EAI/Springer Innovations in Communication and Computing

Muhammad Habib ur Rehman
Davor Svetinovic
Khaled Salah
Ernesto Damiani *Editors*

Trust Models for Next-Generation Blockchain Ecosystems

EAI
RESEARCH MEETS INNOVATION

Springer

EAI/Springer Innovations in Communication and Computing

Editor's Note

The impact of information technologies is creating a new world yet not fully understood. The extent and speed of economic, life style and social changes already perceived in everyday life is hard to estimate without understanding the technological driving forces behind it. This series presents contributed volumes featuring the latest research and development in the various information engineering technologies that play a key role in this process.

The range of topics, focusing primarily on communications and computing engineering include, but are not limited to, wireless networks; mobile communication; design and learning; gaming; interaction; e-health and pervasive healthcare; energy management; smart grids; internet of things; cognitive radio networks; computation; cloud computing; ubiquitous connectivity, and in mode general smart living, smart cities, Internet of Things and more. The series publishes a combination of expanded papers selected from hosted and sponsored European Alliance for Innovation (EAI) conferences that present cutting edge, global research as well as provide new perspectives on traditional related engineering fields. This content, complemented with open calls for contribution of book titles and individual chapters, together maintain Springer's and EAI's high standards of academic excellence. The audience for the books consists of researchers, industry professionals, advanced level students as well as practitioners in related fields of activity include information and communication specialists, security experts, economists, urban planners, doctors, and in general representatives in all those walks of life affected ad contributing to the information revolution.

Indexing: This series is indexed in Scopus, Ei Compendex, and zbMATH.

About EAI

EAI is a grassroots member organization initiated through cooperation between businesses, public, private and government organizations to address the global challenges of Europe's future competitiveness and link the European Research community with its counterparts around the globe. EAI reaches out to hundreds of thousands of individual subscribers on all continents and collaborates with an institutional member base including Fortune 500 companies, government organizations, and educational institutions, provide a free research and innovation platform.

Through its open free membership model EAI promotes a new research and innovation culture based on collaboration, connectivity and recognition of excellence by community.

More information about this series at http://www.springer.com/series/15427

Muhammad Habib ur Rehman • Davor Svetinovic
Khaled Salah • Ernesto Damiani
Editors

Trust Models for Next-Generation Blockchain Ecosystems

Springer

EAI
RESEARCH MEETS INNOVATION

Editors
Muhammad Habib ur Rehman
Department of Biomedical Engineering
& Imaging Sciences
King's College London
London, UK

Davor Svetinovic
Center for Cyber-Physical Systems
Khalifa University of Science
and Technology
Abu Dhabi, UAE

Khaled Salah
Center for Cyber-Physical Systems
Khalifa University of Science
and Technology
Abu Dhabi, UAE

Ernesto Damiani
SEcure Service-Oriented Architectures
Research (SESAR) Group
Università degli Studi di Milano
Crema (CR), Italy

ISSN 2522-8595 ISSN 2522-8609 (electronic)
EAI/Springer Innovations in Communication and Computing
ISBN 978-3-030-75106-7 ISBN 978-3-030-75107-4 (eBook)
https://doi.org/10.1007/978-3-030-75107-4

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

“It’s the time to create well-regulated, community-governed, privacy-preserving, secure, scalable, and inter-operable social computing systems for the next-generations.”

M. H. ur Rehman et al. – 2021

Preface

In 1991, Stuart Haber and W. Scott Stornetta envisioned the concept of a chain of blocks that could be cryptographically secured to enable tamper-resistant data storage. They proposed the concepts of hashing mechanisms and Merkle Tree to group a set of secure documents in a block and linearly connect it with other documents in their system. However, blockchain technology is massively being adopted after the introduction of Bitcoin cryptocurrency, introduced by Satoshi Nakamoto (an anonymous inventor of modern-day cryptocurrencies) in 2008. The blockchain ecosystem evolved after the introduction of transactions in 2012, smart contracts in 2015, and decentralized applications in 2018. Today, blockchain technology is being adopted and implemented in almost every type of personal, business, corporate, and government applications. Considering these trends, Gartner (one of the top market research firms) projected that, in 2030, $ 3.1 trillion dollars of value delivery will be enabled by blockchain ecosystems.

Blockchain enables distributed ledger technologies over consensus-based decentralized computing networks. However, the openness of blockchain technology stacks to multiple competing participants creates mistrust among all participants (i.e., end users, decentralized application developers, blockchain network operators, regulators, and the general public) in the blockchain ecosystems. Considering the fast growth in blockchain research, this book aims to study the trust issues in blockchain ecosystems with a broader perspective to cover the theoretical as well as applied aspects of blockchain technologies. Therefore, this book is structured into 10 chapters.

Müller et al., in Chap. 1, systematically explored a three-step-approach of how to build trust-aware decentralized applications using blockchain technologies. They used business process models with a trust layer as a tool to visualize and analyze trust issues of the decentralized applications' underlying process systematically as a first step. Secondly, different distributed ledger trust patterns were selected and used to mitigate these trust issues. Finally, the trust-enhanced decentralized applications were highlighted. This chapter helps system engineers to systematically create trust-aware decentralized applications with distributed ledgers.

Taimur and Bagdan, in Chap. 2, discussed the contemporary applications of blockchain technology in information auditing, by exploring aspects such as data recording, accuracy, verification, transparency, and overall value of a decentralized blockchain crypto-ledger for auditors. Opportunities for timeliness, completeness, and re-conciliation in appraising regulatory compliance of organizations employing blockchain-based contractual frameworks are also investigated. The authors reviewed the existing and anticipated challenges that blockchain applications pose to traditional regulatory compliance models and the inherent risks for businesses and stakeholders. Also, they highlighted the impact of operational concerns such as decentralized transactions, network complexity, transaction reversals, credential management, software quality, and human resources. Finally, they provided perspective on assurance complexities which are involved in transforming from proprietary to blockchain-based framework while adhering to IT control obligations dictated by three major auditing standards: Sarbanes Oxley Act (SOX), Control Objectives for Information Technologies (COBIT), and ISO/IEC 27001.

To explore participatory and experimental engagements with algorithmic governance, Denisa developed and tested a sandbox for near-future blockchain and satellite services (Lithopy) and presented her results in Chap. 3. The chapter summarized the insights from the final two 2020 workshops with 59 participants deciding on how to regulate a discriminatory blockchain smart contract. The participants expressed a preference for hybrid forms of off-chain and on-chain governance that used industry standards and audits to prevent biases and excesses. The sandbox environment increased the trust in a system through design and policy iterations that probe issues of power, stakes, and interests without compromising on innovation.

Numerous reviews of decentralized applications using blockchain have been published in recent years. However, these have predominantly focused on how applications map to the characteristics of underlying protocols and discuss privacy only in passing. Moritz et al., in Chap. 4, took a more focused perspective by analyzing recently proposed, prominent decentralized applications, categorizing them according to information privacy requirements. Their investigation reported that three dimensions emerged as key differentiators accessibility requirements (i.e., whether an application needs to be publicly accessible), hierarchichality requirements (i.e., the notion of different types of participants with fundamentally different permissions), and requirements on joint verifiability of transaction data (i.e., which subset of an application's data model needs to be verifiable by others). After grouping decentralized applications along these dimensions, it became clear that use cases with closed accessibility requirements are most dominant in the market today. These use cases can establish information privacy through governance, deliberate data modeling, and role-based access management.

Kashif et al., in Chap. 5, investigated the challenge of scalability in blockchain ecosystems to enable the reader to develop an in-depth understanding of scalability issues in blockchain to facilitate the development of scalable blockchains and their adoption in diverse application domains. They introduced scalability as well as its different dimensions such as vertical and horizontal scalability and identified specific parameters that can affect blockchain scalability. Their empirical investigation includes diverse scenarios highlighting the role of parameters such as block size, block generation rate, and transaction processing speed to achieve scalable blockchains.

In a blockchain context, interoperability is the ability to connect multiple networks, thus enabling the exchange of assets, the invocation of smart contracts, and the verification of data, all while ensuring consistency between systems. However, in reality, most of the existing blockchain networks operate in a standalone environment, isolated from other blockchain networks. This causes a lack of communication that further leads to restrictions imposed on data, that is, preventing it from transmitting freely to and from various blockchains regardless of the underlying infrastructure. Manar et al., in Chap. 6, discussed the methodologies used in the available solutions and compared several of their characteristics, including throughput, average block confirmation time, and consensus mechanism. Further, they presented the projects currently implementing these protocols and their future directions.

Dounia et al., in Chap. 7, developed a blockchain-based incident reporting system showing how incident data can be reported and shared through a secure and trusted distributed ledger. Further, they presented algorithms that depict the various interactions among the stakeholders in the reporting network. Finally, they demonstrated the feasibility of their proposed solution through the cost and security analysis while ensuring security, integrity, transparency, and traceability among stakeholders. This study also discussed potential challenges and suggested future research to provide significant insights for the implementation of the blockchain-based incident reporting system in practice.

Meriem et al., in Chap. 8, proposed and implemented a secure firmware update delivery mechanism for IoT devices. To assure a secure firmware update process, the proposed solution was deployed in a blockchain network. This mechanism ensured that the firmware version of the IoT device is verified, while also validating the integrity of the file itself and then downloading the latest version of the firmware update. All these tasks were performed securely through a Hyperledger blockchain network. The main objective of this solution was to mitigate attacks on the firmware update process by ensuring that IoT firmware is up to date and that it has not been modified either during the transfer process or as it is installed on the IoT device.

The absence of a blockchain security control framework is a challenge faced by governments seeking to adopt best security practices for blockchain technologies. Ahmed et al., in Chap. 9, studied the potential of cybersecurity threats associated with establishing and operating a private blockchain consortium for governments. Accordingly, a blockchain security control framework was proposed to guide the government entities to address the risks related to private blockchain consortium.

As the majority of blockchain networks operate in environments that are isolated and disconnected from each other, the inter-blockchain communication and interoperability become prime considerations. Finally, Ilham et al., in Chap. 10, focused on designing and implementing inter-blockchain communication between Hyperledger Fabric networks. Also, they presented the performance evaluation of the proposed scheme in terms of execution time, throughput, and latency. The results showed that the proposed solution reduced the overall performance of Hyperledger Fabric across all measured metrics. However, the overall proposed solution performance was acceptable, given the achieved interoperability and connectivity.

Although the primary audience for this book is the blockchain research community, we tried to make the content accessible to the more general audience including students, researchers, and practitioners in wider computer science and engineering domains. Since this is the first contributed research book that addresses the trust issues in blockchain ecosystems, we aim to include more important and active research topics in the future editions of this book. Finally, we would like to thank and acknowledge the efforts of all contributors including authors, reviewers, and editorial staff for putting their untiring efforts into making this project a success.

Abu Dhabi, UAE Muhammad Habib ur Rehman

Abu Dhabi, UAE Davor Svetinovic

Abu Dhabi, UAE Khaled Salah

Abu Dhabi, UAE Ernesto Damiani
January 2021

Contents

1 **Engineering Trust-Aware Decentralized Applications with Distributed Ledgers** 1
Marcel Müller, Nadine Ostern, Sandro Rodriguez Garzon, and Axel Küpper

2 **Perspectives on Auditing and Regulatory Compliance in Blockchain Transactions** 37
Taimur Bakhshi and Bogdan Ghita

3 **Exploratory RegTech: Sandboxes Supporting Trust by Balancing Regulation of Algorithms with Automation of Regulations** 67
Denisa Reshef Kera

4 **Information Privacy in Decentralized Applications** 85
Moritz Platt, Ruwan J. Bandara, Andreea-Elena Drăgnoiu, and Sreelakshmi Krishnamoorthy

5 **An Empirical Investigation of Blockchain Scalability** 105
Kashif Mehboob Khan, Junaid Arshad, Muhammad Mubashir Khan, and Muhammad Hassan Nasir

6 **Interoperability Among Heterogeneous Blockchains: A Systematic Literature Review** 135
Manar Abu Talib, Sohail Abbas, Qassim Nasir, Fatima Dakalbab, Takua Mokhamed, Khawla Hassan and Khaldoun Senjab

7 **Blockchain-based Incident Reporting System for Patient Safety and Quality in Healthcare** 167
Dounia Marbouh, Mecit Can Emre Simsekler, Khaled Salah, Raja Jayaraman, Samer Ellahham

8 Implementation of Hyperledger-Based Secure Firmware Update Delivery for IoT Devices 191
Meriem Bettayeb, Qassim Nasir and Manar Abu Talib

9 Blockchain Security Framework for Government Private Blockchain Consortium 225
Ahmed Alketbi, Manar Abu Talib, and Qassim Nasir

10 Toward Inter-Blockchain Communication Between Hyperledger Fabric Platforms 251
Ilham Qasse, Manar Abu Talib, and Qassim Nasir

Index 273

Editors and Contributors

About the Editors

Muhammad Habib ur Rehman Center for Cyber-Physical Systems, Khalifa University of Science and Technology, Abu Dhabi, UAE
Department of Biomedical Engineering & Imaging Sciences, King's College London, London, UK

Davor Svetinovic Center for Cyber-Physical Systems, Khalifa University of Science and Technology, Abu Dhabi, UAE

Khaled Salah Center for Cyber-Physical Systems, Khalifa University of Science and Technology, Abu Dhabi, UAE

Ernesto Damiani Center for Cyber-Physical Systems, Khalifa University of Science and Technology, Abu Dhabi, UAE
SEcure Service-Oriented Architectures Research (SESAR) Group, Università degli Studi di Milano, Crema (CR), Italy

Contributors

Ahmed Alketbi University of Sharjah, Sharjah, UAE

Axel Küpper Technische Universität Berlin, Berlin, Germany

Andreea-Elena Drăgnoiu University of Bucharest, Bucharest, Romania

Bogdan Ghita University of Plymouth, Plymouth, UK

Denisa Reshef Kera University of Salamanca, Salamanca, Spain

Dounia Marbouh Khalifa University of Science and Technology, Abu Dhabi, UAE

Fatima Dakalbab University of Sharjah, Sharjah, UAE

Ilham Qasse University of Sharjah, Sharjah, UAE

Junaid Arshad Birmingham City University, Birmingham, UK

Khaldoun Senjab University of Sharjah, Sharjah, UAE

Khaled Salah Khalifa University of Science and Technology, Abu Dhabi, UAE

Kashif Mehboob Khan NED University of Engineering & Technology, Karachi, Pakistan

Khawla Hassan University of Sharjah, Sharjah, UAE

Manar Abu Talib University of Sharjah, Sharjah, UAE

Marcel Müller Technische Universität Berlin, Berlin, Germany

Mecit Can Emre Simsekler Khalifa University of Science and Technology, Khalifa, UAE

Meriem Bettayeb University of Sharjah, Sharjah, UAE

Moritz Platt King's College London, London, UK

Muhammad Hassan Nasir NED University of Engineering & Technology, Karachi, Pakistan

Muhammad Mubashir Khan NED University of Engineering & Technology, Karachi, Pakistan

Nadine Ostern Universität Marburg, Marburg, Germany

Qassim Nasir University of Sharjah, Sharjah, UAE

Raja Jayaraman Khalifa University of Science and Technology, Khalifa, UAE

Ruwan J. Bandara University of Wollongong, Wollongong, NSW, Australia

Samer Ellahham Khalifa University of Science and Technology, Khalifa, UAE

Sandro Rodriguez Garzon Technische Universität Berlin, Berlin, Germany

Sohail Abbas University of Sharjah, Sharjah, UAE

Sreelakshmi Krishnamoorthy Birla Institute of Technology and Science, Pilani, India

Taimur Bakhshi National University of Computer and Emerging Sciences, Karachi, Pakistan

Takua Mokhamed University of Sharjah, Sharjah, UAE

Acronyms

ACL	Access control list
AI	Artificial intelligence
API	Application programming interface
BFT	Byzantine fault tolerance
BGR	Block generation rate
BNA	Business network application
BPM	Business process management
BPMN	Business process model and notation
BSI	British Standards Institute
CIB	Corporate information base
CLI	Command line interface
COBIT	Control objectives for information technologies
CQC	Care quality commission
CRC	Cyclic redundancy check
CSR	Corporate social responsibility
DAG	Direct acyclic graph
DAO	Decentralized autonomous organization
dApp	Decentralized application
DDoS	Distributed denial of service
DHT	Distributed hash table
DID	Decentralized identifiers
DLT	Distributed ledger technology
EA	Ethereum address
EHR	Electronic health record
ERP	Enterprise resource planning
FW	Firmware
GDPR	General data protection regulation
HIE	Health information exchange
HSM	Hardware security module
HTLC	High-time lock contracts
hTMM	Hybrid threat modeling method

IBC	Inter-blockchain communication
IEC	International electrotechnical commission
IoT	Internet of Things
IPFS	Inter-plenary file system
IRS	Incident reporting system
IS	Information system
ISA	International standards on auditing
ISM	Information security management
ISO	International Standardization Organization
ISR	Information security regulation
KPI	Key performance indicator
LN	Lightening network
LRMS	Local reporting management system
MDSE	Model-driven software engineering
MHRA	Medicines and Healthcare Products Regulatory Agency
MITM	Man in the middle
ML	Machine learning
MoH	Ministry of health
MPOS	Multi-token proof of stake
NHS	National Health Service
NIST	National Institute of Standards and Technology
NRLS	National Reporting and Learning System
OTA	Over the air
P2P	Peer-to-peer
PASTA	Process for attack simulation and threat analysis
PBFT	Practical Byzantine fault tolerance
PCN	Payment channel network
PET	Privacy enhancing technology
PKI	Public key infrastructure
QAR	Quality assessment rule
SLR	Systematic literature review
PCI	Payment card industry
PHE	Public Health England
PoW	Proof of work
SCBPE	Smart contract business process engine
SOX	Standards Sarbanes Oxley
SSL	Secure socket layer
TCP	Transmission control protocol
TEE	Trusted execution environment
TeTT	Trust-enhancing technologies and techniques
TLS	Transport layer security
TPS	Transactions per second
WHO	World Health Organization
ZKP	Zero knowledge proofs

Chapter 1
Engineering Trust-Aware Decentralized Applications with Distributed Ledgers

Marcel Müller, Nadine Ostern, Sandro Rodriguez Garzon, and Axel Küpper

1.1 Introduction

Decentralized Applications (dApps) are pieces of software operated by different independent organizations and interact with each other over the Internet to achieve a common business goal. Examples are purchasing systems that interconnect buyers and sellers to conduct an order process or identity management systems used to exchange a verifiable credential about a person, for example, her name or degree, between this person, a party issuing this credential, and another verifying it. In contrast to central applications like web services, which rely on a back-end infrastructure that provides the main business logic and is operated by a service provider, dApps directly communicate in a peer-to-peer (P2P) fashion. The business logic either entirely resides in the peers or is partially outsourced to a decentralized back-end infrastructure maintained by hundreds or thousands of decentralized computing nodes, such as a distributed hash table (DHT) or smart contracts executed on a distributed ledger [1].

dApps are thus a counter-design to centralized platforms provided by large and monopoly-like Internet companies, which often pursue data-centric business models and bind their users through lock-in effects. Users do not influence the business logic that centralized platforms execute "behind the scenes." This leads to different *trust*

M. Müller (✉) · S. R. Garzon · A. Küpper
Service-Centric Networking, Technische Universität Berlin and Telekom Innovation Laboratories, Berlin, Germany
e-mail: marcel.mueller@tu-berlin.de; sandro.rodriguezgarzon@tu-berlin.de; axel.kuepper@tu-berlin.de

N. Ostern
Universität Marburg, Marburg, Germany
e-mail: ostern@staff.uni-marburg.de

M. H. ur Rehman et al. (eds.), *Trust Models for Next-Generation Blockchain Ecosystems*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-75107-4_1

concerns for users. For instance, users of these platforms often lose control over their data generated when using their central applications; they cannot determine the location of data storage, nor are they in control about how their data is processed or with which external actors it is shared. Furthermore, lock-in effects occur in the form of missing or insufficient data portability; when deciding to switch to another service provider, a user's data is either lost or needs to be manually transferred in a cumbersome process. Therefore, dApps are seen as a key blueprint of the future Internet that helps to overcome stakeholder-driven platforms, improve competition, and preserve humans' privacy.

Despite the drawbacks above of the centralized platforms, they are forced to operate according to specific rules in many countries due to legal requirements like the General Data Protection Regulation (GDPR) [2] in Europe. These legal frameworks, as well as the terms and conditions issued by a provider, guarantee a certain level of trust that the service provider will act and behave predictably when handling a user's data and executing tasks during service execution. Therefore, service providers often see themselves as trusted intermediaries, which, in many cases, is part of their value proposition. However, the trust-building does not overcome the disadvantages associated with the use of central applications; the user must accept the provider's conditions or refrain from using the service.

In the absence of a service provider, dApps by nature have no trust-building mechanisms at all. To overcome this problem, the software of many dApps is published as open-source so that the broad public can review and check it for flaws, misuse of data, vulnerabilities, or uncertainties that happened during the design phase. However, software surveillance does not prevent the violation of trust when humans use or misuse this software. They can pretend to be someone else, manipulate data, deny transactions, and many more. Therefore, distributed ledgers have recently been introduced as a decentralized trusted intermediary between peers in many dApps [3].

Distributed ledgers have their origins in the Bitcoin blockchain [4], which is used for peer-to-peer payment transactions without a central bank and preventing double-spending and money printing. Core concepts are a distributed database maintained by each participating node, a consensus protocol applied between the nodes for approving transactions, hashing for securing the ledger against manipulations, and public/private key cryptography. Bitcoin has led to universal distributed ledgers like Ethereum [1] that secure all types of transactions, monetary and non-monetary, and are meanwhile widely used in many domains like finance [5], supply chain management [6], and health [7], to name only a few.

Therefore, today's distributed ledgers are seen as the leading enabler for designing trustful dApps and are regarded as indispensable for their future broad adoption and success. However, the mere integration of a distributed ledger into the transactions occurring between peers is not sufficient to mitigate trust concerns in a dApp per se. Instead, developers are exposed to many design alternatives when developing dApps, for example, on-chain versus off-chain data storage, authentication versus anonymity, or finding a reasonable balance between performance and costs.

We argue that dApps require a trust-by-design approach and propose a trust framework in this chapter that is based on business process models. Business process models have been introduced initially to describe recurring activities within an enterprise. A business process model's core ingredients are activities, gateways, and events that are interconnected by control flows that reflect the temporal sequence of activities. Business processes of different organizations can be interconnected via message exchanges, which result in collaborative business processes. The resulting pattern resembles the decentralized character of dApps; a business process model structures the life cycle of a peer, and it communicates in a collaborative business process with the other peers of the dApp. Thus, we use business process models as a tool for model-driven development to identify trust issues in the life cycle of a dApp and include trust patterns based on distributed ledgers where needed.

This chapter's remaining part introduces a three-step approach for designing trustful dApps based on distributed ledgers and business process models. The next section gives an overview of the theoretical foundations needed for the three-step approach. Afterward, the steps are discussed in detail. Section 1.4 explains how to identify trust issues with trust mining. Section 1.5 builds on that and illustrates how to mitigate trust issues in dApps using distributed ledger *trust patterns*. Afterward, Sect. 1.6 elaborates on unique challenges in the implementation phase of dApps with distributed ledgers. Concluding, Sect. 1.7 discusses the impacts and organizational factors for the trust-aware utilization of dApps.

1.2 Basic Approach

With the advancing digital economy, multi-sided platforms evolved that connect partners who do not know each other and allow them to engage in business together. The platform itself acts as a trust-enhancing intermediary, which increases the partners' confidence in the adherence to agreements. For example, Kickstarter [8] is a crowdfunding platform. It enables users to create campaigns to collect funds for manufacturing a specific product. The campaign creator defines a particular funding goal that needs to be reached until a particular day. Campaign backers invest their money in the project and receive the product after the funding goal is reached and the product has been produced. In this case, Kickstarter acts as an intermediary between the campaign owner and the backers. Through legal frameworks, the platform ensures that the campaign creator, for instance, delivers after the funding goal has been reached or refunds the money to all backers, in case the funding goal has not been reached until its due day. Kickstarter also provides a user interface for campaign owners and backers, payment methods, escrow services, dispute management, and marketing services. Therefore, the platform charges a fee of about 8%[1] of the funded amount.

[1] https://www.kickstarter.com/help/fees?ref=faq-basics_fees.

Centralized platforms like Kickstarter or large e-commerce platforms like Amazon aim to create *lock-in effects* for their users. Lock-in effects originate from a large one-sided value generation from the platform to its customers. For instance, Kickstarter provides a variety of different services to campaign creators, including payment processing, escrow services, reputation systems, marketing through the wide reach of the platform, and managed conflict resolution. The platform provides these services conveniently through a single user interface to the customer. If the campaign creators want to change to another platform for any reason, a complex migration process would be needed. While data may be migrated in a cumbersome process to the new platform, the campaign creator's reputation can hardly be transferred. The reach and the value generated from the platform itself as a marketing instrument create unique benefits specific to the platform. These lock-in effects lead to the platform having much power over prices or practices, while the users have no other choice than to accept the terms. If Kickstarter, for example, decides to increase its platform fees to 10%, campaign creators can either accept these new terms or leave the platform. They would lose the vast range of services and have to overcome the challenges above. In case the migration barrier is high enough, the lock-in effect may lead macro-economically to a *monopoly*-like market structure.

The dApps are an alternative to centralized, monopolistic platforms that often avoid intermediaries [9]. The business value is generated without creating lock-in effects. Typically, business interactions in a dApp happen in a P2P fashion. One major challenge that dApps face is the lack of trust in the interactions. In the case of a decentralized crowdfunding application, backers have to transfer their money directly to the campaign creator. Without the trusted intermediary, they have no confidence that the funds will be used to create the product or whether the funds will be returned if the funding goal is not achieved. Hence, we assume that initially, in every dApp, there is *zero trust* present.

"When we say we trust someone or that someone is trustworthy, we implicitly mean that the probability that he will perform an action that is beneficial or at least not detrimental to us is high enough for us to consider engaging in some form of cooperation with him" is a popular definition of trust by Gambetta [10]. The definition shows with its notion of probability that trust only becomes relevant in situations where *uncertainty* exists. In dApps, where different organizations/parties interact, different uncertainties regarding various *trust concerns* exist because organizations act independently. A trust concern describes the subject of uncertainty. For example, in the crowdfunding case, it is uncertain to the backer whether the campaign creator will return the funds in case the funding goal is not reached.

We need to understand trust concerns in dApps better to improve their trustworthiness by design. Therefore, we propose to use *business process models* with the addition of a *trust layer*. Conceptually, dApps and collaborative business processes are two sides of the same coin, as illustrated in Fig. 1.1. A collaborative business process is a process where different entities from different organizations are working together. A business process model is a formal generic abstraction of the business

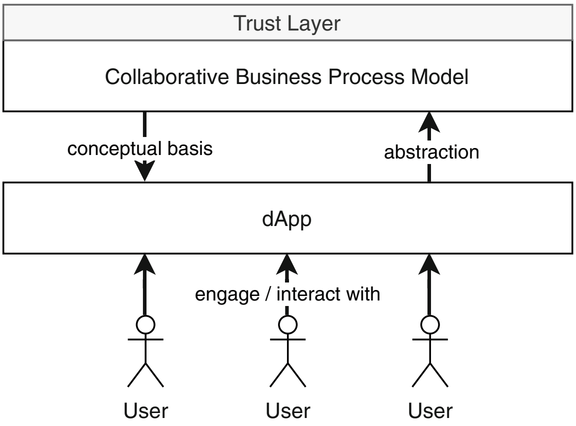


Fig. 1.1 Conceptual relationship between collaborative business processes, dApps, and their users

process it represents. A dApp can be visualized as a collaborative business process. To build trust-aware dApps, system engineers must understand the trust issues in the underlying collaborative processes.

The following sections give a short overview of business process models and the trust layer added to them. They are the conceptual basis for the method of building trust-aware dApps, as presented in this chapter.

1.2.1 Business Process Models

A process model is an artifact in the model-driven software engineering (MDSE) area of *Business Process Management (BPM)* [11]. Process models act as a “blueprint” for different instances (often called cases) of a business process. Conceptually, a process model is a graph that consists of nodes and edges. Nodes are activities, events, or gateways. Activities describe “units of work” in a process and are one of the basic building blocks of process models, while events represent a state of interest within the process. Examples include start events, end events, and error events in a process. Gateways express control flow logic, such as decisions or parallelism in a process. Edges connect different nodes and represent relationships, such as control flow or message flow. In addition to these essential process components, many process modeling languages also allow depicting data flows in a process, for example, with data inputs and data outputs. Collaborative business processes (often also called choreographies) illustrate how different organizations execute different parts of a shared process. Therefore, organizations are often depicted as swim-lanes in a business process model.

For better illustration, Fig. 1.2 illustrates decentralized crowdfunding as a collaborative business process. This example will be utilized as a running example

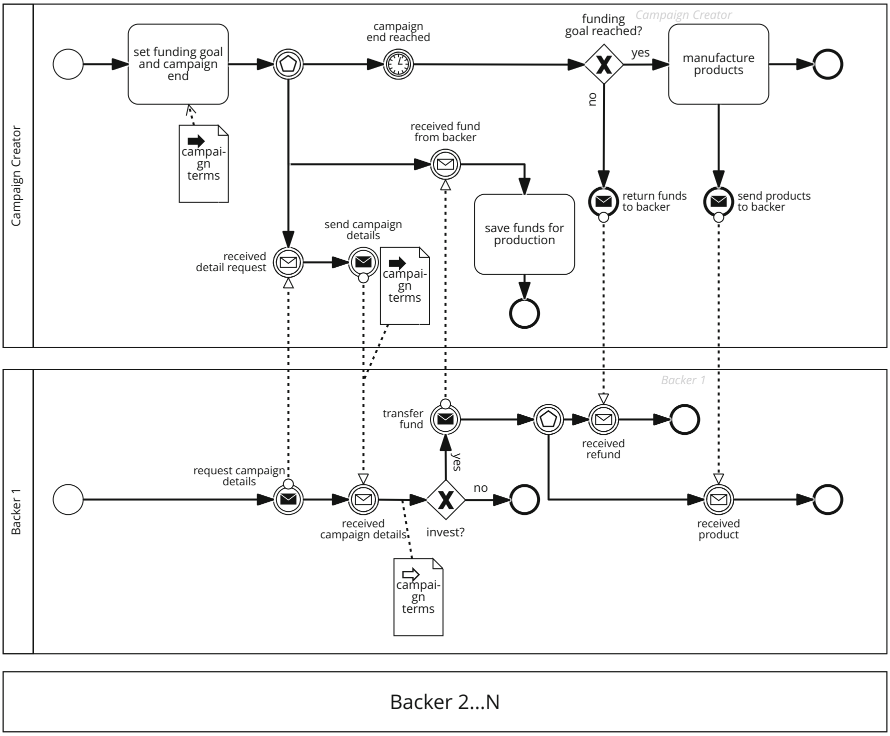


Fig. 1.2 Example business process for decentralized crowdfunding. The interactions between Backer 2..N and the campaign creator are the same as for Backer 1. The arrows were omitted for simplicity

throughout the whole chapter using business process model and notation (BPMN) 2.0 [12] as a business process modeling language. BPMN is utilized as an example, while in general, the presented concepts can be applied to any process modeling language with the described basic elements. The process starts with the campaign creator setting a funding goal, and the campaign ends together with the other terms of conditions. Afterward, backers request the detailed terms for the campaign from the creator. Based on the terms, the backers decide whether to invest in the campaign. If they do not want to engage in the campaign, the process terminates; otherwise, they transfer funds directly to the campaign owner. Every time the campaign creator receives a new fund from a backer, she saves it for the production costs. After the end of the campaign, the creator needs to assess whether the funding goal has been reached. If the goal has not been reached, then the campaign creator returns the funds to all backers, and the process ends. Otherwise, the manufacturing of the products starts. Afterward, the products get shipped to the backers and the process terminates.

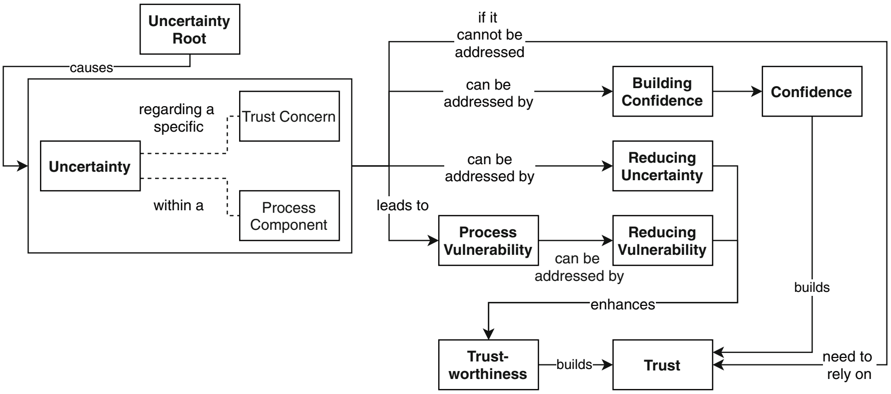


Fig. 1.3 A schematic representation of trust-related concepts for collaborative business processes. Reprint with permissions from [13]

1.2.2 Modeling Trust in Collaborative Business Processes

In the context of dApps, we always start with a *zero-trust assumption*. This means that we systematically describe and analyze the dApp as we would not trust anything. Therefore, business process models as a representation of the dApp's logic can be extended with a trust layer that deals with concepts such as uncertainty and trust [13]. Figure 1.3 shows a conceptualization of the basic trust-related concepts in the context of business processes.

Trust is a social concept. To analyze it on a technical level, we introduce a trust model that mainly focuses on uncertainties. Only if there are uncertainties in a process there is a need for trust after all. *Uncertainty roots* always cause uncertainties in a process. Roots of uncertainty may be employees, software, data, or the organization. Uncertainties are always defined regarding a specific *trust concern*. Trust concerns may be, for example, the integrity of an action or the non-repudiation of it. Uncertainties can be located at a certain *process component* in a business process model. In the crowdfunding example, one uncertainty can be described as follows: the campaign creator (uncertainty root) causes uncertainty regarding the integrity (trust concern) of the assessment of the funding goal (process component). This high-level schema describes in a structured way that there is the possibility that the campaign creator does something wrong, willingly or unwillingly. When uncertainties manifest, they lead to *process vulnerabilities*. Vulnerabilities describe the impact on a process when something undesired happens. For example, the campaign creator may not save the funds for manufacturing the product and instead uses the money to buy beer and disappear afterward. For the backers, this means that they lost all their funds without receiving anything in return. This monetary loss is one example of a process vulnerability.

Uncertainties and vulnerabilities in a process can be addressed in three different ways. Reducing process vulnerability (1) aims to lower the impact *when* something undesired happens in the process. For example, if there is any legal framework between the campaign creator and the backers, the backers can go to court if they do not receive a product and are not refunded. Reducing uncertainty (2), on the other hand, aims to decrease the probability *that* something undesired happens. One example is using an independent escrow service (for example, a smart contract on a distributed ledger) in the dApp. Building confidence (3) in a process aims to address trust issues by adding sources of trust to the process. For example, the backers may use a reputation system to analyze if the campaign creator started similar campaigns before. In case they see that this particular campaign creator was involved in 24 campaigns with 2340 satisfied backers, this increases their confidence. Building confidence does not modify uncertainties or their impact per se. This approach changes how the processes' trustworthiness is perceived. The three methods aim to build trust in a process.

1.2.3 Distributed Ledgers

Distributed ledgers offer unique trust-enhancing capabilities that can be used for trust-enhanced dApps. Distributed ledger technologies (DLTs) originate in the Bitcoin blockchain [4] that enables P2P payments without a central bank while preventing different problems like double-spending or uncontrolled money printing. Therefore, distributed ledgers maintain a distributed database among a network of independent peers. Changes to the database's state are made through transactions submitted to the peers. They ensure the transaction's validity and maintain consensus on the order of transactions. All transactions are cryptographically linked to make it practically impossible to modify or delete past transactions. A blockchain is a specific type of a distributed ledger, where transactions are grouped into blocks. While Bitcoin was the first inception of the concept, many other technical solutions emerged. More technically advanced ledgers, such as Ethereum [1], let users deploy programs called smart contracts. These programs can be triggered through transactions to modify more advanced logic that can trigger any state change to particular pieces of information in the distributed database. These smart contracts are the main building blocks of trust-aware dApps.

1.3 Trust-Aware dApps by Design

This section introduces a methodology to create trust-aware dApps by design. Initially, we have a dApp, and all users have *zero trust* in using it. The goal is to systematically analyze trust issues and mitigate them to create a trust-aware dApp. Therefore, we use business process models as a tool to better understand the

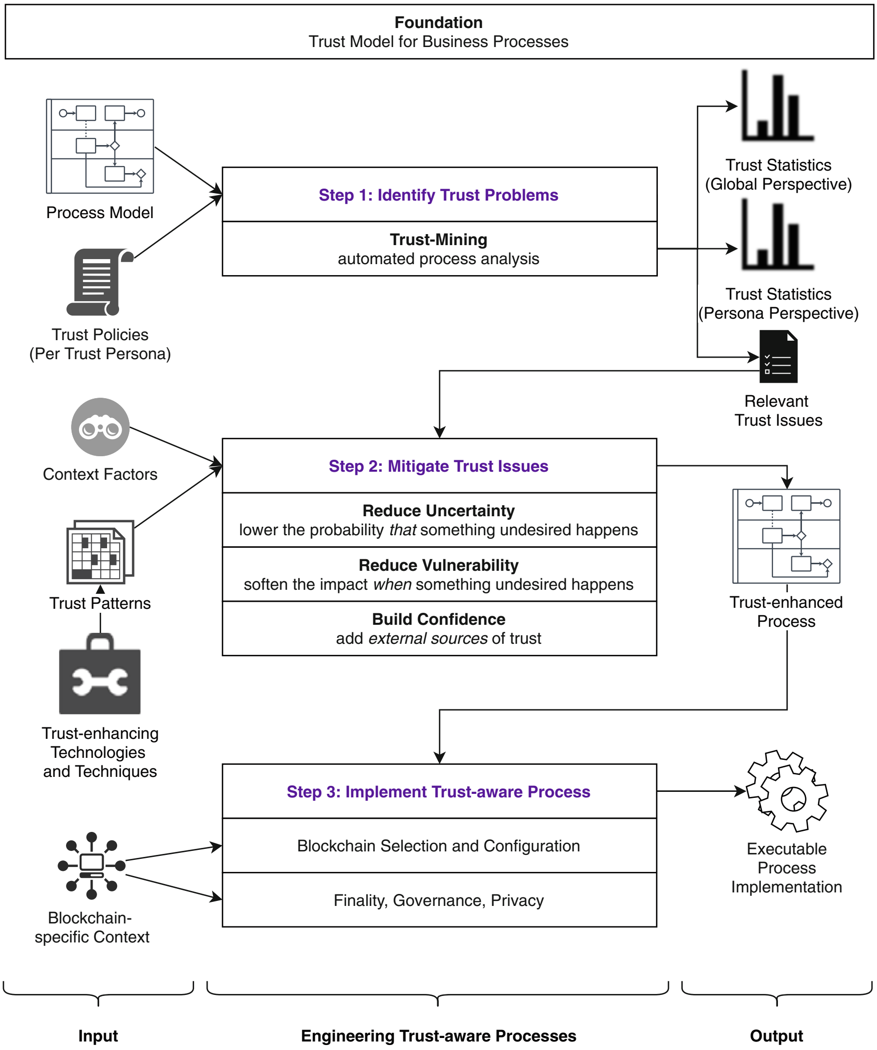

Foundation
Trust Model for Business Processes
Process Model
Trust Policies (Per Trust Persona)
Step 1: Identify Trust Problems
Trust-Mining
automated process analysis
Trust Statistics (Global Perspective)
Trust Statistics (Persona Perspective)
Relevant Trust Issues
Context Factors
Trust Patterns
Trust-enhancing Technologies and Techniques
Step 2: Mitigate Trust Issues
Reduce Uncertainty
lower the probability that something undesired happens
Reduce Vulnerability
soften the impact when something undesired happens
Build Confidence
add external sources of trust
Trust-enhanced Process
Step 3: Implement Trust-aware Process
Blockchain Selection and Configuration
Finality, Governance, Privacy
Blockchain-specific Context
Executable Process Implementation
Input
Engineering Trust-aware Processes
Output

Fig. 1.4 The three steps of building trust-aware dApps with distributed ledgers

trust issues of the dApp's underlying process in a three-step approach. This section discusses the three steps on a high level and localizes them in the general business process management lifecycle. Figure 1.4 depicts an overview of the three-step trust-by-design approach.

The trust model, as introduced in Sect. 1.2, acts as a foundation for all activities in the three-step approach. Therefore, the starting point for developing a trust-aware dApp is modeling the collaborative business process of the dApp.

The main objective of Step 1 is to first *identify trust problems* in the process model. These are of major concern because dApps using a distributed ledger might offer a way to mitigate trust problems. Step 1 uses the *trust mining* method that enables system engineers to systematically analyze trust issues in a collaborative business process. Therefore, trust mining uses the dApp's business process and trust policies of different trust personas as an input. The trust policies describe trust relationships between parties regarding their trust requirements. Trust personas represent a trust perspective onto the system. Trust mining produces a list of relevant trust issues as an output for each of the trust personas.

Step 2 aims to (re-)design the dApp's underlying process in a trust-aware fashion. It uses the list of identified relevant trust issues as its main input. Mitigating trust issues involves reducing uncertainty, reducing vulnerability, and building confidence. Therefore, *Trust-enhancing Technologies and Techniques (TeTTs)* are the foundation. Distributed ledgers are one of these TeTTs and are the focus of this chapter. But in general, there is also a variety of other TeTTs apart from DLTs. As a TeTT, a distributed ledger may be used in different *trust patterns* to mitigate certain trust issues. Trust patterns are design patterns that describe the trust-enhancing capabilities of a certain way of using a technology regarding the collaborative business process's trust model. Step 2 produces a trust-enhanced process model as an output.

Step 3 implements the trust-enhanced dApp based on the process model of Step 2 using a distributed ledger. Activities in Step 3 include system selection as well as coding and configuration of the dApp. Therefore, challenges specific to distributed ledger technologies have to be addressed. These include finality, governance, and privacy, along with all traditional challenges of implementing complex software systems.

The activities in building trust-aware dApps with distributed ledgers are based on the traditional phases of BPM projects. Figure 1.5 shows where the three-step method for building trust-aware collaborations with a distributed ledger attaches to the lifecycle. The *design and analysis phase* models the dApps business process and validates it with the real world process stakeholders. In this phase, Steps 1 and 2 take place. Identifying uncertainty can be carried out after process modeling and identification of trust policies. Step 2 is the design step that mitigates the trust issues on a conceptual level before implementing it. The *configuration phase* deals with the practical implementation of the business process. Choices on system and software components used for the implementation are made, and the process' implementation is tested. In this phase, Step 3 of building trust-aware collaborations enhances the traditional BPM lifecycle. Decisions specific to distributed ledger technologies, such as privacy and ledger configuration (public vs. private, permissioned vs. permissionless, and others), need to be taken into account during the implementation. The configuration phase produces an implemented version as an output that can be executed in the enactment phase. This part of the BPM lifecycle describes the execution of different instances of the process in new processes. The evaluation phase follows the enactment phase and analyzes the executed process instances. After this analysis, there may be additional insights in the process uncovered. Based

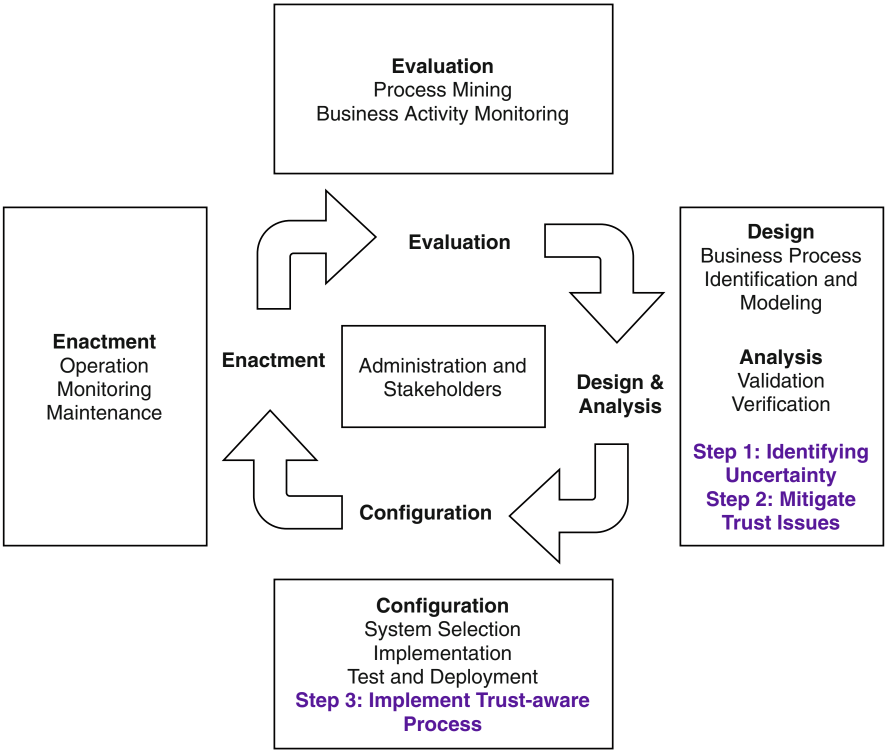


Fig. 1.5 Building trust-aware collaborations in the BPM lifecycle. Enhanced approach based on the model introduced by Weske [11]

on these insights, the next iteration of the BPM lifecycle starts with design and analysis again in an iterative manner.

1.4 Identifying Trust Issues

Before implementing any trust-aware dApp, it is necessary to understand which trust issues are relevant to different trust personas in the dApp's business process. Therefore, we introduce *trust mining* as the primary tool to systematically analyze the dApp's underlying business process concerning their trust issues. It is important to understand the relationship between trust and uncertainty. Trust can hardly be formalized since it is a social concept. On the other hand, uncertainties are concrete instances attributed to parts and roots in a process that implies the need for trust. Hence, trust mining focuses on analyzing uncertainties. The concept comprises four stages as visualized in Fig. 1.6. For trust mining, the *Trust Studio* tool is

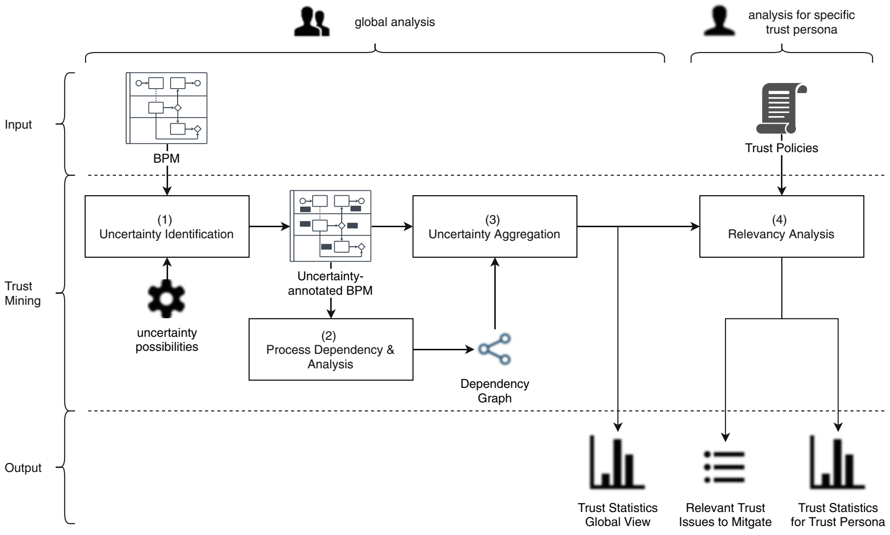

global analysis
analysis for specific trust persona
Input
BPM
Trust Policies
Trust Mining
(1) Uncertainty Identification
Uncertainty-annotated BPM
(3) Uncertainty Aggregation
(4) Relevancy Analysis
uncertainty possibilities
(2) Process Dependency & Analysis
Dependency Graph
Output
Trust Statistics Global View
Relevant Trust Issues to Mitgate
Trust Statistics for Trust Persona

Fig. 1.6 The workflow of trust mining

available as an online implementation[2] that guides the user through the four steps in an automated fashion. The next sections give a comprehensive overview of the four stages of trust mining: uncertainty identification, process dependency analysis, uncertainty aggregation, and relevancy analysis.

