and efforts are underway to standardize these identities on a global scale. Blockchain's tamper-resistant ledger prevents double-spending and removes the necessity for centralized databases, offering a decentralized solution for transacting digital assets and verifying counterparties. Decentralized finance (DeFi) [121] provides an alternative to traditional financial systems, enhancing access and diversity in financial markets. Blockchain not only represents assets but also establishes ownership, enabling secure exchanges and streamlined verification processes. It reduces counterparty risk by allowing direct asset interaction and verification, fostering disintermediation. In financial markets, Primary Markets oversee the issuance of new securities, while Secondary Markets handle the trading of existing securities like stocks and bonds. Secondary Markets manage a larger volume of publicly traded securities and require robust regulation. Established entities such as Security Trading Organizations and Custodial Service Providers play crucial roles in ensuring secure trading within Secondary Markets. This ecosystem comprises various participants, including:

- Investors (Institutional/Brokers): Engage in buying or selling securities by placing orders.
- Trading Organizations (Traders/Market Makers): Match buy and sell orders to execute trades.
- Asset Custodians/Sub-custodians: Protect securities, settle transactions, and provide administrative support.
- Financial Market Utilities (FMUs): Includes CSDs, ICSDs, Clearing Houses, and Central Banks that handle settlements and other essential market functions.

Investors place orders, which Trading Organizations match to execute trades. Trade parameters are reconciled between counterparties to ensure agreement, with settlements handled through FMUs. Custodians, traditionally safeguarding paper securities, have adapted to digitization with the rise of CSDs as electronic repositories. They manage accounts directly or through sub-custodians across jurisdictions [122], facilitating trade settlements on behalf of clients. Tokenization of securities on Blockchain eliminates the need for traditional book-entry records maintained by CSDs, as all trades are settled through Blockchain transactions. Collaboration among ecosystem participants, such as investment banks and private companies, facilitates the issuance of decentralized securities via Blockchain-based marketplaces. This consortium ensures safety and trust by verifying issuers and investors with digitally verifiable credentials. Blockchain maintains the integrity of value exchanges, supporting secure clearing and settlement. Secondary markets must adapt to regulatory and custody complexities, necessitating adjustments in roles like security registries, custodians, and exchanges. Despite these changes, decentralization provides equal access to security holdings information, with custodians conducting additional compliance checks before settlement [123].

Challenges in Decentralizing Secondary Markets -

- Navigating Regulatory Complexities in Transitioning to Decentralized Custody Chains -Transitioning to decentralized custody chains involves navigating a complex landscape of regulatory requirements. Different jurisdictions have varied regulations, and ensuring compliance across these regions can be daunting. The lack of standardized global regulations adds to the complexity, necessitating careful coordination and adherence to local laws to avoid legal pitfalls.
- Achieving Efficient Settlement Across Jurisdictions - Efficient settlement of transactions across different jurisdictions poses significant logistical challenges. Each jurisdiction may have unique financial infrastructure, operational standards, and settlement timelines. Harmonizing these differences to facilitate smooth and timely settlement processes in a decentralized system requires robust mechanisms and coordination.
- Redefining Roles of Intermediaries Like Custodians and Trading Organizations - The shift to decentralized markets necessitates a redefinition of traditional roles played by intermediaries such as custodians and trading organizations. In a

decentralized environment, the responsibilities of these entities must be re-evaluated and adapted to fit the new model. This includes redefining their functions to ensure they continue to provide value, such as security and compliance, within the decentralized framework.

- Establishing Trust in Decentralized Transactions Without Centralized Authorities - One of the core challenges of decentralization is establishing trust in transactions without the presence of centralized authorities. Participants need assurance that the system is secure, transparent, and reliable. Building this trust requires robust security protocols, transparent operations, and mechanisms that guarantee the integrity and authenticity of transactions.
- Implementing Innovative Solutions for Additional Verification Processes in Decentralized Custody Services - In decentralized custody services, innovative solutions are necessary to handle additional verification processes. These solutions must ensure that assets are securely managed and that ownership and transactions are accurately verified. Implementing such solutions involves leveraging advanced technologies like blockchain, smart contracts, and digital identities to maintain high standards of security and reliability.

Addressing these challenges is crucial for the successful decentralization of secondary markets. A comprehensive approach that includes regulatory alignment, efficient logistical solutions, redefined roles for intermediaries, trust-building mechanisms, and innovative verification processes will pave the way for more effective and secure decentralized financial markets.

7.6.2 Comprehensive Solution Topology for Enhancing Efficiency

In contemporary capital markets, a hierarchical structure involving multiple entities leads to duplicated account information and limited visibility into holdings, resulting in settlement delays. Blockchain-driven decentralization addresses these issues by flattening the hierarchy and creating a network of peers with equal visibility. This streamlining facilitates both pre-trade confirmation and post-trade settlement processes. When securities are issued directly on blockchains, traditional registries like Central Securities Depositories (CSDs) become redundant. However, the role of custodians remains essential. Custodians are responsible for safeguarding client assets and

ensuring compliance with government regulations, which is particularly crucial for institutional investors.

In a blockchain context, custodians secure clients' assets through exclusive blockchain-based accounts. This method eliminates the need for reconciliation with accounts held by Financial Market Utilities (FMUs). Instead of maintaining parallel records, all transactions are recorded on the blockchain, providing a single, tamper-proof source of truth. This innovation reduces administrative overhead and enhances the accuracy and security of asset management. Custodians ensure that clients' assets are protected and compliant with regulatory requirements, while also adapting to the decentralized nature of blockchain technology. By leveraging blockchain, custodians can offer enhanced transparency and security, meeting the needs of modern capital markets without the inefficiencies of traditional hierarchical structures. Figure 7.17 illustrates this streamlined approach, highlighting how blockchain integration reduces redundancy and improves overall market efficiency. This new model underscores the critical role of custodians in a decentralized system, ensuring that the transition to blockchain-based securities maintains regulatory compliance and asset protection standards.

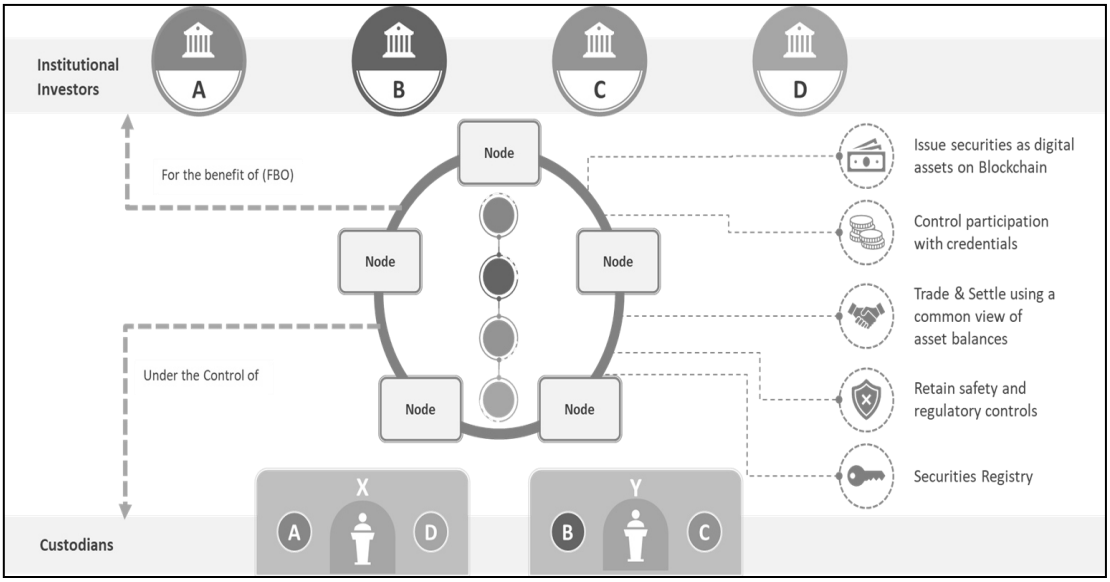


Figure 7. 17 Decentralized capital market structure

This approach guarantees equitable access for all market participants to a unified set of accounts, simplifying the processes of account opening, trading, and settlement directly through the registry or blockchain. By providing a centralized and transparent platform, common post-settlement issues such as duplicate trade matching and delivery failures

due to asset or cash shortages can be significantly minimized, thereby enhancing settlement efficiency. Custodians play a crucial role in this system by conducting various regulatory checks mandated by the jurisdiction. These checks include anti-money laundering (AML) procedures and sanctions screenings of the parties involved in transactions. If any violations are detected during these checks, the settlement processes may be paused to ensure compliance. The proposed solution involves implementing robust methods and operations that enable custodial oversight and regulatory checks while addressing pre and post-settlement challenges. Custodians retain control of the securities until the settlement is complete, ensuring that all counterparties comply with AML and sanctions requirements. Additionally, it is essential that securities remain available for trading on the exchange, with custodians ensuring their exclusive use for legitimate investor purposes. By integrating custodial oversight into the decentralized framework, the solution maintains regulatory compliance and enhances the security and reliability of the trading and settlement processes. Custodians ensure that assets are properly managed and protected throughout the transaction lifecycle, providing a safeguard against potential risks and violations. Overall, this integrated approach streamlines market operations, reduces inefficiencies, and enhances the security and compliance of capital markets. It allows market participants to engage in trading with greater confidence, knowing that robust custodial checks and balances are in place. This system is facilitated by categorizing securities held by the custodian into different states, each representing a specific stage in the transaction lifecycle:

- Normal: These are securities that are freely available and can be used for any purpose, including trading, collateral, or other financial operations.
- Locked: These securities are currently on offer at an exchange. They are reserved and cannot be used for other purposes until the transaction is completed or the offer is withdrawn.
- Committed: These securities have been sold and are now designated for settlement. They are in the final stage of the transaction process, awaiting transfer to the buyer.

By categorizing securities into these distinct states, the system ensures clear and efficient management of assets. This structure prevents potential issues such as double spending or allocation conflicts, as each security's status is transparent and accurately

reflects its current use or commitment. In the Normal state, securities can be utilized for any authorized transaction, providing flexibility to investors and traders. When securities enter the Locked state, they are earmarked for active trading on an exchange, ensuring they are not otherwise allocated or used during this period. Finally, in the Committed state, securities are prepared for settlement, guaranteeing that they are readily available for transfer to fulfil completed sales transactions. This tiered approach to managing securities enhances operational efficiency, reduces the risk of errors, and ensures that all market participants have a clear understanding of the availability and status of their assets. It supports seamless trading and settlement processes, contributing to a more stable and reliable market environment.

Execution of a trade requires the following steps:

Open

1. *Client (the beneficial owner of the securities) informs the custodian to reserve N security tokens and send them to the trading wallet*
2. *Custodian moves N security tokens from Normal to Locked state and informs the exchange*
3. *Exchange adds N proxy security tokens to the client's trading account*
4. *Client opens trading with N security tokens*

Trade

5. *Client agrees to trade N security tokens with a counterparty at a set price*
6. *N proxy security tokens are exchanged between the buyer and seller in the trading account*
7. *N security tokens are moved from Locked state to Committed state in the account secured by the custodian*

Settle

8. *Custodian performs counterparty AML/Sanctions screening*
    a. *If result is negative, N committed security tokens are exchanged for cash tokens with the counterparty custodian in an atomic transaction*
    b. *If result is positive, N committed security tokens are moved back to Normal state*

9. *Proxy tokens in the exchange account are removed*

Both parties involved in a trade, along with their respective custodians, follow the outlined procedures, utilizing both security tokens and cash tokens. This methodology ensures that both tokenized securities and cash are available for settlement before a trade is initiated, thereby guaranteeing settlement upon trade matching. By ensuring that the required securities and cash are tokenized and available beforehand, this process mitigates the risk of settlement failure. Additionally, the simplified custody structure eliminates the need for redundant account updates. This improvement in settlement efficiency is achieved through single automated token exchanges facilitated by smart contracts. Smart contracts automatically execute and enforce the terms of the trade, ensuring that the transfer of tokens occurs seamlessly and accurately once the trade conditions are met. This automation significantly reduces manual intervention and the associated risks and delays. However, final settlement is deferred until the custodians verify the safety and regulatory compliance of the transaction. Custodians play a crucial role in this process, conducting thorough checks to ensure adherence to regulations such as anti-money laundering (AML) and sanctions screening. If any issues or regulatory breaches are detected, the settlement process is halted to prevent non-compliant transactions. This ensures that regulatory controls mandated by the jurisdiction are preserved and enforced. Custodians maintain their essential role in safeguarding assets and ensuring compliance, providing an additional layer of security and trust in the transaction process. Overall, this comprehensive methodology enhances settlement efficiency, ensures regulatory compliance, and maintains the integrity of the trading system. By leveraging tokenization and smart contracts, the process becomes more streamlined, transparent, and secure, benefiting all market participants.

7.6.3 System Design Strategies for Enhancing Decentralized Settlement Efficiency

In this experimental system, the blockchain platform chosen was Ethereum. Ethereum is known for its decentralized and open-source nature, which makes it a robust and secure choice for various blockchain applications. As a public distributed ledger,

Ethereum allows multiple participants to maintain and verify a shared record of transactions without the need for a central authority. This decentralized structure enhances security and transparency, making Ethereum an ideal platform for executing complex transactions and smart contracts. Hyperledger Besu was employed as the Ethereum Virtual Machine (EVM) client in this system. Hyperledger Besu is an enterprise-grade, open-source Ethereum client designed under the Hyperledger project, which is governed by the Linux Foundation. It provides a versatile and performant execution environment for Ethereum smart contracts, supporting both public and private networks. This EVM client connects seamlessly with the Ethereum blockchain, facilitating interactions with the distributed ledger and ensuring reliable execution of smart contracts. Ethereum is more than just a blockchain; it is a comprehensive platform that supports decentralized applications (DApps). Its transparent and immutable ledger system ensures that all transactions are publicly verifiable and traceable. Ethereum's built-in access control mechanisms provide fine-grained permissions, ensuring that only authorized entities can execute specific transactions or smart contracts. This transparency and control are crucial for maintaining the integrity and security of the system. There are multiple Ethereum virtual machine clients available, each capable of interacting with the Ethereum blockchain. These clients connect to the blockchain through JSON-RPC interfaces, which are standardized methods for remote procedure calls. These interfaces enable the clients to interact with the distributed ledger efficiently, executing smart contracts and performing transactions as permitted. The choice of EVM client can affect the performance and security of the system, and in this experimentation, Hyperledger Besu was chosen for its robustness and enterprise capabilities. JSON-RPC interfaces serve as the communication bridge between EVM clients and the Ethereum blockchain. They facilitate the transmission of commands and data, allowing clients to perform various operations such as querying the blockchain, sending transactions, and executing smart contracts. These interfaces are crucial for maintaining the functionality and connectivity of the blockchain system, ensuring that clients can interact with the ledger seamlessly. In the proposed trade system execution, every operation is governed by a smart contract. Smart contracts are self-executing contracts with the terms of the agreement directly written into code. They automatically enforce and execute the terms when predefined conditions are met, reducing the need for intermediaries and minimizing the risk of fraud or error. In this system, smart contracts control all trade operations, ensuring that each action is transparent, verifiable,

and immutable. The system includes a set of participants who perform actions to change states based on trade activities. These participants could include buyers, sellers, custodians, and regulators. Each participant has specific roles and responsibilities, such as initiating trades, verifying transactions, and ensuring compliance with regulations. By defining these roles and actions within smart contracts, the system enhances security and control, providing a reliable framework for digital security exchange and custodial processes. The use of smart contracts and decentralized technologies enhances the security and control of the system. Each transaction is transparently recorded on the blockchain, reducing the risk of fraud and ensuring compliance with regulatory requirements. Custodians play a crucial role in this process by conducting regulatory checks such as anti-money laundering (AML) and sanctions screenings. Settlement processes can be paused if any violations are detected, ensuring that only compliant transactions are executed. Custodians are responsible for ensuring that all transactions comply with relevant regulations. This includes conducting thorough checks on transaction parties and ensuring that all securities and cash are available and verified before trades are initiated. By maintaining control of the assets until settlement, custodians help to prevent non-compliant transactions and ensure the integrity of the financial system. An illustrative figure (referred to as Figure 7.18) could depict the overall architecture and workflow of this experimental system. It would show how Ethereum and Hyperledger Besu interact, the flow of transactions, the role of smart contracts, and the actions performed by various participants. This visual representation would help to clarify the system's design and functionality, highlighting the innovative use of blockchain technology to improve efficiency, security, and compliance in digital security exchanges.

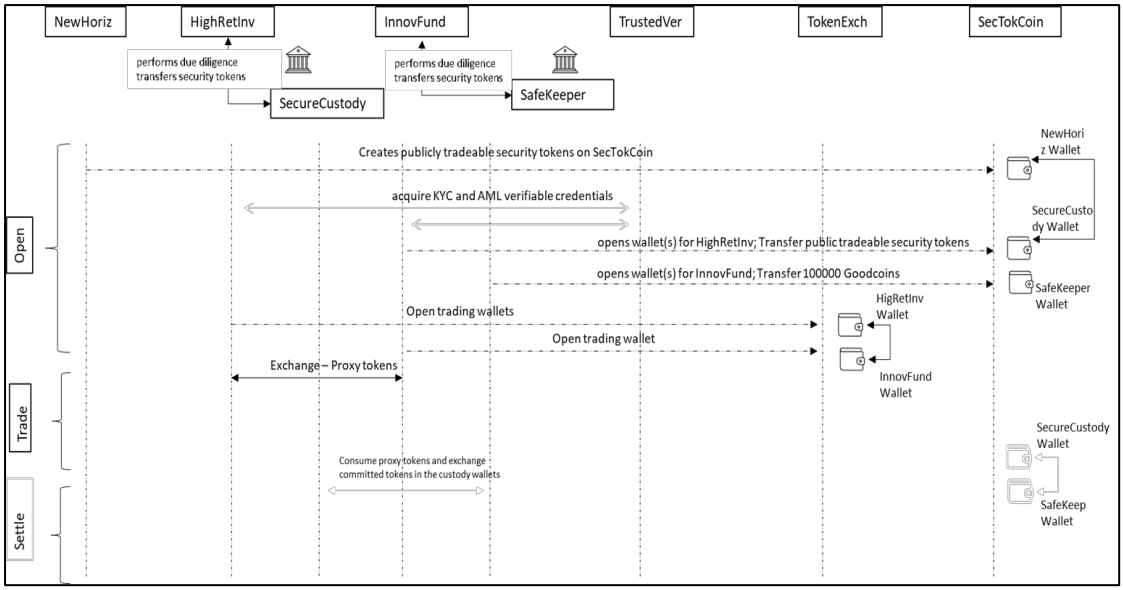


Figure 7. 18 Participants interaction flow

Primary participants –

- HighRetInv: Private investment firm that acquired pre-IPO and IPO shares in NewHoriz.
- InnovFund: Mutual fund that wants to purchase NewHoriz shares.
- SecureCustody: Custodian services provider for HighRetInv
- SafeKeeper: Custodian for InnovFund

Background participants –

- NewHoriz: Tech Startup that went through an IPO.
- WeOfferGoodCoins: A consortium that issues GoodCoin currency backed tokens.
- TokenExch: An exchange that facilitates security token trading.
- TrustedVer: A well-known KYC and AML credential provider
- SecTokChain: A Blockchain network for security tokenization and exchange

Background Setup –

- NewHoriz undergoes an IPO, converting all pre-IPO shares into publicly tradable security tokens on SecTokChain.

- HighRetInv and InnovFund obtain KYC and AML verifiable credentials from TrustedVer.
- SecureCustody and SafeKeeper establish relationships with TokenExch, authorizing wallets held at TokenExch.
- HighRetInv establishes a digital custodial relationship with SecureCustody.
- SecureCustody conducts due diligence, verifying HighRetInv's credentials and setting up wallets on SecTokChain (with private keys held by SecureCustody).
- HighRetInv transfers all publicly tradable NewHoriz security tokens to SecureCustody's custodianship.
- SecureCustody moves the NewHoriz tokens from HighRetInv to the wallet established for HighRetInv on SecTokChain.
- InnovFund purchases GoodCoins from the consortium and obtains KYC and AML credentials from TrustedVer.
- InnovFund establishes a digital custodial relationship with SafeKeeper.
- SafeKeeper conducts due diligence, verifying InnovFund's credentials and setting up wallets on SecTokChain (with private keys held by SafeKeeper).
- InnovFund transfers GoodCoins to SafeKeeper's custodianship.
- SafeKeeper moves the GoodCoins from InnovFund to the wallet established for InnovFund on SecTokChain.
- InnovFund and HighRetInv open trading wallets with TokenExch.
- TokenExch verifies credentials provided by InnovFund and HighRetInv, opening wallets on SecTokChain (with private keys held by corresponding investors).

Background –

- Tokenize NewHoriz stock into 1000000 ATokens (tokenized securities). Henceforth, actual security shall be referred as AToken.
- Transfer 10000 AToken to HighRetInv (Pre-IPO private share holder)
- InnoFund obtains 100000 GTokens (GoodCoin currency backed tokens) from the consortium that issues them. Henceforth, GoodCoin shall be referred to as GToken.

Pre-requisite -

- HighRetInv entrusts 10000 AToken to its custodian, SecureCustody.
- InnovFund entrusts 100000 GToken to its custodian, SafeKeeper.

Action 0 – Initial State

HighRetInv -	*Self Wallet View*		
AToken	0	GToken	0

HighRetInv -	*Custody View*		
AToken	Normal Token = 10000	Locked Token = 0	Committed Token = 0
GToken	Normal Token = 0	Locked Token = 0	Committed Token = 0

InnovFund -	*Self Wallet View*		
AToken	0	GToken	0

InnovFund-	*Custody View*		
AToken	Normal Token = 0	Locked Token = 0	Committed Token = 0
GToken	Normal Token = 100000	Locked Token = 0	Committed Token = 0

State Changes

Precondition: security and cash tokens are available in custody wallets

Action 1 – Open

Primary actors: Investors (HighRetInv & InnovFund)

Secondary actors: Custodians (SecureCustody & SafeKeeper)

Postcondition: Proxy security and cash tokens are made available in the exchange wallets controlled by investors and an equal number of security and cash tokens are locked in the custody wallets.

For HighRetInv -

1. HighRetInv opens with 1000 ATokens.
2. 1000 ATokens are locked in custodial account.
3. 1000 proxy ATokens are assigned to exchange wallet of HighRetInv.

HighRetInv -	*Self Wallet View*		
AToken	1000	GToken	0

HighRetInv -	*Custody View*		
AToken	Normal Token = 9000	Locked Token = 1000	Committed Token = 0
GToken	Normal Token = 0	Locked Token = 0	Committed Token = 0

For InnovFund -

1. InnovFund opens with 10000 GTokens.
2. 10000 GTokens are locked in custodial account.
3. 10000 proxy GTokens are assigned to exchange wallet of InnovFund.

InnovFund -	*Self Wallet View*		
AToken	0	GToken	10000

InnovFund -	*Custody View*		
AToken	Normal Token = 0	Locked Token = 0	Committed Token = 0
GToken	Normal Token = 90000	Locked Token = 10000	Committed Token = 0

Action 2 – Trade (Buy/Sell)

Actors: Investors (HighRetInv & InnovFund)

Postcondition: Proxy tokens are exchanged between the trading partners' exchange wallets, and corresponding tokens in custody wallets are moved from locked to committed state.