1.4.1 Trust Mining Stage 1: Uncertainty Identification

The first step of trust mining is to identify in general, “where” uncertainty exists in a process. Therefore, this step systematically traverses a business process model and annotates each element of the process with *uncertainty possibilities*.

An uncertainty possibility describes which *trust concern* is of relevance for a particular *process element* and what the *root of this uncertainty* is. This is a concrete instance of the trust model previously introduced [13]. As uncertainty roots, human resources, organization, activities, and software components are utilized. As trust concerns, the security properties of information security [14], integrity, confidentiality, availability, and non-repudiation are used together with performance and resilience from the domain of software quality [15]. These seven trust concerns are configuration parameters. If the user of the trust mining concept desires to use different trust concerns for uncertainty possibilities, this is possible. Using trust concerns as configuration parameters illustrates the extensibility of the approach:

[2] www.trust.snet.tu-berlin.de.

if the system engineer wants to examine other trust concerns, these can be easily added. Many other trust concerns can be of interest when analyzing trust concerns in certain fields, such as authenticity, safety, or resistance. Throughout this chapter, we utilize the aforementioned seven trust concerns for illustration. In the same way, also roots of uncertainty are parameters and can be changed or extended.

The set of uncertainty possibilities can be grouped by the focused process component of the uncertainty and can be illustrated as a table. This table is a configuration file and a static input to the uncertainty identification step. The following paragraphs give a semantic overview of the uncertainty possibility list. In addition to trust concerns, process components, and uncertainty roots, the last column states a generic question that describes the uncertainty.

1.4.1.1 Activity-Related Uncertainty

Activities represent units of work in a business process. They can consume data as input and deliver output. Table 1.1 gives an overview of relevant trust concerns. An activity's integrity trust concern describes the uncertainty on whether the activity is executed correctly (integer). In the running example, the integrity trust concern of the "save funds for production" activity describes the uncertainty on whether the funds are saved correctly. If the funds are directly used to buy beer and commit fraud,

Table 1.1 Activity-related uncertainties classified regarding trust concerns, their roots, and questions to ask regarding it

	Trust concern	Process component	Root	Question
Activity-related	Integrity	Activity	Activity, resource, software	Is the activity executed correctly?
	Confidentiality	Activity	Activity, resource, software	Is the internal execution of the activity only visible to authorized resources?
	Availability	Activity	Resource, software	Are the resources and systems, needed for the execution of the activity available?
	Non-repudiation	Activity	Activity, resource, software	When a certain activity is performed, is it non-repudiable?
	Performance	Activity	Activity, resource, software	Is the activity executed within the needed border of time and resource consumption?
	Resilience	Activity	Activity, resource, software	Can the activity handle the case of failure of one of the involved components?

this activity's integrity is violated. The confidentiality of activity is mostly centered around privacy. If the confidentiality of the "decide to invest in campaign" activity in the running example is not assured, other users might be able to reverse-engineer the backers' investment behavior. Availability is concerned with the resources and systems that are needed to execute an activity. If the website to transfer funds from the backer to the campaign creator is not available, the funds cannot be transferred, and the creator potentially misses out on funding. Non-repudiation of activity centers on whether or not the executing organization can later deny that the activity has been executed. An activity's performance trust concern is focused on whether the activity is executed within the desired resource consumption limits. This consumption can mean exceeding a time limit or other resources. For example, if the return of the funds to a backer takes too long, the backer might lose other investment opportunities due to the lack of funds in the meantime. The resilience of activity deals with the handling of failures. If there are problems in the "manufacture products" activity, the backer wants to receive compensation or other products. Most activity-related trust concerns have their roots in the activity and human resources or software in charge of their execution (Table 1.2).

1.4.1.2 Event-Related Uncertainty

Events express the reaching of a specific state of interest in a process. In the running example, a state of interest is that the campaign end has reached or received incoming funds. If an event's integrity is not assured, this means semantically

Table 1.2 Event-related uncertainties classified regarding trust concerns, their roots, and questions to ask regarding it

	Trust concern	Process component	Root	Question
Event-related	Integrity	Event	Activity, resource, software, data	Are the right events emitted from an activity?
	Confidentiality	Event	Activity, resource, software, data	Are the emitted events only visible to those who are authorized to see them?
	Availability	Event	Software, resource	Are the ways to emit an event available once the event occurs?
	Non-repudiation	Event	Activity, resource, software, data	If an event was emitted, can the emitter deny it?
	Performance	Event	Activity, resource, software	Is the evaluation and emission of events executed within the right time and resource consumption constraints?

that the event might not be triggered correctly. The confidentiality trust concern describes whether the event can only be detected by authorized entities. For example, if malicious actors can see that a specific backer transferred some funds, they can send them personalized fishing ads for a fake campaign. The availability trust concerns deal with the availability of all needed tools to trigger the event. If the running example's payment processing system is not available when the backer wants to transfer funds, the campaign creator cannot collect the funds and has a harder time reaching the funding goal. The non-repudiation trust concern becomes relevant when somebody has an interest in denying that an event occurred. For instance, if the campaign manager denies that she ever received funds from a certain backer to "steal" money, this is potentially a significant loss for the backer. The performance trust concerns of an event deal with the time it needs to trigger the event (Table 1.3).

1.4.1.3 Gateway-Related Uncertainty

Gateways in a business process model express decision points or represent parallel behavior. The process components for gateways include (parallel or exclusive) splits and joins. The integrity trust concern is focused on whether the decision is made correctly or not. For example, if the campaign manager assesses that the funding goal has not been reached, the backers might not get their product even though the supply of funds is enough. The confidentiality of the decision deals with the possibility that unauthorized entities can see the logic behind the evaluation of a decision. The availability trust concern at a gateway describes whether all resources,

Table 1.3 Gateway-related uncertainties classified regarding trust concerns, their roots, and questions to ask regarding it

	Trust concern	Process component	Root	Question
Gateway-related	Integrity	Gateway	Resource, software	Are the decisions made as desired?
	Confidentiality	Gateway	Activity, resource, software, data	Is the logic how decisions have been made only visible to authorized actors?
	Availability	Gateway	Resource, software, data	Are the tools needed to evaluate a gateway available?
	Non-repudiation	Gateway	Activity	If a certain gateway was evaluated, is it not deniable?
	Performance	Gateway	Activity, resource	Is the evaluation of gateways executed within the right time and resource consumption constraints?

Table 1.4 Process flow-related uncertainties classified regarding trust concerns, their roots, and questions to ask regarding it

	Trust concern	Process component	Root	Question
Process flow-related	Integrity	Sequence flow	Activity, resource	Are activities executed in the right order (as specified)? Is the process only terminating if it is supposed to be terminating?
	Integrity	Message flow	Activity, resource	Is the message flow between collaborators executed as intended? Is the process only terminating if it is supposed to be terminating?
	Confidentiality	Sequence flow	Activity, resource	Is the (internal) sequence flow and all associated data objects only visible to authorized actors?
	Confidentiality	Message flow	Activity, resource	Is the message flow and all associated data objects only visible to authorized actors?
	Availability	Sequence flow	Activity, resource	Is everything needed to coordinate activities intra-organizationally available?
	Availability	Message flow	Activity, resource	Is everything needed to coordinate activities inter-organizationally available?

tools, or software components needed to evaluate the gateway are available at the time when they are needed. Non-repudiation describes the trust concern that a decision cannot be reverted once it was made. The performance trust concern is centered on whether or not the gateway is evaluated on time (Table 1.4).

1.4.1.4 Process Flow-Related Uncertainty

Edges connect activities, events, and gateways in a process model. They express sequence flows (within an organization) or message flows (for inter-organizational workflows). Integrity regarding process flow is concerned with whether the control flow is executed in the correct order within an organization (sequence flow) or between different organizations (message flow). In the crowdfunding example, the evaluation of the funding goal must happen after the campaign ended. It is not desired that the production directly starts, regardless of the funding goal, because this might lead to situations where the manufacturer cannot deliver the products due to the lack of funds. In this case, the integrity of the sequence flow would not be ensured. The confidentiality trust concern is focused on whether or not the process flow can be seen only by authorized entities. Availability is concerned with whether tools needed to orchestrate the process are available at the desired time.

The discussed uncertainties can be aggregated into one singular uncertainty possibility list. This list is used as a configuration file in the first stage of trust mining. During this first stage, a trust mining software uses the uncertainty possibilities and annotates a process model iteratively. The annotation is done by traversing every process element and adding each uncertainty possibility relevant to the element, as defined in the tables above. The uncertainty-annotated process model is the input to the second step of trust mining.

1.4.2 Trust Mining Stage 2: Process Dependency Analysis

The first stage of trust mining analyzed uncertainties isolated at every process element. In contrast to that, the second stage examines relationship-related uncertainties between different process components and their organizations.

Therefore, trust mining introduces a *data dependency graph* and a *message dependency graph*. The data dependency graph illustrates relationships between the process collaborators, where one collaborator consumes data which another one produces. The dependency shows the potential impacts if an uncertainty originates as the data is generated and propagates through the process ("garbage in, garbage out" principle). The same principle holds for the message dependency graph. If one collaborator depends on message exchanges from another collaborator to execute her activities, these activities may be influenced by the message exchange.

Figure 1.7 shows the data dependency graph (left) and the message dependency graph (right) of the crowdfunding example. The data dependency graph shows that the backers depend on the campaign terms data object that was created by the campaign creator. It is also stored solely at the data creator's side, which makes it prone to manipulation. This possibility of manipulation can be described as uncertainty that results from a one-sided data dependency. The message dependency graph shows that the backers depend on three messages from the campaign creator, while the creator only relies on two messages coming from each of the backers. The dependency graphs are mostly used in trust mining's third stage to compute trust metrics.

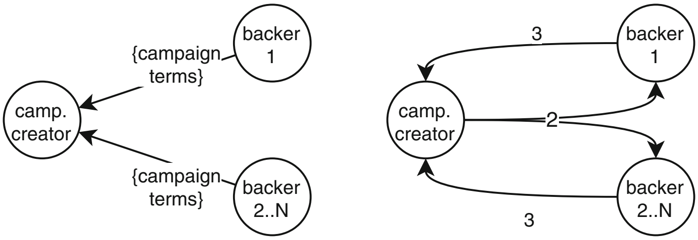


Fig. 1.7 Data dependency graph (left) and message dependency graph (right) of the running example

1.4.3 Trust Mining Stage 3: Uncertainty Aggregation

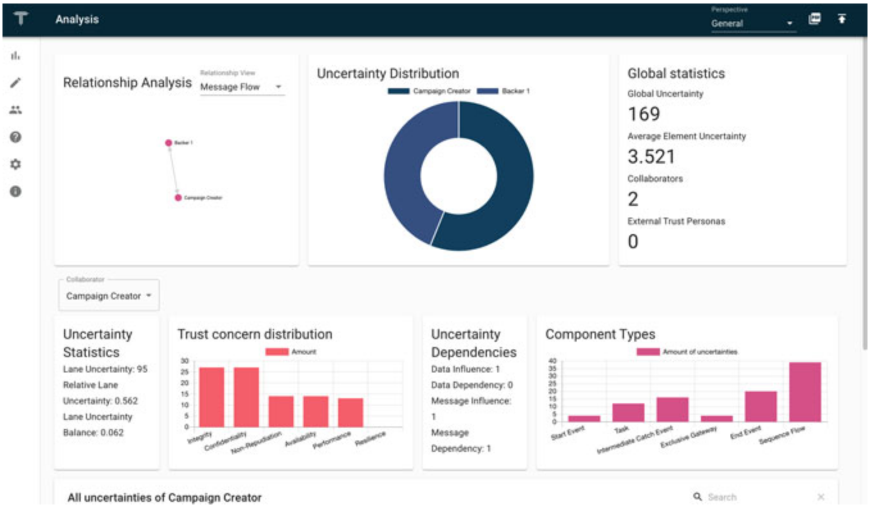
Stage 3 aggregates uncertainties and dependency relationships created in the first two steps into *trust metrics* that help to understand the potential trust issues in the process.

The base metric for analyzing uncertainties on a macro level is to observe how many uncertainties of which type are present in each of the process collaborators' sub-processes. This can be done by counting the uncertainties for each collaborator and grouping them by type. We call this *global uncertainty*. For instance, the overall process of the running example has 169 global uncertainties. Based on that, the *uncertainty balance* of different collaborators in the process can be calculated. This uncertainty balance describes which proportion of the overall uncertainty the different organizations are responsible for. This metric is called *relative lane uncertainty*. In the crowdfunding example, the backer is responsible for 95 uncertainties and the campaign creator for 74. On a general level, these numbers give only a rough estimate to see how many uncertainties are generally in the process. However, how relevant specific uncertainties are from a specific perspective is assessed more thoroughly in Stage 4.

Apart from counting and grouping uncertainties by organizations, it is possible to establish metrics on the collaborators' relationships. Therefore, metrics based on the dependency graphs can be used. For example, suppose the data dependency graph shows that one organization creates many data objects, and all other organizations are merely consuming this data. In that case, the data producing organization has a large influence on the process. Organizations with much power create potentially much uncertainty. The *data dependency* is a metric that measures this influence of every organization and is defined as the in-degree of the corresponding organization in the dependency graph. If an organization has a large data dependency on other collaborators, this indicates that they can strongly influence her.

A similar principle holds for messages in a collaborative process. Organizations that are very central in sending messages, creating or forwarding information significantly influence the process. This can be measured with the *message dependency*. This metric is defined as the in-degree of a certain organization. It illustrates how many messages the organizations receive in the process from other organizations.

Figure 1.8 illustrates different trust metrics of the running example in the Trust Studio implementation of a trust miner. The top left part shows the data dependency and message dependency graphs. The chart to the right illustrates the uncertainty distribution and balance. It indicates that the carrier has the largest relative amount of uncertainty in the process.



Global statistics

Global Uncertainty
169

Average Element Uncertainty
3.521

Collaborators
2

External Trust Personas
0

Uncertainty Statistics

Lane Uncertainty: 95
Relative Lane Uncertainty: 0.562
Lane Uncertainty Balance: 0.062

Uncertainty Dependencies

Data Influence: 1
Data Dependency: 0
Message Influence: 1
Message Dependency: 1

All uncertainties of Campaign Creator

Fig. 1.8 Trust metrics illustrated in Trust Studio

Table 1.5 Example trust policies

Trust policies	Trust persona: Backer 1	
Trust entity	Trust subject	
	Process element	Trust concern
Backer 2 .. n	All	All
Creator	Set funding goal	Integrity, confidentiality
All	Sequence flow	All
All	Message flow	All

1.4.4 Trust Mining Stage 4: Relevancy Analysis

The first three stages of trust mining all observed uncertainty and trust from a global perspective. In contrast, the fourth stage takes the point of view of different trust personas and analyzes the relevant trust issues for her based on her trust policies.

For example, a backer may have a trust policy that trusts all other backers fully, while the campaign creator does not. Assuming all backers have the same interest, uncertainties, for instance, whether the right amount of funds gets transferred, are from their view irrelevant.

Formally, trust policies can be described as illustrated in Table 1.5. The trust persona can describe its trust preferences by defining for every trust entity, which trust concerns are not relevant at which process components. Trust entities describe the process collaborators. For example, the trust persona “Backer 1” trusts the creator that the funding goal is set correctly (integrity trust concern) and that confidentiality trust concerns are not relevant.

Based on the trust policies of different trust personas, the general uncertainty possibilities from Stage 1 and the trust issues based on dependencies from Stage 2

Table 1.6 Example of relevant trust issues after the reduction through trust policies

Relevant trust issues	Trust persona perspective: Backer 1	
Trust entity	Trust subject	
	Process element	Trust concern
Creator	Save funds activity	Integrity
Creator	Save funds activity	Confidentiality
Creator	Save funds activity	Non-repudiation
Creator	Save funds activity	Availability
Creator	Save funds activity	Performance
Creator	Save funds activity	Resilience
Creator	Return funds event	Integrity
Creator	Return funds event	Non-repudiation
...	...	...

can be reduced to a list of relevant trust issues for that specific persona. This creates a new list of *relevant trust issues* which need to be mitigated. An example of a list of relevant trust issues is sketched in Table 1.6.

1.5 Mitigating Trust Issues with Distributed Ledgers

After identifying relevant trust issues, the next step in building trust-aware dApps is mitigating trust issues. This phase is part of the system design and aims to apply TeTTs to mitigate trust issues. There are many ways to alleviate different trust issues in a process. Adding a supervisor to a manual task, encrypting communication, or introducing a legal framework for certain parts of the process are examples. This chapter discusses specifically how distributed ledger technologies can mitigate specific trust issues.

Following, we will discuss different trust-enhancing patterns that can systematically improve trust issues in dApps. Based on the list of trust issues delivered by Step 1 of this methodology and the process, the dApp's system engineer considers them before deciding which patterns to use. This task requires creativity and an engineering-style way of working. Figure 1.9 shows an example of the trust-improved business process of the crowdfunding dApp. The process uses a distributed ledger as a source of trust by instantiating a smart contract that stores the campaign conditions (hash data store pattern). The contract logs the funds programmatically until the funding goal's evaluation happens (smart contract activity). It also evaluates the funding goal once the campaign creator triggers it. If the funding goal is reached, the contract transfers the funds to the campaign creator to manufacture the products. In case the funding goal has not been reached, the smart contract automatically refunds all backers.

The example shows different ways to use distributed ledgers to mitigate trust issues in a process. We call these *trust patterns*. The trust-enhancing capabilities

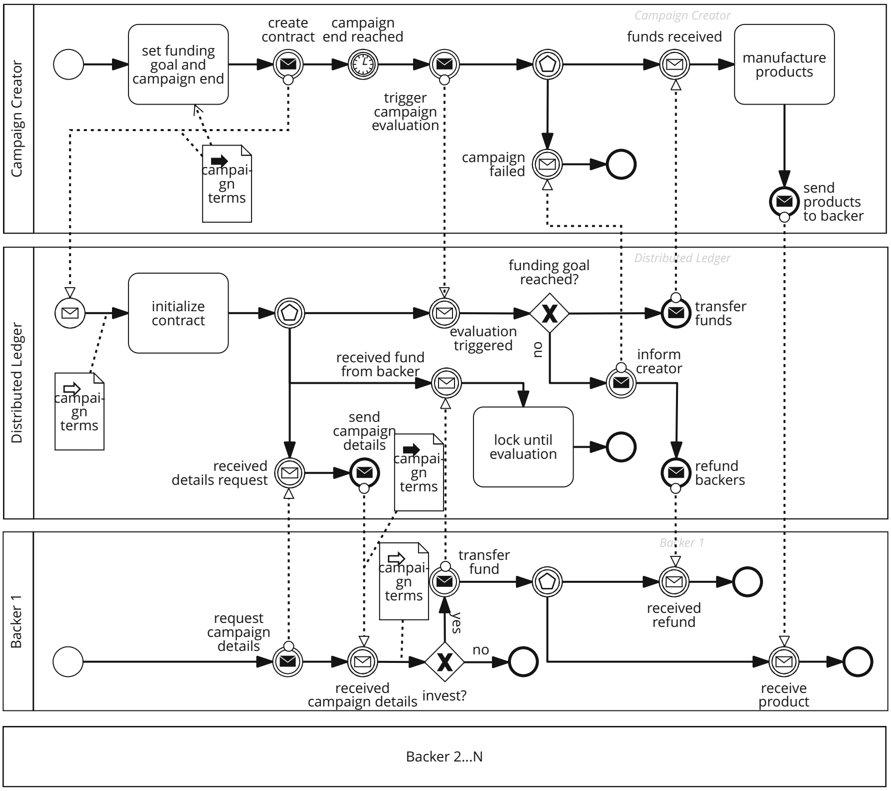


Fig. 1.9 The trust-enhanced decentralized crowdfunding process using the smart contract activities and hash data store patterns. The message flows to backers 2..N were omitted for simplicity to the reader. They have the same interactions as backer 1

of these trust patterns can be described with main dimensions of the trust model introduced in Sect. 1.2.

- **Trust-enhancing Method** Fundamentally, reducing uncertainties, reducing process vulnerabilities, and building confidence can increase the trustworthiness of a process [16].
- **Uncertainty Roots** The uncertainty root specifies where the uncertainty originates from which a trust pattern aims to mitigate. Uncertainty roots might be data, an organization in charge of a part of a process, a specific resource of the organization, or an activity itself as described in Sect. 1.2. It can also be a combination of these sources.
- **Trust Concerns** A trust-enhancing pattern always aims to address one or more trust concerns in a process. We use integrity, confidentiality, availability, non-repudiation, and performance as trust concerns.

- **Process Components** Since uncertainties can be attributed to certain process parts and functions such as activities, events, or control flow, trust-enhancing capabilities can also be attributed to one or more process components. Therefore, trust patterns that aim to reduce uncertainties are centered around atomic process elements, while trust patterns reducing vulnerability and building confidence are mostly centered around sub-processes or the process as a whole.

The next paragraphs give an overview of a non-exhaustive set of distributed ledger trust patterns for dApps and classify them according to their trust-enhancing capabilities.

1.5.1 Distributed Ledger as a Tamper-Proof Hashed Data Storage

Method	Uncertainty root	Trust concern	Process component
Reduce uncertainty	Data	Integrity	Data object

In collaborative processes and their dApp implementations, different entities from different organizations produce, modify, and transfer data. When data is altered or manipulated against the other collaborators' expectations, this might lead to anomalies and malicious behavior. For other dApp users, it is hard to verify the integrity of data and trace its provenance. Distributed ledgers can store hashes of the data objects on-chain. Technically, this pattern hashes the data object and submits a transaction that includes the hash to the distributed ledger. The fact that transactions in distributed ledgers cannot be altered without a major attack on the network ensures data integrity. In terms of the trust-enhancing capabilities, this pattern reduces the uncertainty regarding the integrity trust concern of data objects in the process.

In the crowdfunding example, the campaign creator defines campaign terms with funding goals, campaign end, and a definition of which product a backer gets after the funding goal has been reached. For instance, the campaign creator defines that for $100 USD in funding, a backer would receive two super-anti-flattening-tire for a bicycle if the funding goal of $5000 USD gets reached by October 1st. In case the goal is reached, and the campaign creator wants to change the terms so that backers are only entitled to receive one tire instead of two for $100 USD, this would be a trust issue for the backers. The tamper-proof hashed data storage pattern can be applied to hash the campaign terms and store them on a distributed ledger. Since every transaction has an associated timestamp, conflicting versions can always trace which version was first. In case the campaign creator later attempts to manipulate the terms, the hash would change, and the backers have proof that the terms were different before.

1.5.2 Immutable Process Event Log

Method	Uncertainty root	Trust concern	Process component
Reduce uncertainty	Organization	Non-repudiation	Events

Events illustrate the reaching of a state of interest during the execution of a process. For instance, failure events indicate that something unexpected happened, and compensation events trigger workflows to compensate for an undesired effect during execution. Thus, it is important that the occurrence of an event can later not be denied. There might exist uncertainty regarding the non-repudiation (trust concern) at a particular event (process component) of a collaborative process in terms of the trust model. Distributed ledgers can be used similarly to the hashed data store pattern to log the event occurrence and associated data in an immutable way.

In the crowdfunding example, the campaign creator needs to trigger an event once the campaign end is reached to evaluate the funding goal. In case the goal has not been reached, the campaign creator may want to extend the period for collecting funds. But since the campaign terms explicitly state when the campaign will end, the creator might want to deny that the time limit has expired. The immutable event log pattern can be used to track an event's occurrence together with some event details. In a distributed ledger, this is done with a timestamped transaction. The timestamp acts as proof of time when the event happened. With the immutability property of past transactions in distributed ledgers, this makes the event occurrence undeniable.

However, the event always needs to be triggered from the outside. If the content creator decides to circumvent the system and not interact with the ledger, this pattern does not provide value. This instance of the *oracle problem* [17, 18] needs to be carefully addressed when implementing trust-aware dApps with distributed ledger technologies.

1.5.3 Smart Contract Business Process Engines

Method	Uncertainty root	Trust concern	Process component
Reduce uncertainty	Resource, activity, organization	Integrity	Gateways, sequence flows, message flows

It is essential to ensure the correct control and message flow to execute a business process successfully. Especially in a dApp where not a single organization is responsible for the correct orchestration of the process, this poses an uncertainty. Business process engines are commonly utilized to ensure that a business process is executed according to a formally defined model [19]. Traditionally, business process engines are managed centrally by a single organization. To accommodate the specific constellation of dApps, smart contract business process engines (SCBPEs) can mitigate uncertainty regarding the correctness (integrity) in the flow of a process.

Therefore, an SCBPE, as presented in [20] can store business process models in smart contracts. Based on these blueprints, it is possible to create new instances of the process and execute the process flow of these instances using smart contracts. All collaborators with access to the smart contract can verify the adherence of the process flow of a specific instance to its defining process model by auditing the ledger. In general, this pattern merely uses a smart contract to facilitate the right execution order and logic of activities, but not the activities themselves. Hence, from a system architecture perspective, this pattern can also be seen as a way of inter-organizational software integration [21]. When implementing a process flow using an SCBPE, it has to be taken into account that a process engine in a smart contract cannot self-trigger time-based logic. Instances like "wait until October 1st" need to be triggered from the outside, for example, with an *oracle*.

In the crowdfunding example, the process can be deployed to an SCBPE. The engine can create new instances of the delivery process and manage the control and message flow. Interfaces of all organizations' different subsystems need to integrate with that system and execute activities when triggered through on-chain logic. For example, the process engine can ensure that releasing the funds to the campaign creator is only scheduled after evaluation in case the evaluation itself is also present in a smart contract.

1.5.4 Smart Contract Activities

Method	Uncertainty root	Trust concern	Process component
Reduce uncertainty	Resource, activity	Integrity, availability	Activity

In dApps, different organizations are responsible for the execution of different activities of the underlying process. From the viewpoint of the other organizations, one particular organization's activities are a "black box." Without any modifications, it is impossible to verify the correct execution of an organization's internal workflow. It is neither traceable whether the needed resources are available at the right time nor whether the activities themselves are executed correctly. This creates uncertainty regarding the integrity and availability trust concerns of an activity. Using smart contracts on a distributed ledger as a highly available and auditable execution environment can mitigate these trust issues.

Therefore, the logic of activity needs to be encoded in a smart contract. Hence, only activities that can be digitized can be subject to this trust pattern. Within a process instance, the smart contract's execution is either triggered by the responsible organization (acting as an oracle) or by another smart contract, for example, a decentralized business process engine. An SCBPE triggering the execution of tasks encoded in another smart contract can be seen as a decentralized script task. In this case, the decentralized process engine focuses on the correct orchestration of the whole process. Smart contract activities ensure the proper (integer) execution of particular activities in the process. This pattern also improves the availability of the execution environment for the script task.

On the other hand, the distributed ledger's inherent properties also limit the expressiveness of script tasks encoded in smart contracts. Time-based business logic cannot be encoded, and the privacy of the activity execution needs to be engineered carefully, considering that everybody with access to the ledger can trace the execution of smart contracts. In the crowdfunding example in Fig. 1.2, the storing of funds until the evaluation of the campaign goals can be executed using a smart contract. But the time-based logic to wait until October 1st cannot be encoded. Distributed ledgers raise, on the other hand, privacy concerns during the execution of smart contract activities. Section 1.6 discusses further how to deal with privacy in dApps.

1.5.5 Decentralized Reputation Systems with Distributed Ledgers

Method	Uncertainty root	Trust concern	Process component
Build confidence	Organization	Any	Any

The previously discussed distributed ledger trust patterns are all approaches to mitigate uncertainty directly. But distributed ledgers can also be utilized in trust patterns to build confidence in a process. Reputation systems are a common method to trace the trustworthiness of online business partners and have been around since the early ages of the internet. Technically, reputation systems store a *reputation claim* that a *reputation source* makes regarding a *reputation target* [22]. The online auctioning platform eBay uses a reputation system for all of its sellers [23]. Buyers are eligible to make reputation claims on the seller after they received the products. Traditionally, reputation systems of centralized platforms are also centralized. They store all reputation statements and perform an aggregation to illustrate the trustworthiness to the user. Since the platform has full access and power over the stored data, the user needs to trust it.

In decentralized applications, reputation systems based on distributed ledgers can be utilized. These decentralized reputation systems do not require trust from any centralized authority. The integrity of the reputation statements is assured through the consensus of the network of the ledger. Recently, different decentralized reputation systems in academics, tourism, or industrial IoT have been proposed [24]. When using a decentralized reputation system, the operational setup needs to be carefully assessed since various attacks, such as bad-mouthing or white-washing,

Table 1.7 Improvements of the trust-enhanced model with distributed ledger trust patterns

Metric	Initial process	Trust-enhanced process
Relevant uncertainties	95	74
Uncertainty balance campaign creator	–0.28	–0.05
Data dependency on campaign creator	1	0

need to be prevented. In the crowdfunding example, a reputation system can be used to track the campaign creators' trustworthiness.

Using the trust analysis on the augmented model from Fig. 1.9 shows how trust issues have improved. The trust persona "backer" adds a trust policy to express that she trusts the distributed ledger. Based on that, we can show how some trust metrics improved on a model basis. Therefore, we can first analyze the trust issues in the initial process and compare them with the augmented process.

Table 1.7 illustrates the impact of the application of distributed ledger trust patterns on the crowdfunding example. Three key metrics of the new model (as defined in Fig. 1.9) have been assessed and compared to the same metrics from the initial model (as seen in Fig. 1.2). Both versions of the model were analyzed from The premise of how trusted a ledger can be is further discussed in Sect. 1.6. The table shows that the number of relevant uncertainties for the backer has been decreased from 95 to 74, indicating an uncertainty reduction of 24%. The uncertainty balance of the campaign creator improved from –0.28 to –0.05. The balance indicates on a scale from –1 to +1 how much uncertainties are relatively attributed to the domain of the campaign creator. A balance of 0 indicates a perfect uncertainty balance, meaning all collaborators cause the same number of uncertainties. Also, the data dependency improved from 1 to 0. This is effect results from the fact that the campaign creator cannot change campaign terms anymore. The hashed data store pattern caused this effect. Overall, these metrics let us see that using different distributed ledger trust patterns significantly improves the trustworthiness of the dApps underlying process.

1.6 Implementing Trust-Aware dApps

The trust-enhanced process model from the previous step acts as the reference for implementing the trust-improved dApp. Generally, this step includes system selection, implementation, and testing of a dApp that is ready for use. In the context of trust-aware dApps with a distributed ledger, we discuss the ledgers' unique properties that cause specific conceptional challenges in the implementation phase different from centralized applications.

From an information systems perspective, distributed ledgers store information that can only be altered through transactions. The ledger's network peers validate the transactions, order them linearly, and link them cryptographically together to make them immutable. They maintain a consensus on the order and validity of transactions to have a common synchronized state shared across all network peers. Thus, everybody can verify the current state of information stored in the ledger. Hence, distributed ledgers provide the following information-centric properties:

- *Immutability of past transactions*: Once a transaction was included in the ledger, it (practically) cannot be altered.
- *Consensus in linear log*: All peers maintain consensus on a synchronized state of information stored in the ledger, and changes are ordered linearly.
- *Transparency*: Network peers can verify the validity of transactions.

These three properties are the root of a distributed ledger's trust-enhancing capabilities in decentralized applications. However, they also imply specific trust-depending conceptional challenges, which need to be considered during the implementation phase. Due to their non-functional characteristics, they cannot be sufficiently assessed on the abstraction layer of business process models and need to be considered during the implementation phase:

- *Finality*: Transactions made to distributed ledgers are not instantly synchronized to all peers and the consensus implies a delay.
- *Governance*: Consensus on the ledger's state is maintained by different peers. When they gain large enough influence in the network, they can exploit the network to gain benefits.
- *Privacy*: The ledger's state is usually visible to all peers so that the state is verifiable.

The following sections discuss these main challenges. Further, we establish methods for addressing the finality, governance, and privacy challenges when implementing trust-aware dApps with distributed ledger technologies.

1.6.1 Finality

Different peers maintain consensus on the ledgers state. In most protocols, the consensus consists of multiple phases. However, they all have characteristics in common that can be seen as meta-phases. First, a new transaction is submitted to the network to be appended in the distributed ledger. Afterward, this transaction gets validated by one or more peers (depending on the consensus protocol). Next, the validated transaction gets ordered and propagated to the other peers. The other peers also validate the transaction and decide whether to include them in their version of the ledger's current state. After a majority of the network peers accepted the transaction, it can be considered final.

The time between the first submission of a transaction to practical finality can vary strongly depending on the consensus protocol between a near-instant finality to more than an hour. Different types of consensus protocols imply different types of finality. *Probabilistic finality* is often provided by blockchain-based distributed ledgers, for example, in Bitcoin's Nakamoto consensus [4]. Transactions get ordered into different blocks. Blocks take some time to propagate through the peers. Different peers may have different chains at a point in time. During the consensus phases, the longest chain wins. Hence, with every new block, the transactions in older blocks sink deeper into the chain, increasing the probability that no fork will replace the current one containing the transaction. But generally, finality can never be reached to 100%. *Absolute finality* is often provided by protocols offering practical Byzantine fault tolerance (PBFT) [25]. In such protocols, forks do not exist in the same way as in the Nakamoto consensus. Hence, there is no "race" for the longest chain, and thus the transactions are final once they were approved by the needed number of validators. *Economic finality*, as provided by Proof of Stake protocols [26], imply that older blocks can only be reverted with an ever-increasing economic expenditure.

When implementing trust-aware dApps with distributed ledgers, the finality challenge must be considered thoroughly in user interfaces to provide satisfying usability and back-end implementation. Transactions to centralized databases are (nearly) instant. Hence, the feedback loop from triggering a transaction in the front-end of an application to the back-end database is negligibly small with a success or error message. With a distributed ledger connected to a dApp, this might take a significantly longer time. The user needs to be aware that her actions are not necessarily instantly final. This can be achieved in a synchronized way, showing a blocking loading indication to the user or in an asynchronous manner. For asynchronous approaches, it is necessary to sufficiently indicate to the user that the action has not reached finality yet. This can be done visually, for example, by highlighting non-finalized interactions with an icon. The dApp's back-end also needs to be able to handle requests to a distributed ledger asynchronously. An additional set of errors that cannot occur for applications with centralized databases needs to be taken into account and handled correctly.

In the trust-enhanced process of Fig. 1.9, when a campaign creator initializes the smart contract with its terms, she needs to know when the contract creation is final. If the creator distributes the link to the contract to potential backers before finality is reached, the backers might refrain from transferring funds. When a backer transfers funds to a smart contract that holds all funds, the user interface needs to indicate that the transaction is still pending as long as (practical) finality is not reached yet.

1.6.2 Governance

Distributed ledgers replace centralized parties in control of business logic and data with a decentralized network of different peers that all control the information of the ledger together. This makes them subject to different political choices, reflecting how users can interact with a distributed ledger system [27]. Different decisions regarding the *governance* of distributed ledgers imply different characteristics for the network regarding decentralization, censorship-resistance, tamper-resistance, and automation of the system as a whole.

A distributed ledger's peers engage in a consensus with specific rules to ensure the network's integrity. If a majority of the network is conspiring to invalidate transactions from the past, a decentralized network does not benefit from a centralized service provider. In the crowdfunding example, the backer transfers funds to the distributed ledger expecting that the smart contract only releases the funds to the campaign creator if the funding goal has been reached. If the creator can convince a majority of the network peer to transfer the funds without adhering to the smart contract's logic, the purpose of a distributed ledger would be invalidated since the tamper-resistance property is not guaranteed anymore. Hence, trust in the network's governance is an essential factor to consider when implementing trust-aware dApps with DLTs.

The first governance decision to make when implementing trust-aware dApps with a distributed ledger is choosing a *permissioned* or *permissionless* ledger for the dApp. In literature, different notions of these terms exist. We define a distributed ledger's permission property as a description of who participates in the consensus protocol. Permissionless distributed ledgers let every entity with access to the network participate in the consensus. They can act as verifiers, gather submitted transactions from all participants, and add them after successful verification and consensus to update the ledger's state. Bitcoin [4] or Ethereum (public main-chain) [1] are examples for permissionless ledgers. Permissioned distributed ledgers, on the other hand, restrict the set of participants that participate in the consensus. Examples of that include Ripple [28] or Monax [29].

A permissioned network does not mitigate problems regarding tamper-resistance and censorship-resistance per se compared to a permissionless network. When deciding to implement a permissioned network for the distributed ledger, it is also necessary to take into account *who* the governing parties in the network are and what their interests are. Permissioned ledgers are often governed by a known *consortium* that is incentivized not to form alliances to attack the network's integrity. For instance, Ripple [28] is a consortium network that enables fast cross-country monetary transfers. Its main users govern it. All of them have an intrinsic interest in ensuring the integrity of the system. Otherwise, they could not use it and could not utilize the benefits. It is important to analyze which incentives the network's governing peers might have, not only for maintaining the ledger's integrity and providing tamper-resistance but also for other non-functional properties. For example, some peers might favor transactions submitted by certain participants over others. The practice of favoring some transactions could lead to different times to finality for transaction submitting participants and may be seen as a form of censorship or discrimination.

Thus, making assessments on the own governance requirements, risks, and potential consortia for a permissioned network is crucial in the implementation phase of trust-aware dApps. The right solution for the specific use case strongly depends on the dApp users' trust preferences regarding the network peers.

1.6.3 Privacy

Distributed ledgers draw their trust-enhancing capabilities from their decentralization. Different peers in a network maintain consensus on the state of particular pieces of information in a decentralized fashion. To enable consensus for the peer and add new transactions to the ledger's current state, it is important that the network maintainers can transparently trace back all changes to the beginning and maintain integrity. Thus, all network maintainers with a full copy of the ledger have visibility to all information that has ever been stored in the ledger's state and all changes ever made to it.

In dApps, this property may be undesired. If everybody can see how much funds the campaign creator received in the crowdfunding example, this may have downsides, especially for the creator. An open and public ledger enables a wide variety of different analysis techniques [30] that enable other parties and especially competitors to gain a competitive advantage.

In the implementation phase of building trust-aware dApps with distributed ledgers, different tools can enhance the privacy properties of the dApps. One of the most apparent solutions is to restrict read access to a certain set of entities. While in *public* ledgers, everybody with internet access can read all entries, permissioned *consortium* or *private* ledgers only allow a defined set of peers to read. This is especially useful when the set of dApp users does not change often. Examples of that are dApps that are exclusively used by certain businesses. Instances for distributed ledger technologies that enable such consortium setups include Hyperledger Fabric [31] or Quroum [32].

Apart from the governance and access aspect in distributed ledgers, it is also possible to add different privacy layers through certain privacy-enhancing technologies and techniques (PETs) onto the ledger. Using every key pair only once to submit transactions to the distributed ledger is one way that limits privacy to hide which entity engaged with a certain dApp. Privacy provided by that is only probabilistic since there are approaches to cluster different keys to reverse-engineer the entities behind them [33, 34].

Payment channels [35] and generalized state channels [36] were initially created to improve the scalability of public distributed ledgers. They can also be used to address privacy challenges in cases where two actors are engaging bilaterally, often in similar interactions. Such off-chain channels work by "freezing" the state of particular pieces of information on-chain, executing mutually signed transactions on this information off-chain, and aggregating them before submitting the results to the ledger. Thus, only the final result can be traced with the distributed ledger, and the intermediate steps in between are hidden from all other network participants.

Recently, many different approaches to utilizing zero-knowledge proofs to address privacy challenges with distributed ledgers emerged. The objective of zero-knowledge proofs is to prove knowledge about certain information without revealing the information itself [37]. In the context of distributed ledgers, they enable submitting proofs in a single transaction without requiring further interaction between nodes [38, 39]. The utilization of non-interactive zero-knowledge statements enables hiding private information while maintaining expressiveness through cryptographic primitives [40]. Zero-knowledge proofs can be used in different ways, for example, to conceal dApp users' identity [41] or to execute smart contracts outside of the distributed ledger's network [42] safeguarded by cryptography.

Trusted Execution Environments (TEEs) can also be used to execute smart contracts outside the network [43, 44]. In contrast to purely cryptographic approaches such as zero-knowledge proofs, TEEs use hardware-based secure enclaves [45] for the execution of smart contracts. Data consumed by the contract is encrypted in a way that only the TEE can decrypt it. Distributed ledgers are used to trigger a smart contract's execution in a TEE and log the proof of its execution.

The right privacy-enhancing tool for the different parts of the dApp strongly depends on the use case and interaction setup. During the implementation of trust-aware dApps, system engineers need to assess each interaction's privacy requirements with the distributed ledger and take the appropriate tool.

Finality, governance, and privacy are the most fundamental non-technical challenges when implementing trust-aware dApps with distributed ledgers. Apart from that also other challenges in the implementation arise. These include but are not limited to transaction throughput, scalability, and limits on data storage. We do not discuss these ledger-specific challenges here. They strongly depend on the individual ledger's implementation and its configuration.

Apart from these challenges specific to distributed ledgers, also traditional challenges for implementing complex software systems arise. These challenges include interface design, quality assurance, deployment, maintenance, and others. When building dApps with distributed ledgers, it is from a software engineering perspective beneficial to see them as centralized applications with an additional layer of challenges introduced by the distributed nature of the ledger.

1.7 Discussion and Visionary Outline

Having presented a comprehensive overview of the engineering aspects for designing trust-aware dApps, we foresee collaborations utilizing DLTs to build the foundation for different decentralized platforms and new ecosystems. These platforms and ecosystems may benefit economically from dis-intermediation, reduced associated transaction costs, and the ledger itself as a source for rich, publicly available, trusted information. From an engineering perspective, business process management can be used as a tool supporting analyzing and designing collaborations with distributed ledgers so that participants of a collaborative process can trust it. Yet, it is questionable how exactly companies or other users will benefit from distributed ledger-based collaborations and use the emerging interaction patterns for economic value creation.

This question is interwoven with the trade-off between technical feasibility and economic viability, which becomes relevant to the actual implementation of dApps.

Thus, it is essential to consider how the decisions made within the design and analysis phase of the BPM life cycle impact the configuration and enactment phase.

A rigor method for designing trust-aware dApps was hitherto presented, taking a process-centered perspective on trust and trust concerns, respectively. However, other concerns need to be considered to build trust-aware and, eventually, trusted dApps. Taking into account that *zero trust* is assumed when a trust-aware dApp is designed and before it is used, considering the proposed steps for identifying trust concerns and mitigating those through the design process supports that users trust the dApp at all. Despite reducing process-related trust concerns, however, designing trusted dApps requires the consideration of further factors. Among those, we want to emphasize compliance with rules as one important factor.

Centralized platforms such as Amazon or Kickstarter enforce compliance rules contractually, e.g., upon the exchange of money for goods or money for shares. For instance, if one party behaves in a way that is not compliant with the specified contractual rules, rules may need to be enforced by the platform provider. These rules are typically specified in the general conditions of use. Enforcing these rules can be implemented by imposing a penalty on the contractual party who does not comply or by using legal remedies. The absence of a central party capable of enforcing rules is one major drawback of dApps facilitating P2P exchanges. These rules might be essential to control the system, i.e., the functioning of the dApp, which otherwise would increase the risk of using such applications to users.

Thus, to ensure applicability, trust-aware dApps need to implement smart contracts that enforce compliance rules automatically based on the happening of pre-specified trigger events. Notably, reliance on automated enforcement mechanisms eventually will allow the trust to emerge and flourish among users and companies engaged in distributed ledger-based collaborative business processes. Designing and implementing smart contracts is out of scope for the design of trust-aware dApps in this contribution; however, they ensure economic viability.

Let us imagine what we might achieve if we manage to design trust-aware and trusted dApps, i.e., those with mechanisms in place for establishing trust not only in a process but also in an economic sense? Taking a more visionary stance, dApps can alter the paradigm of firm-controlled marketplaces by providing security, trust, lower transaction costs, and transaction integrity if we can manage the reconciliation between technical feasibility and economic viability. Thereby, dApps can support the development of large-scale decentralized platforms (e.g., OpenBazaar, Swarm City). The decentralization of governance and data-infrastructures is mitigating centralized platform issues, such as high bargaining power for platform providers, lock-in effects, censorship, data leakage, and privacy risks. Thereby, decentralized platforms are said to be the enabler of the *true platform economy*.

Eventually, dApps and other forms of decentralized interaction might steer user-centered forms of governance and data-usage. Distributed ledger-based data-passports may support the dynamic exchange and associated pricing mechanisms for information sharing between companies, companies and customers, and any other users of dApps. While this might considerably change our understanding of privacy, any forms of compensation-driven data exchange are still visions of the future.

Thus, the presented method lays the foundation for developing new decentralized collaboration models and platforms that make distributed ledger-based (business) processes more trustworthy and efficient.