Trade by HigRetInv & InnovFund -

1. HighRetInv initiates sale of 100 ATokens at 50 GTokens each
2. InnovFund accepts and agrees to trade.
3. 100 proxy ATokens are exchanged for 5000 proxy GTokens between exchange accounts of InnovFund and HighRetInv
4. 100 ATokens are moved from Locked to Committed state in the custodial account.
5. 5000 GTokens are moved from Locked to Committed state in the custodial account.

Exchange View			
SELL	AToken	Quantity=100	Unit GToken =50
BUY	AToken	Quantity=100	-

After Trade -State

HighRetInv -	*Self Wallet View*		
AToken	900	GToken	5000

HighRetInv -	*Custody View*		
AToken	Normal Token = 9000	Locked Token = 900	Committed Token = 100
GToken	Normal Token = 0	Locked Token = 0	Committed Token = 0

InnovFund -	*Self Wallet View*		
AToken	100	GToken	5000

InnovFund -	*Custody View*		
AToken	Normal Token = 0	Locked Token = 0	Committed Token = 0
GToken	Normal Token = 90000	Locked Token = 5000	Committed Token = 5000

Action 3 – Settlement

Actors: Custodians (SecureCustody & SafeKeeper)

Postcondition: Consume proxy tokens in exchange wallets and exchange committed tokens in the custody wallets with the counterparty and convert them to Normal tokens in the counterparty wallets.

1. Settlement instruction is sent by HighRetInv and InnovFund to the corresponding custodians.
2. Both custodians perform AML and Sanctions Checks on counterparties
3. If the checks are negative (pass)
4. Committed AToken and GTokens are exchanged between custodian accounts and converted to Normal state.
5. Proxy AToken and GTokens are destroyed from the exchange wallets.

For HighRetInv –

HighRetInv -	*Self Wallet View*		
AToken	900	GToken	0

HighRetInv -	*Custody View*		
AToken	Normal Token = 9000	Locked Token = 900	Committed Token = 0
GToken	Normal Token = 5000	Locked Token = 0	Committed Token = 0

For InnovFund –

InnovFund -	*Self Wallet View*		
AToken	0	GToken	5000

InnovFund -	*Custody View*		
AToken	Normal Token = 100	Locked Token = 0	Committed Token = 0
GToken	Normal Token = 90000	Locked Token = 5000	Committed Token = 0

7.6.4 Tactical Implementation Strategies for Maximizing Market Efficiency

A. Architecture and Technology used –

The architecture and technology mapping of each essential component pertinent to this proof of concept are delineated below. Figure 7.19 aptly illustrates the overall system architecture and the integration of system components to meet the demands of

decentralized business applications. The Besu boot node assumes the crucial role of discovering all peer nodes within the system. Upon the addition of a worker node to the network, the boot node automatically detects it and incorporates it into the network. Through the JSON-RPC API, the blockchain client can execute blockchain operations using Web3 interfaces to meet the application requirements. My focus lies predominantly on the configuration and business process aspects, employing a blockchain architecture based on Hyperledger Besu to validate the concept. Subsequently, the ensuing section outlines my practical experimentation and standard configuration, sufficient to validate the key concept.

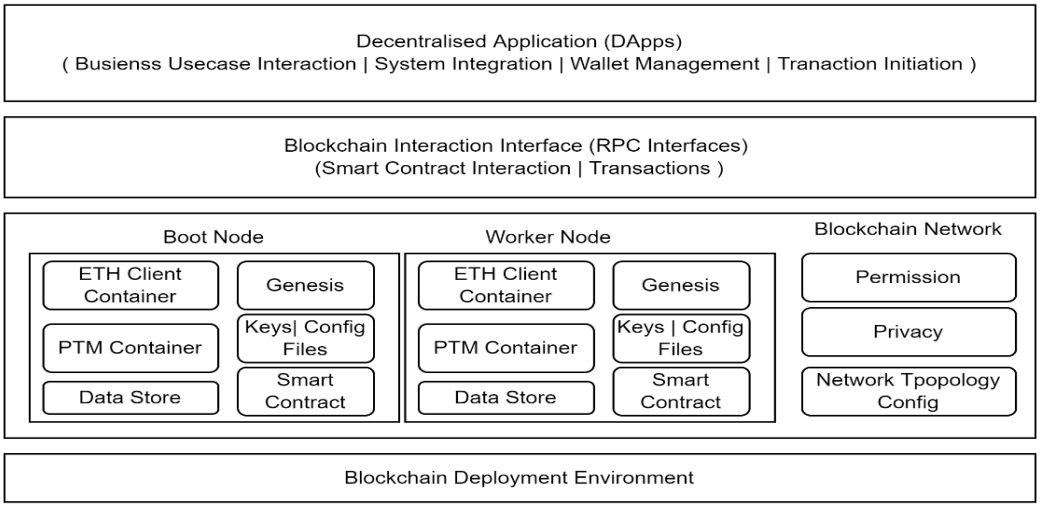


Figure 7. 19 Logical system component architecture

The table below elucidates the technologies employed in developing the proof of concept.

Software	Version	Usage
Hyperledger Besu	22.4.x	Ethereum client for both public and private permissioned network [22]
MetaMask plugin	10.14.x	A crypto wallet for blockchain applications
VS Code	1.6x	Development tool to write code and configuration files
Node.js	16.x	Used to compile web3 codebase and deploy contracts from DApp

Web3JS	1.3.5	JavaScript Web3 library to interact with the Ethereum network to enable JavaScript based application integration.
Solidity	0.8.x	Smart contract development language

Table 7. 6 Software and their usage

B. The execution approach categorizes the below buckets. In this experimentation, the infrastructure setup is initiated, followed by smart contract development and the creation of a decentralized application prototype.

Blockchain Infrastructure Setup - As part of the setup for the Blockchain node infrastructure, the Boot Node was initialized, followed by the initialization of worker nodes. Once the worker nodes were brought online, peers were automatically discovered. All nodes were then brought online and configured to listen and accept client requests. For this proof of concept, a three-node Hyperledger Besu cluster was initiated. As the worker nodes were initialized one by one using peer-to-peer discovery, the network was able to identify the peers and add them into the network.

Worker node –

```
curl -X POST --data '{"jsonrpc":"2.0","method":"admin_addPeer","params":
["enode://a173be1e7b76411e792debe374909afda35c38379dc0bcb4926ed001568c2d40f390d63c2c63c100457a4
f710b6d6b1f1807aef6a1dc31bf7d222182f8af942a@172.31.57.64:30303"],"id":1}' http://127.0.0.1:8546
```

Peering boot and worker nodes –

```
nchronizer | Starting synchronizer.
ownloader | Starting full sync.
argetManager | No sync target, waiting for peers. Current peers: 0
Ethereum main loop is up.
NFO  | FullSyncTargetManager | No sync target, waiting for peers. Current peers: 0
NFO  | FullSyncTargetManager | No sync target, waiting for peers. Current peers: 0
NFO  | FullSyncTargetManager | No sync target, waiting for peers. Current peers: 0
NFO  | FullSyncTargetManager | No sync target, waiting for peers. Current peers: 0
INFO  | FullSyncTargetManager | No sync target, waiting for peers. Current peers: 1
NFO  | FullSyncTargetManager | No sync target, waiting for peers. Current peers: 1
NFO  | FullSyncTargetManager | No sync target, waiting for peers. Current peers: 1
NFO  | FullSyncTargetManager | No sync target, waiting for peers. Current peers: 1
```

After the network has been established, two hot wallets are configured for each of the custodian banks in Metamask [26]. These wallets will be referenced in the smart

contract for the transfer of asset tokens (AToken) or cash tokens (GToken). Regarding Smart Contract Execution, the following smart contracts have been developed as part of the contract implementation and have been deployed on Besu nodes. The "TokenTrading" smart contract serves as the central program responsible for executing all operations initiated from decentralized applications.

It encompasses a set of functionalities essential to this process–

- As part of the function initialization, the contract generates ERC20 tokens and assigns them to custodians on behalf of investors.
- When securities are opened for trading, the contract generates proxy tokens for trading between investors. These tokens are then locked in custodian wallets corresponding to the respective investors.
- Upon execution of the "Trade" function for buying/selling securities, proxy tokens are exchanged, and the status of custodian tokens is updated to "commit."
- Custodians conduct due diligence on the credentials provided by their respective investors through the "TrustVerification" function.
- The "Settle" function facilitates the exchange of actual ERC20 tokens between custodian wallets and consumes the proxy tokens.

"GoodCoin" is an ERC20 token smart contract [28] designed for cash-equivalent transactions during the purchase of security tokens. On the other hand, "SecToken" is another smart contract that generates ERC20 tokens representing digital securities (AToken), traded on an exchange alongside GToken. Likewise, the "ProxyToken" smart contract mints ERC20 proxy tokens corresponding to the quantity of AToken and GToken traded on an exchange

Decentralized Application – Below are critical steps of the decentralized use case.

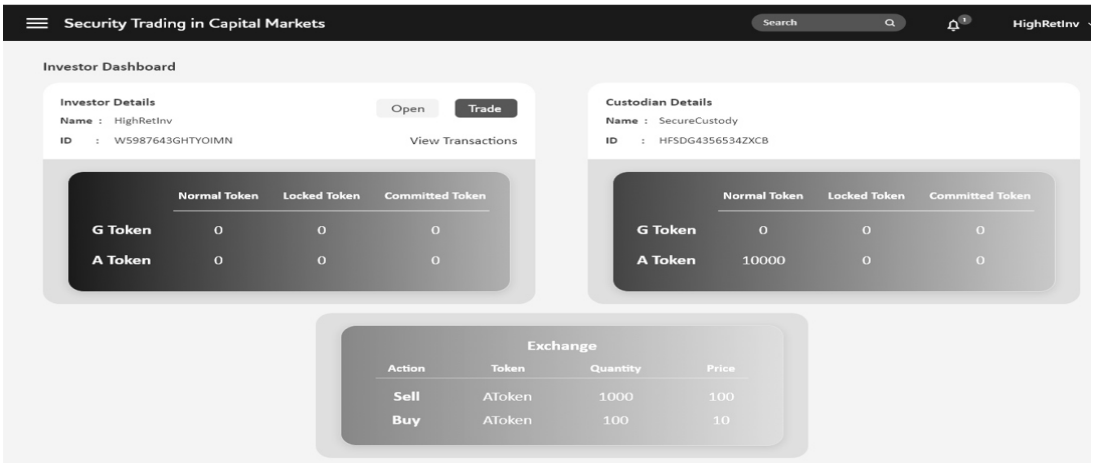


Figure 7. 20 Investor dashboard

In Figure 7.20 Investor (HighRetInv and InnovFund) opens the respective tokens. For that, when the Open action is initiated, the request goes to the respective custodians. Custodian can view the incoming request from the respective investors.

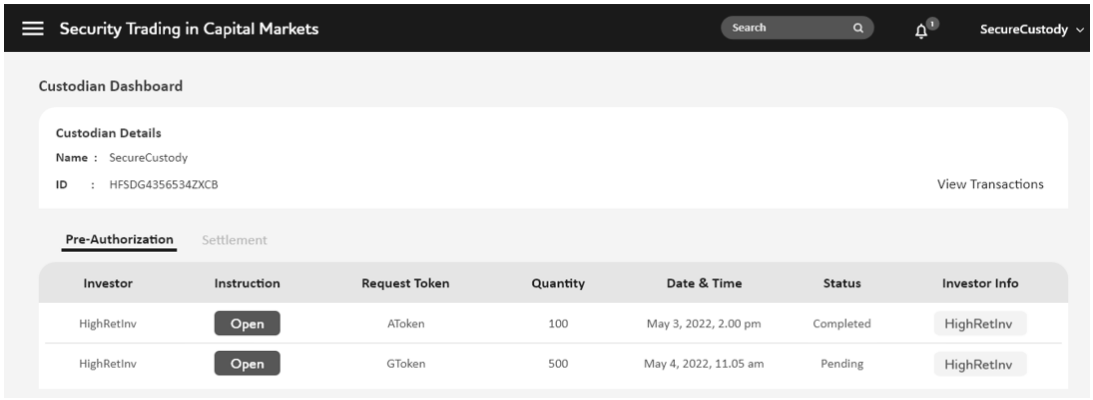


Figure 7. 21 Pre-authorization phase

As part of pre-authorization in Figure 7.21 Custodian can verify initial requests and open requested tokens (AToken or GToken). Once Tokens are locked, the system will create an equivalent amount of Proxy tokens (Proxy AToken and Proxy GToken) and assign to the respective wallets to continue with the trading operations. Investors (Seller and Buyer) now open the trade using proxy tokens (Figure 7.20). When the trade executes, two transfer requests using Proxy tokens (Buy and Sell requests) are executed exchanging the tokens between the buyer and seller (Figure 7.21) and committing the locked tokens in the custodian-controlled wallets.