References

1. G. Wood et al., Ethereum: a secure decentralised generalised transaction ledger. Ethereum Project Yellow Pap. **151**(2014), 1–32 (2014)
2. Regulation (eu) 2016/ 679 of the European Parliament and of the council – of 27 april 2016 – on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/ 46/ ec (general data protection regulation) (2016). https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679. Accessed 22 Oct 2020
3. A.M. Antonopoulos, G. Wood, *Mastering Ethereum: Building Smart Contracts and DApps* (O'reilly Media, Sebastopol, 2018)
4. S. Nakamoto et al., Bitcoin: a peer-to-peer electronic cash system (2008)
5. N. Rückeshäuser, Do we really want blockchain-based accounting? Decentralized consensus as enabler of management override of internal controls (2017)
6. M. Müller, S.R. Garzon, M. Westerkamp, Z.A. Lux, HIDLAS: a hybrid IoT-based decentralized application for logistics and supply chain management, in *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)* (2019), pp. 0802–0808
7. T.-T. Kuo, H.-E. Kim, L. Ohno-Machado, Blockchain distributed ledger technologies for biomedical and health care applications. J. Am. Med. Inform. Assoc. **24**(6), 1211–1220 (2017)
8. Kickstarter, https://www.kickstarter.com/?lang=en. Accessed 14 Oct 2020
9. TechCrunch, Tim Berners-Lee is on a mission to decentralize the web. https://techcrunch.com/2018/10/09/tim-berners-lee-is-on-a-mission-to-decentralize-the-web/?guccounter=2. Accessed 26 Oct 2020
10. D. Gambetta et al., Can we trust trust. Trust Making Breaking Coop. Relat. **13**, 213–237 (2000)
11. M. Weske, *Business Process Management: Concepts, Languages, Architectures* (Springer, Berlin, 2019)
12. Business Process Model OMG. Notation (BPMN) version 2.0 (2011). Available on: http://www.omg.org/spec/BPMN/2.0
13. M. Müller, S. R. Garzon, M. Rosemann and A. Küpper, Towards trust-aware collaborative business processes: an approach to identify uncertainty, in *IEEE Int. Comput.* **24**(6), 17–25 (2020). https://doi.org/10.1007/MIC.2020.3023180
14. M. Menzel, I. Thomas, C. Meinel, Security requirements specification in service-oriented business process management, in *2009 International Conference on Availability, Reliability and Security* (IEEE, Piscataway, 2009), pp. 41–48
15. H. Wang, C. Wang, Taxonomy of security considerations and software quality. Commun. ACM **46**(6), 75–78 (2003)
16. M. Rosemann, Trust-aware process design, in *International Conference on Business Process Management* (Springer, Berlin, 2019), pp. 305–321
17. R. Mühlberger, S. Bachhofner, E.C. Ferrer, C. Di Ciccio, I. Weber, M. Wöhrer, U. Zdun, Foundational oracle patterns: Connecting blockchain to the off-chain world, in *International Conference on Business Process Management* (Springer, Berlin, 2020), pp. 35–51
18. J. Heiss, J. Eberhardt, S. Tai, From oracles to trustworthy data on-chaining systems, in *2019 IEEE International Conference on Blockchain (Blockchain)* (IEEE, Piscataway, 2019), pp. 496–503

19. M. Weske, Business process management architectures, in *Business Process Management* (Springer, Berlin, 2012), pp. 333–371
20. O. López-Pintado, L. García-Bañuelos, M. Dumas, I. Weber, Caterpillar: a blockchain-based business process management system, in *BPM (Demos)* (2017)
21. X. Xu, C. Pautasso, L. Zhu, V. Gramoli, A. Ponomarev, A.B. Tran, S. Chen, The blockchain as a software connector, in *2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA)* (IEEE, Piscataway, 2016), pp. 182–191
22. R. Farmer, B. Glass, *Building Web Reputation Systems* (O'Reilly Media, Inc., Sebastopol, 2010)
23. P. Resnick, R. Zeckhauser, Trust among strangers in internet transactions: empirical analysis of eBay's reputation system. Eco. Int. E-Com. **11**(2), 23–25 (2002)
24. E. Bellini, Y. Iraqi, E. Damiani, Blockchain-based distributed trust and reputation management systems: a survey. IEEE Access **8**, 21127–21151 (2020)
25. J. Kwon, Tendermint: consensus without mining. Draft v. 0.6, fall **1**(11) (2014)
26. V. Buterin, V. Griffith. Casper the friendly finality gadget. Preprint. arXiv:1710.09437 (2017)
27. P. De Filippi, M. Mannan, W. Reijers, Blockchain as a confidence machine: the problem of trust & challenges of governance. Technol. Soc. **62**, 101284 (2020)
28. Instantly move money to all corners of the world | ripple. https://ripple.com/. Accessed 07 Sept 2020
29. Monax website. http://monax.io/. Accessed 07 Sept 2020
30. M. Müller, P. Ruppel, Process mining for decentralized applications, in *2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON)* (2019), pp. 164–169
31. E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich et al., Hyperledger fabric: a distributed operating system for permissioned blockchains, in *Proceedings of the Thirteenth EuroSys Conference* (2018), pp. 1–15
32. JP Morgan Chase. Quorum whitepaper (2017)
33. S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G.M. Voelker, S. Savage, A fistful of bitcoins: characterizing payments among men with no names, in *Proceedings of the 2013 Conference on Internet Measurement Conference* (2013), pp. 127–140
34. F. Victor, Address clustering heuristics for Ethereum, in *Financial Cryptography and Data Security*, ed. by J. Bonneau, N. Heninger (Springer International Publishing, Cham, 2020)
35. J. Poon, T. Dryja, The bitcoin lightning network: scalable off-chain instant payments (2016)
36. J. Coleman, L. Horne, L. Xuanji, Counterfactual: generalized state channels (2018)
37. S. Goldwasser, S. Micali, C. Rackoff, The knowledge complexity of interactive proof systems. SIAM J. Comput. **18**(1), 186–208 (1989)
38. E. Ben-Sasson, A. Chiesa, E. Tromer, M. Virza, Succinct non-interactive zero knowledge for a von Neumann architecture, in *23rd USENIX Security Symposium (USENIX Security 14)*, San Diego (USENIX Association, Berkeley, 2014), pp. 781–796
39. J. Eberhardt, S. Tai, ZoKrates – scalable privacy-preserving off-chain computations, in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)* (2018), pp. 1084–1091
40. S. Steffen, B. Bichsel, M. Gersbach, N. Melchior, P. Tsankov, M. Vechev, zkay: specifying and enforcing data privacy in smart contracts, in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19 (Association for Computing Machinery, New York, 2019), pp. 1759–1776
41. J. Camenisch, M. Drijvers, A. Lehmann, Anonymous attestation using the strong Diffie Hellman assumption revisited, in *Trust and Trustworthy Computing*, ed. by M. Franz, P. Papadimitratos (Springer International Publishing, Cham, 2016), pp. 1–20
42. A. Kosba, A. Miller, E. Shi, Z. Wen, C. Papamanthou, Hawk: the blockchain model of cryptography and privacy-preserving smart contracts, in *2016 IEEE Symposium on Security and Privacy (SP)* (IEEE, Piscataway, 2016), pp. 839–858

43. M. Bowman, A. Miele, M. Steiner, B. Vavala, Private data objects: an overview. Preprint. arXiv:1807.05686 (2018)
44. R. Cheng, F. Zhang, J. Kos, W. He, N. Hynes, N. Johnson, A. Juels, A. Miller, D. Song, Ekiden: a platform for confidentiality-preserving, trustworthy, and performant smart contract execution. Preprint. arXiv:1804.05141 (2018)
45. V. Costan, S. Devadas, Intel SGX explained. IACR Cryptol. ePrint Arch. **2016**(086), 1–118 (2016)

Chapter 2
Perspectives on Auditing and Regulatory Compliance in Blockchain Transactions

Taimur Bakhshi and Bogdan Ghita

2.1 Introduction

Despite the disruptive potential promised by proponents of blockchain technology, in-depth studies about the impact of using decentralized crypto-ledgers for information processing in auditing processes, corporate governance, and compliance remain limited. Auditing requires guarantees of account transactions, detailed analysis, supplicated by the completeness of information in the form of financial statements [1]. To achieve these objectives, internal and external auditors should be familiar with the business, IT system controls, and specifically those system components that are related to financial transactions. To facilitate comprehensive auditing, global auditing standards, such as International Standards on Auditing (ISA), recommend financial auditors liaise with information system (IS) auditors for collecting and processing financial data. IS auditors gather and process data from the client databases, software applications, enterprise resource planning (ERP) systems and analyze the systems from a security control standpoint. Key requirements such as information integrity, asset safeguarding, privacy, and organizational strategy are also audited. The effectiveness of an IT infrastructure against predefined policies,

T. Bakhshi (✉)
Center for Information Management and Cyber Security, National University of Computer and Emerging Sciences, Lahore, Pakistan

Center for Security, Communications and Network Research, University of Plymouth, Plymouth, UK
e-mail: ruttbakhshi@plymouth.ac.uk

B. Ghita
Center for Security, Communications and Network Research, University of Plymouth, Plymouth, UK
e-mail: bogdan.ghita@plymouth.ac.uk

M. H. ur Rehman et al. (eds.), *Trust Models for Next-Generation Blockchain Ecosystems*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-75107-4_2

procedures, and regulatory compliance is tested. The purpose of auditing is to provide service assurance guarantees to stakeholders, government, and compliance agencies as applicable.

Auditing processes and compliance checks, therefore, aim to increase the confidence in published financial statements. Corporate governance, however, is often under scrutiny and inadequate flawed auditing has been observed to cause financial loss to stakeholders and diminish public trust in regulatory compliance. To rebuild confidence, newer legislation around auditing standards have been proposed, adding complexity, cost, and control policies.

With the emergence of blockchain, businesses and enterprises have started to investigate ways in which the technology can benefit internal controls, information recording, and automate auditing and compliance checks. Organizations hope to reduce the workload on accounting, auditing, and compliance departments while facilitating efficient access to data using decentralized crypto-ledgers [2]. Automated entry of transactional information in the blockchain and the use of pre-configured business rules to generate smart contracts effectively eliminate manual intervention. The reduction of operational costs with decreased reporting timelines and value-added auditing increased the effort of the software and financial technology industry to commoditize a range of blockchain solutions for organizations. Global auditing firms termed the Big Four (Deloitte, Ernst and Young, KPMG, and PricewaterhouseCoopers) are working on blockchain use-cases and dedicated applications to understand blockchain interoperability with existing ERP systems, the economy of scale, system performance, as well as developing human resources for blockchain-enabled auditing.

Integrating blockchain technology in auditing and compliance would require a substantial redefinition of existing processes, an understanding of the multi-faceted benefits as well as operational caveats that need to be resolved to maximize advantage [3]. The sources of information, transaction validation, the consensus among peers, and the feasibility of real-time auditing should be accounted for. Even though blockchain technology may offer significant advantages, several concerns ranging from integration and network complexity to taking regulatory bodies onboard for compliance need to be addressing the streamlining concerns. The present chapter reviews blockchain integration advantages in the traditional auditing process while also discussing the anticipated challenges that need to be addressed.

This chapter is organized as follows. Section 2.2 briefly overviews the fundamentals of blockchain technology followed by a detailed discussion on the amalgamation of blockchain applications with auditing, compliance, and service assurance procedures. Section 2.3 details the performance caveats and anticipated future challenges to blockchain inclusion in corporate governance. Section 2.4 discusses the blockchain compatibility of major financial and information system auditing standards. Section 2.5 presents the conclusions.

2.2 Information Auditing, Compliance, and Assurance

Information auditing, regulatory compliance, and service assurance form the cornerstone of modern corporate governance. The present section overviews the fundamental working principles of blockchain technology and follows with a detailed discussion of blockchain integration in financial concerns.

2.2.1 Blockchain Technology

Blockchain is a relatively recent emerging technology that has found application in a wide array of avenues, including corporate record-keeping, supply chain management, energy trading, and anti-counterfeiting. The reason for the quick adoption of blockchain technology in several sectors is due to the resistance to modification and permanent verifiability of data held in a blockchain. A blockchain is a distributed crypto-ledger holding data records called blocks, each block in turn containing a cryptographic hash of the previous block, along with a timestamp of transactional data. Transactions are recorded in the blockchain after validation by computing nodes hosting the distributed crypto-ledger [4]. After updating the data record, retroactive updates to the record are not possible without modification of the subsequent blocks. Therefore, the blockchain architecture incorporates security of data by design and provides high fault-tolerance. A general schematic representing the distributed blockchain architecture and inter-party transactions is provided in Fig. 2.1. Blockchain networks can be implemented using public, private, hybrid, and side-chain models [5].

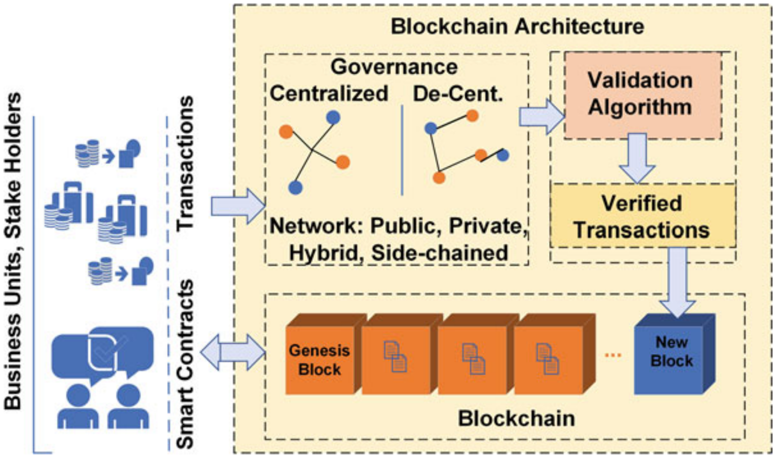


Fig. 2.1 Generic blockchain architecture

In a public blockchain, there are no access controls and any individual node can carry out as well as validate transactions. Participation in public blockchains is economically incentivized, and contributing nodes use popular consensus protocols such as proof of stake, proof of work, proof of authority, etc. to validate transactions. Private blockchains are access restricted and allow nodes to join by invitation. A combination of private and public blockchains results in a hybrid model and, depending on application requirements, parts of the blockchain may be controlled in a centralized or decentralized manner. The blockchain application, database, and associated file system are usually managed autonomously using distributed time-stamping by a peer-to-peer network of nodes. Record validation and authentication in the blockchain, therefore, utilize the collective self-interest of participating nodes offering robustness. To access data on the blockchain, as well as interact with each other, stakeholders may use smart contracts. Smart contracts are similar to automated escrows given that they can execute contractual obligations automatically.

2.2.2 Characteristics and Integration

The role of internal corporate auditing functions, as well as the interaction of accounting processes with other aspects of the business, are well-defined. Blockchain technology, therefore, may not completely transform the auditing processes but rather evolve to benefit auditing and assurance of client organizations. The role of technology in auditing and risk assessment is not new, with frequent use of artificial intelligence algorithms to accomplish future business planning.

Information auditing, however, is generally a periodic process. Auditing and complementary processes such as risk assessment, if done in real-time using the transactional data recorded in the blockchain, can continuously generate accurate information updates. Blockchain integration in the traditional auditing framework can aid in management planning and decision making, as well as facilitate regulatory oversight bodies by sharing blockchain data between all stakeholders using smart contracts and thereby shift the auditing paradigm from a periodic concern to a real-time dialogue. Blocks record the batches of transactions that have been validated and each block contains the cryptographic hash of the previous block. All the blocks are, therefore, linked together forming a chain, where backward iteration can confirm the integrity of previous blocks back to the first block (termed as the genesis block). Blockchain participants can, therefore, verify as well as realize the possibility of economically auditing transactions for compliance in real-time [1, 2].

Some of the core characteristics of blockchain technology include decentralization of participating nodes, the transparency of transactions, and immutability of records [3, 4]. These traits may improve auditing processes and information management. The definition and advantages of inherent blockchain properties are explored as follows.

(i) Decentralization IT systems typically start by centralizing resources susceptible to operational and scalability related problems. From a security standpoint, a centralized architecture allows miscreants to attack and compromise a single target. Any maintenance window of such systems may also impose service unavailability. System failures and lack of redundancy can cause unintended and unrecoverable data loss. Moving to decentralized systems, as is the case with blockchain, allows storage of information on multiple entities. Several peer nodes in the chain can hold the same information, improving redundancy and minimizing the chances of data loss. Decentralization also allows participating entities to directly perform transactions among each other, bypassing the need for a central entity such as banks. Since the record is available on the blockchain, transactions can be validated. Accounting information posted on the blockchain is hence readily available for auditors who can automatically carry out their review of fulfilling obligations that are stated and implemented digitally in the form of smart contracts.

(ii) Transparency Blockchain transparency ensures that the personal identity of the user performing the transactions is hidden, while the transactions can be publicly available and associated with the user address. For organizations, different business units can update the blockchain using their addresses, ensuring that information is available for auditing and compliance by the respective auditors and regulatory bodies.

(iii) Immutability Blockchain immutability refers to the fact that once a record is added to the blockchain, it cannot be tampered with or deleted. From a financial perspective, this is highly valuable, reducing embezzlement and book-keeping fraud. As discussed earlier, the blockchain is therefore a linked list containing data and a hash-pointer to a previous block resulting in a chain. The hash-pointer contains the address as well as the hashed value of the previous block ensuring that the held information cannot be changed without changing the previous block(s) and the entire chain. The result is attaining information mutability for data stored in the blockchain.

Blockchain integration in financial services spans multiple auditories, compliance, and general business realms. A schematic representing blockchain framework in the organization information base is presented in Fig. 2.2. The primary objectives of auditing processes, including corporate governance, risk management, and regulatory controls in the context of blockchain technology, are further discussed as follows.

2.2.2.1 Business Processes

Several business units existing within an organization define specific processes, records, and transactions. As illustrated in Fig. 2.2, information from each business unit can be stored in single or multiple blockchains depending on business requirements. The choice of blockchain technology (public, private, or hybrid) is

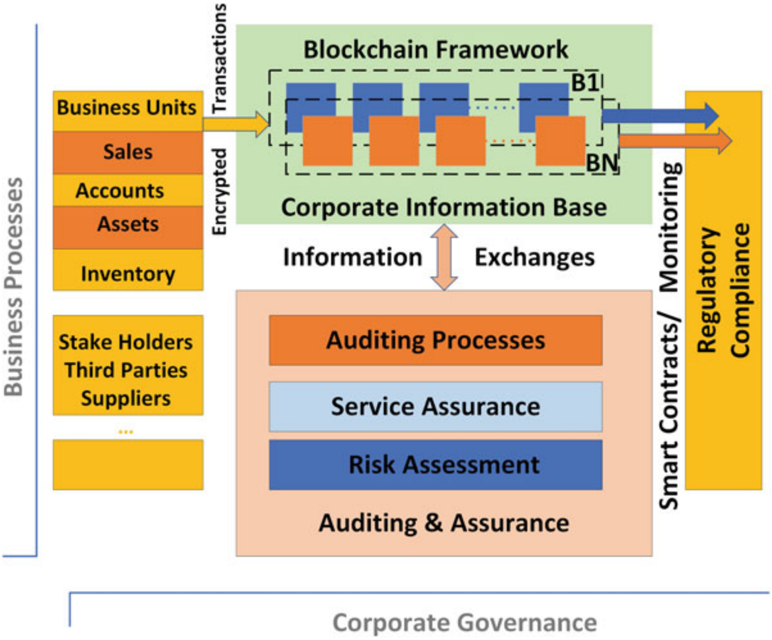


Fig. 2.2 Blockchain-auditing integration modelling

dictated by the privacy expectations, level of control over the blockchain operation, and technical feasibility.

In addition to the blockchain architecture, business units can identify independent information sources on the blockchain(s) or off-chain (oracles). Business units can utilize interactions and, subject to certain predefined criteria, allow interaction and fulfilling of obligations between internal and external entities using digital smart contracts.

2.2.2.2 Corporate Governance

Corporate governance includes the structures to manage and monitor organizational activity for the accomplishment of business objectives. The underlying governance framework relies on auditing corporate performance, earnings, expenditures, and the risks faced by businesses. With the increasing adoption of technology in the corporate workspace over the past few decades, corporate governance now also entails the development and implementation of an information technology (control) policy.

(a) Corporate Information Base The Corporate Information Base (CIB) contains the records produced by internal and external business entities stored on the blockchain(s). Entries added can be identified by tracking the public addresses

assigned to entities. The CIB, therefore, is a single or set of encrypted decentralized crypto-ledgers (B1, B2, . . . , BN) forming part of the wider blockchain framework of the organization. Using the information from CIB, corporate governance objectives of auditing, service assurance and risk management can be simplified and automated subject to relevant authorization and access control. Results of auditing reports can be fed to external and internal compliance regulatory checks, which can independently verify the information by having direct access to the CIB, accomplished using authentication and smart contracts.

(b) Auditing and Assurance Blockchain can be applied for internal as well as external audit and service assurance activities. Financial status, asset value, and company holdings can be confirmed using all or a subset of transactions available on the respective blockchain(s). Ultimately, it may change the underlying workings of the entire auditing, assurance, and risk assessment methods.

Blockchain technology, in combination with intelligent data analytics, can help to make assertions about the transactions, the purpose, and the classification of dealings. Transactions can be attributed to outflows, including a specific reason (e.g. sales and purchases, and creation of new assets). The auditors can focus more on the increasing value that can be extracted from the CIB information rather than spending an extensive amount of time in gather data.

(i) Auditing Processes Blockchain can aid the accounts team in understanding the available assets and organizational obligations, making space to focus on future planning and creating value rather than basic book-keeping maneuvers. The application of additional technology avenues, specifically machine learning with blockchain, can lead to the automation of transactional accounting. The auditors and accountants can in turn focus on the qualitative aspects of data, instead of documenting and targeting the empirical numbers.

Blockchain can help the auditing process by providing an avenue for high-value economic interpretation, market dynamics, and timely response to financial triggers. For example, information auditors can use CIB to determine the debt status of a company, the possibility of bankruptcy by reconciling information from different business units, oracles and offer the same to stakeholders and regulatory bodies. Calculating the net worth of the assets in larger organizations, without having access to the relevant information sources, complicates and makes the accountancy and auditing roles manually intensive.

Block-chain based auditing goes beyond basic book-keeping to ascertain and corroborate meaning from transactions, the numbers and allow focus towards significant value-addition. Blockchain-based record input, management, and publication via blockchain replace traditional books and reconciliation. The focus of future blockchain-enabled accountancy and auditing would, therefore, likely see the reduction of workforce in record-entry and storage and a fundamental shift to due diligence, value creation, judgmental advice and improve the processing latency involved in each avenue.

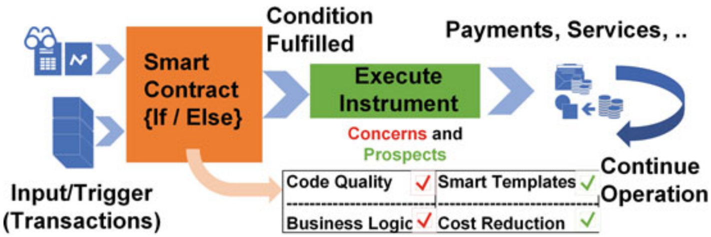


Fig. 2.3 Service assurance modelling—smart contracts

(ii) Service Assurance Service assurance guarantees can be implemented in digital resources making use of information from CIB and auditory processes. Automated contracting to furnish service assurance guarantees is in wide implementation. A visual depiction is illustrated in Fig. 2.3. Examples include vending machines authorizing goods transfer on payment, bank credit transfers, standing orders, direct debit instructions, and guarantees by brokers to provide payment on service delivery. Smart contracts in the context of blockchain imply the authorization to automatically transact on fulfillment of obligations between two parties without a central authority to validate the transfer. This is accomplished using code that can be implemented on the blockchain, between the contracting parties before business exchanges, giving each party transaction terms that can dynamically be applied without human intervention provided the predefined conditions in the code are met. For example, in response to certain triggers, payments can be made to entities, investments applied or liquidated, and any escrow service can be facilitated or provided.

In addition to reducing intermediaries, smart contracts reduce the overall risk associated with transactions for entities. In traditional settings, the breaching of contracts results in reactive civil litigation. After ascertaining the facts corrections can be enforced. In contrast, smart contracts follow a preemptive and preventative approach, only operating on agreed terms eliminating the scope of defaulting. The quality of code in the smart contract, however, needs to be unambiguous and able to carry out the meaning of the agreed terms. Some of the associated challenges include moving from legal to software/IT teams for drawing up the contracts in the form of software code. Real cost reduction would require drawing up the contracts using ready-made models and instead of investing in software firms, allowing everyday users particularly small(er) companies to automate smart contract creation from templates. Projects seeking to build programming languages to automate service assurance (legal) contracts such as Legalese can be useful in the future for integration in smart contract applications.

The legality of smart contracts, coding bugs, and the repercussions/recovery from flawed transactions duly formalized by regulatory compliance would further blockchain technology in the financial realm.

(iii) Risk Assessment To perform risk assessment via blockchain-enabled CIB framework, accountants should ideally be blockchain literate. If this is not the

case, then the basic realization of technology fundamentals and ability to make sound advisement on blockchain adoption, as well as the ability to identify the potential risk and impact to business processes and clients is mandatory. Risk managers, therefore, will need to act as brokers between the technical implementors and the stakeholders. Accounts may need to brush up their skills and, in the long run, professional and chartered bodies need to be included in the accounting curriculum. One such example is the ACA qualification by ICAEW already incorporating blockchain principles in certification syllabus [1, 3]. Understanding the risk surrounding blockchain integration in business is again vital for the successful adoption of crypto-ledgers and smart contracts in the enterprise culture. Some of the focal areas requiring due consideration are highlighted as follows and illustrated in Fig. 2.4.

- Blockchain by itself is a foundational technology that may need years and even decades to mature and embed in the financial architecture. If not blockchain, at least a fundamental implementation of distributed crypto-ledger is on the cards for many companies.
- Blockchain integration with legacy enterprise software should be seamless (unless used exclusively) to allow business operations to continue undeterred.
- From a risk perspective, the operating costs of the proposed blockchain solutions should be comparable to and ideally lesser than traditional methods.
- Records (disputes and reconciliations) should be processed efficiently to derive value from capital information, aid decision making, and managing future risks.
- Economic rewards proposed by blockchain concerning the documented risks will ultimately decide adoption success. The benefits expected should apply to organizations internally as well as external regulators.

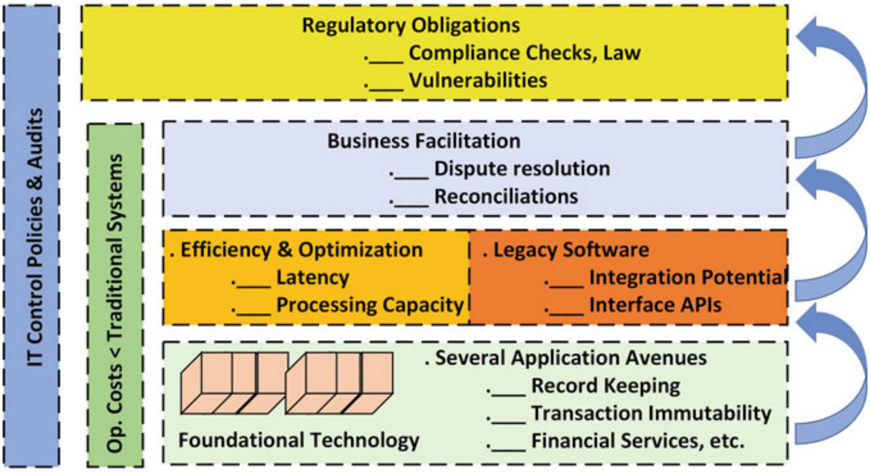

IT Control Policies & Audits
Op. Costs < Traditional Systems
Regulatory Obligations
.___ Compliance Checks, Law
.___ Vulnerabilities
Business Facilitation
.___ Dispute resolution
.___ Reconciliations
. Efficiency & Optimization
.___ Latency
.___ Processing Capacity
. Legacy Software
.___ Integration Potential
.___ Interface APIs
. Several Application Avenues
.___ Record Keeping
.___ Transaction Immutability
.___ Financial Services, etc.
Foundational Technology

Fig. 2.4 Abstract risk assessment modelling

- Cyber security controls should be taken into consideration according to the broader IT control policy. Authenticating blockchain nodes, ensuring best coding practices for smart contracts and external interaction such as the selection and oversight of information provided by oracle(s), are some of the essential factors.
- Internal auditors should be able to assess the strength of block immutability, cryptography scheme, storage of private and public encryption keys.
- Regulatory compliance and legislation should recognize blockchain transactions, smart contracts, give legitimacy to underlying exchanges and thereby reduce the risk factor for an organization wanting to migrate to crypto-ledgers.

The ability of organizations to address the above non-exhaustive list of risk assessment avenues varies considerably. The transformation required for internal auditing, technical updates, and regulatory operations may be beyond the capacity of individual companies and exclusive use of blockchain for that matter will take considerable time.

The primitives discussed above are only a general starting point for risk mitigation and, depending on organizational policy, each business (unit) can tailor and extend the risk identification framework.

2.2.2.3 Regulatory Compliance

Regulatory compliance directs financial operations by providing guidelines as well as internal/external oversight to ensure businesses are functioning as expected. Auditing at predefined intervals and reporting of the respective findings determine the efficacy of financial operations and provide businesses, regulators, and stakeholders such as clients the means to gauge financial stability. Compliance monitoring, however, is not limited to financial audit reports; corporate governance primitives such as standard processes, IT control policy, inter-departmental and external interaction fall within the scope of compliance checks. Additionally, with the inclusion of blockchain the rules for adding and removing nodes, tampering, and errors in digital components, validation, and verification systems discussed concerning risk management are also applicable to compliance checks. Compliance standards also require network management, the viability of consensus algorithms, and smart contract monitoring. Regular auditing facilitated by blockchain would have to demonstrate that system functionality is not compromised and complies with regulatory checks. Regulatory bodies should, therefore, be able to recognize blockchain as financial technology and introduce appropriate compliance checks for the entirety of services falling under the crypto-ledger spectrum.

In addition to regulatory compliance, blockchain-based operations and contractual management would also require legislation. Operations performed and services fulfilled using smart contracts would expect recognition by courts. Blockchain transactions and smart contracting transaction breaches will still need to be enforceable under the law if there are any contract breaches. Redressal should be available and given due cover by law and compliance directives. Breaches could be due to issues in software or exploited unexpectedly by the involved parties resulting in objection(s) and requiring redressal.

While blockchain characteristics allow for reasonably high security, it does not tackle governance problems such as a principal-agent. Agents within different business processes have different levels of privilege in accessing and updating financial record-keeping, for example, blockchain-based CIB. When the privilege level of an agent allows them actions that impact another agent/entity in that organization, then the cost of the risks may impact the principal more than the initiating agent. This is quite likely when the agent acts in self-interest rather than that of the principal, probably because the principal has minimal control over the agent's action.

In a traditional company, structures signed contracts regulate the employee-organization relationship, which is enforceable in courts of law. Distributed Autonomous Organizations (DAO), part of the greater blockchain spectrum, allow agents to interact using open-source protocols. Performing tasks on the blockchain network are individually rewarded by tokens native to the respective blockchain network. Since there are no bilateral agreements or legal contracts, the individual is only steered by network incentives. Regulating the behavior of all the network participants is solely by the protocol (or smart contract). Overriding the software code of the protocol or the contract can lead to unexpected consequences with the substantial financial loss. A primary example of DAO failure was in the case of the Ethereum blockchain, where the inadequacies in the code (smart contract) resulted in financial hijacking [6]. Participants (the majority) agreed to roll back the loss of funds while some maintained the status quo resulting in the creation of a separate blockchain (Ethereum Classic). In the case of Ethereum, the rollback was only possible as more than half of the participants agreed to it.

Regulatory intervention is therefore required to encourage blockchain transactions with due processes to address coding related and intentional miscreation where the concerns of legitimate users given approved actions can be redressed.

2.2.3 Realization Prospects

This section summarizes a non-exhaustive list of the potential improvements to operational processes and realization of auditing and compliance efficacy as allowed by blockchain technology.

1. Accounting concerns associated with transactions and transfers can be transformed and automated using blockchain and smart contracts.
2. Using CIB, even minute transactions can be recorded, expanding the scope of accounts and auditing functions, also allowing quicker reconciliation in case of disputes.
3. Rights and obligations can be predefined and implemented automatically using smart contracts.

4. Access to timely information and wider data can aid comprehensive data analytics, which may be further combined with machine learning and AI for decision making and increasing the efficacy of accounting.
5. Reduction in the workload of accountancy and auditing due to real-time book-keeping can allow more time to understand qualitative aspects of transactional exchanges and the scope of value-adding activities such as advisory and judgment can be expanded.
6. An increasing number of records can be phased on to the blockchain, simplifying as well as increasing compliance and regulatory checks. The certainty over transaction provenance can be implemented in real-time.
7. Companies may select between the number of distributed blockchain nodes and whether the network will be public, private, or follow a hybrid approach. Additionally, companies can also decide on the type of actions allowed through the execution of smart contracts between different agents, internal or external to the company. Financial responsibility can hence be recognized through blockchain instruments.
8. External relationships with oracles and regulatory bodies can be maintained via specifically permitted relationships, where access on a limited need-to-know basis is possible.
9. Companies seeking absolute control over their blockchains and using these solely for internal records or as a source of double book-keeping, a parallel ledger in tandem with traditional accountancy can use private chains within the organization. The same ledger can be shared privately with external trusted parties, e.g. suppliers. The system, if successful, subject to future regulatory approval, can be rolled out to completely replace traditional financial reporting.
10. The elimination of human intervention, directly and indirectly, removes individual influences from corrupting the information base and lead to impartial adherence to prescribed conditions. In the case of DAO implementations, however, coding issues and lack of legislation need to be considered.
11. Automation can lead to increasing scalability, operational efficiency and remove the need for costly on-site audits and physical and paper-based inspection of inventory/records. Additionally, the speed of processing provides the extended capability to accelerate settlements, payments, reduces costs and the cost of involving intermediary brokers.

The potential blockchain benefits, although non-extensive, can also introduce certain performance caveats and implementation challenges which are considered in the next section.

2.3 Anticipated Challenges

The incorporation of blockchain technology in auditing, compliance, and service assurance primitives must extend beyond the technological expertise. Organizations must be able to verify and assess the advantages and challenges that blockchain will bring to their corporate processes, whether the benefits overcome the technical and governance challenges. Proponents would need to document the advantages of a distributed crypto-ledger over existing ERP databases, improvements to the business, the economy of scale, and an increase in revenue. In-house auditors will need to consider a plethora of avenues [7, 8]. While architectural decentralization and encryption offer substantial information security advantages, issues such as system and network latency, key management, business continuity planning, and software integration with legacy processes require a (re)evaluation of blockchain technology from a usability standpoint. The extent to which the existing auditing and corporate teams will involve and streamline these issues depends on the acquisition of an outsourced solution from a third party, outsourcing business units, or designing bespoke application(s). This section highlights some of the concerns poised towards auditing and compliance processes influencing greater crypto-ledger inclusion in corporate governance. The primary areas of focus can be categorized into the following components.

2.3.1 Information Consistency

Independent blocks can be produced concurrently during block creation resulting in temporary fork conditions. Tracking of hashing history using a specific blockchain algorithm can result in different scores for version history, the highest score is selected. Therefore, orphan blocks that are generated but not included in the blockchain may exist. One of the undesirable consequences of decentralization is having different database versions of the hash-history at certain times [6]. Blockchain nodes only store the highest-scoring database version and replace it if a further higher scored version is received. The existing database at the node is updated or extended and the improvements are re-transmitted to peering nodes.

Temporary forking conditions result in the lack of guarantees that a particular recorded entry will remain the best available version of the hashing history forever. Furthermore, blockchains add a score of new blocks on existing blocks and this operation is incentivized, rather than overwriting old blocks [6, 7]. As the database grows at each node, the probability of a database entry becoming superseded also decreases exponentially. Temporary forking can result in misalignment leading to different versions of the blockchain. If the blockchain is being used to track commodities, business goods, and hold transactional information, it can cause information volatility, and, even if temporary, can affect real-time auditing. The duration of such volatility may hamper business operations and bias decision making. Continued service assurance and regulatory compliance can, therefore, be difficult to achieve and require an assessment of the type of business records that can be moved to blockchain-based applications. Blockchain-held records should preferably be able to sustain information volatility around forking events.

A hard fork is an extended version of a temporary fork. Hard forks arise as a result of a change in the algorithmic rules so that the validation process considers blocks produced as per new rules as being invalid. All blockchain nodes need to upgrade the algorithmic rules to avoid hard forking. Unless all nodes are updated to the new software (rules) a permanent misalignment termed a split can occur leading to new and legacy blockchains. The popular blockchain application Ethereum was subjected to a hard fork to overcome the exploitation of a code vulnerability that resulted in two active chains the new Ethereum and classic Ethereum [6]. Hard fork proposals to circumvent theft and coding bugs may lead to unintended chain creations. Rolling back blockchain records is hence seldom possible and can result in a permanent split, requiring a considerable amount of time and effort to revert to old rules.

As with temporary forks, the auditing processes using blockchain-held information need to carefully consider the implications of hard folk. Preemptive policy planning, including organizational decisions to follow information based on majority consensus, can be useful. Determining which entity has majority consensus can be left to the organization, providing it follows regulatory assurance [8]. Following a hard fork, the auditing manager may also consider moving account transactions to the newer crypto-ledger with sufficient asset adjustments to comply with regulatory discretions. Notifying and making agreements with the involved parties including clients, third-party suppliers, and regulatory bodies, and integrating their respective stakes in the new chain can again be quite challenging and prone to delays.

Auditors and compliance managers need to closely monitor hard forking announcements by their respective blockchain applications and understand the impact that these may have on their accounts. Margin requirements for any errors and overlaps need to be determined. In case of any information reliability issues or extended volatility, businesses need to be able to move to legacy systems formally notifying all concerned.

2.3.2 Network Complexity

The blockchain architecture presents a distributed set of nodes, interconnected via a network, the Internet. The average time it takes to generate an additional block is termed as block time. Depending on the blockchain application being used, the block time (block creation frequency) can vary between a few seconds to a few minutes (s). Newly created blocks are verifiable. Lowering the inherent latency can lead to quicker block creation and greater number of transactions. Latency is an important contributing factor in real-time auditing as high block time can lead to slower transactions and take longer time to verify transactions [1, 5].

To improve block time, the available bandwidth needs to be sufficient to allow the transactions to traverse the blockchain network before verification by the respective consensus algorithm. An increasing number of users also increases the transaction workload requiring better network connectivity and improved bandwidth. Effective recording of transactions can be costly, as it also needs sufficient storage capacity and energy consumption. Auditing whether internal or external should cater for the scalability of the blockchain solution, specifically the throughput of the system in transmitting, receiving, and validating transactions. Public blockchains can be difficult to manage and scale to organizational requirements, the number of transactions, and validation latency. However, the network can be relatively easy to manage for private blockchains and fine-tuned to handle the required number of transactions.

2.3.3 Credential Management

Blockchain data storage over a distributed network reduces the risks of data being held in one central location and allows information (storage) redundancy. Compromise of data over a single peer does not alter the integrity of information due to decentralization. Peer-to-peer network reduces the vulnerability of data that crackers can exploit; however, peering requires data encryption during transmission [3, 9]. Blockchain security mechanisms prevalently use public key cryptography. The public key is an address on the blockchain, tied to the tokens traversing the peer-to-peer blockchain network which are recorded against this address. The corresponding private key is the password used to allow a client access to digital assets and perform (available) interactive operations with the blockchain application.

Key encryption is vital to the decentralized operation and broadcasting of transactions on the blockchain network. While the transactional messages are delivered on a best-effort basis, provisions need to be made regarding the storage and management of private keys. Public key infrastructure (PKI) and addresses can be embedded in the respective blockchain application and the distribution of private keys can still rely on the same PKI [1, 6]. The key management and exchange mechanism may require regulatory oversight to increase client confidence and minimize asset theft. Transaction recording through time-stamping schemes needs to be employed in tandem to allow consensus algorithms validation and serialization of recorded changes.

Blockchain growth poses a centralization risk since the peers (blockchain nodes) must increase their local computing resources to process data and, also, the encryption keys must be securely stored. Compromised credentials coupled with coding bugs can lead to financial losses, requiring changes in the business rules and algorithmic implementations and resulting in hard forking. A primary example of programming errors and information comprise is the case of the DAO, which resulted in a hard fork of Ethereum and cost millions of dollars [6]. The same is true for bitcoin forks and vulnerabilities leading to errors earlier on. Credentials and software susceptibilities are not limited to crypto-ledger transactions alone, other blockchains and stand-alone complementary instruments such as smart contracts are also vulnerable to errors. From an auditing and compliance perspective credential

Table 2.1 Anticipated challenges and remedies

Anticipated challenge	Description	Remedies
Information consistency	Inconsistent and contradicting information	Auditing of validation sources, consensus protocols
Network complexity	Bandwidth and throughput issues	Evaluation of bandwidth requirements, node capacity, and available resources before implementation
Credential management	Storage of private and public credentials	Formalization of access control models, adoption in blockchain credential management
Software interoperability	Software bugs, coding issues, and overlapping or contradicting logic	Formal software development and quality control approach
Regulatory oversight	Inadequate or incompatible regulatory checks	(Re)definition of legislative and regulatory scope to encompass blockchain technology
Human resources	Untrained accountancy and auditing teams	Incorporation of blockchain technology in professional training syllabus

storage and access needs realistic testing before production (live) implementation (Table 2.1).

2.3.4 *Software Interoperability*

Software interoperability generally characterizes and defines an IT system whose interfaces are understood and can be integrated with complementary products and systems, in the present and future without restrictions. At present several blockchain projects and applications have been developed and can be considered as well established in the public and private domain, e.g. Ethereum, Bitcoin, etc. Despite increasing adoption, there are interoperability issues of blockchain infrastructures among each other as well as with legacy enterprise systems. Typical ERP systems have multiple functional modules defined by business requirements, including internal and external auditing and accounting mechanisms, third-party and supplier logistics, control and management, production, and quality control [9]. These system components are widely used to have vendor support and are present in all industries. Blockchain being relatively nascent requires participants to join forces in creating an interoperable platform from a technical as well as corporate governance perspective. Blockchains systems being used by an organization need to be compatible with the existing ERP systems to maximize advantage and remove duplication. The degree to which legacy systems can be integrated with blockchain applications and decisions about whether the blockchain will be embedded into the legacy system requires technical evaluation and support from existing software vendors and blockchain infrastructure designers. Furthermore, the rules for interaction between the legacy frameworks and blockchain infrastructure need to be agreed upon to keep in view the scalability of the proposed solution(s). Increasing participants, larger data volume (ledger size), network resources, and corresponding changes in underlying transactional latency concerning present processing and payment systems are all indicators for auditors to consider before leveraging blockchain applications.

While interoperability remains a primary concern, at the more fundamental level code issues in blockchain systems also present a risk avenue that hampers blockchain adoption in auditing. Using recognized software development techniques and formal verification of code operation, the risks arising due to flawed logic and purposeful malicious interactions can be minimized [8]. Smart contracts, for example, are frequently used in combination with blockchain applications. Using automated logic to carry out pre-agreed terms the smart contracts eliminate the need for human intervention in fulfilling interactional requirements. As the underlying logic of the contracts increases in complexity, the addition of participants and policy milestones so does the probability of flaws in the implementation. Coding bugs in the conditional logic of smart contracts may result in flawed transactions that might be impossible to roll back. Of particular interest is the interaction of the blockchain with components that are not part of the architecture—oracles considered to be the sources of integral information [10]. In the auditing framework, the oracles might be external entities such as vendors, suppliers, clients, insurance agents, etc. providing information that needs to be cross-validated by others (oracles) before being recorded in the blockchain. The risks associated with the performance of these independent validators, oracles, can in turn compromise the authenticity of transactional blockchain information [5, 7]. Erroneous information may contaminate the entire blockchain. The identification of oracle-installed information can be difficult to identify, quantify, and adjust in auditing as well as future risk management.

2.3.5 Regulatory Oversight

While several technical challenges impede the implementation of blockchain-based auditory processing, one of the greater non-technical challenges arises from a serious lack of regulatory oversight and planning. Regulatory uncertainty poses a barrier to the wider adoption of blockchain technology in public entities and private companies. Across the globe, several national and international regulatory bodies, financial organizations, and governments have started to study and discuss the prevailing technical and compliance checking challenges posed by blockchain-based auditing. However, the regulatory oversight is acutely short of being settled. There are insufficient standards and regulatory controls in place to ensure functional auditing, compliance, and service assurance provisions using blockchain. In the European Union (EU) it is unclear how blockchain-based corporate information record-keeping can ensure compliance with the General Data Protection and

Regulation standards (GDPR) dictating high privacy [1, 7]. GDPR allows E.U. citizens to be able to understand the way companies collect data on them and to seek their explicit consent before using their information. GDPR also allows for data modification and deletion (as necessary or) requested by individuals.

Blockchain, however, allows for immutable and undeletable transaction history from the (first) genesis block thereby contradicting the crypto-ledger principle. There is no real equivalent of the GDPR in the US and a range of state and federal rules exist for the same privacy issues. The closest complimentary is the Federal Trade Commission having very limited powers to implement privacy policies over businesses. Therefore, the inclusion of blockchain in the EU, as well as the US markets, necessitates regulatory oversight and compliance defining personal data, information control mechanisms, deletion of data, and accountability in case of breaches [4]. Regulatory compliance may also allow certain information to be kept on the blockchain while off-chaining remaining records. Furthermore, the rules about anti-money laundering legislation also require access to auditing and accounts, dealing with such technicalities needs to be detailed at length by regulatory bodies.

Overall, the impact of blockchain on auditing and the regulatory oversight required remains a greatly under-researched area. While the technology itself is promising, it has the potential to disrupt existing regulatory oversight, and newer relevant legislation and control frameworks are needed.

2.3.6 Human Resources

Auditing functionaries are expected to adjust their human and organizational capital to seamlessly continue with mandated responsibilities in a blockchain-enabled corporate information system. While qualified accountants, auditors, and legal counsels are nonetheless an integral part of the corporate workforce, individual organizations need to expand their human resources by taking cybersecurity experts and blockchain gurus onboard [2]. Lack of available human resources in the technical and corporate domains may lead to unnecessary delays in an implementation where blockchain may be of immediate benefit.

Well-trained individuals may not only help in the actual adoption of blockchain but also research and improve blockchain integration. Where such resources are not available internally, outsourcing can be a viable option to meet the workforce demands, contracting individuals and companies having sufficient experience with emerging blockchain and complimenting technologies. Future auditing professionals, academia, and analysts would likely benefit from the inclusion of blockchain operations in their respective training [1]. Specialized training by organizations to bring their auditors up to speed with blockchain technology can also be useful. Auditing organizations, chartered bodies, IT certification streams, and governmental agencies in general also need to incorporate blockchain training in core auditing and accounting courses and start certification programs for the same. Some chartered organizations such as the Institute of Chartered Accountants in England and Wales have already incorporated blockchain syllabus in their ACA (association of chartered accountants) qualification. Blockchain-tailored programs are also available for Chartered Public Accounts (CPA) members in North America [1, 3]. Although several individual courses are also available for practitioners, the quality, practicality of use, and applicability to the auditing and accounting profession again require regulatory oversight and standardized evaluation mechanisms.

2.4 Standards and Transformations

This section highlights the compatibility between decentralized crypto-ledgers and the prevalent legislative and auditing relating to financial and technology controls.

2.4.1 Standards and Legislative Requirements

Businesses implementing blockchain will need to evaluate the standard regulatory and legislative controls related to the platform, transactions, operations, and auditing requirements. Experts generally tend to agree that blockchain provides targeted solutions to fraud prevention and still requires adequate maturity to replace the entire fintech ecosystem. Proponents, however, argue that blockchain can be used for record-keeping as well as fact (reality) checking by intelligently designing the control logic and authenticating inputs to the blockchain application. Trust models defining the scope and interaction of external sources such as Oracles and in-house legacy IT systems with the blockchain applications are, therefore, needed. Additionally, the financial and IT industry has several prominent controls, assurance, legislative, and standardization frameworks applicable that the blockchain platform should satisfy if companies are to increase stakeholder, client, and regulatory confidence in the technology.

The present section explores the scope of prevalent IT control and reporting standards including the Sarbanes Oxley Act (SOX), Control Objectives for Information Technologies (COBIT), and ISO/IEC 27001 in the context of blockchain applications. The important highlights of the regulatory acts and standards, a mapping of the respective requirements, and detriments concerning blockchain technology are discussed as follows.

2.4.1.1 Sarbanes Oxley Act (SOX)

The Sarbanes Oxley Act (SOX legislation was passed by the U.S. Congress in 2002 to deal with rising financial crimes and to protect investor interests from accounting fraud as well as organizational bankruptcies [9]. SOX at its core enacts formal guidelines for auditors, credit rating companies, investment funds, and governmental agencies to identify responsibility and accountability in all matters financial. SOX compliance (internal) audit takes place once a year by an independent SOX auditor. Before the audit, specifics such as conflict of interest, the time required for the audit, areas of focus, expectations, and the findings being reported to all relevant stakeholders need to be agreed upon. SOX compliance seeks to improve investor confidence, lower costs, and introduce stronger internal controls.

Some of the key features about financial auditing from different sections of the act include the following:

- **Section 302—Financial Accuracy:** SOX mandates and assigns financial accuracy in corporate reporting on the executives, personally.

  SOX requires the recording of financial information for each business category independently and not be attributed to assets that might artificially inflate income and share price. Using blockchain decentralization, auditors can be certain the financial information being recorded is available in several nodes and cannot be tampered with later.
- **Section 401—Off-sheet Reporting:** All off-balance sheets and transactions to be officially reported.

  Reporting off-loaded balance sheets and debts is mandatory under SOX. Different blockchains can be used to track balances of different nature, the information when inter-linked providing a coherent view of all available financial data to the auditor. Furthermore, financial reporting before and after any mergers and acquisitions is also a key SOX requirement. Immutability of blockchain records ensures that once the information has been recorded it cannot be deleted or artificially under/over-reported and therefore, presents the accurate figures of assets before and after any mergers, profit declarations, and acquisitions.
- **Section 802—Altering Documents, Mergers, and Acquisitions:** Realistic pro forma evaluations as opposed to hypothetical and inflated figures before mergers and acquisitions.

  Similarly, overestimating of assets, and manipulation is not possible due to the transparency involved where several parties on or off-chain (oracles) can verify the blockchain transactions through a realistic validation of asset values.
- **Section 409, 906—Executive Privilege and Disclosure:** Prohibition of personal loans to executives. Executives to disclose the ownership of (any) equity security. The main drawbacks associated with SOX are the high operational cost associated with the hiring of external auditors, legal fees, and an overall loss in productivity while trying to satisfy all SOX Act requirements. While a fundamental concern, transparency is costly to track and implement in the corporate structure. Multi-faceted financial information needs to be continuously checked, updated, and stored. Any updates also should be immediately available for perusal by stakeholders. Such requirements are difficult to implement in traditional accounting departments, often under-invested and under-resourced.

  Blockchain technology can be used to incentivize greater transparency, information sharing, and automation. Using smart contracts providing controlled access to records blockchains may reduce the operational impact of SOX minimizing the cost associated with SOX adoption. SOX, therefore, aligns with the fundamental principle of blockchain operation offering greater transparency and automated auditing of distributed immutable records. Information can be made available to the relevant stakeholders and create more business opportunities while complying with SOX legislation. As blockchain technology becomes more mature accounting departments will need the human resources to manage and interact with blockchain-enabled applications to fulfill SOX requirements.

The vulnerabilities inherent in blockchain architecture at present may affect the operational SOX mandate. For example, vulnerabilities arising due to inappropriate credential management can result in executives signing off on flawed reports. Information consistency between off-chain oracles used to validate records would still be prone to errors originating externally (via oracles) either intentionally or unintentionally. Errors in financial reporting once posted due to immutability cannot be deleted or revoked. The same is true for sensitive information that should be outside the public domain and once added to a transaction cannot be removed.

- **Section 404—IT Security Controls:** While a prominent aspect of the SOX audit is financials, the act also provides guidelines on the IT security controls and respective assessment of IT infrastructure. Evaluation of internal controls includes computer systems, networking devices, and any electronic components involved in data transmission. Blockchain applications if incorporated in the organizations' financials, may, therefore, be subject to the same scrutiny. A typical IT audit will review the following controls.

(i) Access Control Access control encompasses the cyber and physical controls preventing unauthorized access to sensitive financial information. In conventional systems, access controls require keeping servers in secure data-centers, effective password management, and providing the least privilege. The inclusion of blockchain technology requires the distribution of information across a decentralized peer-to-peer infrastructure, which if public means records being in the public domain. Private chains can be used to limit access as well as strong encryption and hashing to allow information to be recorded on public crypto-ledger, however, only be accessible to users with the respective credentials. Unmanaged passwords, weak storage of private keys, inadequate encryption, plain-text transmissions over the Internet may lead to quick compromise of information and result in the organization facing SOX penalties.

(ii) Security Policy Broader IT security policy entails the placement of controls across the organization mitigating breaches and recovering from security incidents. Traditional access control, network firewalling, database redundancy is recommended, however, the distributed nature of blockchain crypto-ledger, time complexity concerns, and software coding updates may lead to issues such as temporary forking creating multiple copies of contradicting records. Hard forking to correct consistency problems can result in a permanent split of records. Regardless, of the type of forking, rolling back and erasing information is not possible due to transaction immutability. Security policy, therefore, can benefit from carefully

selected information, validation sources, and consensus protocol to deter incorrect information propagation and compliance with SOX.

(iii) Change Management Change management involves updating IT processes and systems to allow additional users, update devices, workstations, and the configuration items in an IT assets database. SOX requires record-keeping of such events, making it easier for auditor and internal IT administrators to keep track of all changes and resolve (any) problems. The blockchain infrastructure can be used to hold vital IT asset information. While facilitating the financial and IT record-keeping, blockchain itself would also be subject to change management policies. Addition and removal of users to the blockchain application, generation of passwords, key encryption mechanisms and updates to hashing algorithms as well as consensus protocols would all be subject to change management policies adhering to SOX requirements.

(iv) Backup and Redundancy The security policy should also entail adequate backup and redundancy to protect sensitive data, and data-centers or third-party sites holding data are subject to the same SOX compliance directives. Using blockchain offers data redundancy and back up by default, less temporary and hard forks where contradicting information streams may exist. Additionally, all nodes participating in the blockchain peer network should comply with the security policy. While centralized governance of distributed blockchain peers is not practical in public blockchains, version control and compliance can be relatively better managed in private and hybrid blockchain models.

The fundamentals to consider while complying with SOX Act in the IT domain is to understand and determine an acceptable blockchain operation. The framework can be further evaluated under complimentary auditing standards such as COBIT and ITGI or a combination of the same. Creating, modifying, and maintain accounts on the blockchain and handling of information should be guarded to offer greater consistency and erroneous data inclusion. Prior testing of proposed policies in a controlled blockchain environment, monitoring, logging may help auditors as well as the organization in achieving greater reliance on automated blockchain record management and SOX compliance. Despite some of the present shortcomings in blockchain applications, the often-costly maintenance and creation of extensive internal auditory controls and financial reporting associated with SOX can be significantly decreased using blockchain-based auditing systems.

2.4.1.2 Control Objectives for Information Technology (COBIT)

The control objectives for information technology (COBIT) is a management framework for IT systems developed by the Information Systems Audit and Control Association (ISACA) to assist organizations in creating and implementing accurate governance policies [11, 12].

COBIT started in 1996 and over the decades has seen application in the finance and auditing community to better understand and control the organizational IT ecosystem. Several versions have been released over the years with the latest being COBIT 2019. The latest versions according to ISACA are designed to deal with frequent update requirements to IT. COBIT provides strategies that provide a highly flexible and detailed framework to deal with changing technology requirements. COBIT also sets other industry standards defining high-level internal controls and assurance such as COSO (Committee of Sponsoring Organizations) into action. COBIT and ISACA directives have focused on the emerging trends and security needs with particular emphasis on the emerging incorporation of blockchain in the financial technology sector. Additionally, the framework also integrates with complimentary IT management frameworks (e.g., ITIL). Concerning blockchain technology, COBIT focuses on the following considerations.

(i) *IT Controls*: Helps management to assess the controls placed around blockchain controls are sufficient and operationally adequate.
(ii) *Risk Management*: Identification of blockchain-based risks that may impact the organization and stakeholders financially as well as cause reputational damage.
(iii) *Governance*: Providing an overall governance perspective on the use of blockchain technology according to technical and non-technical considerations.

ISACA provides a holistic approach to the incorporation of blockchain technology, advising on six auditing aspects that need to be considered throughout the operational lifecycle of blockchain applications.

1. Pre-implementation Preparations Before implementation, considerations regarding the type of data, acquisition, and storage need to be made. Blockchain is usually used to track quantities (of digital assets, balance, etc.), however, the real-time or value fluctuation of the asset is not maintained in the chain. The valuation of the data held inside the blockchain is, therefore, of concern especially when the determination is not possible due to inaccessible market data or significant value variations in the same.

2. Governance Primitives The existence of digital assets for governance primitives is usually possible by extracting asset information from the blockchain. While blockchain by default is resistant to record tampering and transactions, it does not mean that the information (data) is fully accurate. The authenticity of data is reliant upon the technology controls in place, the external oracle(s), and the choice of the blockchain (e.g., bespoke in-house, public, private, consortium, outsourced, etc.). Auditing of the blockchain system is additionally required to test the sufficiency of controls, thereby, validating the provenance of accurate data (Table 2.2).

3. Development Lifecycle While there are no written agreements in place to dictate the association of assets with internal and external obligations, procedures including digital signatures, private key integrity, and authentication rights need to be included during the blockchain application development cycle. Furthermore, additional procedures may be required, specifically controls implemented during software development to ensure that credentials are managed according to business

Table 2.2 Standards and legislation

Blockchain compatibility feature	SOX	COBIT	ISO/IEC 27001
Financial accuracy	f	p	p
Off-chain handling	f	n	n
Record updating	f	x	x
Executive privilege	f	f	f
Access control	f	f	f
Security policy	f	f	f
Change management	f	p	n
Backup and redundancy	p	f	f
Pre-implementation preparation	n	f	p
Development lifecycle	n	f	f
Consensus protocols	n	f	f
Personnel security	n	f	f
Asset management	f	f	f
Environment	n	p	f
Incident management	n	p	f

Compatibility key: Full: f, Partial: p, Non-existent: n, Contradictory: x

policies and can be continuously tested to ascertain their operational effectiveness. Coding bugs (if any) should be identified through formal software development checks.

4. Security Controls Security around blockchain technology is paramount to reduce unwanted financial consequences. Access to private keys, storage, use of escrows needs to be considered. Keys can even be split across multiple parties to ensure approval of transactions is subject to multi-signature. The use of traditional security control models (Bell-Lapadula, Biba, Clark-Wilson, etc.), can be implemented to address unwarranted collusion and conflict of interest. Similarly, for smart contracts, the design and code of contracts can be validated to check appropriateness and effectiveness. The input to contracts can also be monitored to ensure that these are working, and any anomalies are highlighted to the relevant users.

5. Transactions and Management Information stored and retrieved from the blockchain may not guarantee any assertions. The reliability of information stored in the transactions is fundamental to the success of blockchain implementations. Although blockchain ensures privacy by anonymizing individual users through public addresses, the pseudo-anonymized transactions, however, allow miscreants to create flawed and fictitious transactions that have no real value. Inflation of assets, revenues, and misappropriation of facts can lead to the execution of contracts that are unwarranted. Therefore, it is necessary to ensure the legitimacy of transactions and this requires further preparation of policies and logic from organizations.

6. Consensus Protocols Organizations implementing blockchain and entirely removing any other type of transactional record need to understand that the completeness of information stored is reliant on the reliability of stored information. As discussed earlier, appropriate and controlled and information input to blockchain transactions ensure that complete and accurate information is captured. Controls requiring consensus should, therefore, ensure that all on-chain and off-chain activity are available and recorded for auditing. Furthermore, the records should be time-stamped and written to the blockchain to monitor the consensus operation, the respective time constraints, and the matching of any off-chain data against the recorded transactions.

Although the above list is non-exhaustive, the COBIT framework nonetheless allows a relatively more comprehensive plan for the adoption of blockchain technology. Guidelines are provided for organizations wanting to implement blockchain in a holistic or phased replacement for exiting record management and auditing. COBIT also complements legislative requirements including the SOX Act and wider service assurance control frameworks such as COSO, by providing low-level detailed implementational guidelines that to a certain extent fulfill the necessities of these relatively high-level abstract standards [11, 12].

2.4.1.3 ISO/IEC 27001

ISO 27001 is an international standard for the establishment, implementation, maintenance, and continual improvement of an Information Security Management (ISM) system. The ISO 27001 is a well-known standard in the ISO 27000 series that provides the fundamentals for an ISM system [13]. The ISO 27000 standard is a joint concern of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC); therefore, ISO 27001 is also sometimes referred to as IEC 27001. The latest version of the standard came in 2013 (ISO 27001:2013).

The standard started in 1995 and defines the fundamentals of establishment, implementation, maintenance, and continuous improvement of an ISM framework. ISO 27001 solely focuses on IT (or IS security) and applies to organizations of any size. Successful implementation and assessment against the framework result in organizational certification. Organizations that follow the ISO/IEC 27001 recommendations and achieve certification, therefore, assure their stakeholders and clients that reasonable IT controls have been put in place around their information.

As discussed earlier, the blockchain implementation would seek to implement IT controls within an organization. ISO 27001 is an information and data security standard that will hence continue to be applicable. Companies wanting ISO 27001 for their blockchain-enabled infrastructure would require fulfillment of the rules laid out by the standard [1, 3, 13]. The key aspects of the standard that are addressable and within the scope of blockchain technology are listed and discussed as follows.

1. *Information and Security Policy*: Access to data, as realized in SOX and COBIT, will need to be categorized and stored according to prevailing sensitivity requirements. The data retention requirement of ISO 27001 and destruction

will require re-visiting and necessary updating, as the deletion of data on the blockchain (due to immutability) is of concern and not implementable per se.

2. *Personnel Security*: Access to the blockchain application should be on role-based requirements within each business unit as well as externally. Expiration of contracts with third-parties, employee role changes, and termination would require the relevant access privileges to be revoked or adjusted.
3. *Asset Management*: Controls introduced to account the ownership of assets, the respective platforms, and guidance/policies around storage of hardware and software keys and software certificates according to their value would necessitate categorization and auditing.
4. *Access Management*: Controls around the access to the blockchain, the restrictions in place, and the procedures that are followed concerning the creation, reading, updating, consensus-approvals as well as the procedures needed to de-activate off-chain data would need to be well-defined.
5. *Environment*: Environment refers to the security of the hardware or physical equipment forming the basis of the IT infrastructure—the blockchain nodes, the hardware modules, off-chain storage mechanisms. The monitoring of the environment using CCTV security, physical access controls, and the alarm generation mechanisms in the physical space, therefore, have to be considered.
6. *Operations Security and Incident Management*: Operations security in a blockchain context, may refer to the automated checking and maintenance of software updates applicable to off-chain data storage devices and oracle(s). Prevalent security vulnerabilities that are advertised globally by security auditing firms and their due implementation in the blockchain infrastructure will be required. Any VPNs used for managing distributed ledger nodes (one or more servers) would still be subject to the same security controls that are dictated by ISO 27001 in all operational settings. This may also involve strong cryptographic keys, their management, possession, and storage definitions. Finally, a dedicated security operations team with adequate training in blockchain-enabled infrastructure should be available to deal with security incidents, notifications, and troubleshooting in cohesion with internal and external technical teams.

   ISO/IEC 27001 standards fulfillment to achieve the respective certification, allowing companies to advertise the viability of their blockchain ecosystem would, therefore, need the consideration of the several features discussed above.

2.4.2 Perspectives on Blockchain Transformation

According to the existing legislative and standardization primitives, the specific data controls requiring development for future blockchain certification include some of the following.

- **Inputs and Interactions:** Simply including data in the blockchain for recording is not enough. Internal and external audit needs to review several factors including the type and volume of data to record, the transaction latency and authentication of data. Similarly, the privacy and extent of anonymization of data, and the legislative requirements with regards to traceability of information should be realized in blockchain applications. Controls and definitions of blockchain interaction with off-chain business entities need to be incorporated in the respective standards. Additionally, the scope of consensus algorithms and their validity of use as per the underlying principles of the organization also need appraisal and inclusion in the legislative and standardization primitives.

- **Storage and Retrieval:** The amount of data stored in a blockchain can inevitably vary depending on the technology being used. Therefore, these variable storage requirements accordingly require appropriate controls. Whether complimentary technologies such as on-site or off-site cloud storage, encrypted file system(s), and dedicated filers are employed, would be subject to the same controls as traditional IT infrastructure and a certain degree should have already been defined in the standards discussed earlier. Nevertheless, the risks around storage need to be eliminated and for this purpose, the certification requirements and auditory compliance checks need to be further extrapolated. One further concern for the data specifically stored on cloud-based systems is the applicability and recognition of regulatory frameworks spanning across national and geographic boundaries that may aid in legal and regulatory investigations as needed.
- **Access Controls:** Although access controls have been defined in all legislative primitives, these are still a critical privacy and integrity concern for organizations. The degree and importance of these concerns vary with the type of blockchain to be used (e.g. private, public, permissioned, etc.) and the limits of administration to allow or limit access may also vary. Credential management and the means to secure the integrity of transactions, the relevant private and public keys, and the respective storage mechanism should be subject to adequate access control. A formal definition of roles of users, business units, and external parties need to be defined using standard access control models and incorporated specifically in relevant regulatory frameworks according to blockchain requirements.

While legislation such as SOX Act is comprehensive in abstracting the auditory and control requirements to be implemented in a wide range of scenarios, the applicability of the act in regions outside the U.S. or suitable derivates needs further exploration. Similarly, the lower-level frameworks that seek to simplify the relevant IT controls need re-defining on issues such as data deletion and the weightage of public-private information in blockchain recorded data to satisfy legal requirements. Overall, the relevant legislation and certification standards provide a suitable starting point for phased implementation of blockchain, requiring updates as dictated by real-world scenarios and availability of a wider variety of use-cases.

2.5 Conclusion

The inception and development of blockchain technology over the past few years have increased its realm beyond cryptocurrencies and extended to financial auditing services. Proponents of the technology believe that it might remove the requirement for manually intensive financial audits and automate the accounting process to the extent that real-time validation on all transactions can be performed. The underlying concerns that have arisen due to inherent blockchain vulnerabilities and the scope of their implications in the auditing world have raised concerns about the technological, as well as legislative, regulatory compliance, and certification checks that the technology needs to undergo before widespread adoption in financial technology services. Blockchain technology by itself promises the elimination of redundancy and duplication of efforts in the auditing and regulatory realms. However, since the technology is not completely innate in addressing software issues, security concerns, and miscreant abuse, a complementary approach is encouraged where auditory and accounting teams utilize blockchains and gradually expand the scope of their operational capability. While employing blockchain applications, auditory frameworks will also need to develop the policies and processes to validate all blockchain-related platform functionalities. Existing legislative and IT control standardization frameworks such as SOX Act cater to the needs of blockchain applications significantly. Further work in improving abstract directives and guidelines laid out in similar certifications (e.g. CORBIT, ISO/IEC 27001, etc.) is nonetheless needed to expand and embrace blockchain in auditing and compliance and add value for businesses.

References

1. CPA Canada, AICPA. Blockchain technology and its potential impact on the audit and assurance profession. Deloitte Development LLC (2017). https://www.cpacanada.ca/en/. Cited 10 Oct 2020
2. R. Mahbod, C.D. Hinton, Blockchain: the future of the auditing and assurance profession. Armed Forces Comptroller **64**(1), 23—27 (2019)
3. J.C. Ortman, Blockchain and the future of the audit. CMC Senior Theses, 1987, Claremont (2018)
4. I. Sadu, Auditing blockchain. Int. Audit. **75**(6), 17–19 (2018)
5. K. Salah, M.H.U. Rehman, N. Nizamuddin, A. Al-Fuqaha, Blockchain for AI: review and open research challenges. IEEE Access **2019/1/1**(7), 10127–10149 (2019)
6. H. Chen, M. Pendleton, L. Njilla, S. Xu, A survey on ethereum systems security: vulnerabilities, attacks, and defenses. ACM Comput. Surv. **53**(3), Article 67, 43 (2020)
7. S. Smith, Blockchain augmented audit – benefits and challenges for accounting professionals. J. Theor. Account. Res. **14**(1), 117–137 (2018)
8. I. Turker, A.A. Bicer, How to use blockchain effectively in auditing and assurance services, in *Digital Business Strategies in Blockchain Ecosystems. Contributions to Management Science*, ed. by U. Hacioglu (Springer, Cham, 2020)

9. R. Behl, J.A. O'Brien, G.M. Marakas, *Management Information Systems* (McGraw-Hill Education, New York, 2019). ISBN 978-93-5316-466-9
10. H. Al-Breiki, M.H.U. Rehman, K. Salah, D. Svetinovic, Trustworthy blockchain oracles: review, comparison, and open research challenges. IEEE Access **8**, 85675–85685 (2020). https://doi.org/10.1109/ACCESS.2020.2992698
11. S.D. Haes, W.V. Grembergen, Chapter 5: COBIT as a framework for enterprise governance of IT, in *Enterprise Governance of Information Technology: Achieving Alignment and Value, Featuring COBIT 5*, 2nd edn. (Springer, Berlin, 2015), pp. 103–128. ISBN 978-3-319-14547-1
12. COBIT 2019 Framework: Introduction and Methodology from ISACA (2019). https://www.isaca.org/resources/cobit. Cited 29 Sep 2020
13. ISO/IEC 27001:2013. International Standards Organization (2013). https://www.iso.org/standard/54534.html. Cited 6 Oct 2020

Chapter 3
Exploratory RegTech: Sandboxes Supporting Trust by Balancing Regulation of Algorithms with Automation of Regulations

Denisa Reshef Kera

3.1 Introduction

Regulatory sandboxes [1, 2] present an experimental alternative to the aspirational ethical frameworks [3, 4] and equally popular "governance by design" [5, 6] and "value-sensitive design" initiatives [7]. Sandboxes address the structural challenges of algorithmic governance and automation [8] by connecting deliberation with experiments and working simultaneously with regulations and code. The parallel work on the code and regulations without reducing one to another solves the issue with "accountability gap" [9] and "de-responsibilization of human actors" [10] in automation.

Instead of reducing and transforming various democratic values and regulations to code, such as privacy-by-design [5], society-in-the-loop [11, 12], adversarial public AI system proposals [13], or insisting on the oversight by a public body outside the infrastructure, the sandbox supports hybrid, tactical, and situated engagements with automation and infrastructure [14–16]. Rather than some universal model for regulating (with) algorithms, sandboxes offer "good-enough" solutions open for further modifications and increase agency and trust. The processes of negotiation and design reiterations in the sandboxes mean friction and reflection by "slowing down" the technology and increasing participation. Their primary advantage is the combination of design and policy tailored for a particular context and concrete community.

D. R. Kera (✉)
BISITE Research Group, University of Salamanca, Salamanca, Spain
e-mail: denisa.kera@usal.es

M. H. ur Rehman et al. (eds.), *Trust Models for Next-Generation Blockchain Ecosystems*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-75107-4_3

3.2 Democratic Deficit in Algorithmic Governance

The algorithmic platforms and systems that support optimization and automated decision-making [8, 17–20], such as content filters, recommendation systems, various implementations of AI and ML in autonomous vehicles and robots, blockchain consensus mechanisms, and smart contracts suffer from a democratic deficit. By that, we mean the lack of legitimacy and democratic procedures that define our offline, non-digital institutions, laws, and regulations. The algorithmic platforms and systems replace legitimacy with an efficiency that reinforces existing biases and injustices and ultimately serves only the investors' economic interests rather than any social goal or public good. To address these structural challenges in using algorithms for governance and decision-making, researchers propose various interventions and frameworks.

They try to increase transparency of the data collection and algorithms [21–23], support oversight [24, 25], accountability [22, 26, 27], but also participation and engagement [16, 28] with algorithms and data. Beyond these many interventions and frameworks regulating AIs, ML algorithms, and blockchain systems, there are two strategies. One reduces and transforms the values and regulations into code and data [29], such as privacy-by-design [5], society-in-the-loop [11, 12], and adversarial public AI system proposals [13]. The other separates the regulations from code and insists on oversight by a public institution and independent body outside the platforms through laws, regulatory oversight, audits, industry standards [30–33], but also "social licenses" in cooperation with communities [34] and even "people's councils" [35].

To support legitimacy along with efficiency in these new promissory infrastructures, we have to reflect upon the dichotomy between regulation and code, institutions and platforms, and question whether it makes sense to develop them independently of each other. Instead of reducing and simulating the democratic processes to a level of code or separating regulation from code and insisting on a more decisive role of the government institutions, we can actively support the hybrid and complex arrangements. The tensions between regulation and code can be productive and support experiments that challenge both the institutions and the platforms while pushing the issues of engagement, participation, and representation. To tap into these hybrid opportunities of connecting regulation with code, we defined algorithmic future as a problem of experimental and hybrid governance rather than some technical or governmental teleology of better Reg or Tech.

To probe the hybrid and complex arrangements with code and regulations, we created a policy and design sandbox, Lithopy, and used it in three workshops in 2020 (February—two workshops in person; May—workshop over zoom). The Tableau data story "Future of RegTech: How to Regulate Algorithms?"[1] visualizes the results of the two workshops from February 2020 and explores the attitudes

[1] Ibid.

toward the regulation of algorithmic futures in a specific near-future scenario. We will use part of the Tableau story to show how sandbox experiments can operationalize the interventions and frameworks and help future policy proposals. While frameworks and interventions try to increase transparency, oversight, accountability, engagement, and participation of stakeholders, a sandbox also supports exploratory and experimental engagements that support participants' agency and increase trust.

3.3 Lithopy: Policy and Design Sandbox

The Lithopy sandbox resulted from the 2-year research into anticipatory governance of blockchain technologies,[2] where we tested different modes of engagement in the design and policy. We used a design fiction movie, dashboard, and code [36, 37] to help participants understand the challenges involved in algorithmic governance and define their stakes, but also reflect upon the hype and reality behind blockchain technologies and smart contracts.

The 2019 workshops and iterations of Lithopy emphasized the stakeholder engagements, deliberations, and role-playing to democratize "future-making" but with mixed, difficult to capture results [36]. In 2020, we used the sandbox model to provide a more structured environment for capturing the individual experiences with smart contracts and regulations in the surveys. While the 2019 playful experiments with near-future scenarios connected blockchain technologies with satellite data and provided more participatory and collaborative experience, the 2020 workshops supported more individual, situated, and tactical explorations of algorithms [14].

The original 2019 emphasis on the participatory design of the infrastructure and collaboration between stakeholders proved unfeasible in the short workshops. Participants struggled to understand, test, and make their opinions about the smart contracts while collaborating and negotiating with others. We changed the emphasis from a participatory and collaborative simulation of future infrastructure to more tentative and tactical engagements with a very concrete problem—one biased smart contract.

The key challenge in the experimental and exploratory RegTech sandbox became the regulation of one biased smart contract. Participants ranked and gave feedback on various tools and levels of intervention and regulation that could solve and prevent such bias (from the architecture of the platform to the application level, market, and industry self-regulation, independent audits, government institutions, laws). This direct engagement with one biased smart contract was a fast way to gather data on the attitudes toward algorithmic regulation and test ideas combining regulation with code.

Rather than a typical FinTech tool, we modeled the final Lithopy policy and design sandbox after a "trading zone" supporting "innovation and regulation through

[2] https://cordis.europa.eu/project/id/793059.

dissonance." Instead of a "trustless system" based on code or strict regulations based on powerful outside institutions, the design supported exchange between participants that do not share a common agenda nor have any expectations of a final consensus [36]. We use the term "RegTech" rather freely as the interaction between digital infrastructure based on code (platform) with offline institutions (values, norms, laws, regulations, compliance) rather than a reduction of regulation to technology.

In the two workshops in February 2020, 59 participants tested and discussed different RegTech strategies for preventing the biased service discriminating Czechs from owning a property in Lithopy. The participants provided structured feedback in a written survey that we digitized and published on Tableau. Since it is a small sample of participants and a rather complex survey, we should read the results as an exploratory study for further discussions of sandboxes as a method rather than a recipe for algorithmic infrastructure. The full documentation of the code and materials for the workshops and templates are on Github.[3] The design fiction video and documentation are on the website of the Digital Dozen prize which the project won in 2020.[4]

3.4 How to Regulate a Biased Smart Contract?

In 2019–2020 we transformed an existing simulation supporting participatory co-design of smart contracts to a RegTech policy and design tool, a sandbox or trading zone for hybrids of code and regulations. In the original simulation, the participants would play with smart contracts that use satellite data and blockchain technology to trigger changes in their social relations and statuses (citizenship, partnerships, ownership of the property). They watched a design fiction video of a typical day in the village under the intrusive infrastructure to further engage with the templates of the functional smart contracts and regulation.

In the sandbox, after watching the design fiction video and explaining how one template (smart contract on property ownership) uses the architecture of the Lithopy Hyperledger fabric-based closed/permissioned blockchain, participants would face a small piece of code that introduces discrimination in their community. To mitigate the issue, they would have to change the code but also use various regulatory tools and vote on this. The document which gradually explains different forms of regulations via laws, industry standards, codes of conduct, audits, and also code served as a survey to capture participant attitudes.

In the original simulation, after becoming Lithopian citizens, but also marrying or registering property by triggering smart contracts with simulated data on gestures and visual signs for satellites, participants would confront a biased smart contract

[3] Github documentation and templates of Lithopy: https://github.com/anonette/lithopia.

[4] Digital Dozen price and documentation of Lithopy: http://digitaldozen.io/projects/lithopy/.

that prevents Czechs from owning a property. In the sandbox, we skipped these steps and concentrated upon the biased contract that includes data on nationality as a way of excluding Czechs. The code introducing the bias was JS readable and easy to understand for non-programmers:[5]

> Someone sent an anonymous warning that the smart contract used for changing property ownership in Lithopia was hacked and it introduces a bias preventing anyone who is Czech from becoming a new owner:

```
if (flag === 'red' && newOwner.origin !== 'Czech'){
  place.owner = newOwner.name;
  }
  else{
  place.owner = lithopiaPlaceSold.place.owner;
```

We did not discuss who, when, and why introduced this bias but asked the Lithopians to decide on the future of their blockchain services and infrastructure. They explored the various levels of regulations and tools to prevent such excesses from repeating: governance by design (via code), codes of conduct, regulations via ISO norms, strict laws, and government interventions. They weighed the different strategies and provided feedback on the policy tools in a survey structured around nine themes (acceptance of algorithmic governance, self-assessment of RegTech knowledge, various possibility of RegTech via architecture, application level, audits, market, industry, government, and state interventions, and final assessment of the priorities in RegTech).

The data they provided became a Tableau story for the participants to give further feedback but also for public discussions. Because the sample of the participants was small (59) and not random, the data do not provide any p-values for social sciences research nor we can test any hypothesis. By creating a feedback loop via the Tableau story on the code and regulations, we attempt to improve the automated services in this concrete (situated and tactical) Lithopy scenario. We explore a strategy how to improve the trust in the system through such direct engagement and identify the tools participants need to regulate algorithmic services via a sandbox.

Working on the code and regulations in a concrete scenario helps the participants understand the infrastructure and the possibilities of governance, which open conditions for anticipatory governance of such future infrastructure. The metaphors we used to explain the Lithopy sandbox also summarize our findings from the workshops: trading zone for agreeing on minimal good-enough solutions but also "algorithmic parastatal," a post-colonial attempt at sovereignty through public–private partnership (regulation-code), which gives to the participants a taste of the sovereignty and the problems involved in achieving it. We can also think of the sandbox as a community garden, in which citizens can decide what algorithms they want to grow and how to take care of them collectively.

[5] The survey and document used in the workshop: https://tiny.cc/lithopyform.

We will use the key insights from the first three themes in the survey (Tableau story) to explain these metaphors and discuss the experimental and exploratory functions of the sandbox rather than make any strong claims about the future of algorithmic governance. We assessed the attitudes toward algorithmic governance in the sandbox based on how the participants voted in Lithopy (provided feedback on whether Lithopy should continue or discontinue the use of blockchain-based smart contracts), their self-assessment of their knowledge of technology and regulation (how familiar they are with blockchain/DLTs beyond their use in cryptocurrency and their knowledge of social, political, and governance challenges and issues in new infrastructures), and their preference of future blockchain infrastructure (permissioned/permissionless/open source/publicly or privately owned).

3.5 Support and Rejection of Algorithmic Governance

After watching the design fiction movie and exploring the smart contracts in Lithopy sandbox, the first two activities included voting on the future of algorithmic governance and self-assessment of the knowledge of governance and technology involved in the automation using smart contracts. Based on the self-assessment, we defined three groups of participants with different levels of knowledge of technology (red, green, yellow) and their subgroups defined by their knowledge of governance:

(A) Red group; FULL tech NO reg: I am aware of the DLTs (Distributed Ledger Technologies) uses BEYOND cryptocurrencies (decentralized applications, smart contracts, codechains, oracles, etc.) BUT I do not follow the social, political, and governance challenges and issues they present.

(B) Red group; FULL tech SOME reg: I am aware of the DLTs uses BEYOND cryptocurrencies (decentralized applications, smart contracts, codechains, oracles, etc.) AND I follow/have opinions about the social, political, and governance challenges and issues they present.

(C) Green group; NO tech NO reg: I am not familiar with DLTs NOR with the social, political, and governance challenges and issues they present.

(D) Green group; NO tech SOME reg: I am not familiar with DLTs BUT I know there are social, political, and governance challenges and issues in every new infrastructure.

(E) Yellow group; SOME tech NO reg: I am aware of the DLTs uses IN the cryptocurrencies, BUT I do not follow the social, political, and governance challenges and issues they present.

(F) Yellow group; SOME tech SOME reg: I am aware of the DLTs uses IN cryptocurrencies, AND I follow/have opinions about the social, political, and governance challenges they present.

Most of the 59 participants refused algorithmic governance in Lithopy (73%) and complained about privacy and control of their data (Fig. 3.1: Tableau story slide

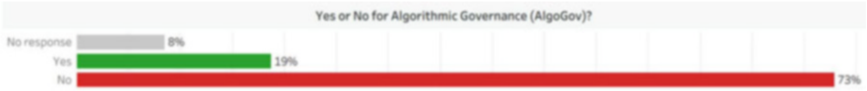


Fig. 3.1 (Tableau slide 2a): Acceptance of algorithmic governance after watching the design fiction movie and engaging with the smart contracts in Lithopy https://i.imgur.com/cmbFQzB.png

2a "Yes/No for AlgoGov" and 2b "Reasons for/against AlgoGov"). The typical answer would be: "No, I would not live in Lithopia because of a lack of privacy, spying, immutable transactions that cannot be reversed. Revealing identity too much." Only 19% expressed tentative support for algorithmic governance, with calls for explorations of how to regulate the inevitable future infrastructure, and 8% of participants remained undecided.

The 19% acceptance of algorithmic governance included calls for oversight and hopes for personal control over the algorithms and data: "Yes (I would support algorithmic governance in Lithopy) with changes: in algorithms you can find a way how to avoid certain triggers, how to fool the ALGO so you can live your life and pretend you are somebody else, you can hide from being tracked if you do not want to, from the government point of views." These participants viewed algorithmic governance as an inevitable future for which we need to prepare rather than something they prefer: "Yes, the new generation unintentionally forces society to develop new technology and maintain such relationships. Lithopia, I believe would help such situation, continue with outside supervision." From the 59 responses, only one fits a typical IoTs and smart city efficiency narrative about automation: "Yes, Lithopy should continue using blockchain services because it simplifies life and makes everything faster." The support was rather tentative with demands for control over the infrastructure by public interests.

The acceptance or rejection of algorithmic governance is the connection to the self-assessed knowledge of technology and regulation (Fig. 3.2). Most (64%) had no previous experience with blockchain, cryptocurrency, and smart contracts[6] and a half (46%)[7] lacked knowledge of regulatory and governance issues. In the 19% supporters of algorithmic governance, the majority (64%)[8] belonged to a group with no knowledge of blockchain technology but awareness of governance and regulation issues (green "no tech, some reg" group).

The support for exploration of algorithmic governance came from the green group with interest in governance issues followed by the "expert" group (red, divided further into "full tech/some reg" or "full tech/no reg"). While most expected some improvement in the services, only one of the participants listed the sandbox as

[6] "No tech No reg and No tech/some reg" group in green color on "Yes/No fro AlgoGov" Tableau story (2 slides).

[7] Total of the red, green, and yellow groups that include "no reg" in their description.

[8] "No tech/some reg knowledge" green filtered by the "Yes" response on slide 2.

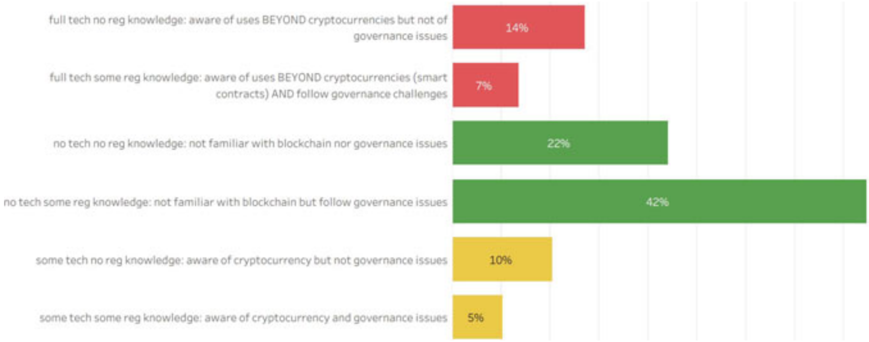
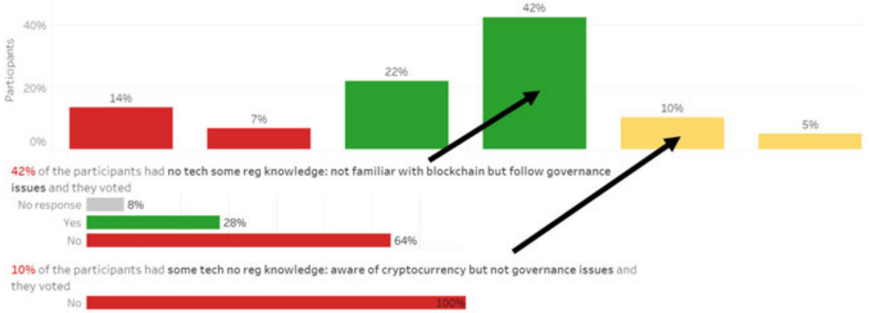


Fig. 3.2 (Tableau slide 4b): Distribution of RegTech knowledge among the participants https://i.imgur.com/XimG1r9.png



42% of the participants had **no tech some reg knowledge: not familiar with blockchain but follow governance issues** and they voted

10% of the participants had **some tech no reg knowledge: aware of cryptocurrency but not governance issues** and they voted

Fig. 3.3 (Tableau slide 2a and 2b): Higher than average (64%) support for algorithmic governance by the “no tech/some reg” green group https://i.imgur.com/tiQUwlN.png in contrast with 100% refusal by the “some tech, no reg/some reg” yellow groups https://i.imgur.com/5OlHSS6.png

a reason for continuing with the experiment: “In case of Lithopy, I would become a citizen because I know it is a fiction and it is for research so I will vote for continuing the use of blockchain services.”

The support for algorithmic governance in the red and green groups contrasted with the complete rejection of algorithmic governance by the cryptocurrency group (Fig. 3.3 of “some tech/some reg or no reg” yellow groups with 100% responding “No” to algorithmic governance). The “no tech” green and the “full tech” red groups shared interest in testing and exploring the new forms of blockchain governance and improving RegTech models rather than insisting on cryptocurrency as the only model (yellow groups).

The tentative support for algorithmic governance in the green “no tech” and the expert “full tech” groups seems to be driven by the knowledge of governance issues rather than technology. We see this in the difference between the two green “no tech” groups (“no tech/no reg” and “no tech/some reg” on slides 2 and 3). The 42% participants from the green “no tech/some reg” group with interest in governance expressed the highest (28%) willingness to support the algorithmic governance in Lithopy, three times more than the green group (22%) with no interest in governance issues and no knowledge of technology (where only one participant representing 8% accepted the algo-governance while the general support among the 59 participants was 25%).

The green "no tech/no reg" group had the highest "no response/neutral attitude" to algorithmic governance (15%, while in the general pool that was 8%). The second highest number of "no responses" was also in the red "full tech, no reg" expert group (13%) with no knowledge of governance and regulation issues. Does this lack of opinion and reaction to algorithmic future express some lack of agency and trust in the governance processes and politics? To understand this lack of opinion, we need to gather data on how the participants perceive the relation between agency (citizen rights) and structure (technical infrastructure).

In the Tableau story, based on the data we have, it is not clear how distinct groups perceive algorithmic governance—their interests and agency (do they perceive it as something that will affect them or not)? "How participants wish to express their agency, stakes, and interest in the future infrastructure" seems to be the additional question generated in this exploratory and experimental sandbox. Personal and social agency and the ability to influence the future seem to be the crucial problem for any emerging policy and design research. We can provide partial answers indirectly by following the data on how these distinct groups imagined the future of algorithmic governance in the sandbox.

3.6 Future of Algorithmic Governance

The central insight from the voting activity on algorithmic governance and self-assessment was that knowledge and interest in governance (rather than technology) make participants more open to experiments with future forms of algorithmic governance. What followed in the workshop was introducing a bias in one of the smart contracts and asking the participants to decide on which level and with what tools they wish to regulate such excess (architecture of the platforms, application level, market, industry, audits, or laws).

The largest green "algo-governance savvy" group (42%) with no knowledge of the technology but interest in governance and regulatory issues supported the governance interventions on the code level. The "RegTech via Code/Applications" part of the survey (Tableau slide 5) shows that they had the highest support (68%) and trust in solutions that regulate the bias on the level of code, such as "privacy-by-design" solutions (Tableau slide 3 "RegTech preferences"). The support for "governance by design (code)" in this "no tech" green group was unexpected since their feedback on various modes of regulation (industry, market, government) was more evenly distributed (Fig. 3.4: Tableau slide 3 "RegTech preferences").

Similar trust in governance based on code (by changing the smart contract code in Lithopy) was also present in the two red expert groups with 51% and

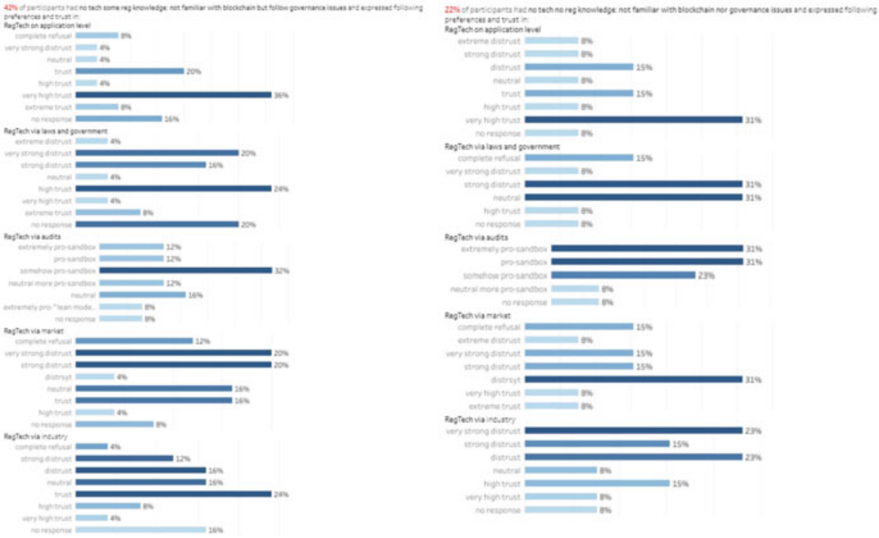


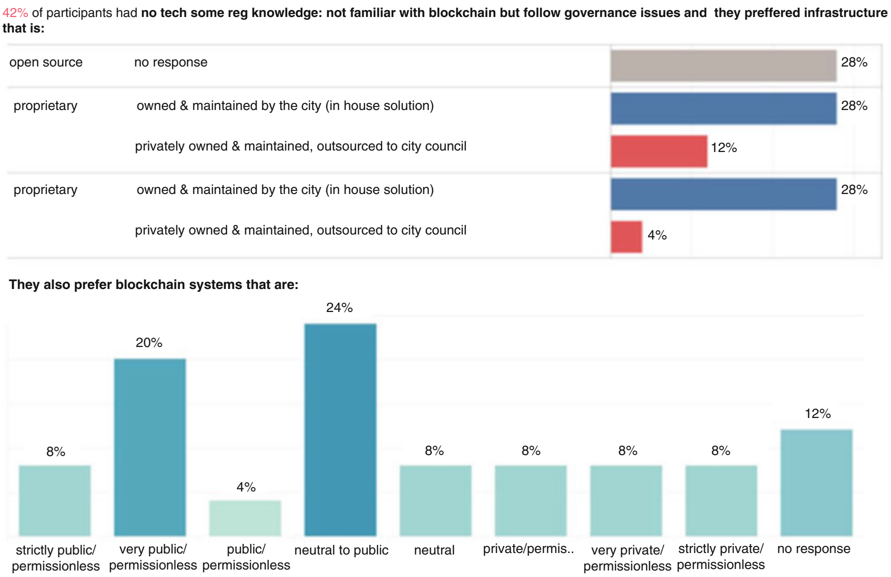
Fig. 3.4 Tableau slide 3 "RegTech preferences" in the green groups

50% participants supporting a change on this level (Fig. 3.8). The "RegTech via Applications" theme shows a similar pattern to the previous general voting on algorithmic governance. The small cryptocurrency/yellow groups had extreme, 0% support for such "governance-by-design" experiments, and 67% participants refused it in the "some tech/some reg" group. The yellow group had the highest 33% ("some tech/some reg") and 50% ("some tech/no reg" group) neutral (no response) reactions.

The consistent support of algo-governance by one of the green groups and the complete rejection by the yellow groups seem to relate to their different expectations about blockchain technologies. We can identify these expectations by having a closer look at the preferred ideal infrastructure summarized in the Tableau story slides 4a, b, and c (theme relates to the question: "How likely would you demand a change of the permissioned blockchain technical infrastructure—architecture of the Lithopia Hyperledger Fabric blockchain network?" responses).

While the yellow groups prefer what they know (cryptocurrency), the largest green "no tech/some reg" group supports an infrastructure that is the opposite of what they experienced in the Lithopy (Fig. 3.5). The group expressed a high trust in open, permissionless, and public blockchains (Lithopy is strictly permissioned) with a strong preference for infrastructure owned and maintained by the city (highest, 56% support for in house solution and permissionless/public blockchain).

The trust in a public (permissionless) blockchain that is owned and maintained by the city was unique to this green group with knowledge of governance ("no tech, some reg"), and it seems to express a hope for a more active role of the government in defining the future public infrastructures. It expresses a preference


42% of participants had **no tech some reg knowledge: not familiar with blockchain but follow governance issues and they preffered infrastructure that is:**

They also prefer blockchain systems that are:

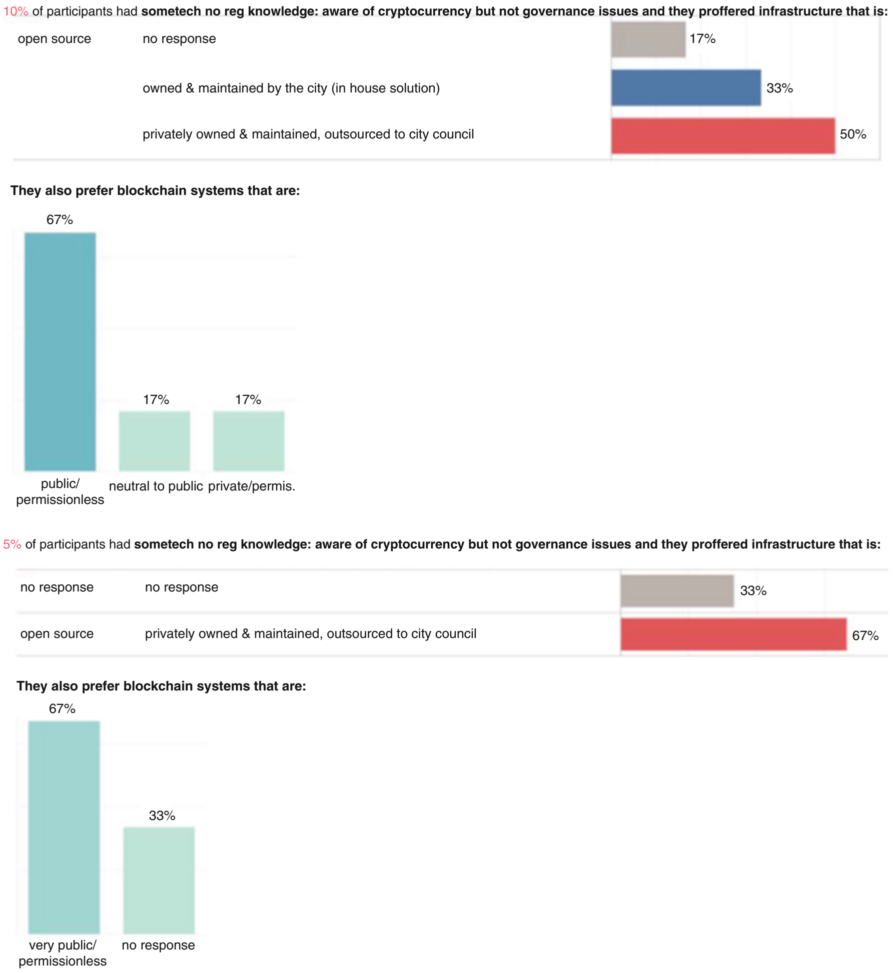
Fig. 3.5 (Tableau slide 4b "Future of AlgoGov")

for an infrastructure that we are seeing implemented as a form of algorithmic sovereignty in places such as Barcelona and Amsterdam as a matter of publicly "owned" algorithms and code: [38] in the DECODE[9] project. Rather than refusing the technology or insisting on a moratorium on the development of such new infrastructure, this group supported the experimental interaction between technology and governance.

The emphasis on city-owned and maintained infrastructure was in sharp contrasts with the two yellow groups (Fig. 3.6). They preferred public permissionless blockchain systems (67–84%) and open source code, but they supported privately owned infrastructure (50–67%). The interest in cryptocurrency seems to confirm strong libertarian tendencies and trust in the reduction of governance to the blockchain (bitcoin) protocol [39–41]. The limited survey confirms that the yellow groups prefer only one type of technology: privately owned public and open blockchains they know from their cryptocurrency experience. Their skepticism toward public institutions seems driven by this insistence on cryptocurrency rather than an interest in exploring further possibilities of DLTs and smart contracts.