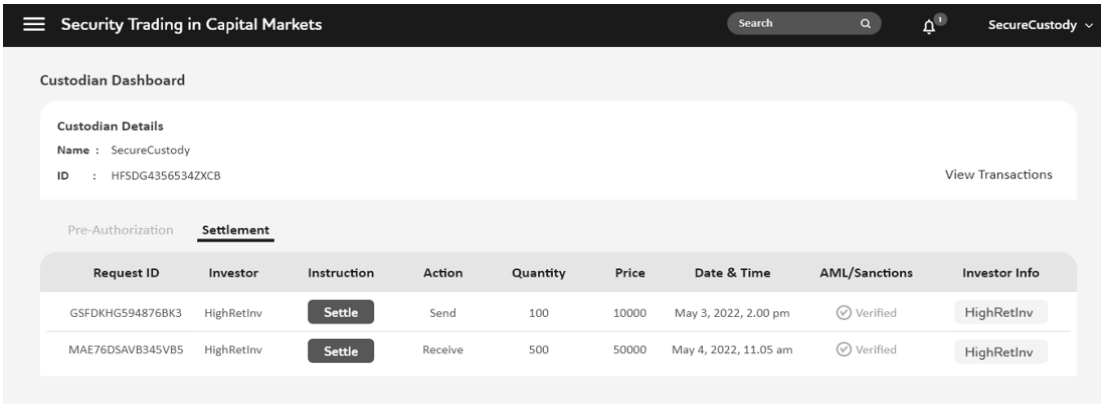


Figure 7. 22 Settlement flow

As part of the Settle step (Figure 7.22), buyer and seller send settlement instructions to the corresponding custodians. As part of this phase of execution, the custodians execute AML, Security, and Regulatory checks. Once verified, the system executes the settlement step exchanging the actual tokens in the custody wallets and removing the proxy tokens from the system.

7.6.5 Conclusion Insights and Future Implications

Looking ahead to future possibilities, several critical steps must be undertaken to harness Blockchain's transformative impact on financial markets effectively. These steps include navigating the shift towards decentralized custody chains, improving cross-border settlement efficiency, redefining intermediary roles, fostering trust in decentralized transactions, and innovating verification processes in custody services. Each of these steps plays a crucial role in the broader adoption and integration of Blockchain technology within the financial ecosystem. Navigating the shift towards decentralized custody chains is a pivotal step. Decentralized custody chains can streamline and secure the custody process, reducing the risk of errors and fraud inherent in traditional, centralized custody models. As Blockchain evolves, it necessitates adept management of regulatory complexities. The proposed solution offers a forward-looking framework designed to facilitate this transition, ensuring that regulatory requirements are met and that the new system integrates seamlessly with existing and emerging market frameworks. By empowering custodians to enforce jurisdiction-specific regulations within Blockchain networks, this approach effectively addresses regulatory challenges, providing a robust foundation for the decentralized future of

financial markets. Improving cross-border settlement efficiency is another significant challenge that must be addressed. Traditional cross-border settlements are often slow and cumbersome, hampered by the complexities of international banking systems. Blockchain's borderless nature offers a solution to these issues. The proposed methodology leverages Blockchain technology to enable rapid and secure settlement mechanisms, transcending geographical barriers and enhancing market liquidity. This capability is critical in a globalized economy where quick and efficient cross-border transactions can significantly impact market operations and investor confidence. Redefining intermediary roles in the decentralized landscape is also essential. Blockchain technology reimagines the functions of intermediaries, such as custodians and trading organizations, by utilizing its inherent transparency and security features. This redefinition can drive significant efficiency gains, reduce transactional costs, and streamline the financial ecosystem. Intermediaries can optimize their functions, focusing on value-added services rather than routine processes that Blockchain can automate. This strategic shift not only enhances operational efficiency but also allows financial institutions to better allocate their resources and capabilities. Fostering trust in decentralized transactions without centralized authorities is central to the success of Blockchain integration. Trust is a cornerstone of financial transactions, and Blockchain's immutable ledger and smart contracts provide the transparency, integrity, and accountability needed to foster this trust among participants. By reducing reliance on intermediaries, Blockchain instills a higher level of confidence in the system, encouraging broader adoption and utilization. Innovating verification processes in decentralized custody services is another critical area of focus. The proposed solution adopts advanced cryptographic techniques and decentralized identity systems to enhance the security and reliability of custody services. This approach not only ensures compliance with regulatory requirements but also strengthens the overall security framework within which financial transactions occur. By addressing the verification challenges head-on, the solution provides a robust mechanism for safeguarding assets and ensuring the integrity of transactions. In brief, the forward-looking solution addresses the multifaceted challenges of transitioning to decentralized custody chains, unlocking Blockchain's full potential in financial markets, and shaping a future marked by trust, efficiency, and innovation. By tackling issues related to regulatory compliance, cross-border settlement, intermediary roles, trust, and verification processes, the proposed framework sets the stage for a transformative shift in how financial markets

operate, paving the way for a more efficient, secure, and transparent financial ecosystem.

Conclusions

Throughout the study of each layer of IoT software architecture, I have meticulously focused on addressing issues that are critical, challenging, and highly demanded in the industry. The rapidly evolving landscape of the Internet of Things (IoT) presents a wealth of opportunities for innovation, particularly through the implementation of decentralized software architectures. Central to this evolution is the development of a robust IoT identity and whitelisting system integrated with decentralized trust management architectures, which collectively enhance security and trustworthiness across diverse IoT ecosystems. This decentralized approach not only strengthens identity verification and access control management but also addresses significant issues of privacy and information transparency, ensuring that data integrity and confidentiality are maintained. Furthermore, the architecture must be capable of managing unforeseen volume spikes in data traffic, necessitating a scalable and resilient data management framework that can adapt dynamically to varying loads at the data ingestion layer. Distributed ledger technology (DLT) plays a pivotal role in this context, offering immutable and transparent records that are invaluable for setting the course of risk management, particularly in industrial applications like parametric insurance businesses where precise and reliable data is crucial. By synergizing these elements—IoT identity and trust management, privacy and access control, adaptive data management, and the strategic application of DLT—a comprehensive decentralized IoT software architecture can be established. This architecture not only leverages the full potential of IoT but also ensures robust security, scalability, and transparency, thereby setting a solid foundation for future advancements and applications in various industries. The conclusion of this thesis underscores the imperative need for such an integrated approach, advocating for continuous innovation and collaboration across technological and regulatory domains to realize a secure, efficient, and trustworthy IoT ecosystem. Below subsections focus on individual aspects of each concept studied, proposed and future directions to this thesis.

A. Innovative Horizons: Synthesizing Conclusions on Research Paths and Opportunities in IoT and Decentralized Software Architecture

In the realm of IoT identity management, the conventional reliance on cryptographic algorithms for device authentication has faced challenges, particularly in terms of time-consuming digital signature production and verification processes. These challenges result in decreased communication speed, and the proposal of a smart card-based next-generation authentication architecture aims to address these issues. This architecture centralizes user authentication, allowing users and the control server to mutually agree to share session keys. However, the centralization of the Identity Provider/Control server introduces a potential single point of failure. The emergence of blockchain technology has offered a decentralized trust management solution, although recent papers maintain centralization in whitelisting components. Research areas such as auto-whitelisting and advancements in identity management architecture become pivotal for an efficient shift from manual whitelisting.

The evolution of IoT edge processing, historically confined to local environments, has seen significant developments with the standardization of communication protocols. Containerized microservice-based architectures have replaced traditional REST/SOAP-based SOA architectures, providing enhanced deployment flexibility. However, challenges arise in handling backpressure, ensuring real-time message delivery, and managing limited resources. The proposal of newer computing introduces potential solutions but requires further research in agent deployment and execution for effective edge processing. The exploration of dynamic handling of real-time processing tasks presents exciting avenues for future research.

The escalating number of sensors in IoT poses challenges to traditional storage and processing approaches. Existing database systems face obsolescence, prompting exploration into alternative systems such as time series, document-based, graph, and

relational databases. Architectural enhancements, including SAGA and CQRS patterns, event sourcing, computing, and change data capture, offer potential solutions to improve read-write performance and optimize data storage processes. Embracing these patterns can extend current capabilities to meet the escalating demands of handling petabytes of data.

Again, the rise in IoT data volume has spurred the emergence of decentralized communication architecture as a transformative solution. By distributing communication and decision-making capabilities across the network, this architecture enhances scalability, resilience, and security. The use of digital twins for predictive maintenance in Industrial IoT highlights the potential of decentralized communication architecture. However, challenges such as interoperability, resource constraints, and governance models must be addressed for widespread adoption. Despite these hurdles, the decentralized IoT communication architecture promises to revolutionize industries, offering enhanced connectivity, data privacy, and user control over personal information. The primary focus should be on designing and architecting the underlying platform to unlock the full benefits of this transformative architecture and create an efficient, secure, and user-centric connected world.

Future research in IoT is set to address challenges in identity management, edge processing, data storage, and decentralized communication architecture. Smart card-based authentication and decentralized trust management through blockchain technology are proposed solutions for identity management. Challenges in edge information processing drive exploration into real-time task handling. The increasing number of sensors prompts research into alternative databases and newer patterns. The transformative potential of decentralized communication architecture is noted, with a focus on overcoming challenges like interoperability and resource constraints. Central to this research is designing a robust underlying platform for a connected world characterized by efficiency, security, and user-centric experiences.

B. Concluding Insights and Future Trajectories in IoT Identity, Whitelisting, and Decentralized Trust Management Architecture

In this research, an innovative blockchain-based software architecture is introduced with the aim of fostering collaboration among service provider participants, addressing the critical issue of electronic waste. The primary goal is to establish a network where participants can achieve common consensus, creating an interoperable and highly reusable ecosystem tailored for Internet of Things (IoT) applications. The proposed architecture's detailed operations and process flow design are meticulously described, and a proof-of-concept implementation is conducted in both a local environment and a test blockchain network to validate the concepts. Despite observed delays in transaction times on public Ethereum and similar test blockchain networks, attributed to the time-consuming hash mining process, the imminent Ethereum upgrade is expected to significantly improve overall blockchain efficiency. The software architecture's core focus is on establishing a foundation for a self-managed, reusable, and interoperable IoT asset whitelisting process, with envisioned future enhancements including standard communication semantics for improved consensus, data isolation, and privacy. The integration of different encryption methods is proposed for added security at the transport and application levels. The experimentation solidifies the concept and foundational software architecture, paving the way for an exciting future roadmap. The solution holds great potential to revolutionize the IoT landscape, fostering sustainability and connectivity.