The cryptocurrency groups in this sandbox were too small to insist on this interpretation, but they rejected any experimental and hybrid form of algorithmic governance. Their feedback and choices were also very "pragmatic" and concrete rather than exploratory. The failures of the Lithopy sandbox (bias in code preventing Czechs from owning a property) did not lead to any reflection or search for better

[9] https://decodeproject.eu/.



10% of participants had **sometech no reg knowledge: aware of cryptocurrency but not governance issues and they proffered infrastructure that is:**

They also prefer blockchain systems that are:

5% of participants had **sometech no reg knowledge: aware of cryptocurrency but not governance issues and they proffered infrastructure that is:**

They also prefer blockchain systems that are:

Fig. 3.6 (Tableau slide 4b "Future of AlgoGov")

regulation or code but a straightforward refusal of this system: "No (I would not support algorithmic governance) because I am Czech and Lithopia does not let us own property"). They expressed strong skepticism toward government institutions that are incapable of catching up with the technology: "No, the infrastructure (drones, satellites, Internet) will always be controlled by others and I do not believe they can be fairly distributed. However, I might change my opinion if confronted with enough evidence. I am biased against any system."

The sceptical attitude was shared by the participants in the red expert group (total of 7%) claiming full knowledge of both technology and regulation: a majority in this group refused algorithmic governance with statements such as "No for the following reasons: no security when triggering smart contracts, the problem with scaling of blockchain services because of speed, too much regulation from the state. Lithopy needs better implementation that allows greater control of data by citizens and solution that makes transactions faster and easier also better security on all levels to protect your property." In the groups with some or full knowledge of technology (yellow and red), the reason for the negative vote in Lithopy was often the permissioned and closed blockchain system. Most participants in these two groups (100% in the red group and 67% in the yellow on slide 4.b) expressed strong support for permissionless, open, and public blockchain systems (compared to the 30% in the general pool slide 4.a).

The trust in algorithmic governance in these groups depends on this preference for open and public blockchain systems, while in the green group with "no tech" knowledge, the attitudes were more diverse, and the emphasis was on publicly owned rather than only "public" (the emphasis was on in house solutions and ownership of the platform by the city government). This "no tech" green group with knowledge of governance supported public blockchains (32%—same as in the general pool) while the "no tech/no reg" green group supported (69%) closed and private blockchains (32 to 37% in the general pool) expressing a more general skepticism toward technology.

The interest and knowledge of governance and regulation make participants more willing to adopt (or remain neutral) alternative governance models that are partially based on some form of automation. The red and green groups with tech expertise ("full tech/no reg") or no knowledge of the technology ("no tech no reg/some reg") dominated the neutral/no response vote on algorithmic governance, 20% was in the "full tech/no reg", while 80% distributed in the two green groups with no knowledge of the technology. The participants in the extreme categories with expertise or no knowledge of the technology seem to be more open or neutral to algorithmic governance prospects. In contrast, the participants with concrete knowledge of cryptocurrency reject these experiments with governance. The narrower ideas of technology make the participants more "conservative" and refusing any explorations of such future modes of governance.

3.7 RegTech Preferences

Most participants in Lithopy workshops expressed support for the sandbox-style regulation of algorithmic governance (Fig. 3.7): 72% across all groups and only 12% were in favor for the "lean-model" where regulation happens after implementation of the smart contract in the real world, and 7% did not respond. The strongest support for sandboxes was in the green "no tech/no reg" group (93%) that also trusted "RegTech via applications (code)" model (62%) (Fig. 3.4). This group expressed a high distrust in the law/government (54%), market (84%), and industry (60%) possibilities to regulate algorithmic governance.

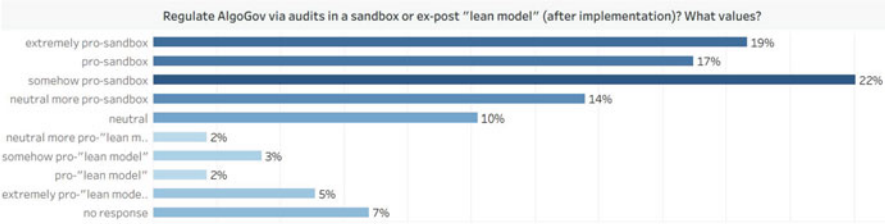


Fig. 3.7 Tableau slide 6 "RegTech via Audits"

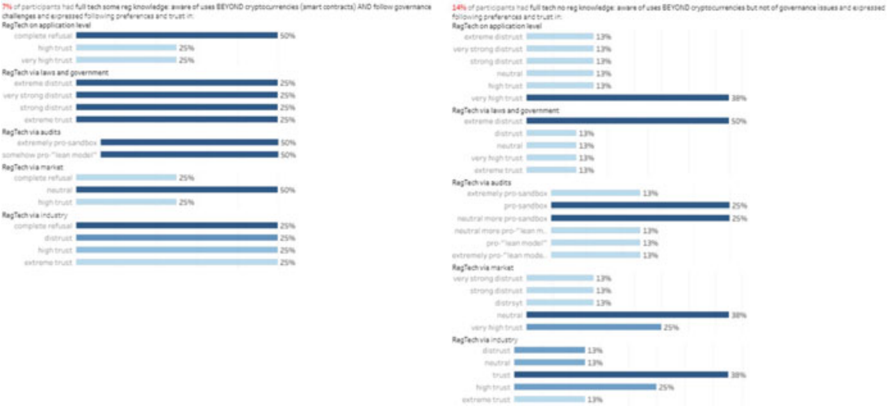


Fig. 3.8 Tableau slide 3 "RegTech preferences" in the red groups

The distrust in these traditional means of regulation was also robust in the green "no tech/some reg" group (Fig. 3.4). The group with knowledge and interest in governance and regulatory issues trusted government and laws (36%) in comparison with the green "no tech/no reg group" (8%). However, some 40% expressed distrust in RegTech via government and laws, 56% expressed various levels of distrust in the market, and strangely 32% distrust industry means of regulation. How to explain this distrust in traditional means of regulation (except industry) by groups with no knowledge of the technology?

We believe that this shows that participants view sandboxes as a new tool, closer to technology, which promises a solution to algorithmic governance's negative effects. Sandboxes mitigate the skepticism toward traditional means of regulation (government, market, industry, non-profit sector) in the general ("no tech") public. This is also supported by the expert group (50–67%). The only exception was the expert red "full tech/no reg" (Fig. 3.8) group with 50% that supports the "lean-model" (50%) of regulations after the infrastructure is implemented for real (Fig. 3.8). Even the other red "full tech/some reg" group supported the lean-model (39%). It is not surprising that the yellow, cryptocurrency group supported

Table 3.1 Summary of responses in the different RegTech categories based on the three groups

% Support of RegTech via	Green; no tech/some reg	Green; no tech/no reg	Red; full tech/some reg	Red; full tech/no reg	Yellow; some tech/some reg	Yellow; some tech/no reg
Trust application/code	64	62	51	50	0	51
Distrust	12	31	39	50	67	0
Trust government/laws	36	8	26	25	0	33
Distrust	40	54	63	75	67	33
Trust in market	36	16	25	25	67	67
Distrust	56	84	25	39	0	17
Trust in industry	52	31	76	50	67	33
Distrust	32	60	13	50	0	0
Audits in sandbox	68	93	63	50	67	67
Lean-model	8	0	39	50	0	0

the sandbox model of the regulation (67%) since similar blockchain testnets are common in the Ethereum community.

While the trust in government as a mean of controlling algorithmic governance was strong in the green "no tech/some gov" group (36%), it was surprising to see much stronger support for RegTech via industry (52%) (Table 3.1). In the workshop, we offered unique examples of such industry RegTech tools, such as corporate ISOs and certification schemes, social responsibility (CSR) strategies, ethics boards committees, and codes of conduct. This RegTech via industry was also supported by the expert red groups (76% in the "full tech/some reg" and 50% in the "full tech/no reg" groups) and cryptocurrency yellow groups (67% in the "some tech/some reg" and 33% in the "some tech/no reg" groups). This positive image of the RegTech via industry is not supported by the present actions of the digital industry (Facebook, etc.), but participants seem to express a strong hope for this to happen. While industry seems to be something most groups agree upon, there were a vast difference in their response to regulation by market forces and government.

The most skeptical toward the markets were the two green "no tech" groups (56% and highest 84% in the "no tech/no reg" group), closely followed by the red groups (25% in the expert "full tech/full reg" that was equal to the support and more substantial 39% in the "full tech/no reg" group). Only the yellow, cryptocurrency groups (Fig. 3.9) seem to have trust in the market forces with 67% support. To summarize, there is a shared skepticism toward the market but also government and most participants support more hybrid forms of regulation that can happen in the industry or via the sandboxes and alternative possibilities of regulation via code.

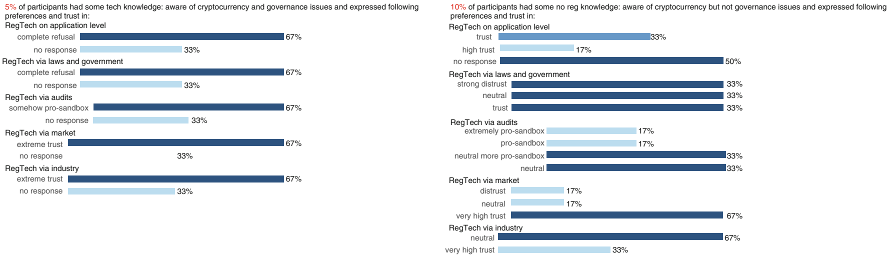


Fig. 3.9 Tableau slide 3 "RegTech preferences" in the yellow groups

3.8 Conclusion

The ambiguous and contentious legacy of the "parastatals," post-colonial attempts at sovereignty through public–private partnerships summarizes well the challenges of hybrid algorithmic governance in the sandboxes. To control the algorithmic "colonialization" by emerging technologies, such as blockchains or AIs, the policy institutions and communities have to engage with the "enemy" in a tentative and contained manner that gives them relative rather than absolute control. Hybrid sandboxes for policy and design are like parastatals or "special economic zones" – environments for experimental governance. They offer a relative autonomy to the algorithms while providing control and feedback on the regulations. Sandboxes offer a pragmatic alternative to the "whitewashing"İ of algorithmic services by most present industry and government frameworks and ethical guidelines. They increase citizens' and communities' agency over their algorithmic futures and provide tools and methods how to negotiate controversies. The pragmatic focus is visible in the feedback that supported "RegTech via industry" efforts by most participants (at least in this sandbox).

The sandbox experiment conducted via Lithopy shows how participants with an interest in governance issues and insufficient knowledge of technology are open to technical interventions and regulation via code and willing to try hybrid and experimental governance models. The sandbox offers these groups a safe space to test the various arrangements and ideas and help them communicate with groups with more extreme (conservative) algorithmic agendas. Because the reasons for supporting or rejecting algorithmic governance seem complex, we see a powerful reason to support such hybrid experiments further.

The hybrid governance arrangements in a sandbox can also support better adaptation and exploration of both regulation and code and test how to embed not only the legal but also cultural and social norms directly into the corporate-owned and machine-readable code. As explored in Lithopy, the sandbox offers an alternative to the ethical guidelines and frameworks supporting ex-ante and ex-post regulations difficult to implement or to the over-promissory or the technocratic "governance-by-design" interventions. To map and evaluate the impact of such algorithmic "parastatals" and governance hybrids of code, data, and regulations, we are using the metaphor of the "trading zones" and "innovation through dissonance" that engage the citizens not only as test subjects or users of future services but actual stakeholders.

The experimental governance in the sandbox as an algorithmic parastatal combined the stakeholder negotiations of regulations with design reiterations. It supported tactical and situated, individual decisions on algorithmic futures that provided feedback for further work. With our Lithopy 2020 workshop participants, we conducted another survey in June 2020 on contact tracing apps during the coronavirus crisis to compare how their attitudes toward algorithmic governance and control changed (it seems they did not—discussed in a forthcoming article). Instead of optimized and efficient algorithmic procedures or strict laws demanding moratoria on specific technologies, sandboxes offer an alternative supporting a democratic process that engages the public and citizens in the decisions about their common future.

References

1. D.W. Arner, J.N. Barberis, R.P. Buckley, FinTech and RegTech in a nutshell, and the future in a sandbox. SSRN Elect. J. (2018). https://doi.org/10.2139/ssrn.3088303
2. E. Gromova, T. Ivanc, Regulatory sandboxes (experimental legal regimes) for digital innovations in BRICS. BRICS Law J. **7**, 10–36 (2020)
3. J. Fjeld, N. Achten, H. Hilligoss, A. Nagy, M. Srikumar, Principled artificial intelligence: mapping consensus in ethical and rights-based approaches to principles for AI. Berkman Klein Center Research Publication **2020**(1) (2020)
4. T. Hagendorff, The ethics of AI ethics: an evaluation of guidelines. Mind. Mach. **30**, 99–120 (2020)
5. A. Cavoukian, Privacy by design – the 7 foundational principles – implementation and mapping of fair information practices, in *Information and Privacy Commissioner of Ontario* (2009), p. 5
6. D.K. Mulligan, K.A. Bamberger, Saving governance-by-design. Calif. Law Rev. **106**, 697–784 (2018)
7. B. Friedman, P. Kahn, Value sensitive design: theory and methods, in *University of Washington Technical* (2002), pp. 1–8
8. J. Danaher, M.J. Hogan, C. Noone, R. Kennedy, A. Behan, A. De Paor, H. Felzmann, M. Haklay, S.M. Khoo, J. Morison, M.H. Murphy, N. O'Brolchain, B. Schafer, K. Shankar, Algorithmic governance: developing a research agenda through the power of collective intelligence. Big Data Soc. **4** (2017)
9. K. Crawford, J. Schultz, AI systems as state actors. Columbia Law Rev. **119** (2020)
10. B.D. Mittelstadt, P. Allo, M. Taddeo, S. Wachter, L. Floridi, The ethics of algorithms: mapping the debate. Big Data Soc. **3**, 205395171667967 (2016)
11. I. Rahwan, Society-in-the-loop: programming the algorithmic social contract. Arxiv preprint (2017)
12. I. Rahwan, Society-in-the-loop: programming the algorithmic social contract. Ethics Inf. Technol. **20**, 5–14 (2018)
13. N. Elkin-Koren, Contesting algorithms: restoring the public interest in content filtering by artificial intelligence. Big Data Soc. **7**, 205395172093229 (2020)
14. J.H.-j. Choi, L. Forlano, D. Kera, Situated automation: algorithmic creatures in participatory design, in *Proceedings of the 16th Participatory Design Conference 2020-Participation (s) Otherwise*, vol. 2 (2020), pp. 5–9
15. K. Shilton, Engaging values despite neutrality. Sci. Technol. Human Values **43**, 247–269 (2018)

16. M. Sloane, E. Moss, O. Awomolo, L. Forlano, Participation is not a design fix for machine learning. Arxiv preprint (2020)
17. S. Hassan, P. De Filippi, The expansion of algorithmic governance: from code is law to law is code. Field Act. Sci. Rep., 88–90 (2017)
18. A. Sinnreich, Four crises in algorithmic governance. Ann. Rev. Law Ethics, **26**, 181–190 (2018). Available at SSRN: https://ssrn.com/abstract=3650593
19. K. Yeung, 'hypernudge': Big data as a mode of regulation by design. Inf. Commun. Soc. **20**, 118–136 (2017)
20. M. Ziewitz, Governing algorithms: myth, mess, and methods. Sci. Technol. Human Values **41**(1), 3–16 (2016)
21. L.D. Introna, Algorithms, governance, and governmentality: on governing academic writing. Sci. Technol. Hum. Values **41**, 17–49 (2016)
22. J.A. Kroll, J. Huey, S. Barocas, E.W. Felten, J.R. Reidenberg, D.G. Robinson, H. Yu, Accountable algorithms. Univ. Pennsylvania Law Rev. **165**, 633–705 (2017)
23. M.K. Lee, A. Jain, H.J.I.N. Cha, S. Ojha, D. Kusbit, Procedural justice in algorithmic fairness: leveraging transparency and outcome control for fair algorithmic mediation. Proc. ACM Human Comput. Inter. **3** (2019)
24. D.L. Burk, Algorithmic fair use. Univ. Chicago Law Rev. **86** (2019)
25. B. Shneiderman, The dangers of faulty, biased, or malicious algorithms requires independent oversight. Proc. Natl. Acad. Sci. U. S. A. **113**, 13538–13540 (2016)
26. R. Binns, Algorithmic accountability and public reason. Philos. Technol. **31**(4), 543–556 (2018)
27. N. Diakopoulos, Accountability in algorithmic decision making. Commun. ACM **59**, 56–62 (2016)
28. M.K. Lee, D. Kusbit, A. Kahng, J.T. Kim, X. Yuan, A. Chan, D. See, R. Noothigattu, S. Lee, A. Psomas, A.D. Procaccia, WeBuildAI: participatory framework for algorithmic governance. Proc. ACM on Human Comput. Inter. **3**, 1–35 (2019)
29. M. Hildebrandt, Algorithmic regulation and the rule of law. Philos. Trans. R. Soc. A Math. Phys. Eng. Sci. **376** (2018)
30. L. Edwards, M. Veale, Enslaving the algorithm: from a "right to an explanation" to a "right to better decisions"? IEEE Sec. Priv. **16**(3), 46–54 (2018)
31. F. Fitsilis, Administrative and judicial decisions on advanced algorithms, in *Imposing Regulation on Advanced Algorithms* (Springer, Berlin, 2019), pp. 25–53
32. F. Fitsilis, *Imposing Regulation on Advanced Algorithms* (Springer International Publishing, New York, 2019)
33. B.W. Wirtz, J.C. Weyerer, C. Geyer, Artificial intelligence and the public sector — applications and challenges. Zenodo preprint (2018)
34. M. Aitken, E. Toreini, P. Carmichael, K. Coopamootoo, K. Elliott, A. van Moorsel, Establishing a social licence for financial technology: reflections on the role of the private sector in pursuing ethical data practices. Big Data Soc. **7**, 205395172090889 (2020)
35. D. McQuillan, People's councils for ethical machine learning. Soc. Media Soc. **4**, 205630511876830 (2018)
36. D.R. Kera, Anticipatory policy as a design challenge: experiments with stakeholders engagement in blockchain and distributed ledger technologies (BDLTS), in *Advances in Intelligent Systems and Computing*, vol. 1010 (Springer, Berlin, 2020), pp. 87–92
37. D.R. Kera, M. Kraiński, J.M.C. Rodríguez, P. Sčourek, Y. Reshef, I.M. Knoblochová, Lithopia: prototyping blockchain futures, in *Conference on Human Factors in Computing Systems – Proceedings* (2019)
38. D. Roio, *Algorithmic Sovereignty*. PhD thesis, University of Plymouth (2018). http://hdl.handle.net/10026.1/11101
39. J. Flood, L. Robb, Trust, Anarcho-capitalism, blockchain and initial coin offerings. SSRN Elect. J. (2017)
40. B.D. Trump, E. Wells, J. Trump, I. Linkov, Cryptocurrency: governance for what was meant to be ungovernable. Environ. Syst. Decis. **38**, 426–430 (2018)
41. A. Wright, P. De Filippi, Decentralized blockchain technology and the rise of lex cryptographia. SSRN preprint (2015)

Chapter 4
Information Privacy in Decentralized Applications

Moritz Platt, Ruwan J. Bandara, Andreea-Elena Drăgnoiu, and Sreelakshmi Krishnamoorthy

4.1 Introduction

The setting-in of “crypto winter,” erasing close to two-thirds of the total global market capitalization of cryptocurrency in 2018 [1, p. 99], put a damper on the inescapable Blockchain[1] hype of the mid-2010s. Yet, many use cases exist in which the technology can live up to its promise of process optimization, transparency, and cost reduction by ensuring the immutability and verifiability of shared data. Privacy concerns, while fundamental to Blockchain applications in highly regulated enterprise environments, have only been secondary aspects of early Blockchain systems.

[1] For this chapter, the term “Blockchain” refers to any type of “Distributed Ledger Technology,” even if it does not make use of the “block” concept, first described by Nakamoto [2].

M. Platt (✉)
Department of Informatics, King’s College London, London, UK
e-mail: moritz.platt@kcl.ac.uk

R. J. Bandara
Faculty of Business and Law, University of Wollongong, Wollongong, NSW, Australia
e-mail: hmrjb180@uowmail.edu.au

A.-E. Drăgnoiu
Department of Computer Science, University of Bucharest, Bucureşti, Romania
e-mail: andreea-elena.panait@drd.unibuc.ro

S. Krishnamoorthy
Department of Computer Science and Information Systems, Birla Institute of Technology and Science, Pilani, India
e-mail: h20180130@pilani.bits-pilani.ac.in

M. H. ur Rehman et al. (eds.), *Trust Models for Next-Generation Blockchain Ecosystems*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-75107-4_4

Recently, many disruptive technologies such as big data, the Internet of Things (IoT), artificial intelligence (AI), online platforms, and Blockchain technology have evolved, rivetingly rendering our world increasingly hyperconnected. In this hyperconnected environment, we are increasingly interacting with machines to carry out our tasks, while machines interact with one another to deliver different tasks at the same time. Moreover, many parts of individuals' lives today take the shape of a digital trace or are datafied [3]. With *datafication*, there is a dramatic shift in the collection, storage, processing, and transmission of individuals' data among numerous parties [4, 5]. Also, the same digital traces have now resulted in the extensive commoditization of data and commercial surveillance, or what is referred to as "data capitalism" [6]. For instance, the rapid growth of online platform-based e-commerce, such as Amazon and eBay, has intensified the number of online transactions conducted and, consequently, the scale, scope, and velocity of the consumer data transmitted [7, 8]. Moreover, other technologies that facilitate these transactions, such as virtual assistants (e.g., Siri, Alexa) and other IoT-enabled devices, have allowed these devices to interact with sellers on behalf of consumers and also to deal with consumers' data [4, 9]. Therefore, these disruptive technologies have created a complex ecosystem in which various interconnected technologies and numerous actors mediate in the relationship between a consumer and a seller [10].

The emergence of these disruptive technologies and mass commoditization of personal data have generated irreversible social effects, including algorithmic discrimination and loss of autonomy and privacy [3]. This chapter focuses on threats to privacy in the context of Blockchain. Therefore, we focus on information privacy, as opposed to physical privacy. Information privacy is concerned with the flow of information–what, by whom, why, and how information is collected and used [10]. Privacy issues are overwhelming as diverse types of sensitive data are collected by different players, not only for primary purposes but also for unintended and secondary purposes, without the proper consent and knowledge of individuals. For instance, third parties, such as data brokers, buy, compile, transform, and resell data on a massive scale—these players are not known to most of the individuals [8, 11]. There is no difference in the Blockchain context. As Zamani and Giaglis [12] assert, "rather than complete disintermediation, the most probable scenario is that of new types of intermediaries finding previously unthinkable roles to play in mediating Blockchain-based economic transactions." Hence, there is a necessity to understanding privacy in the utilization and governance of Blockchain technology.

Misunderstanding the information privacy properties of Blockchain can have serious consequences, as seen in the context of deanonymization of Bitcoin transactions [13]. Furthermore, in academia, as well as in industry, confusion around the appropriateness of Blockchain technology in privacy-sensitive scenarios is widespread. This may take effect in two ways: by applying Blockchain solutions to scenarios in which they do not provide a benefit and are threatening to privacy and by dismissing Blockchain in scenarios where it would be appropriate. Previous work has rarely focused on issues of privacy exclusively, and where it has, it was approached by categorizing solutions, not requirements. Analyzing requirements, however, allows us to move beyond the practical constraints of implementation and to focus on the nature of the underlying business problem.

We aim to contribute to the literature on trust issues in Blockchain ecosystems by answering the following research questions: what are the key categories in the context of privacy that allow us to differentiate between decentralized applications? What are prominent combinations of these categories among popular decentralized applications? Can any recommendations for future work be derived from these findings?

Answering these questions will benefit researchers in Blockchain protocol design by showing what privacy requirements future protocols will be expected to address. It will also benefit the wider community, including practitioners, by providing a taxonomy of privacy requirements, thus making novel applications classifiable in terms of their privacy characteristics. Finally, answering the research questions will help identify the need for advanced cryptographic tools in next-generation Blockchain ecosystems.

4.2 Previous Work

The wider literature provides the following types of outputs relevant to this work:

1. Taxonomies of Blockchain platforms. These often compare implementation details including consensus mechanisms and cryptographic methods or conceptual goals like accessibility (private vs. public Blockchains).
2. Papers that describe novel information-processing systems using Blockchain.
3. Reviews analyzing several applications often focus on certain sectors or industries. These works commonly categorize applications according to traits declared relevant by the authors.
4. Taxonomies of applications that structure the body of knowledge in Blockchain application research.
5. Meta-reviews that combine the results of multiple reviews, oftentimes applying quantitative methods to them.

In this chapter, we analyze application papers and reviews to derive a taxonomy of Blockchain applications. The distinguishing factor between our work and the previous literature of the same type is that we focus exclusively on characteristics relevant to privacy by analyzing recent publications from 2018 and beyond.

The wider literature provides similar works, such as the taxonomy published by Labazova et al. [14] that defines eight characteristics of Blockchain systems. Among those, "read/write access," "consensus mechanism," and "anonymity level" are defined. A recently published taxonomic decomposition of Blockchain-based technologies by Tasca and Tessone [15] names data privacy as a differentiating factor by differentiating between "built-in" data privacy architectures, i.e., those in which data is obfuscated or encrypted as part of the core protocol, and "add-on" data privacy architectures, i.e., those that rely on off-chain methods to keep sensitive information private.

In addition to these taxonomies, in recent years, comprehensive reviews of Blockchain applications discussed in academia have been published. In one of the few quantitative outputs in the field, the authors evaluate in which domains Blockchain is widespread using a systematic literature review [16]. In a similar paper, nine main sectors in which applications are deployed are described [17]. Within the privacy realm, the authors focus on applications that enhance data security and on infrastructure around digital identity services and solutions that combine hardware and software. Despite these efforts, there is a lack of studies approaching privacy in Blockchain from a perspective of requirements.

4.3 Privacy-Preserving Techniques in Blockchain Applications

The development to date of use cases in Blockchain can be categorized in two phases [18]: *Blockchain 1.0*, in which cryptocurrencies, notably the experimental peer-to-peer electronic cash system “Bitcoin” [2], dominate, and *Blockchain 2.0*, in which “smart contracts” are used to enable more complex applications. “Smart contracts,” a concept introduced by Szabo [19], entail encoding contractual clauses in program code, thus making them automatically enforceable by a Blockchain system. To evaluate data through smart contracts, it has to be validatable. This introduces a fundamental threat to information privacy, since, depending on the consensus protocol of the Blockchain and its governance structure, a wide variety of—potentially untrustworthy—actors might act as validators.

4.3.1 Privacy Techniques for Blockchain 1.0

Blockchain technology was not conceived to tackle issues of information privacy: cryptocurrencies, the archetypal applications implemented on Blockchains, did rely on the public replication of all transaction data in the clear, an approach that is contradictory to the goal of giving participants control over who can access their data. Blockchain technology extensively uses asymmetric cryptography, i.e., pairs of public and private keys, where the public key usually corresponds to the public address of the participant and the private key is used to sign their transactions. The fundamental assumption of the inventors of these early stage systems was that pseudonymous transactions, identifiable only by an obscure public key, would not allow for conclusions to be drawn about the identity of a user in the physical world. This assumption proved to be incorrect, as evidenced by the wide body of research around transaction deanonymization [13].

The decentralized nature of Blockchain facilitates trust and transparency across the network, thereby preventing changes to transaction history. Moreover, the risk of data loss is small, as Blockchain transactions are stored in a decentralized peer-to-peer network that replicates data. While this has an upside for use cases that preliminarily require tamper-proof data replication and redundant data storage, the downside is that data cannot easily be removed. This puts privacy at risk in scenarios where the data stored on the chain is sensitive. What constitutes sensitive data is not always obvious, with recent experiments showing that even hashed data can be considered personally identifiable information [20].

The fundamental conflict between verifiability and privacy has been approached in three ways in Blockchain 1.0 applications.

Applications can address it implicitly, by composed architectures in which they combine on-chain elements with off-chain elements. Here, sensitive data, such as personally identifiable information, would be stored in private, centralized data repositories and only be loosely referenced on-chain. This architectural approach implies that, even if deanonymization of on-chain data occurs, no sensitive data is made available as it would be stored on-chain.

In addition to these architectural measures, privacy can be addressed on the protocol level through anonymization sequences that utilize middle persons to conceal the origins of transactions. This means that, while it can be proven that a pseudonymous entity has participated in some form in an exchange of funds, the chain of the provenance of cryptocurrency holdings is obfuscated.

Lastly, some Blockchains make use of advanced cryptographic algorithms, such as ring signatures or zero-knowledge proofs that allow parts of the data model, in the case of cryptocurrencies commonly the amounts, to stay private. These approaches are not mutually exclusive, and application architects can combine them arbitrarily.

4.3.2 Privacy Techniques for Blockchain 2.0

Blockchain 2.0 is characterized by an explosion in the complexity of the applications deployed. While earlier applications could only use a small set of select functionalities around sending and receiving cryptocurrency, this new phase allowed arbitrary logic to be expressed in smart contracts. From the diversity of data associated with this more complex logic, challenges arose for privacy, as now, data was no longer limited to the manageable domain of cryptocurrency but extended to highly sensitive domains. Modern Blockchain applications, therefore, are much more intricate in terms of their data model, which is why new technologies for privacy preservation are the subject of research.

A more flexible user model is one of the main aspects characterizing this phase. Where previously a public key was the predominant means of identification, the integration with various off-ledger systems meant that participants needed a convenient way to transact on a Blockchain, based on identity information issued outside of it. This need encouraged the appearance of privacy-preserving identity management models, e.g., the self-sovereign identity (SSI) management model. Those models benefit from the intrinsic advantages of Blockchain technology, since central authorities are not required to control and manage applications and, instead, this responsibility is passed on to the user [21]. Users can privately identify themselves when transferring digital assets through different means, including decentralized identifiers (DID), DID documents, verifiable claims, identity attributes, or proofs of identity [22]. Moreover, to preserve privacy in the SSI model, users can make use of zero-knowledge proofs (ZKPs), proving their identity to a verifier through data stored on the Blockchain.

ZKP can also be applied outside of the context of user identity: a protocol in which the *prover* assures a *verifier* about a statement's validity without revealing other information is useful in many application contexts. The processing of encrypted data (e.g., through homomorphic encryption) and trusted execution environments are additional techniques for maintaining information privacy. Homomorphic encryption is an encryption method that enables participants and validators to perform calculations and process encrypted data without previously decrypting it. The results of the calculations are in an encrypted form whose decrypted output is the same as if the operations had been executed on plain data. Trusted execution environments constitute tamper-resistant processing contexts that allow for data to be processed privately under isolated execution conditions.

All of these techniques allow for the decoupling of the *verifiability* of data from its *visibility*. This means that a verifier might be able to conclude the correct state of data on the Blockchain without being able to understand its meaning. While these technologies only start to receive commercial attention in the context of Blockchain applications, they have the potential to have a significant impact on application architectures.

4.4 Applications

The variety of applications in Blockchain is unwieldy. To illustrate the size of the market, the number of smart contracts deployed to Ethereum, a popular smart contract platform, can serve as an indication. The "Smart Corpus" project, a repository of Ethereum smart contract code, alone is growing at a rate of 100 smart contracts per day [23]. To make sense of this number, the relationship between smart contracts and applications needs to be understood. Oliva et al. [24] describe how some smart contracts constitute stand-alone applications, while others form part of the back ends of larger applications. This suggests that the number of applications is smaller than the number of smart contracts deployed. Taking all that into account, the actual number of viable applications that are in use on the Ethereum platform is likely much smaller than the number of smart contracts deployed, but still sizable. Attempts to enumerate all the applications on the Ethereum platform in particular, let alone throughout the diverse Blockchain ecosystem, are therefore unfeasible.

With this chapter, we do not intend to provide a comprehensive review, as recent years have already produced extensive works to this effect (cf. Sect. 4.2). Instead, we want to represent a broad selection of use cases that have attracted academic and commercial attention. Methodologically we approach this selection through a literature review, focusing on the most prominent fields in which Blockchain applications have been proposed.

Using the meta-review provided by Jaoude and Saade [16] as an approximation of the relevancy of Blockchain in a given field, we identify five focus areas: Internet of Things (IoT), energy, finance, healthcare, and government. Within these areas, a literature review is conducted to discover prevalent applications. To create a universal taxonomy, it is desirable to also consider areas that have had less attention in academia. We, therefore, take miscellaneous notable applications outside of the focus areas into account.

Applications in IoT, specifically smart home and smart healthcare, have strong privacy implications [25] as they are concerned with personally identifiable information. The principle of the inviolability of the home is legally safeguarded in many contexts and the sensitiveness of patient data in healthcare settings places high regulatory demands on applications deployed in these contexts. Here, data must not leak beyond those that are authorized, authorization must be revocable, and a reliable audit trail must be kept.

Use cases in the energy field often overlap with those in the IoT field. They mostly revolve around energy trading and metering. The privacy requirements of these use cases are not as pronounced as those of other fields such as healthcare. Except for applications of peer-to-peer trading [26], they constitute business-to-consumer or business-to-business transactions where the relevant data is metering or payment data. While this allows for conclusions to be drawn about consumer behavior, this data is not as sensitive as other forms of personally identifiable information.

The key technologies operating on Blockchain are cryptocurrencies that facilitate the distributed transfer of currency tokens. As early Blockchain applications, financial use cases started in the consumer-to-consumer space, where, due to the pseudonymous nature of account-related information, privacy concerns were secondary. When institutional entities started to engage with Blockchain, the problem of exchanging data that was legally or commercially sensitive emerged. In addition to these privacy requirements, use cases in the financial industry often have high-performance requirements that might conflict with privacy-preserving cryptographic techniques.

The sensitive nature of healthcare data makes it a highly relevant domain when it comes to information privacy, specifically when tangible negative consequences can arise from privacy violations [64] or where data collected could be abused for mass surveillance [62]. Healthcare data has a privileged status in many regulatory contexts. Blockchain holds promise as integration technology for medical data, as healthcare environments are known for the siloed nature of the data repositories used [27]. Fine-grained access control is commonly required where patient data is concerned, and a reliable and tamper-proof audit trail of data access is often a legal requirement.

Use cases in government take different forms. Some are focused on intra-government communication and assume the trustworthiness of all participants in the Blockchain, thereby alleviating privacy concerns, others supply services directly to their constituents. Where this is the case, privacy requirements materialize similarly to business-to-consumer contexts, in which a central entity (here, the government) interacts with entities on a lower hierarchical level (here, the constituents). In the latter case, privacy can be addressed via central validating functions. A noteworthy use case, with characteristics that differ from that, is the use case of public votes, where a central validator (i.e., the current government) might be disincentivized from acting as an honest validator. These use cases are among the most challenging from a privacy perspective, as they require technical solutions to safeguard the secrecy of the ballot while having to prevent electoral fraud.

4.5 Taxonomy

Against the background of the works described in Sect. 4.2, this taxonomy aims to provide additional value by focusing on aspects relevant to privacy exclusively. Distilling findings from the wider literature into a minimal taxonomy of aspects that are drivers for privacy in Blockchain requires considerable interpretation. Outputs from the wider information systems community can serve as a starting point for developing a concise set of dimensions.

While Blockchain is often presented as “an institution ensuring the fairness of decisions by voting based on a consensual procedure” [28], it has been shown that whether an application can deliver on this promise is heavily dependent on the underlying consensus algorithm [28, 70]. Whether an explicit governance structure that forms the foundation of a use case exists is, therefore, a crucial decision gate. It is covered by the “accessibility” dimension (cf. Sect. 4.5.1).

Bélanger and Crossler [29, p. 1035] recommend to researchers in the field of information privacy to focus on studying information privacy concerns “beyond the individual,” focusing on the “organizational level.” This seems particularly appropriate since this chapter predominantly discusses “business-to-business” and “business-to-consumer” scenarios. From this recommendation emerges the “hierarchicality” dimension (cf. Sect. 4.5.2).

Lind and Tyler [30, p. 200] advance some practical recommendations to researchers in organizational design. Among those is the recommendation to focus on procedures, and especially their fairness. Most applications studied in this chapter are implemented via smart contracts (cf. Sect. 4.1). Smart contract theory suggests fairness is a given, through the assumption that deterministic computer contracts do not leave room for interpretation [31]. The reality is, however, that smart contracts are by no means without ambiguity and that many vulnerabilities can arise from this [32]. Since creating validation logic that performs well in an adversarial environment like a public Blockchain is so challenging, the advice to practitioners is commonly to embrace minimalistic designs. This also enables data minimization, a technique to limit data collection and subsequent processing. With an initiative for simplicity of contract logic and data minimization comes the decision of which parts of an application data model need to be made available for validation. These aspects are further distinguished in the "verifiability" dimension (cf. Sect. 4.5.3).

4.5.1 Accessibility

The application taxonomies summarized earlier as part of the section on previous work (cf. Sect. 4.2), directly or indirectly, include accessibility as a differentiating factor. The wider literature commonly differentiates between "permissioned" and "permissionless" Blockchains. Gamage et al. [33, p. 114] characterize permissioned Blockchains as systems in which write access is restricted to a limited subset of participants, and permissionless Blockchains as systems in which "anyone can openly read data, inspect data, and participate in validation and writing of the data in accordance with [a] consensus protocol." While a permissionless approach is needed to implement the original vision of creating censorship-resistant systems without formal governance structures, such an approach is not required to solve many common business problems.

Table 4.1 compares open and closed accessibility requirements. While these align closely with the permissioned/permissionless dimension described above, in reality, these do not always coincide. Most commonly, deviations occur when permissionless technology is applied to use cases with closed accessibility requirements.

Table 4.1 Different governance paradigms imply different accessibility requirements

Accessibility	Description
Open	Open use cases are designed to allow any participant with the technical means (i.e., access to client software and network connectivity) to partake in the use case. No formal manual admission process is needed
Closed	Closed use cases require central entities or consortia to admit participants to the network. Admission criteria are flexible and depend on the use case, ranging from minimal (e.g., basic identity checks) to extensive (e.g., proof of an established business relationship with the application operator)

Accessibility is relevant to privacy as it defines who, in principle, has access to data on a Blockchain. In a closed system, only participants explicitly admitted can read and write data.

Usually, closed requirements are satisfied by systems with a single "gatekeeping" authority or a consortium (also referred to as a "committee" or "federation") of authorities. While there are diverse potential implementations of such systems, they are usually built around "proof-of-authority" consensus protocols or use dedicated transaction validation and ordering components, thus not requiring any consensus at all, since they are operated by a single logical entity. Intuitive insight is that decentralization comes at a cost [34]. Whether the advantages of Blockchain in closed scenarios (transactional transaction semantics, atomic ordering of transactions, immutability, etc.) outweigh the cost of decentralization is a topic of ongoing debate with many viewpoints. Wüst and Gervais [35] argue that whether a Blockchain provides value in such a scenario is dependent on the appropriateness of utilizing a trusted third party. They argue that should the use case allow for it, an "always online" trusted third party can have write operations delegated to it and can function as a verifier for state transitions, thus making decentralization—and thereby Blockchain—unnecessary.

4.5.2 Hierarchicality

Bitcoin's predecessor "B-Money" was conceived before a backdrop of crypto-anarchy and contemplated a future in which governments as central authorities are "not [only] temporarily destroyed but permanently forbidden and permanently unnecessary" [36]. As such, early use cases in Blockchain could be characterized as libertarian and anti-authoritarian. The only element introducing the notion of hierarchicality was mining, a process in which those who invest in computing resources are in return more likely to qualify for a position of power on the network.

Hierarchicality is relevant to privacy as it shows who can claim a legitimate interest when inspecting data on the Blockchain. Actors on a higher hierarchy level often require more visibility of application data than those on a lower level.

Modern use cases, i.e., those involving the corporate realm and specifically regulated industries, however, prescribe fine-grained roles and responsibilities. The original idea of a system in which participants all have identical permissions

Table 4.2 Different applications bring with them different requirements for hierarchical relationships between participants

Hierarchicality	Description
Flat	Use cases that imply a flat hierarchy assume that all participants have the same intrinsic permissions. The absence of hierarchical relationships suggests a good fit with traditional public Blockchain architectures
Two layer	A common requirement in many of the use cases analyzed is the need for a superordinate member on the network that assumes a controlling function or acts as a trusted interface to external components
Tall	Those use cases that exhibit multiple layers of controlling functions require a tall hierarchy, akin to a role-based access model. This is often necessary to reflect more complex trust relationships (e.g., customer, operator, regulator), which occur in complex corporate use cases or regulated industries

and in which their identity "cannot be linked to their true names or physical locations" [36] does not reflect the realities in these use cases. Rather, these modern use cases exhibit varying degrees of hierarchical relationships between participants (cf. Table 4.2)

4.5.3 Verifiability

One of the key contributions of Blockchain is the strong validity and immutability guarantee provided by smart contracts. In the case of early "Blockchain 1.0" [18] platforms, specifically cryptocurrencies like "Bitcoin," the data being validated is limited to account balances and transfers of balances. Even this simple data model can enable complex use cases if more complex downstream activities are conducted in "off-chain" systems.

Not all aspects of an application data model need to be validated *on-chain*. For many use cases, it is sufficient to merely *reference* data in *off-chain* systems on the Blockchain. As such, this requirement plays a decisive role in privacy as it determines what data is stored *on-chain* and thereby potentially visible to others.

While more powerful "Blockchain 2.0" [18] platforms allow for the implementation of the entirety of a business's complex application logic "on-chain," this is rarely advisable from a design perspective for reasons of cost, speed, and privacy. In terms of data privacy, many Blockchain applications face a dilemma: to utilize the validation potential of smart contracts, data needs to be inspectable by others, but

Table 4.3 Viewing transaction data as sets of attributes, requirements for different levels of data validation and visibility can be defined

Verifiability	Description
None	In cases where Blockchain is employed not for its validation guarantees but as a message-passing infrastructure only, there is no on-chain validation requirement (i.e., in the form of smart contracts)
Fully bilateral	Fully bilateral validation implies that all participating parties must be able to validate all data in a joint data model. This is the case where there is no significant need for individual record-keeping by either participant
Partially bilateral	In a partially bilateral use case, significant parts of the joint data are not required to be shared with counterparties for validation purposes. This is the case in use cases that require joint mechanisms on a low level (e.g., account balances, sensor readings, etc.) but not on a high level (e.g., inventory)
Supported bilateral	This class is similar to the "partially bilateral" requirement outlined before. In addition to both parties validating some shared data, the use case mandates a third party, often an operator or neutral third party, acting as an oracle
Validator driven	This requirement introduces a third party as the exclusive validator of participant data. Here, participants would not be involved in validating counterparty states but would rely on a third party to do so. Such a requirement can occur in a scenario where participants aim to keep their data private, or because it is the exclusive responsibility of the validator to ensure consistent data

sharing data might be against the best interests of the party holding the data (e.g., because it constitutes an industrial secret) or even illegal.[2] Consequently, which parts of the data model need to be made available to which validators is a key question for Blockchain use cases (cf. Table 4.3).

ZKPs, homomorphic encryption and trusted execution environments (cf. Sect. 4.3) have been used to address this dilemma. These techniques enable computation on data (and thereby its validation) without revealing the data to the one performing the computation. Therefore, verifiability is not to be confused with visibility. It merely implies that data must be in a suitable format for it to be programmatically verified, but not through which mechanisms it will be verified.

4.6 Results

Having analyzed a large number of use cases (cf. Table 4.4), we find the majority of Blockchain applications reviewed have closed accessibility requirements. These

[2] GDPR has been discussed extensively in the context of Blockchain and was found to be incompatible with certain Blockchain technologies [37].

Table 4.4 This table categorizes the uses cases discussed by their accessibility requirements (*AR*), hierarchicality requirements (*HR*), and joint verifiability requirements (*JVR*)

Source	Application	*AR*	*HR*	*JVR*
Agbo [38]	Medical record management	●	◰	◭
	Remote patient monitoring	●	◰	⬚
	Drug supply chain management	●	◰	◭
	Facilitation of biomedical research	●	◰	◭
	Health data analytics	●	□	■
	Health insurance management	●	◰	◬
Albrecht et al. [39]	Peer-to-peer energy trading	●	□	◭
	Wholesale energy trading	●	◰	◭
	Asset management of energy production resources	●	◰	◬
	Certification of "green" electricity	●	◰	◭
	Management of an electric grid	●	▤	◬
Alladi et al. [40]	Vehicle-to-vehicle energy trading	●	◰	◭
	Vehicle-to-grid energy trading	●	◰	◭
Aristidou and Marcou [41]	Data sharing between government and non-government entities	●	◰	◬
	Citizen identity management	●	◰	
	Transparent tax collection and distribution	●	◰	◭
	Business-to-government coordination on supply chain management	●	◰	◭
Bishr [42]	Settlement of payments between government entities and the private sector	◌	◰	■
Bürer et al. [43]	Peer-to-peer energy trading	◌	◰	◬
	Electrical load management	●	◰	◬
	Electrical vehicle charging	◌	□	◬
	Trading of renewable energy certificates	●	◰	◬
	Trading of energy	●	□	◭
	Cryptocurrency enabled electricity meters	●	◰	◬
	Crowdfunding of energy supply	◌	◰	◬
Chang et al. [44]	Letter of credit system	●		▧
Dhillon et al. [45]	Public voting	●	◰	
Dick and Praktiknjo [46]	Electricity wholesale trading	●	◰	◭
Gatteschi et al. [47]	Exchange of claim data and claims settlement	●	▤	◬
	IoT-based alerting infrastructure	●	◰	◭
	Data-driven premium calculation	●	▤	◭
	Pay-per-use insurance	●	◰	◭
Hölbl et al. [48]	Medical record sharing	●	◰	◭
	Medical record management	●	◰	◭
	Access control to digital medical resources	●		⬚
	Medical resource access audit trail	●		◭
	Drug supply chain management	●	◰	◭
Khurshid [49]	Essential medical goods supply chain management	●	▤	◭
	Medical contact tracing	◌	◰	⬚
Kim and Sarin [50]	Transaction settlement		◰	
	Dark pools		□	▧
Kouhizadeh and Sarkis [51]	Storage of historic sustainability metrics		◰	◬
	Recording of supplier development investments	●	◰	◭
	Product supply chain	●	□	◬
	Management of logistics networks	●	◰	◬
	Tracking of production activities	●	□	▧
	Time management for product development activities	●	◰	◭
	Commercial ride-sharing initiatives		□	▧
	Tamper-proof waste reduction metrics record keeping	●	□	▧
Kshetri and Voas [52]	Allocation of aid funds	●	◰	▧
Lendák et al. [53]	Censorship-resistant web archiving	◌	□	■

(continued)

Table 4.4 (continued)

Source	Application	*AR*	*HR*	*JVR*
Li et al. [54]	Monetization of personal items	●	◰	
	Automatic billing for transportation fees		◰	
	Building project-related document management	●		⬚
	Crowdfunding of construction projects	◌	◰	◬
Lu et al. [55]	Reconciliation of payments along the oil and gas value chain	●	◰	◬
	Oil and gas exploration and development data management	●		⬚
	Oil and gas supply chain management	●	▤	◭
Lux et al. [56]	Decentralized public key infrastructure	◌	◰	▧
Mackey et al. [57]	Medical record management	●	▤	◭
	Electronic patient identity	●	◰	⬚
	Management of devices in the "Internet of Medical Things"	●	◰	
Mashamba-Thompson and Crayton [58]	Pandemic self-tracking and reporting	◌	◰	⬚
Mengelkamp et al. [26]	Management of a microgrid	●	◰	◭
Nakamoto [2]	Cryptocurrency	◌	□	■
Ngubo et al. [59]	Decentralized thermometry in a smart building	●	◰	◭
Niya et al. [60]	Scientific publication management	◌	□	■
Perboli et al. [61]	Food supply chain management	●	◰	▧
M. Platt et al. [63]	Cryptocurrency exchanges	◌	◰	◬
Radanović and Likić [65]	Medical record management	●	▤	◭
	Health insurance management	●	◰	◭
	Facilitation of biomedical research	●	◰	◭
	Healthcare procurement process management	●	▤	◭
	Medical education credentialing	●	◰	◭
Sgantzos and Grigg [66]	Tamper-proof management of AI input and output data	◌	□	⬚
	Direct distribution of software	◌	□	■
	Pre-trained AI networks as a service	◌	□	▧
	Crowd sourcing AI agent training	◌	□	
Shen and Pena-Mora [67]	Clinical trial data management	●	▤	◭
	Medical record management	●	▤	◭
	Drug supply chain management	●	◰	◭
	Health insurance claims processing	●	◰	◭
Sigwart et al. [68]	Temperature-controlled supply chains ("cold chains")	●	◰	◭
	Personal health reporting through smart health devices	●	◰	◭
	Event log for autonomous vehicles	●	◰	◭
de Souza et al. [69]	Transparent government payments	●	◰	▧
Valtanen et al. [71]	Peer-to-peer energy trading	●	◰	◭
Varma [72]	Micropayments	◌	□	■
	Pre-/post-trade systems	●	◰	▧
	Smart contract-based investments	◌	◰	▧
	Credit data registries		◰	◬
	Intra-organizational ERP	●	□	◬
	Decentralized autonomous organizations	◌	□	▧
Verhoeven et al. [73]	Sea freight supply chain management	●	◰	◭
	Trade of farming goods	◌	□	
	Coffee supply chain management	●	◰	◭
	Cross-country/cross-organization supply chain management		▤	
	Food supply chain management	◌	◰	◬
Whitaker [74]	Collective ownership of works of art	◌	◰	■
Yrjölä [75]	Broadband capacity sharing orchestration	●	◰	◬
Zhang et al. [76]	Cross-border payments	●		
	Digital asset registries	●	◰	◬

Accessibility requirements: ◌—Open, ●—Closed; Hierarchicality requirements: □—Flat, ◰—Two Layer, ▤—Tall; Verifiability requirements: ⬚—None, ▧—Partially Bilateral, ■—Fully Bilateral, ◬—Supported Bilateral, ◭—Validator Driven applications target bounded user communities and therefore require a formal admissions process. This is especially noticeable in the field of medicine, where data exchanged is highly sensitive and extends a preexisting relationship in the real world. The field of finance, the birthplace of Blockchain technology, has less strict accessibility requirements and provides more applications that are accessible publicly.

Few of the use cases analyzed have flat hierarchicality requirements. Of those that have, many are in the financial domain. Most use cases, however, have some notion of hierarchy. Two-layer hierarchicality requirements are most common. This is often the case where the operator of a closed Blockchain network assumes special privileges. Here, we observe relationships between a supplier and a vendor, government and citizen, hospital and patient, or underwriter and policyholder. We also find hierarchicality in IoT scenarios, where an operator might deploy devices like meters or sensors. Subsequently, readings from these devices would be trusted system-wide due to the trustworthiness of the operator. Tall hierarchies are less common and are found most commonly in regulated contexts, for example, in healthcare, where applications might target a national health service, a hospital, and a patient, thus creating a three-level tall hierarchy. Most applications with tall hierarchicality requirements appear in the form of closed applications, whereas applications with flat hierarchies are more likely to have open characteristics.

We find various verifiability requirements. We find few applications at the extreme ends of the verifiability spectrum: applications without validation requirements are most commonly found when Blockchain is used as message-passing infrastructure or for immutable data storage. We find applications with full validation requirements, most commonly target digital assets like cryptocurrencies. Most applications rely on the verifiability of parts of the data model only.

4.7 Discussion

The prevalence of use cases that have supported bilateral or partial bilateral verifiability requirements shows that application designers frequently achieve privacy by keeping the number of data attributes shared deliberately small, thereby practicing data minimization. In some cases, however, this is not done to improve privacy but comes as a side effect of optimizing the performance of a system by limiting the volume of the data processed. Many applications, specifically those in highly regulated domains or those that deal with confidential data, address data privacy explicitly by strictly limiting access to the system. Few use cases implement an open design that allows arbitrary unchecked entities to participate. The hierarchical nature of many of the use cases analyzed enables centralized data validation. This means that data does not have to be made available for verification to entities on the same hierarchy level. Consequently, access to parts of the data can be restricted to parts of the network, thereby minimizing the exposure of potentially sensitive data for the benefit of improved data privacy. These findings suggest that most of the

Blockchain applications analyzed can address privacy through governance (i.e., by limiting access to the system), deliberate data modeling (i.e., keeping sensitive data off-chain), and role-based validation structures.

While the number of applications analyzed is comparatively large, this chapter can only provide a qualitative result. This is because applications are selected via non-systematic review of academic literature and not from industry sources that, arguably, are significant drivers of innovation in the field. Furthermore, use cases analyzed are not comparable in terms of maturity (i.e., use cases range from abstract designs to production-ready applications). Also, deriving requirements from high-level descriptions of applications requires considerable interpretation.

An unexpectedly large number of applications analyzed have closed characteristics, expose some hierarchicality, and have validator-driven requirements. It is unclear whether use cases with these characteristics are best addressed via Blockchain technology or whether other, less complex, architectures like immutably auditable databases operated by trusted third parties could satisfy those requirements equally well. Replacing a decentralized system with a third party entrusted with storing data centrally makes the privacy properties of a system significantly easier to analyze. Furthermore, the limited awareness of novel cryptographic approaches (e.g., ZKPs, homomorphic encryption, or trusted execution environments) to keep data private was surprising. It shows that application designers are not yet ready to entrust sensitive data to them. It can be assumed that these technologies will be more widely accepted once they become more mature. Future work should focus on analyzing the influence of these novel technologies on the privacy properties of the use cases identified. This will show whether their application can contribute to more open system designs.

4.8 Conclusion

In this chapter, we have investigated the prevalent requirements relating to privacy in the Blockchain ecosystem. To help with this, we have focused on three aspects of system architecture: (1) the accessibility of a system that impacts those who, in principle, have access to shared data, (2) the hierarchicality of a system, which entails whether privileged entities with broad data inspection permissions can exist, and (3) the data validation requirements of a system, which define the aspects of an application's data model, which need to be verifiable by others. We find that most systems have closed accessibility requirements, exhibit some degree of hierarchicality, and share only part of their data model for validation purposes. These characteristics allow operators to restrict access to data by governance (i.e., by only allowing trusted and vetted actors access to a system). Works that address privacy considerations directly often focus exclusively on the accessibility dimension. Since most works analyzed discuss governance-based approaches to questions of privacy, designs in which emerging cryptographic techniques could be usefully applied are rare. This work concentrates on privacy requirements in Blockchain, thus contributing to the wider literature that takes a broader perspective. While the number of applications analyzed is extensive, this work is purely qualitative. Future work should focus on the applicability of emerging cryptographic techniques that allow for operations on encrypted data, as these can further benefit data privacy.

References

1. O. Doe-Bruce, Blockchain and alternative sources of financing, in *Cryptofinance and Mechanisms of Exchange* (Springer, Berlin, 2019), pp. 91–111
2. S. Nakamoto, Bitcoin: a peer-to-peer electronic cash system (2008). Accessed 10 Oct 2020
3. M. Flyverbom, R. Deibert, D. Matten, The governance of digital technology, big data, and the internet: new roles and responsibilities for business. Bus. Soc. **58**(1), 3–19 (2017)
4. R. Bandara, M. Fernando, S. Akter, Privacy concerns in e-commerce: a taxonomy and a future research agenda. Elect. Mark. **30**(3), 629–647 (2019)
5. R. Bandara, M. Fernando, S. Akter, Explicating the privacy paradox: a qualitative inquiry of online shopping consumers. J. Retail. Consum. Serv. **52**, 101947 (2020)
6. S.M. West, Data capitalism: redefining the logics of surveillance and privacy. Bus. Soc. **58**(1), 20–41 (2017)
7. R. Bandara, The ethics of online privacy in the data-driven marketplace: a power-responsibility equilibrium and construal level theory perspective. PhD Thesis, School of Management, Operations and Marketing, University of Wollongong (2020)
8. R. Bandara, M. Fernando, S. Akter, Managing consumer privacy concerns and defensive behaviours in the digital marketplace. Eur. J. Mark. **55**, 219–246 (2020). https://doi.org/10.1108/ejm-06-2019-0515
9. S. Baswani, J. George, A. Townsend, Connected homes and online retail: the case of IoT enabled online shopping, in *Proceedings of the 24th Americas Conference on Information Systems*, AMCIS, New Orleans (Association for Information Systems, Atlanta, 2018)
10. R. Bandara, M. Fernando, S. Akter, Addressing privacy predicaments in the digital marketplace: a power-relations perspective. Int. J. Consum. Stud. **44**(5), 423–434 (2020)
11. C.-L. Yeh, Pursuing consumer empowerment in the age of big data: a comprehensive regulatory framework for data brokers. Telecommun. Policy **42**(4), 282–292 (2018)
12. E.D. Zamani, G.M. Giaglis, With a little help from the miners: distributed ledger technology and market disintermediation. Ind. Manag. Data Syst. **118**(3), 637–652 (2018)
13. X. Lv, Y. Zhong, Q. Tan, A study of bitcoin de-anonymization: graph and multidimensional data analysis, in *Proceedings of the 5th International Conference on Data Science in Cyberspace*, DSC, Hong Kong (IEEE, Piscataway, 2020), pp. 339–345
14. O. Labazova, T. Dehling, A. Sunyaev, From hype to reality: a taxonomy of blockchain applications, in *Proceedings of the 52nd Hawaii International Conference on System Sciences*, HICSS, Wailea (University of Hawaii at Manoa, Association for Information Systems, Honolulu, 2019), pp. 4555–4564
15. P. Tasca, C.J. Tessone, A taxonomy of blockchain technologies: Principles of identification and classification. Ledger **4** (2019). https://doi.org/10.5195/ledger.2019.140
16. J.A. Jaoude, R.G. Saade, Blockchain applications – usage in different domains. IEEE Access **7**, 45360–45381 (2019)
17. F. Casino, T.K. Dasaklis, C. Patsakis, A systematic literature review of blockchain-based applications: current status, classification and open issues. Telematics Inform. **36**, 55–81 (2019)
18. M. Swan, *Blockchain* (O'Reilly Media, Sebastopol, 2015)
19. N. Szabo, Formalizing and securing relationships on public networks. First Monday **2**(9) (1997)

20. M. Marx, E. Zimmer, T. Mueller, M. Blochberger, H. Federrath, Hashing of personally identifiable information is not sufficient, in *Beiträge der 9. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V.*, Konstanz (Gesellschaft für Informatik e.V., Bonn, 2018), pp. 55–68
21. A.-E. Panait, R.F. Olimid, A. Stefanescu, Identity management on blockchain – privacy and security aspects (2020). Accessed 10 Oct 2020
22. J. Bernal Bernabe, J.L. Canovas, J.L. Hernandez-Ramos, R. Torres Moreno, A. Skarmeta, Privacy-preserving solutions for blockchain: review and challenges. IEEE Access **7**, 164908–164940 (2019)
23. G.A. Pierro, R. Tonelli, M. Marchesi, An organized repository of Ethereum smart contracts' source codes and metrics. Fut. Int. **12**(11), 197 (2020)
24. G.A. Oliva, A.E. Hassan, Z.M. Jiang, An exploratory study of smart contracts in the Ethereum blockchain platform. Empir. Softw. Eng. **25**(3), 1864–1904 (2020)
25. K. Sreelakshmi, A. Bhatia, A. Agrawal, Securing IoT applications using blockchain: a survey (2020). Accessed 10 Oct 2020
26. E. Mengelkamp, J. Gärttner, K. Rock, S. Kessler, L. Orsini, C. Weinhardt, Designing microgrid energy markets. Appl. Energy **210**, 870–880 (2018)
27. R. Ranchal, P. Bastide, X. Wang, A. Gkoulalas-Divanis, M. Mehra, S. Bakthavachalam, H. Lei, A. Mohindra, Disrupting healthcare silos: addressing data volume, velocity and variety with a cloud-native healthcare data ingestion service. IEEE J. Biomed. Health Inform. **24**(11), 3182–3188 (2020)
28. L. Smorgunov, Blockchain and a problem of procedural justice of public choice, in *Communications in Computer and Information Science* (Springer International Publishing, Basel, 2018), pp. 13–23
29. F. Bélanger, R.E. Crossler, Privacy in the digital age: a review of information privacy research in information systems. MIS Quart. **35**(4), 1017 (2011)
30. E. Allan Lind, T.R. Tyler, *Procedural Justice in Organizations* (Springer US, Boston, 1988), pp. 173–202
31. M. Cannarsa, Interpretation of contracts and smart contracts: smart interpretation or interpretation of smart contracts? Europ. Rev. Priv. Law, 773–785 (2018)
32. G. Destefanis, M. Marchesi, M. Ortu, R. Tonelli, A. Bracciali, R. Hierons, Smart contracts vulnerabilities: a call for blockchain software engineering? in *Proceedings of the International Workshop on Blockchain Oriented Software Engineering*, IWBOSE, Campobasso (IEEE, Piscataway, 2018), pp. 19–25
33. H.T.M. Gamage, H.D. Weerasinghe, N.G.J. Dias, A survey on blockchain technology concepts, applications, and issues. SN Comput. Sci. **1**(2), 114 (2020)
34. W. Wang, D.T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, D.I. Kim, A survey on consensus mechanisms and mining strategy management in blockchain networks. IEEE Access **7**, 22328–22370 (2019)
35. K. Wüst, A. Gervais, Do you need a blockchain? in *Proceedings of the 2018 Crypto Valley Conference on Blockchain Technology*, CVCBT, Zug (IEEE, Piscataway, 2018), pp. 45–54
36. W. Dai, b-money (1998). Accessed 17 Nov 2020
37. A. Van Humbeeck, The blockchain-GDPR paradox. J. Data Protect. Priv. **2**(3), 208–212 (2019)
38. C. Agbo, Q. Mahmoud, J. Eklund, Blockchain technology in healthcare: a systematic review. Healthcare **7**(2), 56 (2019)
39. S. Albrecht, S. Reichert, J. Schmid, J. Strüker, D. Neumann, G. Fridgen, Dynamics of blockchain implementation – a case study from the energy sector, in *Proceedings of the 51st Hawaii International Conference on System Sciences*, HICSS, Waikoloa Village (Association for Information Systems, Atlanta, 2018), pp. 3527–3536
40. T. Alladi, V. Chamola, J.J.P.C. Rodrigues, S.A. Kozlov, Blockchain in smart grids: a review on different use cases. Sensors **19**(22), 4862 (2019)
41. C. Aristidou, E. Marcou, Blockchain standards and government applications. J. ICT Stand. **7**(3), 287–312 (2019)

42. A.B. Bishr, Dubai: a city powered by blockchain. Innov. Technol. Gov. Glob. **12**(3–4), 4–8 (2019)
43. M.J. Bürer, M. de Lapparent, V. Pallotta, M. Capezzali, M. Carpita, Use cases for blockchain in the energy industry opportunities of emerging business models and related risks. Comput. Indust. Eng. **137**, 106002 (2019)
44. S.E. Chang, H.L. Luo, Y. Chen, Blockchain-enabled trade finance innovation: a potential paradigm shift on using letter of credit. Sustainability **12**(1), 188 (2019)
45. A. Dhillon, G. Kotsialou, P. McBurney, L. Riley, Voting over a distributed ledger: an interdisciplinary perspective (2020)
46. C. Dick, A. Praktiknjo, Blockchain technology and electricity wholesale markets: expert insights on potentials and challenges for OTC trading in Europe. Energies **12**(5), 832 (2019)
47. V. Gatteschi, F. Lamberti, C. Demartini, C. Pranteda, V. Santamaría, Blockchain and smart contracts for insurance: is the technology mature enough? Fut. Int. **10**(2), 20 (2018)
48. M. Hölbl, M. Kompara, A. Kamišalić, L.N. Zlatolas, A systematic review of the use of blockchain in healthcare. Symmetry **10**(10), 470 (2018)
49. A. Khurshid, Applying blockchain technology to address the crisis of trust during the COVID-19 pandemic. JMIR Med. Inf. **8**(9), e20477 (2020)
50. S. Kim, A. Sarin, Distributed ledger and blockchain technology: framework and use cases (2018). Accessed 20 Dec 2020
51. M. Kouhizadeh, J. Sarkis, Blockchain practices, potentials, and perspectives in greening supply chains. Sustainability **10**(10), 3652 (2018)
52. N. Kshetri, J. Voas, Blockchain in developing countries. IT Prof. **20**(2), 11–14 (2018)
53. I. Lendák, B. Indig, G. Palkó, WARChain: blockchain-based validation of webarchives, in *Proceedings of the 10th International Workshop on Socio-Technical Aspects in Security*, STAST, Guildford, ed. by T. Groß, L. Viganò (Springer, Berlin, 2020)
54. J. Li, D. Greenwood, M. Kassem, Blockchain in the built environment and construction industry: a systematic review, conceptual models and practical use cases. Autom. Constr. **102**, 288–307 (2019)
55. H. Lu, K. Huang, M. Azimi, L. Guo, Blockchain technology in the oil and gas industry: a review of applications, opportunities, challenges, and risks. IEEE Access **7**, 41426–41444 (2019)
56. Z.A. Lux, D. Thatmann, S. Zickau, F. Beierle, Distributed-ledger-based authentication with decentralized identifiers and verifiable credentials, in *Proceedings of the 2nd Conference on Blockchain Research and Applications for Innovative Networks and Services*, BRAINS, Paris (IEEE, Piscataway, 2020), pp. 71–78
57. T. Mackey, H. Bekki, T. Matsuzaki, H. Mizushima, Examining the potential of blockchain technology to meet the needs of 21st-century Japanese health care: viewpoint on use cases and policy. J. Med. Int. Res. **22**(1), e13649 (2020)
58. T.P. Mashamba-Thompson, E.D. Crayton, Blockchain and artificial intelligence technology for Novel Coronavirus Disease-19 self-testing. Diagnostics **10**(4), 198 (2020)
59. C. Ngubo, M. Dohler, P. Mcburney, Blockchain, IoT and sidechains, in *Proceedings of The International Multiconference of Engineers and Computer Scientists*, IMECS (IAENG, Hong Kong, 2019), pp. 36–140
60. S.R. Niya, L. Pelloni, S. Wullschleger, A. Schaufelbuhl, T. Bocek, L. Rajendran, B. Stiller, A blockchain-based scientific publishing platform, in *Proceedings of the 2019 International Conference on Blockchain and Cryptocurrency*, ICBC Seoul (IEEE, Piscataway, 2019), pp. 329–336
61. G. Perboli, S. Musso, M. Rosano, Blockchain in logistics and supply chain: a lean approach for designing real-world use cases. IEEE Access **6**, 62018–62028 (2018)
62. M. Platt, A. Hasselgren, J.M. Román-Belmonte, et al., Test, trace, and put on the Blockchain?: A viewpoint evaluating the use of decentralized systems for algorithmic contact tracing to combat a global pandemic. JMIR Public Health Surveill. **7**, e26460 (2021) https://doi.org/10.2196/26460

63. M. Platt, F. Pierangeli, G. Livan, S. Righi, Facilitating the decentralised exchange of cryptocurrencies in an order-driven market, in *Proceedings of the 2nd Conference on Blockchain Research and Applications for Innovative Networks and Services*, BRAINS, Paris (IEEE, Piscataway, 2020), pp. 30–34
64. W.N. Price II, I.G. Cohen, Privacy in the age of medical big data. Nat. Med. **25**, 37–43 (2019). https://doi.org/10.1038/s41591-018-0272-7
65. I. Radanović, R. Likić, Opportunities for use of blockchain technology in medicine. Appl. Health Econ. Health Policy **16**(5), 583–590 (2018)
66. K. Sgantzos, I. Grigg, Artificial intelligence implementations on the blockchain. use cases and future applications. Fut. Int. **11**(8), 170 (2019)
67. C. Shen, F. Pena-Mora, Blockchain for cities—a systematic literature review. IEEE Access **6**, 76787–76819 (2018)
68. M. Sigwart, M. Borkowski, M. Peise, et al., A Secure and Extensible Blockchain-Based Data Provenance Framework for the Internet of Things. Pers. Ubiquit. Comput. (2020). https://doi.org/10.1007/s00779-020-01417-z
69. R.C. de Souza, E.M. Luciano, G.C. Wiedenhöft, The uses of the blockchain smart contracts to reduce the levels of corruption: some preliminary thoughts, in *Proceedings of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age*, dg.o (ACM, New York, 2018)
70. C. Tozzi, Decentralizing democracy: approaches to consensus within blockchain communities. TEKN **16**, 181–195 (2019). https://doi.org/10.5209/tekn.64523
71. K. Valtanen, J. Backman, S. Yrjola, Blockchain-powered value creation in the 5G and smart grid use cases. IEEE Access **7**, 25690–25707 (2019)
72. J.R. Varma, Blockchain in finance. Vikalpa J. Decis. Makers **44**(1), 1–11 (2019)
73. P. Verhoeven, F. Sinn, T. Herden, Examples from blockchain implementations in logistics and supply chain management: Exploring the mindful use of a new technology. Logistics **2**(3), 20 (2018)
74. A. Whitaker, Art and blockchain: a primer, history, and taxonomy of blockchain use cases in the arts. Artivate J. Enterp. Arts, 21–47 (2019)
75. S. Yrjölä, Analysis of blockchain use cases in the Citizens Broadband Radio Service spectrum sharing concept, in *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering* (Springer International Publishing, Basel, 2018), pp. 128–139
76. L. Zhang, Y. Xie, Y. Zheng, W. Xue, X. Zheng, X. Xu, The challenges and countermeasures of blockchain in finance and economics. Syst. Res. Behav. Sci. **37**(4), 691–698 (2020)

Chapter 5
An Empirical Investigation of Blockchain Scalability

Kashif Mehboob Khan, Junaid Arshad, Muhammad Mubashir Khan, and Muhammad Hassan Nasir

5.1 Introduction

A Blockchain system exemplifies distributed ledger technologies. It enables a digital transaction's data to be stored in the form of a chained block with the consensus of verifying nodes, also known as *miners*. A blockchain-based system has inherent properties of decentralization, transparency, trust-less-trust, immutability, and traceability since it allows each participating node to have an equal opportunity to influence the ledger. The transparency and immutability are enforced by allowing the nodes to view and maintain the ledger which can only be altered with the consensus of other mining nodes. Although Bitcoin [1] still represents the most popular application of blockchain, the use of blockchain has witnessed significant attention across diverse application domains including healthcare, finance, e-government [2], and supply chain management [3]. Features of blockchain such as security, immutability, and decentralization have a profound role in this. Emerging blockchain applications have been facilitated by research on blockchain beyond cryptocurrencies where significant efforts have been made to devise different

K. M. Khan
Department of Software Engineering, NED University of Engineering & Technology, Karachi, Pakistan
e-mail: kashifmehboob@neduet.edu.pk

J. Arshad (✉)
School of Computing and Digital Technology, Birmingham City University, Birmingham, UK
e-mail: junaid.arshad@bcu.ac.uk

M. M. Khan · M. H. Nasir
Department of Computer Science & IT, NED University of Engineering & Technology, Karachi, Pakistan
e-mail: mmkhan@neduet.edu.pk

M. H. ur Rehman et al. (eds.), *Trust Models for Next-Generation Blockchain Ecosystems*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-75107-4_5

business models, its applicability in various domains, and to optimize its adaptability for different applications. Furthermore, several platforms such as Ethereum [4], Multichain [5], and Hyperledger [6] have been developed to facilitate the adoption of blockchain technology beyond cryptocurrency and have had a profound impact in achieving its wide-spread adoption.

A significant factor in the successful adoption of blockchain is its ability to facilitate large-scale transactions which require a scalable solution. For instance, applications such as public voting require concurrent processing of a large number of transactions [7]. Such scenarios present opportunities for further research into the ability of blockchain to facilitate the development of scalable applications and challenges associated to achieve this aim. For instance, a common belief is to treat blockchain as a black box which can be used to develop decentralized applications that are inherently efficient. Deep analysis of several important parameters including the rate of block generation and block processing, and block size is to be performed in order to build an efficient blockchain-based decentralized application.

Among many interesting applications of blockchain the electronic voting (e-voting) is the one that harnesses the potential benefits of blockchain technology which includes anonymity, immutability, and non-repudiation. These features of blockchain are extremely important to ensure privacy and authenticity of voters and the integrity of overall electoral system. There have been several important research contributions that have attracted the utilization of distributed ledger technology in electronic voting system [8–10]. The structure of blockchain enables us to gracefully map the voters and candidates from the practical electoral system into the blockchain. Furthermore, every vote caste is represented by a transaction within blockchain which is stored on the ledger in an immutable manner. To ensure immutability blockchain has a strong hashing mechanism that tightly binds the successive transactions into a chain of blocks which is computationally infeasible to modify illegally by an adversary.

In this chapter we present our research to highlight important issues related to the scalability of blockchain-based applications, as it is the primary requirement of majority of such applications including the electronic voting. Our research is based on an e-voting scenario that is implemented on a permission-less blockchain testbed. The voting environment created in our experimentation includes small number of users for which the three important parameters have been tested by introducing several variations. These parameters include the block size, the level of difficulty for mining a block, and the average time to create a block. The research results show an interesting trade-off among these parameters. The experimental testbed is evaluated against the population of remote clients (voting machines) with their network connectivity from different locations and the volume of transactions. These factors have obvious impact on the overall performance of the blockchain-based system.

The rest of the chapter is organized as follows. The motivation to achieve scalable blockchains along with the state of the art to achieve scalable blockchains is presented in Sect. 5.2. Section 5.3 presents a comprehensive discussion to define different aspects of scalability within blockchains along with a critical overview of existing approaches to achieve scalable blockchains. Details of the e-voting system model are presented in Sect. 5.4 along with details of testbed for evaluation. Section 5.5 presents a detailed description of the scenarios used to identify important factors affecting scalable blockchains along with an insightful discussion of trade-offs involved. Section 5.6 presents a summary of the efforts along with a discussion on open challenges in achieving scalable blockchains.

5.2 Blockchain Scalability

5.2.1 The Case for Scalable Blockchains

Although Bitcoin represents a major milestone in the adoption of blockchain technology, the underpinning concepts and mechanisms date back even further. For instance, the pseudonyms adopted by blockchain were first proposed by Chaum in 1983 [11], whereas the Proof of Work (PoW) used by Bitcoin and other applications as consensus protocol was developed by Adam Back in 2002 [12]. Alongside Bitcoin, recent developments such as Smart contracts have attracted significant attention leading to its adaption as a cutting-edge paradigm beyond cryptocurrencies in domains including finance, supply chain, Internet of Things (IoTs), etc. Emergence of platforms such as Ethereum [4], Multichain [5], and Hyperledger [6] have a profound role in this.

The use of blockchain in diverse domains has also led to the identification of number of research challenges which can affect its successful adoption. Scalability is one such challenge and has become a prime concern for research and business communities. Specifically, blockchain performance has been compared with VISA which can achieve a maximum throughput of 2000 transactions per second. On the contrary, Bitcoin, blockchain's most prominent application, can only achieve a small fraction of this. Further, performance-hungry applications such as a public voting system require a high transaction throughput [7], motivating researchers to investigate mechanisms, and methods to enhance the scalability of blockchain-based applications.

However, in order to assess blockchain's ability to support scalable applications that are capable of achieving high performance throughput, a rigorous analysis of various characteristics including block size, transaction processing speed, and block generation rate. For instance, the size of a block in Bitcoin was set to 1 MB that supports a peak block generation rate of 10 blocks per minute. Increasing the block size enhances the capacity of transactions to be included within a block. However, it may lead to an increase in the block propagation time. Conversely, a decrease in the block size may improve the latency but can also result in the generation of multiple forks [13]. Such concerns necessitate trade-offs between parameters such as block size, efficiency, and security which require in-depth analysis to facilitate development of scalable blockchain-based applications.

5.2.2 State of the Art Within Scalable Blockchains

The research contribution of Corman et al. [14] on the scalability of the blockchain-based applications has a significant importance in which they focus on the practical constraints such as transaction mining time, transaction processing speed, as well as costs incurred in running blockchain nodes (electricity, storage, etc.). The authors also highlight the impact of blockchain scalability and security. For example, malicious activities inside the blockchain system such as forking, generating malleable transactions, and realizing the double spending scenario can be mitigated by carefully improving the block size.

It is important to note that the mining rewards similar to Bitcoin may disrupt the smooth processing of transactions in a block and adding that block gracefully into the blockchain. Such disruption is induced by the greedy behavior of miners in order to gain more and more financial rewards against mining. Such behavior has been highlighted by Karame [15] and Zheng et al. [16]. Similarly, there are other legitimate factors, such as network propagation delays, which have been highlighted by Karame [15] in the context of scalability and security of a blockchain network. Introducing the delays and utilization of Bitcoin network bandwidth can provide opportunities for the attackers (greedy miners) to prevent a victim from acquiring a block.

As proposed by Segregated Witness [17], one approach to increase the blockchain throughput can be to increase the size of the block. However, as identified by Zheng et al. [16], increasing the block size may reduce the block generation rate that may in turn increase the chances of forking. As pointed out by Zheng et al. [18] that increasing the block size may lead towards centralization of blockchain resources as many of the mining nodes might not be able to store high volume of data. The transaction processing rate and network latency has an interesting trade-off that has been explained by the authors in order to achieve scalability while preserving the security of the blockchain.

Another interesting proposal is the implementation of the so-called GHOST or pairwise ledgers by Xu et al. [19]. However, these pairwise ledgers such as R3's Corda24 have problems concerning to their adoption because it is not easy to realize a globally replicated database in order to ensure the availability of data. One possible solution to such constraint is to increase the number of nodes that are capable to execute a full copy of ledger as part of the decentralized blockchain network.

The assessment of the effectiveness of several on-chain solutions to achieve scalability in blockchain has been studied by Vukoli et al. [20] and Bano et al. [21]. Their study also includes the well-known consensus algorithms in blockchain such as Proof of Work (PoW) and Byzantine Fault Tolerance (BFT) in the context of scalability of the overall blockchain system. As the consensus mechanism has a significant impact on major factors, such as block generation rate and block size, on which the scalability of blockchain depends. It has been found that BFT is better in performance when there are small number of nodes; however, proof of work based distributed ledgers are scalable in term of the number of nodes at the cost of performance degradation. The design based techniques of blockchain proposed by Bano et al. [21] present the issues pertaining to the scalability of Bitcoin-NG [22] and Practical BFT with Sharding transactions.

Kim et al. [23] explains scalability as a function of few important parameters in which he includes the transaction processing rate, the space required for the storage of chain and the associated fee related to a particular cryptocurrency application. The authors categorize different methods for improving scalability such as on-chain, off-chain, and side-chain. Important on-chain solutions include Bi Block, Segregated Witness [17] and Merkelized Abstract Syntax Tree (MAST) [24]. The Big Block method that was implemented by Bitcoin Unlimited increases the block size that maximizes the transmission limit and minimizes the transmission cost. In off-chain method a multi sign address is used by creating a channel between the sender and receiver by ensuring the stake of both the parties. Here the exchange of transaction is done off-chain removing the requirement of transaction fee as well as time delays. However, the channel creation if off course an on-chain process that has a transaction fee associated with the main blockchain. Finally, side-chain is also a method for minimizing the issues related to the scalability where the main functionalities of blockchains are aggregated.

Mattias et al. [25] present the non-cryptocurrency applications of blockchain in several interesting areas. Their work is unique in a sense that majority of the contributions for scalability of blockchain are based on Bitcoin. The authors address the scalability with a range of security vulnerabilities of blockchain system that is decentralized in nature. This contribution presents very important results for diversified applications of blockchain where scalability is required.

5.3 Factors Affecting Blockchain Scalability

In this section, we identify and define different factors affecting blockchain scalability. We first describe the different dimensions of blockchain scalability, i.e., horizontal and vertical dimensions of scalability. We then categorize and describe approaches to blockchain scalability in Sect. 5.3.2.

5.3.1 Dimensions of Blockchain Scalability

The world has witnessed significant adoption of blockchain technology in the past 10 years due to its inherent features including security, immutability, and decentralization. Especially after the arrival of Smart Contracts in blockchain 2.0 which extended its applicability beyond cryptocurrencies. The blockchain is now being widely applied in diverse fields including finance, healthcare, asset management, etc. With this wide-spread adaption, scalability has emerged as one of the prime concerns for the blockchain community. Without sufficient scalability,

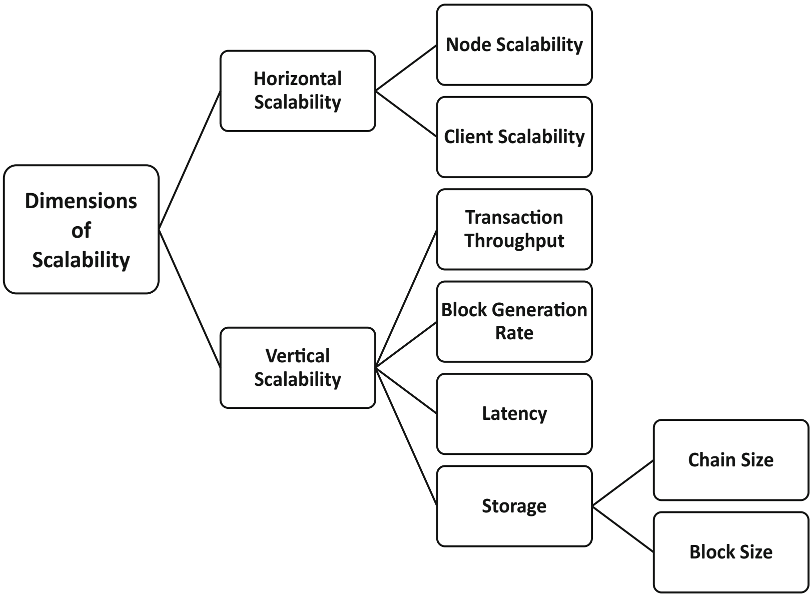


Fig. 5.1 Dimensions of blockchain scalability

the blockchain-based systems will hardly become mainstream and will not be able to realize their true disruptive potential [7].

However, scalability is a crucial trade-off characteristic and a composite term with several potentially conflicting interpretations. It refers to enhancing the capabilities of current processes to facilitate the stakeholders (including users, miners, developers, etc.) and the systems. Consequently, this section identifies and defines major factors that can be used to describe the scalability of a blockchain system. We define these as different dimensions of blockchain scalability and summarize them in Fig. 5.1.

5.3.1.1 Horizontal Scalability

Horizontal scalability refers to the ability of a blockchain to add more nodes and clients thereby expanding the network of participants. It is a measure of facilitating the maximum number of nodes or clients without degradation in the efficiency and performance of the blockchain. For example, the proof of work (PoW) based consensus approaches can deliver better performance with an increased number of nodes, whereas the performance of Byzantine fault tolerance (BFT) based approaches degrades with an increasing number of nodes. Both families of consensus approaches offer decent client scalability and support thousands of clients.

- **Client scalability**
  The clients are the application programs that submit transactions on the behalf of a user and client scalability refers to the ability of a blockchain system to facilitate an increasing number of clients without affecting the overall performance.
- **Node scalability**
  Node scalability refers to the ability of a blockchain system to facilitate an increasing number of nodes without affecting the overall performance.

5.3.1.2 Vertical Scalability

Vertical scalability refers to enhancing the capabilities of nodes to achieve efficiency. The vertical scalability can be enhanced by adjusting the parameters such as block size, parallel mining, lightning, sharding, etc. Few aspects of vertical scalability such as throughput, block generation rate, latency, storage scalability are described below:

- **Transaction throughput**

  The transaction throughput expressed as transaction per second (TPS) is the major component of scalability. It is the rate at which valid transactions are committed and added to the block after establishing consensus between the stakeholders (miners) by the blockchain.

  The significant rise in the volume of transactions does not allow current implementations such as bitcoin, to fulfill the requirements of a transaction hungry environment. The traditional blockchain(bitcoin) can support a transaction rate of up to seven transactions per second [26]. While other implementations, like Ethereum, [4] and Bitcoin cash [26], perform better and can process up to 20 and 60 TPS, respectively. They are still way behind the rate which is expected, such as legacy Visa which can process up to 24,000 TPS [27].
- **Block generation rate**

  In the blockchain, the multiple transactions are grouped, depending upon the block size, to form a block. The Block generation rate (BGR) is the frequency at which a new block is mined, produced, and added to the blockchain. The block generation is a resource expensive process in terms of mining and is solely dependent upon the size of the block and consensus efficiency. For example, the legacy Nakamoto's algorithm (bitcoin) has a fixed block size of 1 MB and requires each new block to have a unique hashing puzzle solution than the previous block to be added to the chain. This results in a requisite of 10 min to generate a block. Ethereum has a better BGR which is between 10 to 20 s.
- **Latency**

  Typically, the latency is measured as the time delay between input and output. Low network latency is of paramount importance in the blockchain. The latency can refer to two different delays, i.e., network latency and transaction latency. The network latency is the measure of delay between initiating a transaction request and getting the confirmation of acceptance of the transaction from the network. As more blocks are added to the main-chain, the transaction becomes more authentic. The transaction's latency is the measure of consensus efficiency which is the ability to process and execute large volumes of transactions.

- **Storage scalability**

  The storage scalability can further be divided into block size management and chain size management
- **Block size**

  The block size is the limit of a block to accommodate the number of transactions and related information. For example, in a typical bitcoin network, the block size currently stands at 1 MB, which can accommodate more than 500 transactions. The increase in block size is an important factor to improve TPS but may hurt the block generation rate. On the other hand, decreasing the block size may lead to the formation of forks. Therefore, there should be a trade-off between TPS and BGR while selecting the block size.
- **Chain size**

  The blockchain is a decentralized ledger in which the amount of transactions is significantly increasing. A node, before participating in the network, requires enough storage to download the chain to have a complete view of the network. The bitcoin blockchain has already exceeded 160 GB. This means a miner needs to download a huge amount of data locally to participate in the network. The increasing number of transactions will also increase the synchronization time (Table 5.1).

Table 5.1 Scalability approaches

S. No.	Approach	Selected existing efforts
1	Payment Channel Network (PCN)	Malavolta et al. [28], Poon et al. [29], Radian Network[30]
2	Sharding	Dang et al. [31], Kan et al. [32], Kogias et al [33], Li et al. [34], Luu et al. [35], Ozyilmaz et al. [36], Ren et al.[37], Yu et al. [38], Zamani et al. [39]
3	Blockchain Delivery Network	Guobiao et al.[40], Kalrman et al. [41], Kuzmanovic et al.[42]
4	Parallel Processing	Gao et al. [43], Hazari et al. [44]
5	Hardware-Assisted Approaches	Liu et al. [45], Lind et al. [46], Sanka et al. [47]
6	Redesigning	Dennis et al. [48], Ehmke et al. [49], Fan & Chai [50], Fan et al. [51], Gupta et al. [52], Ittay Eyal et al. [22], Min et al. [53], Otte et al. [54], Thakkar et al.[55], Zhang et al. [56]

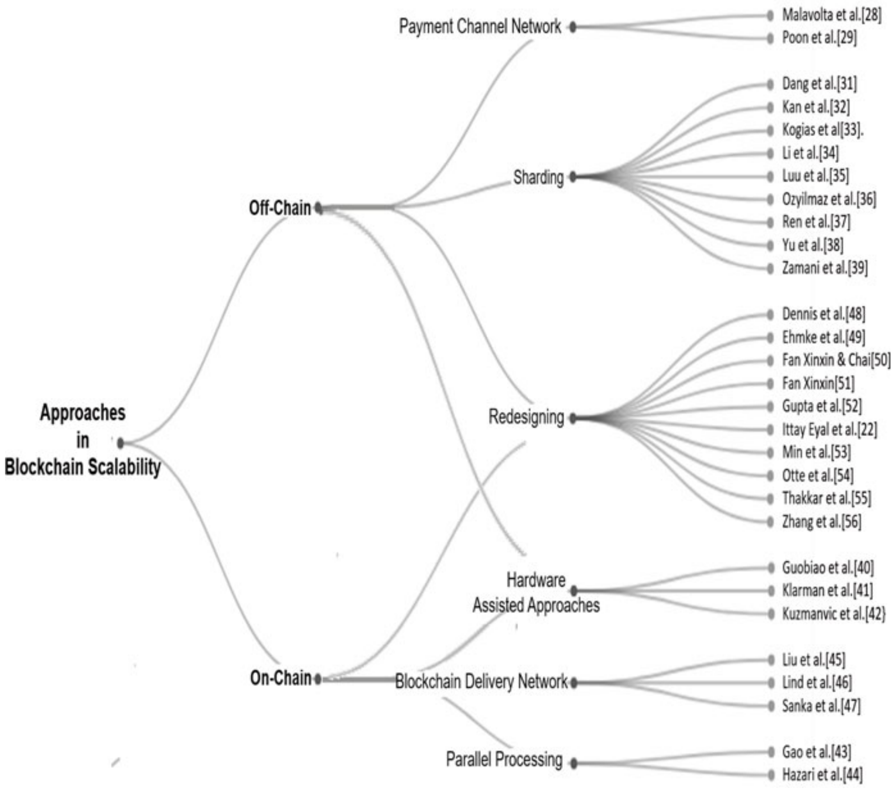


Fig. 5.2 Blockchain scalability approaches

5.3.2 *Blockchain Scalability Approaches*

As highlighted in Sect. 5.2.2, several efforts have been made to improve the state of the art with respect to scalable blockchains. In this section, we present a classification that can be used to categorize existing efforts (Fig. 5.2).

5.3.2.1 On-Chain Solutions

The on-chain approaches enhance the scalability by adjusting the internal ingredients of the blockchain. This can be achieved by improving the network latency or optimizing the transaction or message size. Some of the On-chain solutions are described below:

- **Blockchain pipelining**
  Blockchain pipelining is a process of adding blocks in the main-chain without

being verified by other nodes. The final decision for the validity of block for the formation of the main-chain is done through voting among nodes as a separate layer.

- **Blockchain delivery network**
  Some solutions rely on a delivery network that includes cut-through routing enabled gateways or cloud delivery networks without disturbing the centralized nature of blockchain to achieve a better transaction throughput, or a collaborative cloud storage mechanism to achieve storage scalability.
- **Adjustment in block size**
  Another approach to attain scalability is to increase/decrease the block size. But, too much increase in block size results in more transactions per block but leads to an increase in propagation time. On the other hand, too much decrease in block size may result in increased BGR with improved latency. However, it can lead to a frequent generation of forks.

5.3.2.2 Off-Chain Solutions

The off-chain solutions process transactions outside the blockchain to lighten the load from the main-chain. These solutions are backed by creating ways that execute on the top of the blockchain. The solutions such as using payment channels (LN or Raiden network) or sharding are the famous off-chain solutions that have significant positive effects on horizontal as well as vertical scalability.

- **Payment channel networks**
  The payment channel is a concept to allow multiple off-chain transactions between parties by creating a micro-payment channel without having to commit all transactions publically on the main-chain. Consequently, resulting in increased throughput by removing the load from the main-chain. In a typical payment channel network, two parties can perform an unlimited number of transactions. Only two transactions are required for the main-chain to update a record that can be informed upon completion of all transactions between parties or upon the requirement of an on-chain transaction. Even the parties which are not directly connected can commit the transactions through intermediaries. The most popular implementations of PCNs are Lightning Network (LN) and Raiden Network (RN) for Bitcoin and Ethereum, respectively.
- **Sharding**
  Typically, a mining node on a blockchain's network is responsible for storing all the states, which includes critical information like account balance and history of all transactions. This considerably slows the transactions throughput linearly. Sharding is a well-established concept in databases to divide a huge database into smaller and manageable portions to increase its efficiency. In a blockchain, it refers to the horizontal division of the main-chain into multiple independent partitions known as "shards." Each partition or shard is responsible for storing its state. Although the sharding is considered to be an off-chain solution, in reality,

it does not attempt to move the transactions off the chain but rather divides the main-chain into multiple independent groups so that the mining nodes are not responsible for mining every broadcasted transaction on the network. Each shard works separately as an independent blockchain in the network but has a strong cryptographic binding with the main-chain appearing as a Markle tree. These shards can be brought back to the main-chain at any time. This approach brought a significant increase in transaction throughput and a considerable drop in latency.

- **Hardware-assisted approaches**
  Many solutions use specialized trusted hardware devices either to improve consensus or to speed up the transaction process to solve scalability issues within the blockchain. The hardware includes high processing machines or Trusted Execution Environment (TEEs) to efficiently handle transactions with accuracy and speed.
- **Parallel mining/processing**
  Traditional blockchain is based on solo mining and offers limited transaction throughput and scalability. The parallel mining approach is used to improve blockchain scalability by concurrent mining of multiple blocks without changing the basic structure of the blockchain.
- **Redesigning blockchain**
  Some solutions rely on redesigning the blockchain to propose an alternative scalable DLT, such as Graphchain which uses DAG for a non-linear generation of blocks. on the other hand, some solutions are focused on designing an efficient strategy such as a novel consensus framework to address various aspects of scalability.

5.4 Use-Case for Empirical Investigation of Blockchain Scalability

To observe the impact of scalability with e-voting as an application example, a different set of experiments have been conducted by building different cases over the proposed testbed to closely monitor certain controlled variations in the blockchain processing domain and to achieve an optimum level of performance. Each case corresponds to different conditions for evaluation. These cases differ in several ways including block generation rate, different values for maximum block size, etc. In each of these cases, various transactions were carried out to transfer a vote from a voter's address to the candidate's address. Based upon the output data of these cases, the response of the blockchain was recorded and analyzed to understand the relationship between transaction mining time and the waiting time of a transaction in the pool. Similarly, using the experimental results of case one and case two, a critical empirical analysis was performed to compute various scalability related parameters like the maximum number of voting transactions a block can contain under these case, an average number of voting transactions a block currently contains in case one and two, size of the voting transaction, the maximum speed of transaction processing the voting system in case one and two can support, and the current operational transaction processing speed of the system.

5.4.1 System Model and Implementation

The process begins when a voter moves his voting token to the candidate's address. A system model may be divided into two major zones (vote casting and its administration). A voting token in the blockchain is analogous to a vote in the real-life election process. This voting token is assigned to every voter. Similarly, a candidate's address is a public address on the blockchain which will be used by the voters to transfer the vote to their desired candidates.

To keep the clarity in the scheme and to focus on the problem at an initial level, a total of six voters were made to vote for one candidate while the remaining four voters were made to vote for the other candidate, out of the overall population of 10 voters. Initially, this simple scenario was followed up as the focus is more towards analyzing the response of the blockchain platform for transaction processing and not on the application running on it.

Since the domain of the research is voting, the asset in the voting scenario may be referred to as a vote (a voting asset). So, after creating votes as assets initially in the public blockchain and then in the private blockchain, these voting assets are then transferred to an address which will act as an authority to transfer the right of the vote (by performing transaction of moving voting asset) to the eligible candidates(one vote to each candidate). The vote has been made immutable and indivisible. This is achieved while creating a voting asset on the blockchain (using the Multichain platform) by setting the attributes in such a way that the minimum unit of the vote is set to 1. Secondly, while assigning these votes to the voters, care has been taken to move only one vote to the voter through an automated offline process as mentioned in the previous section. The balance against the address of the vote issuing authority is then set to 0 which is the only address of blockchain that has the authority to distribute votes by having the right on this particular voting asset.