Expanding the ecosystem further involves extending the solution beyond e-waste to encompass various waste management processes and recycling initiatives. Robust reward and incentive systems are developed to motivate user participation and promote responsible waste management practices. Integrating the platform with existing e-waste management infrastructure and recycling facilities enhances data exchange and

streamlines workflows. Considering open-sourcing the platform code and documentation encourages community contributions and wider adoption. Real-world implementation and pilot projects are crucial for validating the solution. Collaborations with e-waste management entities, governments, or NGOs are sought for launching pilot projects, testing the solution in authentic scenarios. Economic and environmental impact analyses are conducted to assess the benefits of the platform in reducing e-waste generation and promoting sustainability. Regulatory considerations ensure compliance with blockchain technology and e-waste management regulations, preparing the platform for market readiness. Looking toward the long-term vision, the platform is positioned as a pivotal force in transforming the e-waste industry towards a more sustainable and circular economy model. Strategies for scaling the platform and facilitating global adoption are developed to address the escalating e-waste challenge. The ultimate goal is to contribute to the development of a more sustainable and environmentally conscious future, where circularity and responsible waste management practices prevail. In pursuing these future directions, the research endeavours to refine existing findings and set the stage for a collaborative, efficient, and sustainable approach to e-waste management empowered by blockchain technology.

C. A Comprehensive Conclusion and Forward Outlook on IoT Privacy, Information Transparency, and Access Control Management

Illustrating the significance of robust privacy protections, permissions, and access management within public blockchain networks, this study delves into an Internet of Things (IoT) supply chain scenario. By scrutinizing the latest research papers on privacy control mechanisms and architecture in private blockchains, the findings underscore the acceptability of public network restrictions tied to centralized control, semi-private transaction handling, closed network data architecture, closed group transaction access control, and mass adoption. The recommendation advocates the

implementation of access control and privacy features for public networks through the utilization of privacy groups. Leveraging the Ethereum (Besu) node's private transaction manager, users can actively participate in multiple privacy groups within the same network. To validate the core concept, a software infrastructure in the cloud was established, running 3 + 3 Besu and Tessera nodes to execute private transactions. Experimental results reveal decentralized application-initiated transactions that are selectively visible to a subset of nodes based on privacy group and access control settings. Smart contracts equipped with on-chain permissioning offer access to permission nodes and accounts, empowering network administrators to dynamically regulate vulnerabilities and threats through web-based node and account additions or revocations. However, for comprehensive validation, this architecture demands testing across various verticals, employing a broader software infrastructure with distinct Besu and Tessera nodes running on each virtual machine instance. Only through this extensive validation can a permissioned distributed ledger be established, ensuring robust user privacy protection.

The future trajectory of this research presents an exciting roadmap for further exploration and refinement. Building upon the established foundation, future endeavours should extend the validation process across diverse verticals, employing an expansive software infrastructure with distinct Besu and Tessera nodes running on each virtual machine instance with isolated, rotational key management with native cloud PaaS integration. This comprehensive testing will offer insights into the scalability and adaptability of the proposed architecture, ensuring its applicability across varied use cases. Furthermore, the integration of emerging technologies and advancements, such as zero-knowledge proofs and homomorphic encryption, could be explored to enhance the privacy and security features of the network. Collaborations with industry stakeholders and practitioners would be instrumental in aligning the research outcomes with real-world scenarios, addressing practical challenges, and fostering the adoption of privacy-focused blockchain solutions. Additionally, investigating the potential implications of regulatory frameworks on privacy-centric blockchain networks and proposing compliance measures would contribute to the holistic development of the research. The goal is to establish a future-proof, permissioned distributed ledger that not only safeguards user privacy but also aligns seamlessly with evolving technological landscapes and regulatory landscapes.

D. Drawing Conclusions and Charting Future Directions in Data Ingestion Architecture for Unforeseen Volume Spikes

In this extensive study, a thorough examination has been conducted to delve into the merits and challenges inherent in handling real-time, unpredictable data streams originating from various Internet of Things (IoT) sources. Addressing the dynamic nature of these data surges, a novel approach grounded in microservices architectural principles has been introduced. This groundbreaking microservices-based application architecture is further fortified through the integration of a data surge protection mechanism, allowing for the dynamic rebalancing of application instances. Consequently, the system exhibits a remarkable capacity to seamlessly adapt to abrupt increases in incoming data. Moreover, a comprehensive exploration has been undertaken to scrutinize the scalability of this rebalancing strategy as it transitions from on-premises environments to the more flexible and scalable realm of Platform as a Service (PaaS). This migration to the cloud environment not only streamlines the establishment of a more versatile infrastructure but also carries the potential to significantly reduce platform consumption costs, a critical consideration for cloud-operating enterprises. Looking ahead, a visionary outlook is articulated for the continued evolution of this strategy into an adaptive rebalancing approach. This evolution involves the seamless integration of the strategy with existing load balancing frameworks within PaaS environments, positioning it as a versatile and generic component. Such adaptability allows for its seamless assimilation into diverse PaaS architectures. By continually adapting to evolving data dynamics and optimizing resource utilization, this adaptive rebalancing strategy emerges as a promising avenue for augmenting the efficiency and cost-effectiveness of data-intensive applications in both contemporary and future PaaS ecosystems.

The future trajectory of this study unfolds across several dimensions, each aimed at further developing, refining, exploring, and applying the proposed strategies to address the evolving landscape of data-intensive applications in the cloud. The refinement of

the microservices architecture stands as a pivotal direction, focusing on the analysis of performance bottlenecks and the optimization of microservices to adeptly handle increasingly complex and demanding data surges. Concurrently, the enhancement of the data surge protection mechanism takes center stage, with a vision to incorporate machine learning algorithms for predictive capabilities, enabling proactive resource scaling ahead of potential bottlenecks. The expansion of Platform as a Service (PaaS) support envisions a broader integration scope, reaching across various PaaS platforms and cloud providers to maximize applicability. The exploration and integration phase involve strategic collaborations, incorporating adaptive load balancing frameworks, investigating containerization and serverless computing technologies, and exploring the adaptability of the strategy for edge computing applications in real-time scenarios like autonomous vehicles and smart cities. Real-world application and testing are emphasized through case studies across diverse industries, benchmarking against existing solutions, and considering open sourcing to foster community contributions. Looking into the long-term vision, the study aspires to develop the strategy into a self-adaptive and autonomic system, positioning it as a foundational component for emerging technologies such as real-time analytics, artificial intelligence, and the Internet of Everything (IoE). Advocacy for standardization and adoption seeks to simplify cloud application development, while enabling future technologies positions the strategy as a linchpin for the evolution of data-intensive applications in the cloud, ushering in a more efficient, scalable, and cost-effective era.

E. Concluding DLT's Role and Setting the Course for Risk Management for Parametric Insurance Business

The adoption of parametric insurance signifies a transformative paradigm shift in risk management strategies, particularly in addressing the challenges associated with predefined events. Unlike traditional insurance models that predominantly compensate actual losses incurred, parametric solutions focus on proactively covering the risk of the event itself. This innovative approach operates as a catalyst for growth in the fiercely competitive insurance landscape, creating a mutually beneficial scenario for both insurers and the insured. The primary focus of parametric insurance is on swift payouts and addressing losses challenging to model accurately. The study undertakes a comprehensive exploration of parametric insurance through specific use cases, with a notable emphasis on the transport and Quick Service Restaurant (QSR) sectors. The objective is to construct and execute a platform solution deeply rooted in decentralized immutable distributed ledger technology.

The proposed architectural solution integrates a range of functionalities, including robust system services, seamlessly integrated APIs, event-triggering mechanisms, streamlined claim initiation processes, and near real-time auto claim processing. This integration establishes an immutable audit trail, a critical feature for logistics businesses and organizations seeking transparency and accountability in their operations. The application of parametric insurance to address challenges such as trip delays showcases a detailed system design, technological integrations, and dynamic state modifications aligned with various trip delay criteria. This not only streamlines the claims process but also provides a proactive and efficient approach to risk management. Moreover, the study acknowledges the inherent issues in centralized claim processing systems, intricacies of existing processes, and difficulties in quantifying losses within supply chains. By leveraging external data sources to validate claims, the platform ensures accuracy and efficiency in processing, further enhancing its reliability. The end-to-end

application, featuring user interfaces for trip management, claim processing, and payouts, reflects a user-centric design that prioritizes accessibility and ease of use.

In conclusion, this study not only highlights the potential of parametric insurance in addressing complex risks but also positions it as a transformative force in the insurance industry. The proposed platform not only meets the immediate needs of insurers and policyholders but also sets the stage for the evolution of insurance practices in the face of increasingly complex and unpredictable risks. As the insurance landscape continues to evolve, this parametric insurance model demonstrates resilience and adaptability, marking a significant step towards a more efficient, transparent, and responsive risk management framework.

Looking ahead, the trajectory for advancing **parametric insurance platforms** involves a strategic expansion beyond the confines of the transport and Quick Service Restaurant (QSR) sectors. The exploration of opportunities in diverse industries such as agriculture, energy, and healthcare are pivotal, aiming to customize use cases and platform interfaces according to the unique demands of each sector. This expansion further delves into the feasibility of extending parametric coverage to address a broader spectrum of intricate risks within existing sectors, fostering adaptability to evolving challenges. The **technological frontier emphasizes** the integration of advanced analytics and artificial intelligence, heightening the precision of risk prediction and dynamic pricing within parametric contracts. Synergies with other innovative risk management solutions are also explored, envisioning a holistic approach to risk mitigation. Seamless integration with established insurance company systems and processes is essential, ensuring easy adoption and robust data exchange for enhanced interoperability. **Regulatory and legal considerations** form a crucial aspect, necessitating a thorough analysis of regulatory landscapes across diverse regions and jurisdictions. The establishment of standardized claim validation protocols and procedures is advocated to uphold fairness, transparency, and ethical practices in the industry. An active push for supportive legal and regulatory frameworks is envisioned, aiming to encourage widespread adoption and continuous development of parametric insurance solutions. In the realm of platform **enhancements and scalability**, a fortified focus on security and privacy features ensures data integrity, instilling user confidence in the reliability of the system. Robust scaling mechanisms are implemented to accommodate the growth in user base and transaction volume, ensuring uninterrupted

performance during periods of increased demand. The development of flexible deployment options catering to on-premises, cloud-based, and hybrid environments aims to provide adaptability to diverse user preferences.

This versatile platform can extend its utility beyond the insurance industry to various sectors, requiring only an upgrade to the logical service layer. The envisaged platform capabilities mark the future trajectory for the current piloted concept.

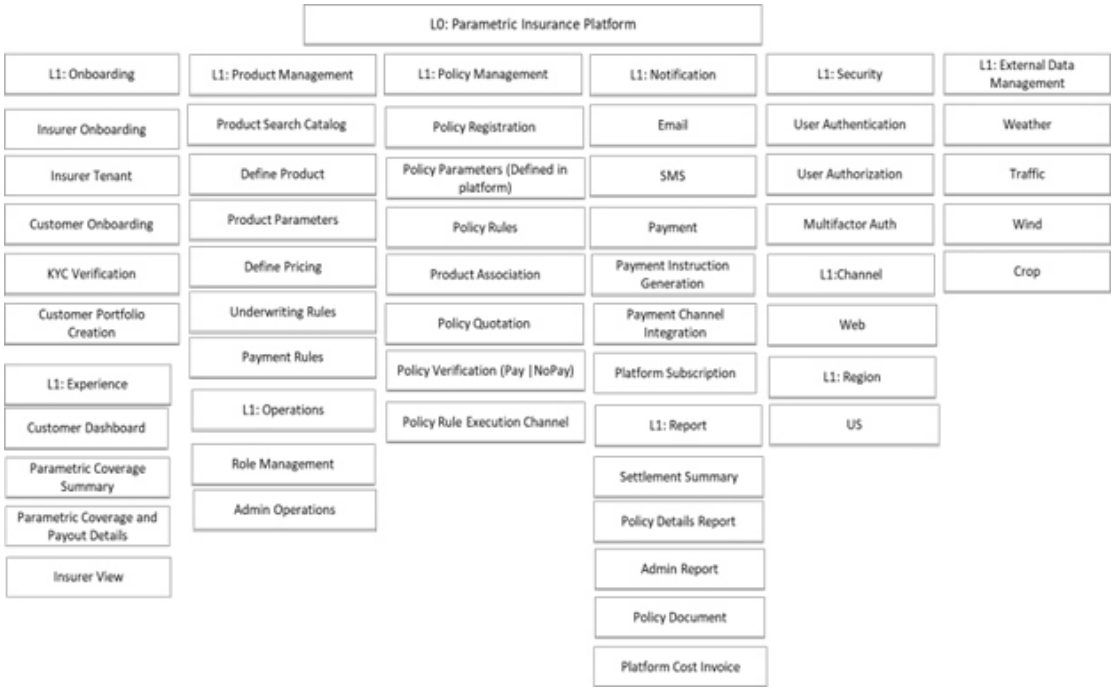


Figure 8. 1 Conceptualized decentralized parametric insurance platform capabilities

The long-term vision positions parametric insurance platforms as catalysts for extensive adoption and integration of decentralized ledger technology within the insurance industry. Advocacy for collaborative efforts and industry-wide standardization contributes to the establishment of a secure and efficient ecosystem for parametric insurance solutions. Ultimately, the study envisions contributing to the development of a resilient, responsive, and sustainable risk management infrastructure, shaping the future landscape of the insurance industry.

References

[1] Kenaza, R., Khemane, A., Bendjenna, H., Meraoumia, A., & Laimeche, L. (2022). Internet of Things (IoT): Architecture, applications, and security challenges. In *2022 4th International Conference on Pattern Analysis and Intelligent Systems (PAIS)* (pp. 1–5). IEEE. https://doi.org/10.1109/PAIS56586.2022.9946918

[2] Minerva, R., Biru, A., & Rotondi, D. (2015). Towards a definition of the Internet of Things (IoT) (Rev. 1). IEEE IoT Initiative. https://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf

[3] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials, 17(4), 2347–2376. https://doi.org/10.1109/COMST.2015.2444095

[4] Ebrahim, N. S. (2023). Complexity of IoT world – Review of challenges and opportunities in application development. In *2023 International Conference on Smart Computing and Application (ICSCA)* (pp. 1–5). IEEE. https://doi.org/10.1109/ICSCA57840.2023.10087783

[5] Bangare, P. S., & Patil, K. P. (2022). Security issues and challenges in Internet of Things (IoT) system. In *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)* (pp. 91–94). IEEE. https://doi.org/10.1109/ICACITE53722.2022.9823709.

[6] Aaqib, M., Ali, A., Chen, L., et al. (2023). IoT trust and reputation: A survey and taxonomy. *Journal of Cloud Computing, 12*(42). https://doi.org/10.1186/s13677-023-00416-8

[7] Su, R., Sfar, A. R., Natalizio, E., Moyal, P., & Song, Y.-Q. (2022). Ensuring trustworthiness in IoIT/AIoT: A phase-based approach. IEEE Internet of Things Magazine, 5(2), 84–88. https://doi.org/10.1109/IOTM.001.2100190

[8] Liu, Y., Wang, J., Yan, Z., Wan, Z., & Jäntti, R. (2023). A survey on blockchain-based trust management for Internet of Things. *IEEE Internet of Things Journal, 10*(7), 5898–5922. https://doi.org/10.1109/JIOT.2023.3237893

[9] Eisenbarth, J.-P., Cholez, T., & Perrin, O. (2021). A comprehensive study of the Bitcoin P2P network. In *2021 3rd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)* (pp. 105–112). IEEE. https://doi.org/10.1109/BRAINS52497.2021.9569782

[10] Vivek Anand, M., Mithun, S., Dhivya Shree, L. S., & Ranjith, M. (2023). Survey on connecting to the decentralized storage using IPFS protocol with Web 3 technology. In *2023 International Conference for Advancement in Technology (ICONAT)* (pp. 1–4). IEEE. https://doi.org/10.1109/ICONAT57137.2023.10080423

[11] Fong, D. K. Z., Selvarajah, V., & Nabi, M. S. (2022). Secure server storage based IPFS through multi-authentication. In *2022 International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC)* (pp. 1–7). IEEE. https://doi.org/10.1109/ASSIC55218.2022.10088338

[12] Bader, F., Radoveneanu, A., & Ragab-Hassen, H. (2011). A new security architecture for BitTorrent. In *2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications* (pp. 451–455). IEEE. https://doi.org/10.1109/TrustCom.2011.58

[13] Bhat, A., Nor, R. M., Mansor, H., & Amiruzzaman, M. (2021). Leveraging decentralized Internet of Things (IoT) and blockchain technology in international trade. In *2021 International Conference on Cyber Security and Internet of Things (ICSIoT)* (pp. 1–6). IEEE.

https://doi.org/10.1109/ICSIoT55070.2021.00010

[14] Steichen, M., Fiz, B., Norvill, R., Shbair, W., & State, R. (2018). Blockchain-based, decentralized access control for IPFS. In *2018 IEEE International Conference on Internet of Things (iThings), IEEE Green Computing and Communications (GreenCom), IEEE Cyber, Physical and Social Computing (CPSCom), and IEEE Smart Data (SmartData)* (pp. 1499–1506). IEEE. https://doi.org/10.1109/Cybermatics_2018.2018.00253

[15] Jonny, Kriswanto, & Toshio, M. (2021). Modeling IoT and Big Data implementation. In *2021 International Conference on Information Management and Technology (ICIMTech)* (pp. 645–650). IEEE. https://doi.org/10.1109/ICIMTech53080.2021.9535084

[16] Lv, W., Meng, F., Zhang, C., Lv, Y., Cao, N., & Jiang, J. (2017). A general architecture of IoT system. In *2017 IEEE International Conference on Computational Science and Engineering (CSE) and Embedded and Ubiquitous Computing (EUC)* (pp. 659–664). IEEE. https://doi.org/10.1109/CSE-EUC.2017.124

[17] Islam, M. M., Nooruddin, S., Karray, F., & Muhammad, G. (2023). Internet of Things: Device capabilities, architectures, protocols, and smart applications in healthcare domain. *IEEE Internet of Things Journal, 10*(4), 3611–3641. https://doi.org/10.1109/JIOT.2022.3228795

[18] Fan, C., Khazaei, H., Chen, Y., & Musilek, P. (2019). Towards a scalable DAG-based distributed ledger for smart communities. In *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)* (pp. 177–182). IEEE. https://doi.org/10.1109/WF-IoT.2019.8767342.

[19] Gartner. (2017). 8.4 billion connected things will be in use in 2017. *Gartner Newsroom*. https://www.gartner.com/newsroom/id/3598917

[20] OAuth Security Workshop. (2018, March). *OAuth Security Workshop 2018*.

https://st.fbk.eu/osw2018

[21] Bhawiyuga, A., Data, M., & Warda, A. (2017). Architectural design of token-based authentication of MQTT protocol in constrained IoT device. In *2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA)* (pp. 1–4). IEEE. https://doi.org/10.1109/TSSA.2017.8272933

[22] N. V., R., & K. P., M. (2020). Survey on state of art IoT protocols and applications. In *2020 International Conference on Computational Intelligence for Smart Power System and Sustainable Energy (CISPSSE)* (pp. 1–3). IEEE. https://doi.org/10.1109/CISPSSE49931.2020.9212227

[23] Sivaraman, V., Gharakheili, H. H., Vishwanath, A., Boreli, R., & Mehani, O. (2015). Network-level security and privacy control for smart home IoT devices. In *2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)* (pp. 163–167). IEEE. https://doi.org/10.1109/WiMOB.2015.7347976

[24] El-Hajj, M., Fadlallah, A., Chamoun, M., & Serhrouchni, A. (2019). A survey of Internet of Things (IoT) authentication schemes. *Sensors, 19*(5), 1141. https://doi.org/10.3390/s19051141.

[25] Pelaez, R. M., Cruz, H. T., Michel, J. R., Garcia, V., Mena, L. J., Felix, V. G., & Brust, A. O. (2019). An enhanced lightweight IoT-based authentication scheme in cloud computing circumstances. *Sensors, 19*(9), 2098. https://doi.org/10.3390/s19092098

[26] Yu, S., Park, K., & Park, Y. (2019). A secure lightweight three-factor authentication scheme for IoT in cloud computing environment. *Sensors, 19*(16), 3598. https://doi.org/10.3390/s19163598.

[27] Lee, J., Yu, S., Park, K., Park, Y., & Park, Y. (2019). Secure three-factor authentication protocol for multi-gateway IoT environments. *Sensors, 19*(10),

2358. https://doi.org/10.3390/s19102358.

[28] Mendez, D., & Yang, B. (2018). Blockchain-based whitelisting for consumer IoT devices and home networks. In *Proceedings of the 1st Workshop on Blockchain-enabled Networked Sensor Systems* (pp. 13–18). ACM. https://doi.org/10.1145/3241815.3241853.

[29] Hammi, M. T., Hammi, B., Bellot, P., & Serhrouchni, A. (2018). Bubbles of Trust: A decentralized blockchain-based authentication system for IoT. *Computers & Security, 78*, 126–142. https://doi.org/10.1016/j.cose.2018.06.004

[30] Marchal, S., Miettinen, M., Nguyen, T., Sadeghi, A.-R., & Asokan, N. (2019). AuDI: Toward autonomous IoT device-type identification using periodic communication. *IEEE Journal on Selected Areas in Communications, 37*(6), 1402–1412. https://doi.org/10.1109/JSAC.2019.2904364.

[31] Emery, V., Gray, J., & Fragale, D. (n.d.). *Automoni for trusted IoT* (Version 0.9.4a) [White paper]. https://doi.org/139778950.10

[32] Hanada, Y., Hsiao, L., & Levis, P. (2018). Smart contracts for machine-to-machine communication: Possibilities and limitations. In *2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)* (pp. 130–136). IEEE. https://doi.org/10.1109/IOTAIS.2018.8600854

[33] Pant, T. (2019). Ingesting IoT and sensor data at scale. *Hackernoon*. https://hackernoon.com/ingesting-iot-and-sensor-data-at-scale-ee548e0f8b78

[34] Gartner. (2018). How IoT impacts data and analytics. *Gartner Research*. https://www.gartner.com/smarterwithgartner/how-iot-impacts-data-and-analytics

[35] Ray, P. P. (2016). A survey on Internet of Things architectures. *EAI Endorsed*

Transactions on Internet of Things, 2(7), e2. https://doi.org/10.4108/eai.1-12-2016.151714.

[36] Wang, D., Lee, S., Zhu, Y., & Li, Y. (2017). A zero human-intervention provisioning for industrial IoT devices. In *2017 IEEE International Conference on Industrial Technology (ICIT)* (pp. 1271–1276). IEEE. https://doi.org/10.1109/ICIT.2017.7915546

[37] Adegbija, T., Lysecky, R., & Kumar, V. V. (2019). Right provisioned IoT edge computing: An overview. In *Proceedings of the Great Lakes Symposium on VLSI 2019 (GLSVLSI '19)* (pp. 1–6). ACM. https://doi.org/10.1145/3299874.3319338

[38] Mach, P., & Becvar, Z. (2017). Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials, 19*(3), 1628–1656. https://doi.org/10.1109/COMST.2017.2682318

[39] Mocnej, J., Seah, W., Pekar, A., & Zolotová, I. (2018). Decentralized IoT architecture for efficient resources utilisation. *IFAC-PapersOnLine, 51*(6), 168–173. https://doi.org/10.1016/j.ifacol.2018.07.148.

[40] Villari, M., Fazio, M., Dustdar, S., Rana, O., & Ranjan, R. (2016). Osmotic computing: A new paradigm for edge/cloud integration. *IEEE Cloud Computing, 3*(6), 76–83. https://doi.org/10.1109/MCC.2016.124.

[41] Carnevale, L., Celesti, A., Galletta, A., Dustdar, S., & Villari, M. (2019). Osmotic computing as a distributed multi-agent system: The body area network scenario. Internet of Things, 5, 100064. https://doi.org/10.1016/j.iot.2019.01.001

[42] Richardson, C. (2016, March 8). Refactoring a monolith into microservices. Eventuate, Inc. https://www.nginx.com/blog/refactoring-a-monolith-into-microservices/

[43] The New Stack. (n.d.). What led Amazon to its own microservices architecture. https://thenewstack.io/led-amazon-microservices-architecture.