Figure 5.3 represents the basic infrastructure of the testbed that has been deployed for executing different cases (as mentioned above) for carrying out scalability stress testing. Seed node is responsible for initialization the core blockchain while connected nodes ensure the decentralization of the blockchain network. The mining process is carried out by all the nodes (in case 1 and 2). The mining pool is shared among connected and seed nodes. All participating nodes are fully running nodes and keep their local copy of the complete blockchain as well. In the two presented cases, blockchain data was created and manipulated directly on the blockchain instead of moving it from the application so that initially an

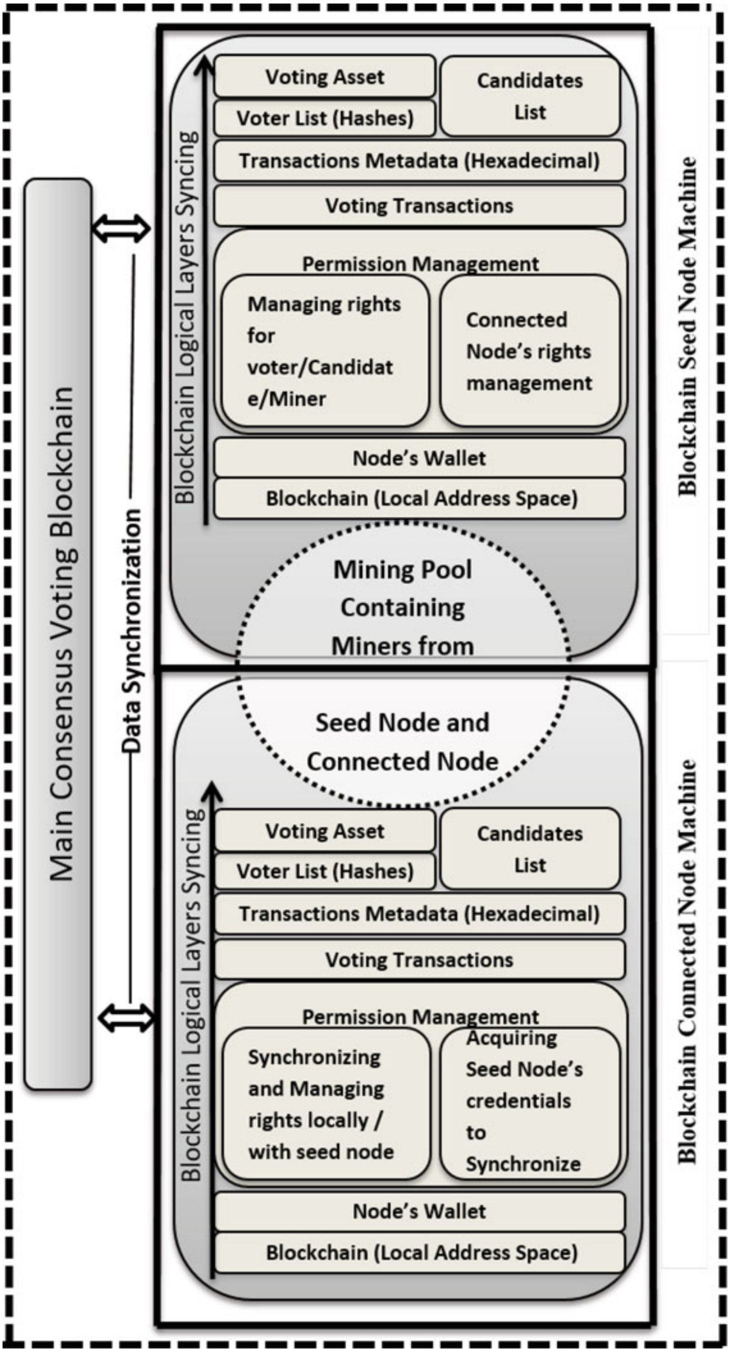

Main Consensus Voting Blockchain
Data Synchronization
Blockchain Logical Layers Syncing
Voting Asset
Voter List (Hashes)
Candidates List
Transactions Metadata (Hexadecimal)
Voting Transactions
Permission Management
Managing rights for voter/Candidat e/Miner
Connected Node's rights management
Node's Wallet
Blockchain (Local Address Space)
Blockchain Seed Node Machine
Mining Pool Containing Miners from Seed Node and Connected Node
Blockchain Logical Layers Syncing
Voting Asset
Voter List (Hashes)
Candidates List
Transactions Metadata (Hexadecimal)
Voting Transactions
Permission Management
Synchronizing and Managing rights locally / with seed node
Acquiring Seed Node's credentials to Synchronize
Node's Wallet
Blockchain (Local Address Space)
Blockchain Connected Node Machine

Fig. 5.3 Testbed for proposed system implementation

ideal relationship may be worked upon for having empirical analysis without any programming delay.

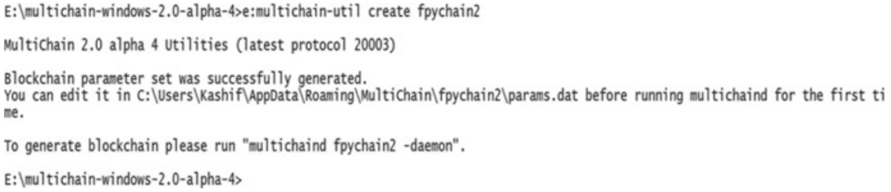
5.5 Evaluation and Analysis

This section presents the experimentation conducted to investigate the factors affecting the scalability of the blockchain-based e-voting use-case. As part of the experimentation, different scenarios were established against varying values for block size, block creation rate (to find out significant trade-offs between block size), block generation rate, and the total number of voting transactions per second.

5.5.1 Experimentation Scenario 1

The experimentation with a voting population of 10 voters (as an appropriate size to demonstrate a prototype voting process) has been conducted in a hypothetically more ideal environment where the network latency was not of much concern because the voting transactions have not been performed through remote voting clients. With these assumptions this scenario is useful to highlight the level of difficulty for a miner to mine a block. In addition, it also reveals the size of the block that contains the transactions and the average rate of the creation of block in the blockchain. This limited scale voting process is helpful to draw inferences regarding the behavior of the system concerning their existing and theoretical limits of transaction processing speed. This approach also reveals the significant trade-off between transaction block size, block generation rate, and transaction processing speed. Overall, it is easy to determine how scalable the system can be made and what possible compromises may be needed to match the desired result of a given voting scenario. The blockchain was created and initialized on Multichain [5] platform using the following command in Fig. 5.4.

A configuration file is created in the installation directory after the creation of blockchain. This file contains the parameters which are required to initialize the blockchain and therefore cannot be altered once the blockchain starts running. The

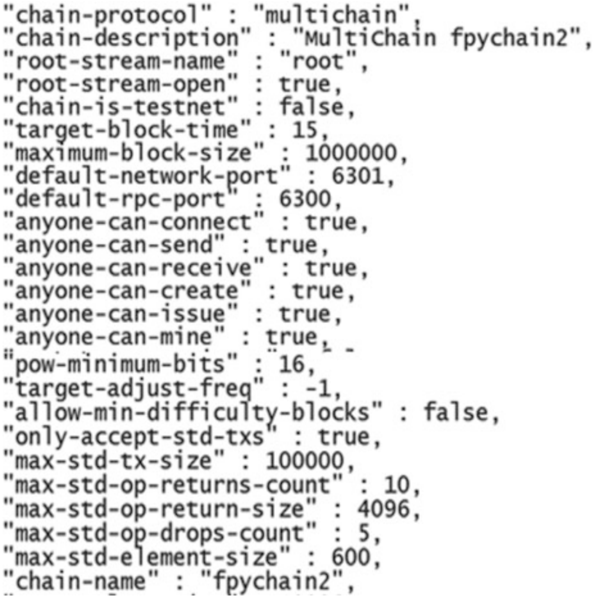


```
E:\multichain-windows-2.0-alpha-4>e:multichain-util create fpychain2

MultiChain 2.0 alpha 4 Utilities (latest protocol 20003)

Blockchain parameter set was successfully generated.
You can edit it in C:\Users\Kashif\AppData\Roaming\MultiChain\fpychain2\params.dat before running multichaind for the first ti
me.

To generate blockchain please run "multichaind fpychain2 -daemon".

E:\multichain-windows-2.0-alpha-4>
```

Fig. 5.4 Blockchain initialization for case 1

Table 5.2 Initial parameter values for case 01

	Blockchain parameters				
Platform	Difficulty	No. of miners	Block generation rate(s)	Max. allowable size (MB)	Bits required for PoW
Windows	00001526	10	15	1	16



```
"chain-protocol" : "multichain",
"chain-description" : "MultiChain fpychain2",
"root-stream-name" : "root",
"root-stream-open" : true,
"chain-is-testnet" : false,
"target-block-time" : 15,
"maximum-block-size" : 1000000,
"default-network-port" : 6301,
"default-rpc-port" : 6300,
"anyone-can-connect" : true,
"anyone-can-send" : true,
"anyone-can-receive" : true,
"anyone-can-create" : true,
"anyone-can-issue" : true,
"anyone-can-mine" : true,
"pow-minimum-bits" : 16,
"target-adjust-freq" : -1,
"allow-min-difficulty-blocks" : false,
"only-accept-std-txs" : true,
"max-std-tx-size" : 100000,
"max-std-op-returns-count" : 10,
"max-std-op-return-size" : 4096,
"max-std-op-drops-count" : 5,
"max-std-element-size" : 600,
"chain-name" : "fpychain2",
```

Fig. 5.5 Blockchain specifications for case 01

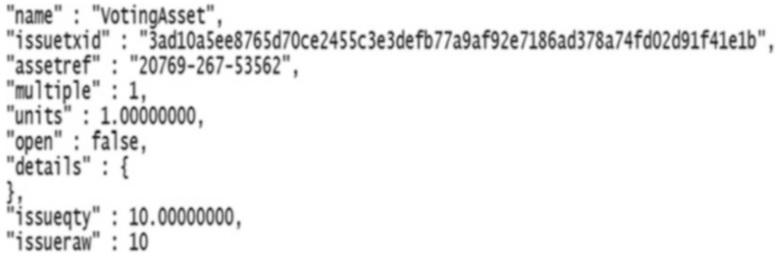
detail of these parameters is shown in Table 5.2. In the first run of the experiment, the time interval for adding new blocks to the blockchain was set at 15 s. This means that at a regular interval of 15 s, unconfirmed transactions will be processed, packed, and appended to the existing chain of blocks in the form of a new block. Here, the maximum memory size of the block is limited to 1 megabyte. To make the blockchain up and running, there have been a total of 60 blocks at the start of the blockchain (which is set by default in Multichain).

Table 5.2 shows the initial parameter values for observing blockchain behavior related to scalability. Figure 5.5 shows the specification of these parameters at Multichain blockchain output terminal.

Here, the interest is to derive the results which may enable us to test and investigate the impact of the above parameters on key performance indicators of scalability, which includes transactions processing per second, the size of the transaction, the maximum and average number of current transactions which can be accommodated, and the number of bytes the block is currently taking for the experimentation condition according to this test case. These parameters are very crucial in identifying the scalability related constraints of blockchain process execution. If the blockchain generation rate is not made compatible with the overall system's desired expected output, some serious performance degrades may be experienced. For instance, if it is kept at a rate that is too high, there may exist a situation in which too many transactions keep waiting in the pool of unconfirmed transactions which can ultimately overflow the volume of the memory pool of the node. This situation may cause the blockchain to either respond slowly or even become unresponsive at all. Similarly, in a situation where the blocks are generated at a very high rate, the blockchain is stressed out and starts mining the empty blocks even without having a transaction. Such situation is very common in the private blockchains. Hence, an optimum level must be set to keep a matching condition of equilibrium between the rate of incoming transactions and the rate at which the transactions are mined into the new blocks. This can be decided by the nature of the application which is built on blockchain [23].

As mentioned earlier that Multichain has been used here as a platform for carrying out the experimentation work. The addresses of candidates, voters, and vote issuing authority have been generated using the command "getnewaddress" on "Multichain-cli" executable. Figure 5.6 shows the creation of voting asset which was later on assigned to voter hashes so that these voters may cast these votes to their candidate using their hashes (Table 5.3).

Let the arrival time of the transaction to come to the unconfirmed pool is T_au, time taken by a node for adding the transaction to its local wallet is T_nw, total time taken for confirmation of the transaction from unconfirmed pool to enter into



```
"name" : "VotingAsset",
"issuetxid" : "3ad10a5ee8765d70ce2455c3e3defb77a9af92e7186ad378a74fd02d91f41e1b",
"assetref" : "20769-267-53562",
"multiple" : 1,
"units" : 1.00000000,
"open" : false,
"details" : {
},
"issueqty" : 10.00000000,
"issueraw" : 10
```

Fig. 5.6 Voting asset for case 01

Table 5.3 Sample voting transactions

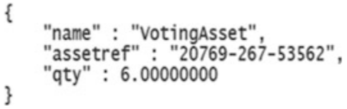
S. No	Voter hash	Candidate hash	Transaction hash
01	1Zq8GjEd5NUnXqqQjsnKF zuj5v92fvt4qekfsQ	1SS6H6G7aU96oMerxDu752 FYAsn4ZaKVHZVyNR	b9c8432b858c47d23ac44a44a64ca0c 44cbc11c5b5ef8216cef9934af651ef7d
02	1Nz4aWjykexrxgeJq2x RLixrmj4hU3oKaHtyCD	1SS6H6G7aU96oMerxD u752FYAsn4ZaKVHZVyNR	8de26636e4b0c035b55cd82cea119d1 1e24321c7b771779720b69bd74a925b6b
03	16ckT5efpKVzu6fM7d6 TvU5Uae7fgAxqv5Q2TK	1SS6H6G7aU96oMerxDu 752FYAsn4ZaKVHZVyNR	7a763820bb758ac0d060da5be8c7598e 43a495e608f654e4f3e1a3fe8d9e4b3d
04	1bPRUUm8EeYA1cGLX55 85RNAdWhxj3uyahQ1z6	1SS6H6G7aU96oMerxDu7 52FYAsn4ZaKVHZVyNR	1ee2338ae825b3b78c4a21492ea23f8e 03cb5723d3bd3216dbde6bb39f3e78ed
05	1DtQn2g3MCNLtHZxGhUj 67Kc4SWTeXi5jTeeCy	1SS6H6G7aU96oMerx Du752FYAsn4ZaKVHZVyNR	91674fdb026e0a8fb0e407ea2ee69eb 4123377ed9a0ee58c93d9e50db0fe2f09

Table 5.4 Voting transaction details

Tx No.	Tx ID	$T_{au} = T_{nw}$	T_{ad}	T_{ctn}	B_s
01	b9c8432b858c47d23ac44a44a64ca0c44cbc11c5b5ef8216cef9934af651ef7d	1542300652	1542300661	9	516
02	8de26636e4b0c035b55cd82cea119d11e24321c7b771779720b69bd74a925b6b	1542300713	1542300724	11	517
03	7a763820bb758ac0d060da5be8c7598e43a495e608f654e4f3e1a3fe8d9e4b3d	1542300750	1542300761	11	516
04	1ee2338ae825b3b78c4a21492ea23f8e03cb5723d3bd3216dbde6bb39f3e78ed	1542300785	1542300787	02	517
05	91674fdb026e0a8fb0e407ea2ee69eb4123377ed9a0ee58c93d9e50db0fe2f09	1542300994	1542301002	08	516

Table 5.5 Voting transaction details

S. No	Candidate hash	Asset name	Asset reference	Quantity
01	1SS6H6G7aU96oMerxDu752FYAsn4ZaKVHZVyNR	Voting asset	20769-267-53562	6
02	1YetjPS8i6km12QavaMX2PJkRU1LuBmcYcG541	Voting asset	20769-267-53562	4



```
{
    "name" : "VotingAsset",
    "assetref" : "20769-267-53562",
    "qty" : 6.00000000
}
```

Fig. 5.7 Final result count for candidate in case 1

blockchain is T_ctn, timestamp of the block into the main blockchain in which the transaction was confirmed is T_ad, and the block size is B_s in bytes, refer to Table 5.4. Here time is in UNIX style time stamp. The final statistics of votes are shown in Table 5.5 followed by the output screen at blockchain in Fig. 5.7. As shown in Table 5.5, the first set of parameters displays the balance of the first candidate while the second set of parameters shows the total vote count for the second candidate. Table 5.4 shows the blockchain response under the settings, presented in case 01.

5.5.1.1 Graphical Analysis

This section analyses the graphs which have been generated with the experimental data under the conditions defined for case 01. The study tries to investigate the issues related to the blockchain performance and scalability by observing the behavior of these graphs.

The graph in Fig. 5.8 shows that new transactions are arriving at different times but are added immediately to the node's wallet upon arrival. So, the graph shows that there is no delay in the transition during its journey from pool to the blockchain. When this behavior is observed in the context of throughput and blockchain

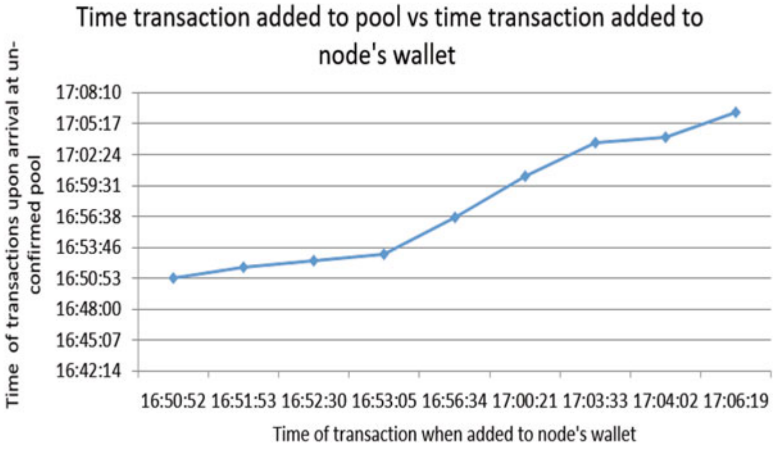


Fig. 5.8 Time taken by miners to add unconfirmed transactions from pool to their local wallet

performance, which is characterized with transaction processing speed, maximum number of transactions per block, etc., it may be considered that the transactions movement from pool to node's wallet is not a major factor of consideration in the case 1. This may also be applicable in majority of other cases in general, as it will be shown through the calculation for determining transaction processing speed.

But there are some other (although indirect) factors that may affect the behavior of this scenario in the blockchain. These factors include pool size and the number of dependent transactions. According to the official documentation of Multichain platform there is no fixed barrier in limiting the capacity of the pool but off course nodes have constraints in picking up transactions from other nodes. Another thing to ponder here is about the dependent transactions such as multiple transactions from the same node address which are dependent on each other. If the subsequent dependent transactions start to arrive earlier than its previous transactions on which it depends then the earlier arrived transactions cannot be confirmed until its previous transaction gets into the block.

The graph in Fig. 5.9 represents the arrival of voting transactions at different times into the node's wallet. These transactions were picked up by the miners to first begin the race of mining those transactions. Therefore, the transaction processing depends mainly upon the proof of work that is performed by the miner while keeping the average block generation rate constant. By observing the pattern of the graph it can be said that it is not necessarily important that the transactions which arrive earlier to the pool take a shorter time to add to the block even though when the difficulty is kept constant among all the transactions.

If the focus is to get the throughput of the system, transaction processing speed can be one of the vital factors to be worked upon. Figure 5.10 elaborates the time taken by each voting transaction in the experiment. Since the target block time is set to 15 s, this means that each block will be generated at an interval of 15 s. This

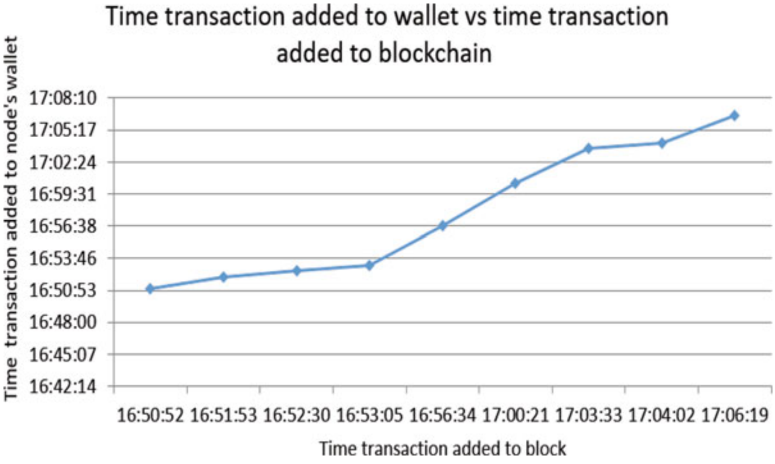


Fig. 5.9 Time taken by miners to confirm transactions into the block

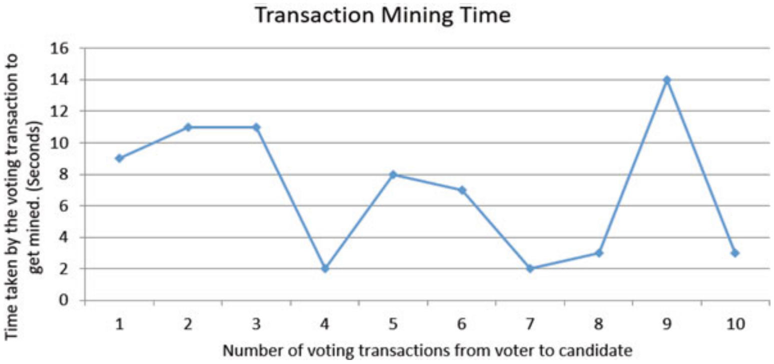
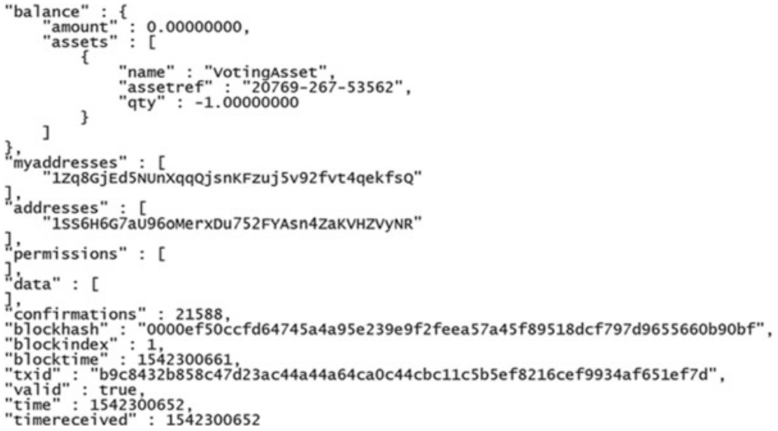


Fig. 5.10 Transaction mining time

interval is an average time of block generation which means that if a block is mined much earlier than the target block time then the rest of the blocks will be mined at a rate to restore the average rate of block generation. The difference in transaction delays is due to the time taken by the proof of work. The difficulty level has been kept constant throughout all the readings so that it may not affect the dataset and a direct relationship may be built between the number of transactions and their processing time to check the scalability of the system. The total amount of time taken by a transaction to be published also depends upon its arrival time (in the case of our Multichain platform). Suppose, in this experiment where the maximum target block time is 15 s, a transaction comes into a wallet and a block is about to be published within 3 or 4 s to keep the average time of 15 s, then the transaction (if it does not depend upon any unconfirmed transaction) will most likely be published in the main blockchain within those 3 or 4 s, provided that the block size does not exceed the maximum size limit. The situation may be entirely different when the same transaction would arrive just after the latest block was added.

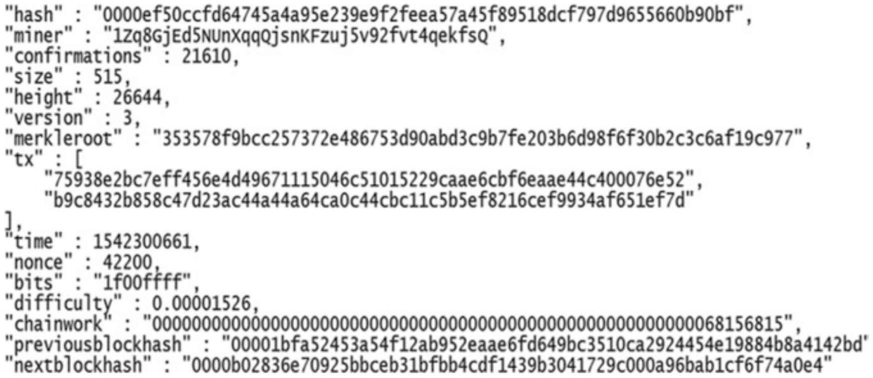
5.5.1.2 Scalability Test for Case 1

To test how scalable the system is, let us represent the size of the voting transaction as T_s, block generation rate as B_t, the maximum amount of data that can be held by the block as B_{d*max}, and the average current size of block for our voting transaction as B_{d*avg}. To calculate the maximum number of transactions that can be accommodated per block T_{s*max} and the current average number of transaction T_{s*avg} per block, $T_s = 250$ bytes has been considered (voting transaction size is 250 bytes). The size of the transaction has been calculated as per the method which is described in Multichain documentation. According to this, the number of hexadecimal characters is counted by issuing "getrawtransaction" command to the Multichain node. The number of characters is then divided by 2 to get the size in bytes. Since the experimental work has been done on Multichain blockchain, therefore the size of the transaction may be obtained by taking the arithmetic sum of all the hexadecimal characters and dividing the sum by 2. In this case, the size is found to be 250 bytes. Similarly, as mentioned above (in Table 5.2), block generation rate has been set to 15 s that is B_t=15 s and B_{d*max} = 1 MB while B_{d*avg} in this test run is 516 bytes. The average number of transactions contained in a single block, in this case, $T_{(s*avg\_block)}$ is 2. The B_{d*avg} and T_{s*avg} can be found by issuing "getblock" command to Multichain command editor such as it has been shown in Fig. 5.11 (for transaction detail) and in Fig. 5.12 (B_{d*avg} and T_{s*avg}) for the output of transaction number one in voting hash table (see Table 5.4).



```
"balance" : {
    "amount" : 0.00000000,
    "assets" : [
        {
            "name" : "VotingAsset",
            "assetref" : "20769-267-53562",
            "qty" : -1.00000000
        }
    ]
},
"myaddresses" : [
    "1Zq8GjEd5NUnXqqQjsnKFzuj5v92fvt4qekfsQ"
],
"addresses" : [
    "1SS6H6G7aU96oMerxDu752FYAsn4ZaKVHZVyNR"
],
"permissions" : [
],
"data" : [
],
"confirmations" : 21588,
"blockhash" : "0000ef50ccfd64745a4a95e239e9f2feea57a45f89518dcf797d9655660b90bf",
"blockindex" : 1,
"blocktime" : 1542300661,
"txid" : "b9c8432b858c47d23ac44a44a64ca0c44cbc11c5b5ef8216cef9934af651ef7d",
"valid" : true,
"time" : 1542300652,
"timereceived" : 1542300652
```

Fig. 5.11 A transaction from voter to candidate



```
"hash" : "0000ef50ccfd64745a4a95e239e9f2feea57a45f89518dcf797d9655660b90bf",
"miner" : "1Zq8GjEd5NUnXqqQjsnKFzuj5v92fvt4qekfsQ",
"confirmations" : 21610,
"size" : 515,
"height" : 26644,
"version" : 3,
"merkleroot" : "353578f9bcc257372e486753d90abd3c9b7fe203b6d98f6f30b2c3c6af19c977",
"tx" : [
    "75938e2bc7eff456e4d49671115046c51015229caae6cbf6eaae44c400076e52",
    "b9c8432b858c47d23ac44a44a64ca0c44cbc11c5b5ef8216cef9934af651ef7d"
],
"time" : 1542300661,
"nonce" : 42200,
"bits" : "1f00ffff",
"difficulty" : 0.00001526,
"chainwork" : "0000000000000000000000000000000000000000000000000000000068156815",
"previousblockhash" : "00001bfa52453a54f12ab952eaae6fd649bc3510ca2924454e19884b8a4142bd'
"nextblockhash" : "0000b02836e70925bbceb31bfbb4cdf1439b3041729c000a96bab1cf6f74a0e4"
```

Fig. 5.12 Specification of block containing transaction of Fig. 5.11

Table 5.6 Initial parameter values for case 02

	Blockchain parameters				
Platform	Difficulty	No. of miners	Block generation rate(s)	Max. allowable size (MB)	Bits required for PoW
Windows	1.0000	10	60	1000	32

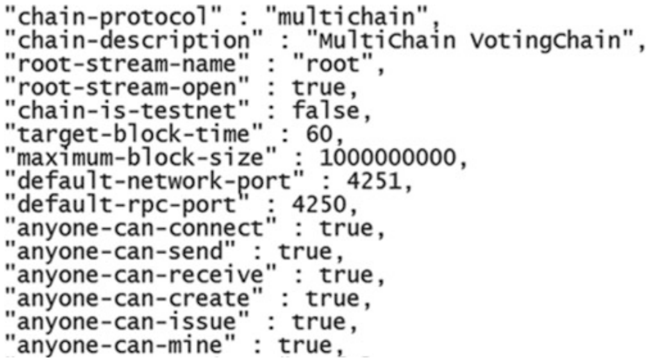
5.5.2 *Case 2*

In case 2, the same system has been tested under the stress of larger block size, increased block generation rate, and above all, the required amount of bits in proof of work has been extended to 32 bits. This implies that on average the miner has been asked to compute 2Î6 hashes to mine a block. These variations in the values of the parameters significantly affected the overall behavior of the system in comparison to case 1. Table 5.6 shows the modified parameter values for carrying out the scalability test in case 2.

To conduct a test for case 2, this time blockchain has been initialized with the parameters as shown in Table 5.6. Figure 5.13 shows the blockchain parameters on Multichain blockchain console.

Using the same conventions for denoting blockchain behavior as it has been shown earlier for case 1 in Table 5.4, Table 5.7 records the responses for case 2 following the same experiment of transferring vote from voters to candidates but this time under the settings of Table 5.7 for case 2.

The transaction number in Table 5.7 represents the same combination and order of voter and candidate hash as it has been shown in Table 5.3. Table 5.8 shows the final candidates' votes in the wallet of the blockchain node.



```
"chain-protocol" : "multichain",
"chain-description" : "MultiChain VotingChain",
"root-stream-name" : "root",
"root-stream-open" : true,
"chain-is-testnet" : false,
"target-block-time" : 60,
"maximum-block-size" : 1000000000,
"default-network-port" : 4251,
"default-rpc-port" : 4250,
"anyone-can-connect" : true,
"anyone-can-send" : true,
"anyone-can-receive" : true,
"anyone-can-create" : true,
"anyone-can-issue" : true,
"anyone-can-mine" : true,
```

Fig. 5.13 Blockchain specifications for case 02

Table 5.7 Voting transaction details

Tx No.	$T_{au} = T_{nw}$	T_{ad}	T_{ctn}	B_s
01	1543813800	1543817161	3361	5468
02	1543813851	1543817161	3310	5468
03	1543813898	1543817161	3263	5468
04	1543813938	1543817161	3223	5468
05	1543814081	1543817161	3080	5468
06	1543814231	1543817161	2930	5468
07	1543815716	1543817161	1445	5468
08	1543816072	1543817161	1089	5468
09	1543816105	1543817161	1056	5468
10	1543816316	1543817161	845	5468

Table 5.8 Final candidate votes in blockchain wallet

S. no	Candidate hash	Asset name	Asset reference	Quantity
01	1PxXxADdtsrG25cGBHqD8UugaTWbiPZ YSULiWS	Voting asset	3-265-63893	6
02	1WJkwH6ZYQnXBivdnGjoSqeC58Absmhp Ax5gLs	Voting asset	3-265-63893	4

5.5.2.1 Graphical Analysis

It can be seen in Fig. 5.14 that the transactions move instantly to node's wallet upon arrival. The noticeable thing here is that even if the transactions are issued with delays (as it is the case of reading number 7 when we move from left to right in the above graph), it has no impact on the time it takes to move the transaction into wallet.

Similarly, in Fig. 5.15, it can be seen that the transactions are getting confirmed to the same block due to increased block generation rate which was issued subsequently and also were independent of each other.

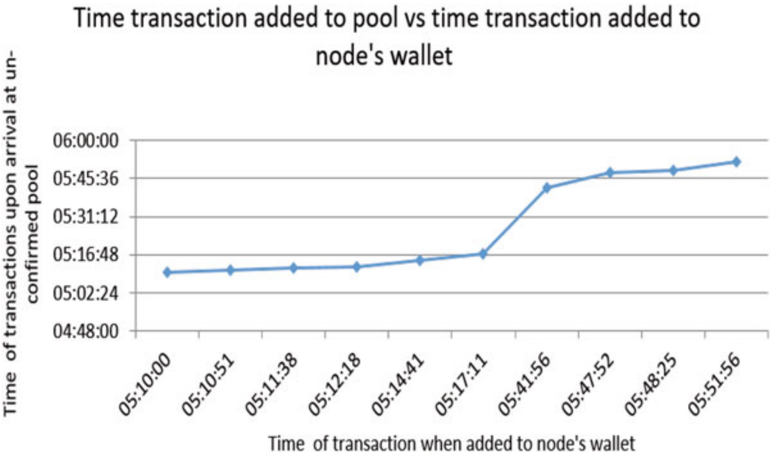


Fig. 5.14 Time taken by miners to add unconfirmed transactions from pool to the node wallet

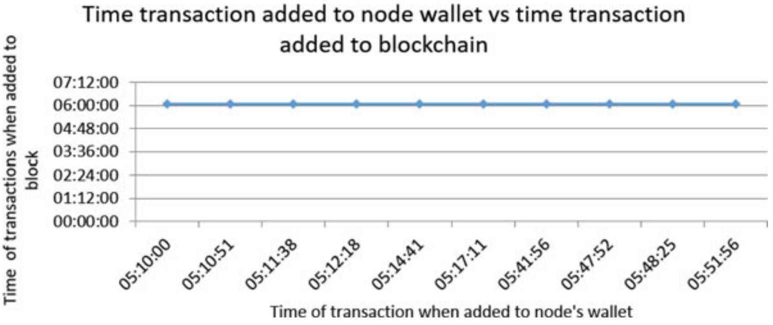


Fig. 5.15 Time taken by miners to confirm transactions into the block

To completely understand Fig. 5.16, it should be analyzed in the context of Fig. 5.13 also. The graph in Fig. 5.16 shows that the transactions which are coming late to the system surprisingly getting mined earlier than the previous transaction even though these transactions do not depend upon each other. The obvious and fundamental reason is that the new transactions are arriving at a time when there is some time left for the block to be added (four times more than in case 1). Another interesting point here to note is that in case 2, not only the block can contain more transactions but also this time the miner will have to work even harder as the proof of work has now been increased to 32 bits.

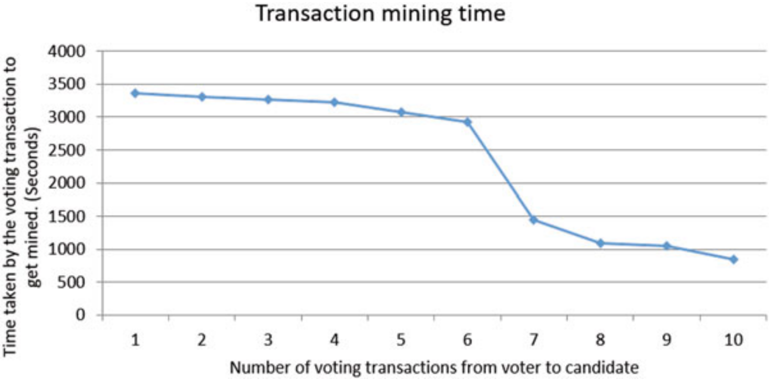


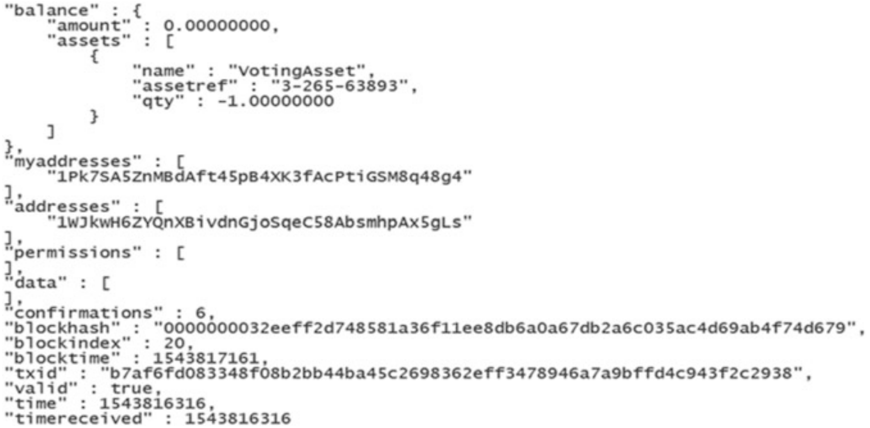
Fig. 5.16 Transaction mining time

5.5.2.2 Scalability Test for Case 02

In this case T_s has same value as it was before, i.e., 250 bytes, B_{d*max} is increased to 1GB while B_{d*avg} in this test run is 5468 bytes. Also in case 2, B_t has been set to 60 s and a number of required bits for proof of work has been set to 32 bits which moved the difficulty level to 1 (referring to Table 5.6).

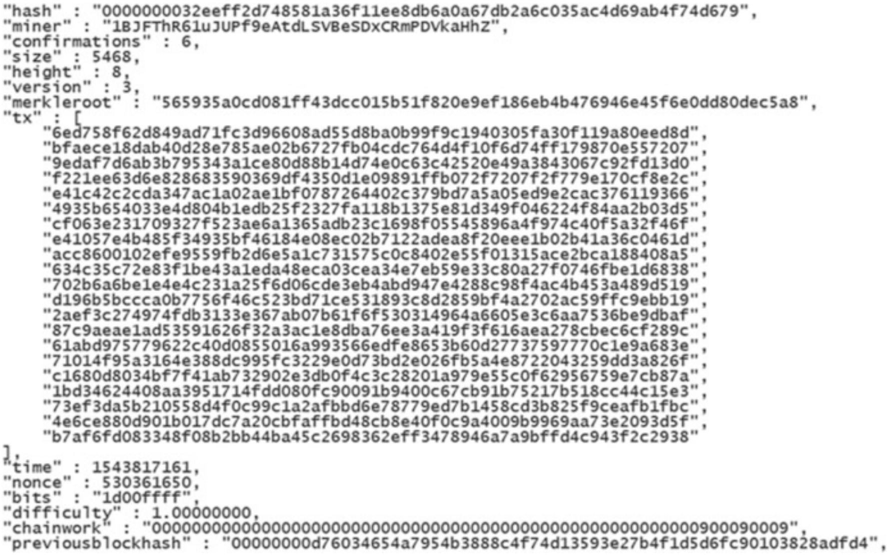
5.5.3 Comparative Analysis for Scalability Tests for Case 01 and 02

Scalability is a critical factor to determine the strengths and weaknesses of the system for the selection of optimum performance under various scenarios. As the study for case 1 and case 2 show that transaction processing speed is a key factor in evaluating the scalability of a system. There is also another aspect that is needed to look into while scaling the system. This is the context of the application domain in which blockchain is running. A significant ratio must be needed to maintain between the number of transactions that can be added to a block and the speed by which the new transactions are added to the block, i.e., it is $T_{(s*max\_block)}$ Vs. B_t. Since, if the ratio between the two is not maintained at an optimum level, there may arise a situation where the transactions will be started to confirm at a much lower rate due to large block size which will enforce many transactions to wait until the block is ready to be added to the main blockchain at a time to keep the average block generation as specified in the blockchain. This means that block generation rate and block size affect the rate at which transactions are coming to the pool and the rate at which these transactions are confirmed to the block. Therefore extreme care is needed to be taken in adjusting these parameters to fully capitalize the potential of blockchain in accordance with the condition (Figs. 5.17 and 5.18).



```
"balance" : {
    "amount" : 0.00000000,
    "assets" : [
        {
            "name" : "VotingAsset",
            "assetref" : "3-265-63893",
            "qty" : -1.00000000
        }
    ]
},
"myaddresses" : [
    "1Pk7SA5ZnMBdAft45pB4XK3fAcPtiGSM8q48g4"
],
"addresses" : [
    "1WJkwH6ZYQnXBivdnGjoSqeC58AbsmhpAx5gLs"
],
"permissions" : [
],
"data" : [
],
"confirmations" : 6,
"blockhash" : "0000000032eeff2d748581a36f11ee8db6a0a67db2a6c035ac4d69ab4f74d679",
"blockindex" : 20,
"blocktime" : 1543817161,
"txid" : "b7af6fd083348f08b2bb44ba45c2698362eff3478946a7a9bffd4c943f2c2938",
"valid" : true,
"time" : 1543816316,
"timereceived" : 1543816316
```

Fig. 5.17 A transaction from voter to candidate



```
"hash" : "0000000032eeff2d748581a36f11ee8db6a0a67db2a6c035ac4d69ab4f74d679",
"miner" : "1BJFThR61uJUPf9eAtdLSVBeSDxCRmPDVkaHhZ",
"confirmations" : 6,
"size" : 5468,
"height" : 8,
"version" : 3,
"merkleroot" : "565935a0cd081ff43dcc015b51f820e9ef186eb4b476946e45f6e0dd80dec5a8",
"tx" : [
    "6ed758f62d849ad71fc3d96608ad55d8ba0b99f9c1940305fa30f119a80eed8d",
    "bfaece18dab40d28e785ae02b6727fb04cdc764d4f10f6d74ff179870e557207",
    "9edaf7d6ab3b795343a1ce80d88b14d74e0c63c42520e49a3843067c92fd13d0",
    "f221ee63d6e828683590369df4350d1e09891ffb072f7207f2f779e170cf8e2c",
    "e41c42c2cda347ac1a02ae1bf0787264402c379bd7a5a05ed9e2cac376119366",
    "4935b654033e4d804b1edb25f2327fa118b1375e81d349f046224f84aa2b03d5",
    "cf063e231709327f523ae6a1365adb23c1698f05545896a4f974c40f5a32f46f",
    "e41057e4b485f34935bf46184e08ec02b7122adea8f20eee1b02b41a36c0461d",
    "acc8600102efe9559fb2d6e5a1c731575c0c8402e55f01315ace2bca188408a5",
    "634c35c72e83f1be43a1eda48eca03cea34e7eb59e33c80a27f0746fbe1d6838",
    "702b6a6be1e4e4c231a25f6d06cde3eb4abd947e4288c98f4ac4b453a489d519",
    "d196b5bccca0b7756f46c523bd71ce531893c8d2859bf4a2702ac59ffc9ebb19",
    "2aef3c274974fdb3133e367ab07b61f6f530314964a6605e3c6aa7536be9dbaf",
    "87c9aeae1ad53591626f32a3ac1e8dba76ee3a419f3f616aea278cbec6cf289c",
    "61abd975779622c40d0855016a993566edfe8653b60d27737597770c1e9a683e",
    "71014f95a3164e388dc995fc3229e0d73bd2e026fb5a4e8722043259dd3a826f",
    "c1680d8034bf7f41ab732902e3db0f4c3c28201a979e55c0f62956759e7cb87a",
    "1bd34624408aa3951714fdd080fc90091b9400c67cb91b75217b518cc44c15e3",
    "73ef3da5b210558d4f0c99c1a2afbbd6e78779ed7b1458cd3b825f9ceafb1fbc",
    "4e6ce880d901b017dc7a20cbfaffbd48cb8e40f0c9a4009b9969aa73e2093d5f",
    "b7af6fd083348f08b2bb44ba45c2698362eff3478946a7a9bffd4c943f2c2938"
],
"time" : 1543817161,
"nonce" : 530361650,
"bits" : "1d00ffff",
"difficulty" : 1.00000000,
"chainwork" : "0000000000000000000000000000000000000000000000000000000900090009",
"previousblockhash" : "00000000d76034654a7954b3888c4f74d13593e27b4f1d5d6fc90103828adfd4",
```

Fig. 5.18 Specification of block containing transaction of Fig. 5.18

5.6 Summary and Open Challenges

This chapter has focused on investigating the challenge of scalable blockchains, identifying constraints which can affect the performance of a blockchain-based system. We used blockchain-based e-voting as a use-case to conduct in-depth empirical analysis of the system. The experimentation used varying settings with respect to the level of difficulty for a miner to mine a block, the size of the block that contains the transactions and the average block generation rate. We use the outcomes of this experimentation to draw inferences regarding the behavior of the system with respect to their existing and theoretical limits of transaction processing speed. The experimentation also revealed the significant trade-off between transaction block size, block generation rate, and speed of the processing of a transaction. Overall, this chapter highlights potential avenues to achieve scalable solutions using blockchains while highlighting potential trade-offs among different parameters.

5.6.1 Open Issues and Challenges

This chapter has contributed towards identifying methods to achieve scalable blockchain; however, there remain challenges which require further work. We present prominent such challenges below.

- Sharing is one of the most prominent approaches to address the challenge of scalability. However, existing sharding-based approaches including Elastico[35], Omniledger [33], and Rapidchain [57] are either suitable for public (permission-less) blockchain scenarios or require a trusted hardware setup to reduce communication overhead [31]. Moreover, the Communication Cost Per Transaction (CCPT) in the order of O(n) is one of the prime objectives to blockchain scalability [58]. However, this is achieved either by compromising one or more important factors including reliability and decentralization, or require a TEE (Trusted Execution Environment) setup including trusted hardware and rational behavior of all nodes. Furthermore, sharding approaches also lack a BFT mechanism to prevent a shard from being taken over by malicious adversary for non-crypto currency based blockchains scenarios.
- The performance of consortium (permissioned) blockchain can be significantly enhanced by using a powerful trusted hardware setup which consequently, decreases the Block Generation Rate (BGR) and subsequently, leads to improved transaction throughput[31, 45–47, 59]. However, an attractive incentive mechanism, in permission-less scenarios, must be in place to encourage the owners of mining towards the use of resourceful hardware.
- Consensus algorithm is a critical component within blockchain stack and has a fundamental role in achieving scalable operation. In this respect, existing work such as [60, 61] have conducted a feature-based comparison of consensus algorithms. However, an empirical comparative study is deemed essential for evaluating the performance and applicability of specific consensus algorithms in various scenarios.