[44] Butzin, B., Golatowski, F., & Timmermann, D. (2016). Microservices approach for the internet of things. In 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA) (pp. 1–6). IEEE. https://doi.org/10.1109/ETFA.2016.7733707

[45] Alam, M., Rufino, J., Ferreira, J., Ahmed, S. H., Shah, N., & Chen, Y. (2018). Orchestration of microservices for IoT using Docker and edge computing. IEEE Communications Magazine, 56(9), 118–123. https://doi.org/10.1109/MCOM.2018.1701233

[46] Yu, R., Kilari, V. T., Xue, G., & Yang, D. (2019). Load balancing for interdependent IoT microservices. *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, 298–306. https://doi.org/10.1109/INFOCOM.2019.8737450

[47] Richardson, C. (2016). *Pattern: Microservice chassis*. https://microservices.io/patterns/microservice-chassis.html

[48] Romanov, E. L., & Troshina, G. V. (2017). The IoT-architecture on the principles of reactive programming. *2017 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)*, 317–322. https://doi.org/10.1109/SIBIRCON.2017.8109897

[49] Lv, H., Ge, X., Zhu, H., Wang, C., Yuan, Z., & Zhu, Y. (2019). Design and implementation of reactive distributed Internet of Things platform based on actor model. *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 1993–1996. https://doi.org/10.1109/ITNEC.2019.8729169

[50] Lira, C., Mello, B., & Prazeres, C. (2019). Reactive microservices for the Internet of Things: A case study in fog computing. *Proceedings of the 2019 ACM Symposium on Applied Computing*, Article 10. https://doi.org/10.1145/3297280.3297402

[51] da Cruz, M. A. A., Rodrigues, J. J. P. C., Al-Muhtadi, J., Korotaev, V. V., & de Albuquerque, V. H. C. (2018). A reference model for Internet of Things middleware. *IEEE Internet of Things Journal, 5*(2), 871–883. https://doi.org/10.1109/JIOT.2018.2796561

[52] Patel, M., & Bhise, M. (2019). Raw data processing framework for IoT. In *2019 11th International Conference on Communication Systems & Networks (COMSNETS)* (pp. 695–699). IEEE. https://doi.org/10.1109/COMSNETS.2019.8711408

[53] Smidt, H., Thornton, M., & Ghorbani, R. (2018). Smart application development for IoT asset management using graph database modeling and high-availability web services. In *Proceedings of HICSS 2018*. https://doi.org/10.24251/HICSS.2018.725

[54] Di Martino, S., Fiadone, L., Peron, A., Riccabone, A., & Vitale, V. N. (2019). Industrial Internet of Things: Persistence for time series with NoSQL databases. In *2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)* (pp. 340–345). IEEE. https://doi.org/10.1109/WETICE.2019.00076

[55] Naqvi, S. N. Z., Yfantidou, S., & Zimányi, E. (2017). Time series databases and InfluxDB. *Studienarbeit*, Université Libre de Bruxelles, https://cs.ulb.ac.be/public/_media/teaching/influxdb_2017.pdf

[56] Chen, B., Eck, B., Fusco, F., Gormally, R., Purcell, M., Sinn, M., & Tirupathi, S. (2018). Castor: Contextual IoT time series data and model management at scale. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)* (pp. 1487–1492). IEEE.

https://doi.org/10.1109/ICDMW.2018.00213

[57] Siow, E., Tiropanis, T., Wang, X., & Hall, W. (2018). TritanDB: Time-series rapid Internet of Things analytics. *arXiv. https://arxiv.org/abs/1801.07947*

[58] Gupta, H., Xu, Z., & Ramachandran, U. (2018). DataFog: Towards a holistic data management platform for the IoT age at the network edge. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 2018)*, Boston, MA, July 10, 2018

[59] Nolan, M., McGrath, M. J., Spoczynski, M., & Healy, D. (2019). Adaptive industrial IoT/CPS messaging strategies for improved edge compute utility. In *IoT-Fog '19: Workshop on Fog Computing and the IoT* (pp. 1–5). ACM. https://doi.org/10.1145/3313150.3313220

[60] Pillai, A., Gaddam, C. P., & Khwaja, A. (2019). A service-oriented IoT architecture for disaster preparedness and forecasting system. *Internet of Things, 100076*. https://doi.org/10.1016/j.iot.2019.100076

[61] Yu, W., Dillon, T. S., Mostafa, F., Rahayu, W., & Liu, Y. (2019). A global manufacturing big data ecosystem for fault detection in predictive maintenance. *IEEE Transactions on Industrial Informatics*. https://doi.org/10.1109/TII.2019.2915846

[62] Fuller, Aidan & Fan, Zhong & Day, Charles. (2019). Digital Twin: Enabling Technology, Challenges and Open Research.

[63] Karanjkar, N., Joglekar, A., Mohanty, S., Prabhu, V., Raghunath, D., & Sundaresan, R. (2018). Digital twin for energy optimization in an SMT-PCB assembly line. In *2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)* (pp. 85–89). IEEE. https://doi.org/10.1109/IOTAIS.2018.8600830

[64] Gómez Berbis, J., & Amescua-Seco, A. (2019). SEDIT: Semantic digital twin

based on industrial IoT data management and knowledge graphs. In *Proceedings of the 2nd International Conference on Internet Science* (pp. 172–184). Springer. https://doi.org/10.1007/978-3-030-34989-9_14

[65] Heller, D., & Meierhofer, J. (2019, September 13). An architectural approach for service value creation with the digital twin. *2nd Smart Services Summit*, Zürich

[66] Makarov, V. V., Frolov, Y. B., Parshina, I. S., & Ushakova, M. V. (2019). The design concept of digital twin. In *2019 Twelfth International Conference "Management of Large-Scale System Development" (MLSD)* (pp. 1–4). IEEE. https://doi.org/10.1109/MLSD.2019.8911091

[67] Tao, F., Qi, Q., Wang, L., & Nee, A. Y. C. (2019). Digital twins and cyber–physical systems toward smart manufacturing and Industry 4.0: Correlation and comparison. *Engineering, 5*(4), 653–661. https://doi.org/10.1016/j.eng.2019.01.014

[68] Lu, Y., Liu, C., Kevin, I., Wang, K., Huang, H., & Xu, X. (2020). Digital twin-driven smart manufacturing: Connotation, reference model, applications, and research issues. *Robotics and Computer-Integrated Manufacturing, 61*, 101837. https://doi.org/10.1016/j.rcim.2019.101837

[69] Souza, V., Cruz, R., Silva, W., Lins, S., & Lucena, V. (2019). A digital twin architecture based on the industrial Internet of Things technologies. In *2019 IEEE International Conference on Consumer Electronics (ICCE)* (pp. 1–2). IEEE. https://doi.org/10.1109/ICCE.2019.8662081

[70] Santamaria, A. F., Raimondo, P., Tropea, M., De Rango, F., & Aiello, C. (2019). An IoT surveillance system based on a decentralized architecture. *Sensors, 19*(6), 1469. https://doi.org/10.3390/s19061469

[71] Spezzano, G. (2019). Editorial: Special issue Swarm Robotics. *Applied Sciences, 9*(7), 1474. https://doi.org/10.3390/app9071474

[72] Costa, L. C. P., Rabaey, J., Wolisz, A., Rosan, M., & Zuffo, M. K. (2015). Swarm OS control plane: An architecture proposal for heterogeneous and organic networks. *IEEE Transactions on Consumer Electronics, 61*(4), 454–462. https://doi.org/10.1109/TCE.2015.7382337

[73] Priyan, M. K., Gandhi, U., Manogaran, G., Sundarasekar, R., Chilamkurti, N., & Varatharajan, R. (2018). Ant colony optimization algorithm with Internet of Vehicles for intelligent traffic control system. *Computer Networks, 144*, 154–162. https://doi.org/10.1016/j.comnet.2018.07.001

[74] Costa, L., Ccori, P., Corazza, G., Guinezi, M., Fedrecheski, G., & Zuffo, M. (2019). Swarm Assistant: An intelligent personal assistant for the Swarm. In *2019 IEEE International Conference on Consumer Electronics (ICCE)* (pp. 1–2). IEEE. https://doi.org/10.1109/ICCE.2019.8662010

[75] Kang, S., Park, J., & Chung, K. (2019). An MQTT-based context-aware autonomous system in oneM2M architecture. *IEEE Internet of Things Journal, 6*(6), 10586–10594. https://doi.org/10.1109/JIOT.2019.2919971

[76] Girma, A., et al. (2020). IoT-enabled autonomous system collaboration for disaster-area management. *IEEE/CAA Journal of Automatica Sinica, 7*(5), 1249–1262. https://doi.org/10.1109/JAS.2020.1003291

[77] Adams, A., & Adam, B. (2017). *The twelve-factor app*. Retrieved August 2023, from https://12factor.net/

[78] Claeys, T., Rousseau, F., & Tourancheau, B. (2017). Securing complex IoT platforms with token-based access control and authenticated key establishment. In *2017 International Workshop on Secure Internet of Things (SIoT)* (pp. 1–9). IEEE. https://doi.org/10.1109/SIoT.2017.00006

[79] Peyrott, S. E. (2017). *JWT handbook*. Auth0,

https://auth0.com/resources/ebooks/jwt-handbook

[80] Ethereum Foundation. (n.d.). *Ethereum for Java developers*. Retrieved from https://ethereum.org/en/developers/docs/programming-languages/java/

[81] Bhatt, S., Pham, T. K., Gupta, M., Benson, J., Park, J., & Sandhu, R. (2021). Attribute-based access control for AWS Internet of Things and secure industries of the future. *IEEE Access, 9*, 107200–107223. https://doi.org/10.1109/ACCESS.2021.3101218

[82] Karimibiuki, M., Aggarwal, E., Pattabiraman, K., & Ivanov, A. (2018). DynPolAC: Dynamic policy-based access control for IoT systems. In *2018 IEEE 23rd Pacific Rim International Symposium on Dependable Computing (PRDC)* (pp. 161–170). IEEE. https://doi.org/10.1109/PRDC.2018.00027

[83] Hassija, V., Chamola, V., Gupta, V., Jain, S., & Guizani, N. (2021). A survey on supply chain security: Application areas, security threats, and solution architectures. *IEEE Internet of Things Journal, 8*(8), 6222–6246. https://doi.org/10.1109/JIOT.2020.3025775

[84] Chang, V., Baudier, P., Zhang, H., Xu, Q., Zhang, J., & Arami, M. (2020). How blockchain can impact financial services: The overview, challenges and recommendations from expert interviewees. *Technological Forecasting and Social Change, 158*, 120166. https://doi.org/10.1016/j.techfore.2020.120166

[85] He, Q., Lin, H., Xiao, F., Hu, J., & Wang, X. (2021). Blockchain-based access control model to preserve privacy for students' credit information. In *2021 17th International Conference on Mobility, Sensing and Networking (MSN)* (pp. 105–111). IEEE. https://doi.org/10.1109/MSN53354.2021.00030

[86] Premkumar, R., & Sathya, P. S. (2021). A blockchain-based framework for IoT security. In *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 409–413). IEEE.

https://doi.org/10.1109/ICCMC51019.2021.9418485

[87] David, S., & Canessane, A. (2021). A centralized blockchain-based data security system for electrical energy against attacks. In *2021 International Conference on Communication, Control and Information Sciences (ICCISc)* (pp. 1–4). IEEE. https://doi.org/10.1109/ICCISc52257.2021.9484898

[88] Mingxiao, D., Xiaofeng, M., Zhe, Z., Xiangwei, W., & Qijun, C. (2017). A review on consensus algorithm of blockchain. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 2567–2572). IEEE. https://doi.org/10.1109/SMC.2017.8123011

[89] Lorido-Botran, T., Miguel-Alonso, J., & Lozano, J. A. (2014). A review of auto-scaling techniques for elastic applications in cloud environments. *Journal of Grid Computing, 12*(4), 559–592. https://doi.org/10.1007/s10723-014-9314-7

[90] Butzin, B., Golatowski, F., & Timmermann, D. (2016). Microservices approach for the internet of things. In *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)* (pp. 1–6). IEEE. https://doi.org/10.1109/ETFA.2016.7733707

[91] Fowler, M. (n.d.). *Microservices guide*. Retrieved from http://martinfowler.com/microservices/

[92] Kulkarni, S., Bhagat, N., Fu, M., Kedigehalli, V., Kellogg, C., Mittal, S., Patel, J. M., Ramasamy, K., & Taneja, S. (2015). Twitter Heron: Stream processing at scale. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data* (pp. 239–250). ACM. https://doi.org/10.1145/2723372.2742788

[93] Abouee-Mehrizi, H., & Baron, O. (2016). State-dependent M/G/1 queueing systems. *Queueing Systems: Theory and Applications, 82*(1–2), 121–148.

https://doi.org/10.1007/s11134-015-9465-1

[94] Butzin, B., Golatowski, F., & Timmermann, D. (2016). Microservices approach for the internet of things. In *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)* (pp. 1–6). IEEE. https://doi.org/10.1109/ETFA.2016.7733707

[95] Fowler, M. (n.d.). *Microservices guide*. Retrieved from http://martinfowler.com/microservices/

[96] Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., & Byers, A. H. (2011). *Big data: The next frontier for innovation, competition, and productivity*. McKinsey Global Institute. https://www.mckinsey.com/~/media/mckinsey/business%20functions/mckinsey%20digital/our%20insights/big%20data%20the%20next%20frontier%20for%20innovation/mgi_big_data_exec_summary.ashx

[97] Weber, K., Otto, B., & Österle, H. (2009). One size does not fit all—A contingency approach to data governance. *ACM Journal of Data and Information Quality (JDIQ), 1*(1), Article 4. https://doi.org/10.1145/1515693.1515697

[98] Davenport, T. H., Barth, P., & Bean, R. (2012). How 'big data' is different. *MIT Sloan Management Review, 54*(1), 43–46. https://sloanreview.mit.edu/article/how-big-data-is-different/

[99] Beath, C., Becerra-Fernandez, I., Ross, J. W., & Short, J. (2012). Finding value in the information explosion. *MIT Sloan Management Review, 53*(4), 18–20. https://sloanreview.mit.edu/article/finding-value-in-the-information-explosion/

[100] Brandes, U., Lerner, J., & Pich, C. (n.d.). *Graph Markup Language (GraphML), Chapter 16*. University of Konstanz & Swiss Re. https://www.uni-konstanz.de/algo/publications/belp-g-13.pdf

[101] Brandes, U., & Group (n.d.). GraphML specification. Retrieved from http://graphml.graphdrawing.org/specification.html

[102] Purswani, P. (2021). Blockchain-based parametric health insurance. In *2021 IEEE Symposium on Industrial Electronics & Applications (ISIEA)* (pp. 1–5). IEEE. https://doi.org/10.1109/ISIEA51897.2021.9510001

[103] Allied Market Research. (2022, May). *Parametric insurance market by type, by industry vertical: Global opportunity analysis and industry forecast, 2021–2031*. Retrieved from https://www.researchandmarkets.com/reports/5640302/parametric-insurance-market-by-type-by-industry

[104] Poor queuing, servers blamed for mega-sale crashes. (2014, October). *The Hindu BusinessLine*. Retrieved from https://www.thehindubusinessline.com/info-tech/poor-queuing-servers-blamed-for-mega-sale-crashes/article64240096.ece

[105] Flipkart fumbles on the big day as server fails. (2014, October). *The Hindu BusinessLine*. Retrieved from https://www.thehindubusinessline.com/info-tech/flipkart-fumbles-on-the-big-day-as-server-fails/article64240270.ece

[106] Amazon down for thousands of users. (2022, December). *Mint*. Retrieved from https://www.livemint.com/companies/news/amazon-down-for-thousands-of-users-details-here-11670428739244.html

[107] Nita, S. L., & Mihăilescu, M. I. (2023). Elliptic curve–based query authentication protocol for IoT devices aided by blockchain. *Sensors,*

23(1371). https://doi.org/10.3390/s23031371

[108] Yousefnezhad, N., Malhi, A., Keyriläinen, T., & Främling, K. (2023). A comprehensive security architecture for information management throughout the lifecycle of IoT products. *Sensors, 23*(3236). https://doi.org/10.3390/s23063236

[109] Dehalwar, V., Kolhe, M. L., Deoli, S., & Jhariya, M. K. (2022). Blockchain-based trust management and authentication of devices in smart grid. *Cleaner Engineering and Technology, 8*, 100481. https://doi.org/10.1016/j.clet.2021.100481

[110] Kiourtis, A., Mavrogiorgou, A., & Kyriazis, D. (2023). A computer vision–based IoT data ingestion architecture supporting data prioritization. *Health Technology, 13*, 391–411. https://doi.org/10.1007/s12553-023-00748-0

[111] Naghib, A., Jafari Navimipour, N., Hosseinzadeh, M., et al. (2023). A comprehensive and systematic literature review on the big data management techniques in the Internet of Things. *Wireless Networks, 29*, 1085–1144. https://doi.org/10.1007/s11276-022-03177-5

[112] Huang, X., Fan, J., Deng, Z., Yan, J., Li, J., & Wang, L. (2021). Efficient IoT data management for geological disasters based on big data-turbocharged data lake architecture. *ISPRS International Journal of Geo-Information, 10*(11), 743. https://doi.org/10.3390/ijgi10110743

[113] Gkonis, P., Giannopoulos, A., Trakadas, P., Masip-Bruin, X., & D'Andria, F. (2023). A survey on IoT-edge-cloud continuum systems: Status, challenges, use cases, and open issues. *Future Internet, 15*(12), 383. https://doi.org/10.3390/fi15120383

[114] Bixio, L., Delzanno, G., Rebora, S., & Rulli, M. (2020). A flexible IoT stream processing architecture based on microservices. *Information, 11*(12), 565.

https://doi.org/10.3390/info11120565

[115] Hao, M., Qian, K., & Chau, S. C.-K. (2023). Privacy-preserving blockchain-enabled parametric insurance via remote sensing and IoT. *arXiv.* https://doi.org/10.48550/arXiv.2305.08384

[116] European Insurance and Occupational Pensions Authority. (2021, April 29). *Discussion paper on blockchain and smart contracts in insurance*. https://www.eiopa.europa.eu/consultations/discussion-paper-blockchain-and-smart-contracts-insurance_en

[117] Nadler, M., Bekemeier, F., & Schär, F. (2022). DeFi risk transfer: Towards a fully decentralized insurance protocol. *arXiv.* https://doi.org/10.48550/arXiv.2212.10308

[118] Cao, S., Johnson, H., & Tulloch, A. (2023). Exploring blockchain-based traceability for food supply chain sustainability: Towards a better way of sustainability communication with consumers. *Procedia Computer Science, 217*, 1437–1445

[119] Zhao, L., Shen, S., & Zhao, Z. (2024). Planning decentralized battery-swapping recharging facilities for e-bike sharing systems. *Sustainable Cities and Society, 101*, 105118

[120] Heeß, P., Rockstuhl, J., Körner, M. F., et al. (2024). Enhancing trust in global supply chains: Conceptualizing digital product passports for a low-carbon hydrogen market. *Electronic Markets, 34*, Article 10. https://doi.org/10.1007/s12525-024-00690-7

[121] Vysya, V. N., & Kumar, A. (2019). *Perspective: Blockchain adoption in financial services*. https://www.infosys.com/industries/financial-services/white-papers/documents/blockchain-adoption-financial-services.pdf

[122] Davis Polk & Wardwell LLP. (2016). *The custody services of banks* [White

paper]. Retrieved from https://www.davispolk.com/sites/default/files/20160728_tch_white_paper_the_custody_services_of_banks.pdf

[123] Mori, T. (2016). Financial technology: Blockchain and securities settlement. Journal of Securities Operations & Custody, 8(3), 208–227. https://ideas.repec.org/a/aza/jsoc00/y2016v8i3p208-227.html

[124] Šarac, M., Pavlović, N., Bacanin, N., Al-Turjman, F., & Adamović, S. (2021). Increasing privacy and security by integrating a blockchain secure interface into an IoT device security gateway architecture. *Energy Reports, 7*, 8075–8082. https://doi.org/10.1016/j.egyr.2021.07.078

[125] Chang, V., Baudier, P., Zhang, H., Xu, Q., Zhang, J., & Arami, M. (2020). How blockchain can impact financial services: The overview, challenges and recommendations from expert interviewees. *Technological Forecasting and Social Change, 158*, Article 120166

[126] Choo, K.-K. R., Yan, Z., & Meng, W. (2020). Blockchain in Industrial IoT Applications: Security and Privacy Advances, Challenges and Opportunities. *IEEE Transactions on Industrial Informatics*, *16*(6), 4119 - 4121. https://doi.org/10.1109/tii.2020.2966068

[127] Shi, G., Hao, H., Lei, J., & Zhu, Y. (2021). Application security system design of Internet of Things based on blockchain technology. In *2021 International Conference on Computer, Internet of Things and Control Engineering (CITCE)* (pp. 134–137). https://doi.org/10.1109/CITCE54390.2021.00033

[128] Pal, S., Dorri, A., & Jurdak, R. (2022). Blockchain for IoT access control: Recent trends and future research directions. *Journal of Network and Computer Applications*, Article 103371. https://doi.org/10.1016/j.jnca.2022.103371

[129] Agarwal, V., & Pal, S. (2020). Blockchain meets IoT: A scalable architecture

for security and maintenance. In *2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)* (pp. 53–61). https://doi.org/10.1109/MASS50613.2020.00017

[130] Ma, M., Shi, G., & Li, F. (2019). Privacy-oriented blockchain-based distributed key management architecture for hierarchical access control in the IoT scenario. *IEEE Access, 7*, 34045–34059. https://doi.org/10.1109/ACCESS.2019.2904042

[131] He, Q., Lin, H., Xiao, F., Hu, J., & Wang, X. (2021). Blockchain-based access control model to preserve privacy for students' credit information. In *2021 17th International Conference on Mobility, Sensing and Networking (MSN)* (pp. 105–111). https://doi.org/10.1109/MSN53354.2021.00030

[132] Joshi, S., Pise, A. A., Shrivastava, M., Revathy, C., Kumar, H., Alsetoohy, O., & Akwafo, R. (2022). Adoption of blockchain technology for privacy and security in the context of Industry 4.0. *Wireless Communications and Mobile Computing, 2022*, Article 4079781. https://doi.org/10.1155/2022/4079781

[133] Li, P., Feng, W., Yan, Z., Li, Y., Zhou, X., & Shimizu, S. (2021). Privacy preservation in permissionless blockchain: A survey. Digital Communications and Networks, 7(3), 295–307. https://doi.org/10.1016/j.dcan.2020.05.008

[134] Gai, K., Wu, Y., Zhu, L., Zhang, Z., & Qiu, M. (2020). Differential Privacy-Based Blockchain for Industrial Internet-of-Things. *IEEE Transactions on Industrial Informatics*, *16*(6), 4156-4165. Article 8874972. https://doi.org/10.1109/TII.2019.2948094

[135] Lu, Yunlong & Huang, Xiaohong & Dai, Yueyue & Maharjan, Sabita & Zhang, Yan. (2019). Blockchain and Federated Learning for Privacy-Preserved Data Sharing in Industrial IoT. IEEE Transactions on Industrial Informatics. PP. 1-1. 10.1109/TII.2019.2942190.

Hirew Dutta
8/7/25

Parama Bhaumik
8/7/2025

Associate Professor
Dept. of Information Technology
JADAVPUR UNIVERSITY
Block-LB Plot-8, Sector-3
Salt Lake, Kolkata-700 106, India