References

1. S. Nakamoto, Bitcoin: a peer-to-peer electronic cash system. Cryptography Mailing list at https://metzdowd.com (2009)
2. K.M. Khan, J. Arshad, M.M. Khan, Secure digital voting system based on blockchain technology. Int. J. Electron. Gov. Res. **14**(1), 53–62 (2018)
3. N. Nizamuddin, K. Salah, M. Ajmal Azad, J. Arshad, M.H. Rehman, Decentralized document version control using Ethereum blockchain and IPFS. Comput. Elect. Eng. **76**, 183–197 (2019)
4. Ethereum Project. Blockchain app platform. Available at: https://ethereum.org/en/ [last visited: 5th June 2021]
5. Multichain. Open platform for blockchain applications. Available at: https://www.multichain.com/ https://ethereum.org/en/ [last visited: 5th June 2021]
6. Hyperledger. Hperledger fabric. Available at: https://www.hyperledger.org/ [last visited: 5th June 2021]
7. K.M. Khan, J. Arshad, M.M. Khan, Investigating performance constraints for blockchain based secure e-voting system. Fut. Gen. Comput. Syst. **105**, 13–26 (2020)
8. N. Kshetri, J. Voas, Blockchain-enabled e-voting. IEEE Softw. **35**(04), 95–99 (2018)
9. X. Sun, Q. Wang, P. Kulicki, M. Sopek, A simple voting protocol on quantum blockchain. Int. J. Theor. Phys. **58**(1), 275–281 (2019)
10. B. Shahzad, J. Crowcroft, Trustworthy electronic voting using adjusted blockchain technology. IEEE Access **7**, 24477–24488 (2019)
11. D. Chaum, Blind signatures for untraceable payments, in *Advances in Cryptology* (Springer, Berlin, 1983), pp. 199–203
12. A. Back et al., Hashcash-a denial of service counter-measure (2002)
13. T. McConaghy, R. Marques, A. Müller, D. De Jonghe, T. McConaghy, G. McMullen, R. Henderson, S. Bellemare, A. Granzotto, Bigchaindb: A scalable blockchain database, white paper, BigChainDB. (2016)
14. K. Croman, C. Decker, I. Eyal, A.E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. Sirer, D. Song, R. Wattenhofer, On scaling decentralized blockchains, in *Financial Cryptography and Data Security. FC 2016*, vol. 9604 (2016), pp. 106–125
15. G. Karame, On the security and scalability of bitcoin's blockchain, in *The 2016 ACM SIGSAC Conference* (2016), pp. 1861–1862
16. Z. Zheng, S. Xie, H.-N. Dai, X. Chen, H. Wang, Blockchain challenges and opportunities: a survey. Int. J. Web Grid Ser. **14**, 352 (2018)
17. Segregated Witness Proposal, Available at: https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki [last visited: 5th June 2021]
18. Z. Zheng, S. Xie, H. Dai, X. Chen, H. Wang, An overview of blockchain technology: Architecture, consensus, and future trends, in *2017 IEEE International Congress on Big Data (BigData Congress)* (2017), pp. 557–564
19. X. Xu, I. Weber, M. Staples, L. Zhu, J. Bosch, L. Bass, C. Pautasso, P. Rimba, A taxonomy of blockchain-based systems for architecture design, in *2017 IEEE International Conference on Software Architecture (ICSA)* (2017), pp. 243–252
20. M. Vukolić, The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication, in *Open Problems in Network Security*, ed. by D. Camenisch, J. Kesdoğan (Springer International Publishing, Cham, 2016), pp. 112–125
21. M.A.-B. Shehar Bano, G. Danezis, The road to scalable blockchain designs. Login **42**(4), 31–42 (2017)
22. I. Eyal, A.E. Gencer, E.G. Sirer, R.V. Renesse, Bitcoin-ng: a scalable blockchain protocol, in *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, Santa Clara (USENIX Association, Berkeley, 2016), pp. 45–59
23. S. Kim, Y. Kwon, S. Cho, A survey of scalability solutions on blockchain, in *2018 International Conference on Information and Communication Technology Convergence (ICTC)* (2018), pp. 1204–1207

24. J. Rubin, M. Naik, Merkelized abstract syntax trees(2014). Available at: raw.githubusercontent.com
25. M. Scherer, Performance and scalability of blockchain networks and smart contracts. Master's Thesis, UmeÃe University, Department of Computing Science (2017)
26. C. Li, P. Li, D. Zhou, W. Xu, F. Long, A. Yao, Scaling Nakamoto Consensus to Thousands of Transactions per Second. arXiv:1805.03870 [cs] (2018)
27. S. Goswami, Scalability analysis of blockchains through blockchain simulation (2017)
28. G. Malavolta, P. Moreno-Sanchez, C. Schneidewind, A. Kate, M. Maffei, Anonymous multi-hop locks for blockchain scalability and interoperability, in *NDSS* (2019)
29. J. Poon, T. Dryja, The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. Technical Report
30. Radian Network, Available at: https://raiden.network/ [last visited: 5th June 2021]
31. H. Dang, T. Dinh, D. Loghin, E.-C. Chang, Q. Lin, B. Ooi, Towards scaling blockchain systems via sharding, in *SIGMOD '19: Proceedings of the 2019 International Conference on Management of Data* (2019), pp. 123–140
32. L. Kan, Y. Wei, A.H. Muhammad, W. Siyuan, G. Linchao, H. Kai, A multiple blockchains architecture on inter-blockchain communication, in *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)* (2018), pp. 139–145
33. E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, B. Ford, Omniledger: a secure, scale-out, decentralized ledger via sharding, in *2018 IEEE Symposium on Security and Privacy (SP)* (2018), pp. 583–598
34. W. Li, A. Sforzin, S. Fedorov, G.O. Karame, Towards scalable and private industrial blockchains, in *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, BCC '17 (Association for Computing Machinery, New York, 2017), pp. 9–14
35. L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, P. Saxena, A secure sharding protocol for open blockchains, in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (2016), pp. 17–30
36. K.R. Özyılmaz, H. Patel, A. Malik, Split-scale: scaling bitcoin by partitioning the UTXO space, in *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)* (2018), pp. 41–45
37. Z. Ren, E. Zekeriya, A scale-out blockchain for value transfer with spontaneous sharding (2018)
38. Y. Yu, R. Liang, J. Xu, A scalable and extensible blockchain architecture, in *2018 IEEE International Conference on Data Mining Workshops (ICDMW)* (2018), pp. 161–163
39. M. Zamani, M. Movahedi, M. Raykova, Rapidchain: scaling blockchain via full sharding, in *CCS '18: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (2018), pp. 931–948
40. G. He, W. Su, S. Gao, Chameleon: a scalable and adaptive permissioned blockchain architecture, in *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)* (2018), pp. 87–93
41. U. Klarman, S. Basu, A. Kuzmanovic, E.G. Sirer, Bloxroute: a scalable trustless blockchain distribution network (2018). https://bloxroute.com/wp-content/uploads/2019/01/whitepaper-V1.1-1.pdf
42. A. Kuzmanovic, Net neutrality: unexpected solution to blockchain scaling. Queue **17**, 20–78 (2019)
43. Z. Gao, L. Xu, L. Chen, N. Shah, Y. Lu, W. Shi, Scalable blockchain based smart contract execution, in *2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS)* (2017), pp. 352–359
44. S.S. Hazari, Q.H. Mahmoud, A parallel proof of work to improve transaction speed and scalability in blockchain systems, in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)* (IEEE, Piscataway, 2019), pp. 0916–0921
45. J. Liu, W. Li, G.O. Karame, N. Asokan, Scalable byzantine consensus via hardware-assisted secret sharing. IEEE Trans. Comput. **68**(1), 139–151 (2019)

46. J. Lind, O. Naor, I. Eyal, F. Kelbert, E.G. Sirer, P. Pietzuch, Teechain: A secure payment network with asynchronous blockchain access, in *Proceedings of the 27th ACM Symposium on Operating Systems Principles*, (2019), pp. 63–79
47. A. Sanka, R.C.C. Cheung, Efficient high performance FPGA based NoSQL caching system for blockchain scalability and throughput improvement, in *2018 26th International Conference on Systems Engineering (ICSEng)* (2018), pp. 1–8
48. R. Dennis, G. Owenson, B. Aziz, A temporal blockchain: a formal analysis, in *2016 International Conference on Collaboration Technologies and Systems (CTS)* (2016), pp. 430–437
49. C. Ehmke, F. Blum, C. Friedrich, Proof-of-property – a lightweight and scalable blockchain protocol (2018)
50. X. Fan, Q. Chai, Roll-DPoS: a randomized delegated proof of stake scheme for scalable blockchain-based internet of things systems, in *MobiQuitous '18: Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services* (2018), pp. 482–484
51. X. Fan, Scalable practical byzantine fault tolerance with short-lived signature schemes, in *Proceedings of the 28th Annual International Conference on Computer Science and Software Engineering*, CASCON '18 (IBM Corp., Berkeley, 2018), pp. 245–256
52. H. Gupta, D. Janakiram, Colosseum: a scalable permissioned blockchain over structured network, in *BCC '19: Proceedings of the Third ACM Workshop on Blockchains, Cryptocurrencies and Contracts* (2019), pp. 23–25
53. X. Min, Q. Li, L. Liu, L. Cui, A permissioned blockchain framework for supporting instant transaction and dynamic block size, in *2016 IEEE Trustcom/BigDataSE/ISPA* (2016), pp. 90–96
54. P. Otte, M. de Vos, J. Pouwelse, TrustChain: A Sybil-resistant scalable blockchain. Futur. Gener. Comput. Syst. **107**, 770–780 (2020)
55. P. Thakkar, S. Nathan, B. Viswanathan, Performance benchmarking and optimizing hyperledger fabric blockchain platform, in *2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)* (IEEE, Piscataway, 2018), pp. 264–276
56. K. Zhang, H. Jacobsen, Towards dependable, scalable, and pervasive distributed ledgers with blockchains, in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)* (2018), pp. 1337–1346
57. X. Boyen, C. Carr, T. Haines, Graphchain: a blockchain-free scalable decentralised ledger, in *BCC '18: Proceedings of the 2nd ACM Workshop on Blockchains, Cryptocurrencies, and Contracts* (2018), pp. 21–33
58. G. Wang, Z.J. Shi, M. Nixon, S. Han, Smchain: a scalable blockchain protocol for secure metering systems in distributed industrial plants, in *Proceedings of the International Conference on Internet of Things Design and Implementation*, IoTDI '19 (Association for Computing Machinery, New York, 2019), pp. 249–254
59. R. Dennis, J.P. Disso, An analysis into the scalability of bitcoin and ethereum. Third International Congress on Information and Communication Technology. (Springer, Singapore, 2019), pp. 619–627
60. L.M. Bach, B. Mihaljevic, M. Zagar, Comparative analysis of blockchain consensus algorithms, in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (IEEE, Piscataway, 2018), pp. 1545–1550
61. N. Chaudhry, M.M. Yousaf, Consensus algorithms in blockchain: comparative analysis, challenges and opportunities, in *2018 12th International Conference on Open Source Systems and Technologies (ICOSST)* (IEEE, Piscataway, 2018), pp. 54–63

Chapter 6
Interoperability Among Heterogeneous Blockchains: A Systematic Literature Review

Manar Abu Talib, Sohail Abbas, Qassim Nasir, Fatima Dakalbab, Takua Mokhamed, Khawla Hassan, and Khaldoun Senjab

6.1 Introduction

Blockchain technology has gained a great deal of attention in both industry and analysis in recent years. The rapid growth of this technology has led to the development of numerous different blockchain platforms and decentralized applications which serve not only cryptocurrency domains but also fields such as banking, supply chain management, healthcare, and other fields [64]. However, some concerns prevent the potential mass adoption of blockchain applications. The most common issue is the lack of interoperability. As many blockchain networks exist as isolated systems enabling all operations to be conducted locally, their structures and policies prevent communication between different blockchains, limiting their ability to transfer to and from various blockchains regardless of differences in language, interface, and execution platform. This issue attracted the attention of many researchers and blockchain application developers. They proposed and designed many different solutions to overcome this incompatibility problem and provide interoperability to blockchain systems.

We include the following definition of an "interoperable blockchain architecture" using the NIST [80] definition of blockchain to explain the importance of interoperability for blockchain systems. Interoperable blockchain architecture is a group of distinguishable blockchain systems, each representing a specific distributed data ledger, where multiple heterogeneous or homogeneous blockchains can execute atomic transactions and where data recorded in one blockchain ledger is available, verifiable, and referenced by another foreign transaction in a semantically

M. Abu Talib · S. Abbas · Q. Nasir · F. Dakalbab · T. Mokhamed · K. Hassan (✉) · K. Senjab
University of Sharjah, Sharjah, United Arab Emirates
e-mail: mtalib@sharjah.ac.ae; sabbas@sharjah.ac.ae; nasir@sharjah.ac.ae; falbab@sharjah.ac.ae; u17106067@sharjah.ac.ae; u16106525@sharjah.ac.ae; u16200379@sharjah.ac.ae

M. H. ur Rehman et al. (eds.), *Trust Models for Next-Generation Blockchain Ecosystems*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-75107-4_6

compatible nature. In another definition of interoperable blockchain proposed by P. Lafourcade and M. Lombard-Platet [47], interoperability is the ability of two blockchains to work together by executing and validating transactions, sending assets from one participant to another in different chains, or invoking and executing smart contracts. Blockchain interoperability solutions can be categorized into four main types [64]:

- **Sidechain or relay chain solution** A separate blockchain system connected to the main blockchain (mainchain/parent chain) which has the main functionality of verifying and reading data for another blockchain. The blockchains are interconnected through a two-way peg mechanism. The two-way peg mechanism allows digital assets to transfer from a mainchain to a sidechain and vice versa at a fixed or otherwise deterministic exchange rate [2]. Another term, *Federated Peg*, was introduced by the authors of "Enabling Blockchain Innovations with Pegged Sidechains" [2], referring to a mechanism that uses functionaries to validate and sign the data blocks by Block signers and the pegs by Watchmen. This network acquires the property of being secure. However, sidechains are limited to homogeneous blockchain systems [64].
- **Blockchain router solution** This technique involves some blockchain entities or nodes to serve as routers for transmitting transactions across various blockchain networks [47]. The design concept of this solution is derived from the routing architecture of the Internet.
- **Smart contracts** This approach uses a smart contract or a set of smart contracts to create a kind of inter-communication protocol among multiple different blockchain networks [7, 20, 48]. This method provides interoperable and secure data sharing and access control [20].
- **Industrial solutions** This category uses a collection of trusted validators to validate and confirm transactions. Many modern industrial projects use validators to ensure and guarantee the state of the node and its integrity [64].

Our Contributions:
The primary purpose of this study is to provide a systematic review and conduct a comprehensive study of existing solutions for the creation of interoperable blockchains. Also, this review examines the methodologies used in the available solutions to reach blockchain interoperability and compares them in terms of several factors, including performance (throughput, average block confirmation time), the method used to achieve interoperability, strengths, challenges, and possible future directions. We expect that this study will allow researchers to have a clearer understanding of the various existing interoperable blockchain mechanisms and to situate them about the recent research carried out on this topic. Furthermore, we will present the projects currently implementing these protocols.

The rest of the chapter is organized as follows: In Sect. 6.2, we discuss the literature review. Section 6.3 illustrates the methodology used to conduct the systematic literature review (SLR), which consists of planning and conducting the research. In Sect. 6.4, we discuss the results and analysis. Finally, Sects. 6.5 and 6.6 present the conclusion, the limitations of this review, and directions for future works.

6.2 Literature Review

Currently, blockchain technology operates as a series of stand-alone networks without the ability to communicate with other blockchain networks, exchange external data, or autonomously perform transactions. Thus, the interoperability of blockchain is one of the most critical and challenging aspects of blockchain technologies. Motivated by this challenging aspect, we studied reviews that discussed innovation in interoperable blockchain. For example, Ilham et al. [64] provided an overall study of inter-blockchain communication. They reviewed all available inter-blockchain communication solutions and classified the available solutions into four main types: blockchain routers, sidechains, industrial solutions, and smart contracts. Furthermore, they provided a comparison where they discussed the weaknesses and strengths of each type. In another example, Rafael et al. [6] presented an extensive survey on all aspects of blockchain interoperability. They introduced the area of interoperability research, delved into the background of the domain, and defined and discussed various architectures and standards. Moreover, they presented existing solutions in three main categories: cryptocurrency-directed approaches, blockchain engines, and blockchain connectors. Additionally, they presented the advantages of a multiple-blockchain approach through a case study. They showed the various challenges related to the development of interoperable blockchain.

On the other hand, Stefan et al. [69] mentioned the need for blockchain interoperability and its benefits in improving the paradigm from current blockchain technology to an open system that allows different blockchain systems to communicate with one another. They review the aspects of cross-blockchain token transfers and smart contract invocation and interaction. Furthermore, Liping Deng et al. [21] presented a paper that outlines the importance of cross-blockchain and details multi-signature wallet concepts. Furthermore, they concentrate on the study of the latest relevant cross-chain technologies and active ventures. Peter Robinson [65] raised a review that looks at cross-chain communication usage scenarios, and specifically at atomic swaps, values transfers, and reading and writing state pinning. Additionally, he presents key cross-chain classification techniques, which include locked hash time contracts, block header relaying, relay chains and threshold structures, and communication chains. Moreover, Babu et al. [59] presented a paper that classifies digital crypto-assets for interoperable deployment. The authors categorized crypto-assets based on their features and purpose and provided an interoperability scenario for specified crypto-asset classes. In another paper, Richard Barnes [5] surveyed existing architecture for interoperability and smart contract language. Barnes also described variables that impact tokenized asset portability. Finally, he suggested a maturity model for portability that can be used to determine the existing state of technology and business infrastructure support.

In this systematic literature review, we have provided analyses that are different from those in the surveys mentioned above. Our study differs in various aspects from the related work in that we provide a full review of all proposed methods and solutions related to inter-blockchain communication and a precision comparison of each solution with its strengths and limitations. We also discuss the performance and evaluation metrics applied to each method and the applications and contexts of use. Finally, we present future directions related to inter-blockchain communication.

6.3 Methodology

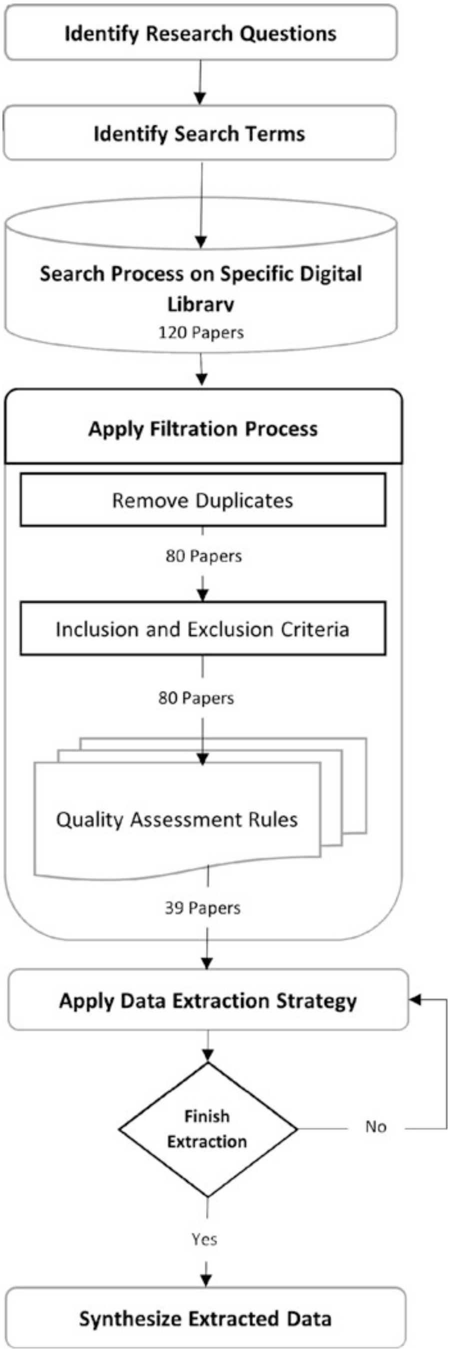
We conducted a systematic literature review (SLR) based on Kitchenham and Charters' methodology [44]. This method consists of three main stages: planning, conduction, and reporting. In each stage, there are several processes and steps involved. In the planning stage, the following six steps are included: (I) identify your research questions based on the objectives you are planning to achieve in your study, (II) define your search strategy, (III) create criteria for your selection, (IV) set up your quality assessment rules, (V) define your techniques for data extraction, and (VI) specify how you will synthesize extracted data. Furthermore, the following subsection will include a detailed description of the steps mentioned. Figure 6.1 illustrates the search methodology applied in this research.

6.3.1 Research Questions

In this study, our main objective is to review blockchain interoperability research area. The following research questions are raised to achieve this objective (Sect. 6.2):

3.1.1 RQ1: What is the methodology used to create inter-blockchain communication?
RQ1 aims to identify the methods and solutions applied by researchers to achieve blockchain interoperability.
3.1.2 RQ2: What are the strength and limitations of interoperable blockchain?
RQ2 aims at presenting the advantages and disadvantages of the methodologies proposed by researchers.
3.1.3 RQ3: What are the performance and evaluation metrics?
RQ3 aims to show the performance metrics used in evaluating the methods and solutions.
3.1.4 RQ4: What is the application and context of usage of inter-blockchain?
RQ4 is concerned with the application and context of the blockchain interoperable blockchain solution.
3.1.5 RQ5: What are the future directions of inter-blockchain?
RQ5 aims to present future direction for blockchain interoperability.

Fig. 6.1 Applied research methodologies


Identify Research Questions
Identify Search Terms
Search Process on Specific Digital Librarv
120 Papers
Apply Filtration Process
Remove Duplicates
80 Papers
Inclusion and Exclusion Criteria
80 Papers
Quality Assessment Rules
39 Papers
Apply Data Extraction Strategy
Finish Extraction
No
Yes
Synthesize Extracted Data

6.3.2 Search Strategy

The process for choosing the search term was as follows:

3.1.1. The research questions identified the main search terms.
3.1.2. Additional search terms were derived with the same meanings of the main search terms such as blockchain interoperability, cross-blockchain communication, multi-blockchain, and heterogeneous blockchain communication.
3.1.3. The search findings are constrained by Boolean operators (ANDs and ORs).
3.1.4. The search words used in this study refer to interoperable blockchain communications.

The digital libraries (journals and conference papers) used are listed as follows: GoogleScholar, Elsevier, Springer, the Association for Computing Machinery (ACM) Digital Library, and the Institute of Electrical and Electronics Engineers (IEEE) Library. Moreover, we found that the Cornell University journal includes several research papers that met our selection criteria.

Following our inclusion/exclusion criteria, we collected 39 scientific papers and 37 projects. The scientific papers included 24 conference papers and 15 journal papers.

6.3.3 Study Selection

Our initial search produced a collection of 90 scientific papers based on our search terms. Moving on, we filtered the results to verify that we included papers related to our subject. In our scheduled daily meetings, the filtration mechanism was addressed by the coauthors. The following table explains the filtration and selection processes:

Stage 1: Delete all duplicated papers from various digital collections.
Stage 2: Eliminate irrelevant papers by applying inclusion and exclusion criteria.
Stage 3: Remove review and survey papers from the collection.
Stage 4: Apply quality assessment rules that allow only qualified papers to be included.
Stage 5: Search for more papers from the sources mentioned in selected papers and repeat the processes for the newly added papers.

Table 6.1 addresses the inclusion and exclusion criteria that were applied in this study to provide the best possible answers for the proposed research questions. Finally, 39 papers were selected after the filtration stages.

6.3.4 Quality Assessment Rules (QARs)

This is the final step in determining the finalized list of papers to be included in the SLR. This is an important stage that aims to determine the quality of the collected research papers. Thus, 10 QARs are determined and marks are given to each paper

Table 6.1 Inclusion and exclusion criteria

Inclusion criteria	Exclusion criteria
Include only journals and conference papers	Exclude papers with no clear publication information
Include inter-blockchain communication solutions	Exclude non-refereed articles
Include studies that discuss solutions for communication in heterogenous blockchain	Exclude articles that include blockchain and it is not related to interoperability
	Exclude all digital resources, which do not discuss inter-blockchain

out of a total value of 10. Scores for each QAR are applied as follows: "fully answered" = 1, "above average" = 0.75, "average" = 0.5, "below average" = 0.25, and "not answered" = 0. The summation of the marks achieved for the 10 QARs is the paper's total ranking. Finally, we chose to retain only papers assigned a score of 5 or higher; otherwise, we excluded the paper from the SLR collection.

QAR1: Are the study objectives recognized?
QAR2: Are inter-blockchain backgrounds well defined?
QAR3: Are the specific context and usage of blockchain clearly defined?
QAR4: Are the strengths of the proposed methods well explained?
QAR5: Are the limitations of the proposed methods well explained?
QAR6: Are the methods well designed and justifiable?
QAR7: Are the evaluation metrics reported?
QAR8: Are the evaluation metrics compared to those of other methods?
QAR9: Are the evaluation metrics of the proposed methods suitable?
QAR10: Overall, does the study enrich the academic community or industry?

6.3.5 Data Extraction Strategy

In this stage, the final list of papers was analyzed to extract the necessary information to answer the set of research questions. The information extracted from each paper included information such as the authors, year of publication, the title of the paper, type of paper (whether it is from a conference or a journal), methodology applied for blockchain interoperability communication, and this methodology's strengths and limitations, contexts in which it can be applied, and future directions. It is important to note that not all papers collected were able to answer all the research questions.

6.3.6 *Synthesis of Extracted Data*

In this stage, we employed numerous processes to gather evidence to answer the research questions to synthesize the information obtained from selected papers. Furthermore, we utilized the narrative synthesis method to answer all research questions. Narrative synthesis refers to the method used to tabulate and visualize the findings of the research questions through pie charts, bar charts, and diagrams.

6.4 Results and Analysis

We analyze the findings of this study in this segment. This subsection describes the selected scientific papers and projects collected to answer the research questions stated above. In the following five parts, the findings of each research question are explored in detail. A total of 39 scientific papers and 37 projects that carried out inter-blockchain communication were selected. Furthermore, according to Figs. 6.4 and 6.5, the collected scientific papers were published between 2016 and 2020, while the projects had taken place between 2015 and 2020. As mentioned above, a quality assessment rule criterion was applied and the scores of the selected papers and projects are shown in Tables 6.12 and 6.13. The list is presented in Appendix A, Table 6.14. Furthermore, the full table of the quality assessment rule is in Table 6.15 in Appendix A.

6.4.1 *Design and Implementation Methods*

In this section, we address RQ1, which aims to find the methods and solutions raised in scientific papers for heterogeneous blockchain interoperability.

6.4.1.1 Scientific Papers

Depending on the type of interoperable solution used, a portion of platforms and protocols has been developed. The inter-blockchain protocol consists of the rules programmed to define intercommunication policies between blockchains. A blockchain platform is a group of interoperable blockchain technologies that are used as a base to create communication with other blockchain networks. In this part, we discuss the types of inter-blockchain proposed by selected scientific papers. Figure 6.2 shows that the proportion of inter-blockchain platforms and protocols studied in the selected articles was approximately equal, recorded at 46% and 39%, respectively. However, for the remaining 15%, we could not define the solution type.

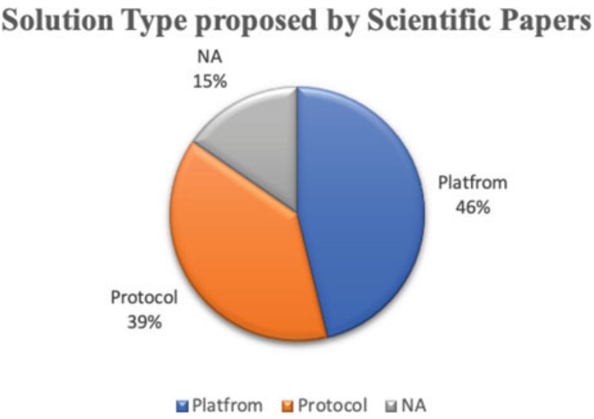


Fig. 6.2 Solution type in scientific papers

Table 6.2 Methodologies proposed by scientific papers

Inter-blockchain Solution	Description	Paper ID	Freq.
Router	Some blockchain entities or nodes serve as routers to send requests between different blockchain networks	A1, A2, A3,A4, A9, A21, A30	7
Sidechain	The main blockchain is linked to the independent blockchains (sidechains) The asset may be transferred using a two-way peg process	A8, A10, A17, A19, A20, A23, A33, A34, A37	9
Smart contract	To build interoperable protocols between independent blockchain networks, a smart contract or a series of smart contracts is used	A5, A6, A7, A11, A12, A25, A29, A35, A39	9
Atomic cross-chain swap	This approach utilizes the following basic mechanisms: • Multi-signatures • Hash-locks • Time-locks • Basic scripting	A15, A18, A22, A24, A28, A31, A32, A36	8
Multi-tokens Proof of Stake (MPoS) consensus protocols	MPoS modified version of PoS It supports the staking mechanism with multiple tokens in a cross-chain ecosystem	A14	1

As shown in Table 6.2, we identified five techniques that had been applied by researchers in the development of inter-blockchain communication.

In this review, the most frequent approaches used to create inter-blockchain communication are the sidechain, smart contract, atomic cross-chain swap, and router methods.

Sidechains are emerging mechanisms that allow one chain to safely use tokens and digital assets on a different chain. A two-way peg, also known as a bridge, allows the transfer of assets.

Atomic swaps are based on **Hashed Time-Lock Contracts (HTLCs)**, which utilize the following basic mechanisms [72]:

- Multi-signatures: a signature-based condition where transactions must be signed by two or more entities, thereby confirming and accounting for multi-signature transactions by signing parties.
- Hash-locks: used on two blockchains for linking transactions. Both locks are designed with the same hash function and are programmed with the same hash, so the password unlocking one hash-lock releases the password used to open the hash-lock on the other chain.
- Time-locks: a time-based condition restricts a transaction from being returned after a particular amount of time has passed. The period can be proportional to the publishing time of the transaction on the blockchain, or it can be an absolute time.
- Basic scripting: the purpose of basic scripting is to ensure that a transaction is initiated only if multiple (or committed) conditions are met. For example, the conditions may include the expiration of the period defined by the time-lock and the provision of a specific signature, or the release of both passwords to unlock a hash-lock.

Blockchain router is another approach that can connect various network blockchains in the same manner as the Internet network. In the blockchain router network, a blockchain plays the function of a router that analyzes and transmits connection requests according to the communication protocol, retaining a dynamic communication layout of the blockchain network.

6.4.1.2 Projects

The Fig. 6.3 below shows the percentage of inter-blockchain projects developed in one of the two types: platform or protocol. The review reveals that, with a percentage of 57%, the most frequent inter-blockchain implementations are the platform type, while inter-blockchain protocols were used in 40% of the projects. For the remaining 3% of the projects, the solution type was not specified.

According to Table 6.3, we have identified eight inter-blockchain project solutions. Most of the projects use the sidechain structure and atomic cross-chain swap technique as a solution for blockchain interoperability. However, some projects developed their internal architecture to achieve blockchain interoperability.

The ICON [37] project aims to link numerous blockchain communities across its platforms. Nexus and ICON Republic are part of the ICON structure. A Nexus is a collection of separate independent blockchains that are connected through the ICON Republic.

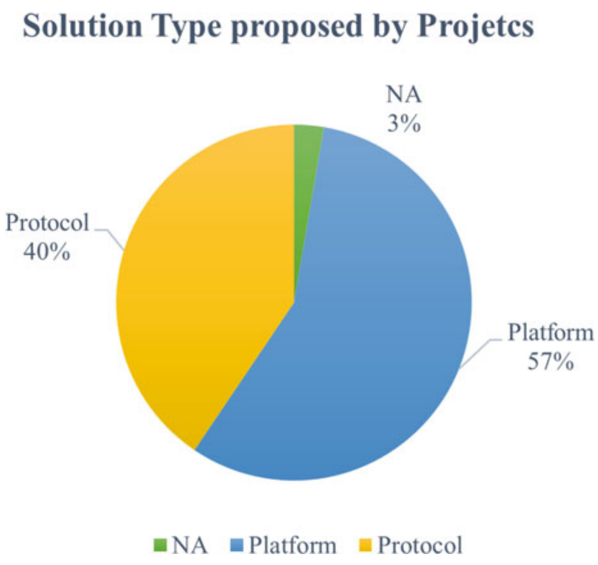


Fig. 6.3 Solution type by projects

The ArcBlock [51] project provides an environment for open blockchain applications to be developed and implemented. The project consists of three key components: Open Chain Access Protocol, Chain Adapter, and Blocklet. Open Chain Access Protocol provides an abstract layer for accessing different blockchain underlayers. Chain Adapter acts as a converter for switching blockchain underlayer protocols into the shared APIs specified in the Open Chain Access Layer protocol. Blocklet manages smart contracts, oracles, management of capital, and business logic off-chain. Blocklet interacts with blockchains through ArcBlock Open Chain Protocol.

The Cosmos [56] project is based on an Inter-Blockchain Communication (IBC) protocol. The architecture of the project consists of two major elements: center hubs and zones. The individual blockchains are zones, while the hubs allow for connections between various zones.

6.4.2 Strengths and Limitations

In this section, we address RQ2, which concerns the strengths and limitations of the solutions and methods raised in the previous subsection.

6.4.2.1 Scientific Papers

Table 6.4 presents the strengths and limitations of the collected research articles. Please note that not all research articles are included in this research question.

Table 6.3 Methodologies proposed by projects

Inter-blockchain solution	Description	Project ID	Freq.
Atomic cross-chain swap	This approach utilizes the following basic mechanisms: multi-signatures, hash-locks, time-locks, and basic scripting	P11, P13, P15, P20, P26, P27, P29, P30, P31, P35	10
Sidechain	The main blockchain is linked to independent blockchains (sidechains) and the asset may be transferred using a two-way peg process	P1, P2, P3, P4, P12, P18, P19, P21, P22, P23, P34	12
Router	Blockchain entities or nodes serve as routers, sending requests between different blockchain networks	P5	1
Atomic cross-chain swap	This approach utilizes the basic mechanisms: multi-signatures, hash-locks, time-locks, and basic scripting	A15, A18, A22, A24, A28, A31, A32, A36	8
Smart Contract	To build interoperable protocols between independent blockchain networks, a smart contract or a series of smart contracts is used	P16, P17, P24, P28, P32	5
Bringing	A specially programmed component enables the intercommunication between different blockchains and controls the validation of the transactions	P8, P10, P14	2
Nexus and ICON Republic	A Nexus is a collection of multiple blockchain networks ICON Republic portal is the link connecting independent blockchains (Nexus)	P9	1
Open Chain Access Protocol, Chain Adapter, Blocklet	Open Chain Access Protocol: enables different blockchains to interact with one another by assessing blockchains underlayers. Chain Adapter: converts the blockchain into the standardized APIs specified in the Open Chain Access Layer under layer protocols. Blocklet: manages different types of applications (smart contracts, oracles, management of capital and properties, and business logic off-chain)	P36	1
Zones, Hub, IBC protocol	Zones are different separate blockchains connected to Hub through IBC protocol	P7	1

Moreover, we found that most of the limitations related to inter-blockchain communication fall into the categories of security, privacy, lack of control, scalability, and lack of support for hybrid systems. This allows us to surmise that those limitations will open the gate for researchers to think of them as future directions to solve those challenges for inter-blockchain communication. On the other hand, most of the strengths found in the proposed methods involved achieving communication between different chains, allowing scalability for any interoperable blockchain network, and building securable, cheap, and fast solutions. Furthermore, scientific papers focused on the efficiency, feasibility, and flexibility of their solutions.

Table 6.4 Strengths and weaknesses of scientific papers

Strength	Paper ID	Limitation	Paper ID
Enable communication between blockchain systems	A1, A2, A6, A9, A10, A31	Privacy	A1, A3, A16, A20
No third party needed	A2, A4	Security	A2, A7, A16, A20, A24
High throughput	A5, A17	Scalability	A7, A26
Efficiency, feasibility, flexibility	A8, A11, A17, A21, A26, A27, A35, A37	Need an efficient consensus algorithm	A13, A16
Support heterogeneous consensus protocol	A5	Efficiency	A10
Atomic cross-chain compatibility	A7, A8, A11, A18	Throughput affected	A4
Scalability	A3, A13, A15, A17, A32	Lack of control	A19
Security	A10, A11, A13, A14, A17, A27, A31, A32, A33, A34, A37	Data store is in the provider's local database	A6, A10
Support multi-token users	A14	Does not support token transfer	A18
Low overhead, latency	A17	Exchange crypto assets in a seamless manner	A29, A18
Improve access control	A20	Lack of trust	A35
Ease of use and adoption	A27	Long lag for reconciliation	A36
No single point of failure	A34	Slow access time	A38
Improve data storage	A20	Restricted by latency and gas fee cost of the other blockchain platform	A39

6.4.2.2 Projects

In this section, we address the strengths and limitations of the collected projects that apply and achieve interoperability between blockchain networks. Table 6.5 presents each project's strengths and weaknesses in their ability to provide solutions and methods for inter-blockchain communication. According to Table 6.5, the strengths of most of the collected projects involved achieving interoperable connections and communications between two blockchain networks. Furthermore, most projects were able to achieve high scalability and security and to reduce the transaction costs of their solutions. On the other hand, some interesting limitations were revealed by the Plasma project, which addressed the issue of mass exit. This project demonstrated a situation where several users simultaneously attempted to release their Plasma chains, flooding the root chain and leading to network congestion.

Table 6.5 Strengths and limitations of projects collected

Strength	Project ID	Limitation	Project ID
Successful communicating and interoperable connection	P5, P7, P10, P11, P14, P15, P18, P23, P33, P34, P35	Compatibility	P37
High scalability	P1, P3, P7, P8, P9, P13, P14, P36	Relies on PoW, which is inefficient	P1
Secure	P1, P2, P14, P18, P24, P25, P31, P34	Network congestion (mass exit problem)	P3
Reduce transaction cost	P1, P4, P13, P31, P36	Geographic concentration	P4
Fast	P9, P13, P18, P19	Low TPS rate	P6
Easily adaptable	P8, P9, P14, P25	Gets slower with an increase in validators	P7
Confidential transaction	P2, P18, P27	Focused and designed for Korea	P9
Efficient and decentralized	P22, P26, P24	Scaling	P10
Trustless reutilization	P12, P24, P28	Not completely decentralized	P11
Compatible with scaling solution	P3, P8	Limited to digital assets	P12
Coin interchangeable with other units of the same coin	P6, P24	Uses a large amount of processing power	P14
Optimization	P12, P36	No rigorous technical documents	P19
High performance	P9, P36	Support interaction between a maximum of two blockchains networks	P23
Real-time value exchange	P29	Strict requirement	P27
Full control of assets	P31	No instant atomic swap	P28
Encryption algorithms supported	P34	The sender and receiver must know the private asset transfer key	P29
Cloud supported	P36	Does not support negotiation protocol	P31
Increasing bitcoin utilization	P1	…	…
Enable payment across different networks	P15	…	…
Asset issuance, flexible configuration	P2	…	…

Table 6.6 Performance metrics frequencies among scientific papers

Scientific papers	Quantity
Cost (computational cost and monetary cost)	6
I/O overhead	4
Process time	4
Transaction per second	3
Gas per transaction	3
Latency	3
Security risk	2
Scalability	2
Throughput	1
Speed	1
CPU utility	1
Query time	1

6.4.3 Performance and Evaluation Metrics

In this section, we present the performance and evaluation metrics applied to test the quality of the methods and solutions proposed by the researchers. Moreover, we determine which metrics have been applied most and discuss the metrics in greater detail.

6.4.3.1 Scientific Papers

In this section, we discuss the performance metrics most applied by the selected scientific papers. Table 6.6 presents the metrics and the frequency of each metric applied in the collected scientific papers. Please note that most of the papers applied more than one performance metric to evaluate the performance of their proposed solution. On the other hand, some papers did not apply any experiments to their solutions. Moreover, the most frequently used performance metrics applied were solution cost, I/O overhead, and processing time.

6.4.3.2 Projects

In this section, we present the performance metrics used by the projects that implement cross-blockchain. Table 6.7 lists the metrics applied and presents the frequency of their application in the collected projects. The most frequently applied performance metric was transaction per second, with 14 projects applying this performance to test the quality of their projects. Block time, cost, and block confirmation were also applied by several projects each. Please note that some of the projects applied several performance metrics.

Table 6.7 Performance metrics frequencies among projects

Projects	Quantity
Cost (computational cost and monetary cost)	6
Transaction per second	14
Block time	5
Cost	5
Block confirmation	4
Block size	2
Process time	1
Gas limit	1
Latency	1
Transaction confirmation	1
Transaction per block	1
Throughput	1

6.4.4 Application and Context of Usage of Inter-blockchain

In this section, we aim to identify the applications and use cases of inter-blockchain to address RQ4. Inter-blockchain is an evolving technology that enhances multiple structures in different areas and applications. In this regard, the applications that implement this technology must give independent blockchains the ability to connect and communicate with one another. Inter-blockchain can be used in a wide variety of applications. In this review, we defined multiple distinct applications in the selected papers and projects

6.4.4.1 Scientific Papers

Table 6.8 represents the list of inter-blockchain applications discussed in scientific papers. Moreover, the tables provide detailed information on the frequency with which a given inter-blockchain application is used in the selected papers.

As shown in Table 6.8, the review indicates that healthcare and finance and payment are the applications that most often implement inter-blockchain in the selected documents, with a percentage of 15% and 17%, respectively. However, it is obvious from the review that inter-blockchain can be applied in a variety of fields.

6.4.4.2 Projects

In this subsection, we discuss the applications used in the selected projects integrating blockchain interoperability. In Table 6.9, it can be seen that inter-blockchain communication was frequently integrated in decentralized exchange and finance and payment applications, with proportions of 12% and 21%, respectively. Business

Table 6.8 Inter-blockchain applications among scientific papers

Application	Paper ID	Freq.	Percentage
Arbitrary blockchain system	A2	1	2%
Asset transfer application	A5, A15	2	4%
Can be applied in many fields	A1, A3, A10, A11, A16, A24, A28, A29, A31, A32, A35	11	23%
Chain communication	A14, A17	2	4%
Cloud computing	A15	1	2%
Decentralized exchange	A7, A21, A34	3	6%
Finance and payment applications	A5, A8, A12, A13, A15, A26, A27, A34	8	17%
Gaming	A19, A39	2	4%
Healthcare	A6, A20, A22, A23, A33, A36, A37	7	15%
N/A	A4, A9, A38	3	3%
Retail services	A5	1	2%
Security systems	A18	1	2%
Smart contract applications	A25	1	2%
Storage services	A34	1	2%
Supply chain system	A5, A27, A30	3	6%

and supply chain systems are also applications that quite often implement inter-blockchain communication, with a percentage of 8%.

6.4.5 *Future Direction of Inter-blockchain*

In this section, we present future directions for innovation in inter-blockchain communication. Generally, it seems clear that more blockchain technology will be adopted in the ecosystems over time. There is currently a lack of interoperable and scalable solutions available to develop decentralized applications. Furthermore, there is a continuing gap between theoretical and practical applications, since much of the work currently underway is mostly conceptual. Recent developments in this field make interoperability a fact that needs to be addressed.

6.4.5.1 Scientific Papers

Table 6.10 shows the future directions for blockchain interoperability discussed in the selected scientific papers. As shown in Table 6.10, most of the researchers' future directions involve improving the security and privacy of their proposed solutions. Moreover, many researchers in the field plan to verify connections with formal methods, implement different network topologies, and improve efficiency and performance.

Table 6.9 Inter-blockchain applications among projects

Application	Project ID	Freq.	Percentage
Artificial intelligence systems	P35, P36	2	2%
Asset transfer applications	P13, P16, P18, P25, P33	6	5%
Business applications	P9, P10, P11, P13, P14, P32, P33, P36	9	8%
Can be applied in many fields	P11	1	1%
Chain communication	P8, P14, P21, P23, P25	5	4%
Cloud computing	P36	1	1%
Cyber-physical systems	P36	1	1%
Decentralized asset trading	P11, P17, P25, P30, P35	5	4%
Decentralized exchange	P1, P4, P11, P13, P14, P20, P23, P25, P26, P29, P31, P34, P35	14	12%
Development of commercial applications	P13, P19, P28, P30, P36	5	4%
Finance and payment applications	P1, P3, P4, P5, P6, P7, P8, P9, P11, P13, P14, P15, P16, P18, P21, P24, P26, P28, P29, P30, P32, P33, P35, P36, P37	25	21%
Gaming	P1, P4, P8, P14, P19, P21, P36	7	6%
Government organizations	P9, P36	2	2%
Healthcare	P5, P9, P25	3	3%
Internet of Things	P8, P11, P14, P15, P32	5	4%
N/A	P2, P22, P27	3	3%
Retail services	P1, P15	2	2%
Security systems	P9	1	1%
Smart contract applications	P8, P24, P32, P34	4	3%
Storage services	P8, P12, P16, P35	4	3%
Supply chain system	P1, P5, P8, P11, P14, P15, P21, P29, P32, P33	10	8%
Voting Systems	P1, P4, P26	3	3%

6.4.5.2 Projects

In this part, we present the future directions of the projects that currently implement cross-blockchain communication. Table 6.11 shows the future directions discussed in the project documentation. Various projects are planning to form more strategic partnerships and providers, to develop and consider more protocols, to generalize recursive SNARKs/STARKs to boost the security of the project, and to upgrade the consensus engine and smart contract system.

Table 6.10 Future direction of scientific papers

Future Direction	Paper ID
Improve security and privacy	A2, A3, A5, A16, A18, A21, A31, A34
Verify connection with formal methods	A2, A4, A24, A37
Discuss different network topologies	A4, A27, A28, A36
Improve efficiency and performance	A10, A25, A30, A38
Design an efficient consensus algorithm or improve it	A3, A27, A32
Add access control	A2, A30, A31
Add cloud computing platform	A22, A36
Add encryption	A2
Improve adaptability, scalability	A6
Improve data storage	A10
Atomic swaps on other blockchains	A7
Transaction-related error handling	A21
Reduce cost and support token transfer	A25
Analysis of the behavior of dishonest trustee	A26
Analysis of the behavior of irrational observer	A26

Table 6.11 Future direction of projects

Future direction	Project ID
Generalize recursive SNARKs/STARKS to increase security	P3, P21
Form strategic partnerships and add more products and providers	P9, P14
Protocol consideration for development	P17, P36
Upgrade consensus engine and smart contract system	P19, P37
Enable trustless messaging	P8
Work on other Hyperledger projects	P15
Translate script operation onto discrete logarithm information	P6
Focus on the number of extensions	P8
Scaling to more bridged chains	P17
Increase scalability and usability and enable para threads	P8
Transaction and storage fees	P19
Release of public API facilitating automatic shifting	P20
Evolve more refined approaches for veto contrast	P27
Add more features and increase the number of governing members	P34
Implement chain adapter for bitcoin, Ethereum, and Hyperledger	P36
Support Windows Azure, Google Compute Engine, IBM Bluemix	P36

6.5 Limitation of This Review

This systematic literature review is restricted to journals, conference papers, and studies related to inter-blockchain. By applying our research strategy at the first stage of the review, we filtered out a significant number of research papers that were found to be irrelevant. This guaranteed that the selected research articles fulfilled the criteria of this review. However, we assume that this analysis may have been further improved by considering additional references. Our pool of data may have been constrained by our stringent quality assessment criteria, which included only relevant papers that could provide synthesized findings.

6.6 Conclusion

In this SLR, we analyzed and compared the methodologies used in the current solutions for achieving blockchain interoperability. We examined several factors such as performance metrics (throughput, transmission time, block confirmation time), the strengths and limitations of the proposed solutions, and potential future directions. Our conclusion is summarized as follows:

- RQ1 shows that approximately 46% implement platform-based inter-blockchain communication and 39% implement protocol-based solutions in scientific papers. Moreover, 5 approaches were identified and the most frequently used to create inter-blockchain communication in scientific papers are sidechain, smart contract, atomic cross-chain swap, and router methods. On the other hand, for projects collected, we found that with a percentage of 57%, the most frequent inter-blockchain implementations are implemented as a platform type, and with a percentage of 40%, inter-blockchain protocols were created. Adding more, most of the projects leverage from the sidechain structure and atomic cross-chain swap technique as a solution for blockchain interoperability.
- RQ2 found that most of the limitation of inter-blockchain communication falls into the category of security, privacy, lack of control, scalability, and not supporting hybrid systems, as well as most of the strengths that have been found in the proposed methods were achieving the communication between different chains, allowing scalability for any interoperable blockchain network, and building securable, cheap, and fast solutions. On the other hand, for projects, we found that they were able to achieve high scalability and security and reduce the transaction cost of their solutions.
- RQ3 discussed the most applied performance metrics in scientific papers and projects. We found that cost performance is the most applied metric in scientific papers, whereas transmission per second is applied the most by the projects.
- RQ4 mentioned the most applied application of inter-blockchain. The most applied context for scientific papers and projects is finance and payment applications. Adding more, most of the scientific paper's context can be applicable in many fields.
- RQ5 presented the future directions for the scientific papers and projects. We found that the most common future direction for scientific papers is improving the security and privacy of their proposed solutions. Alternatively, for projects,

we did not find a lot of similarities, but some of them were working with other network topologies such as Hyperledger (Tables 6.12 and 6.13).

As part of our future work, we intend to implement and develop some of the inter-blockchain solutions, and we plan to conduct some experiments to test and enhance some of the blockchain interoperability problems discussed in Sect. 6.4.2 and improve the performance of our solution. Moreover, providing a discussion of the technical background of our inter-blockchain approach will be a focus in our future work (Tables 6.14 and 6.15).

Acknowledgments For this SLR study, Dr. Manar Abu Talib and her coauthors wish to thank the University of Sharjah and OpenUAE Research and Development Group. We also thank our research assistants for their contribution to the selection, summary, and interpretation of the research articles for this SLR study.

Appendix

See Figs. 6.4 and 6.5 and Tables 6.12 to 6.15.

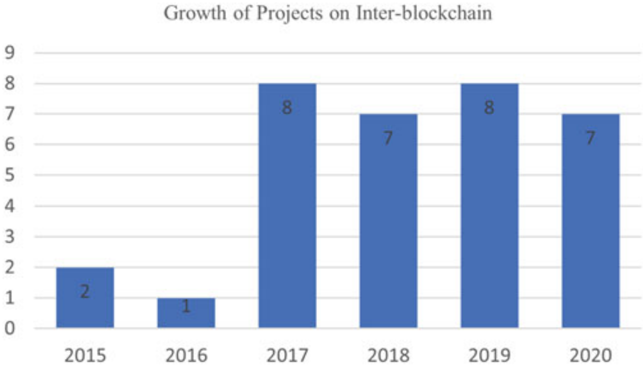


Fig. 6.4 Growth of projects based on years

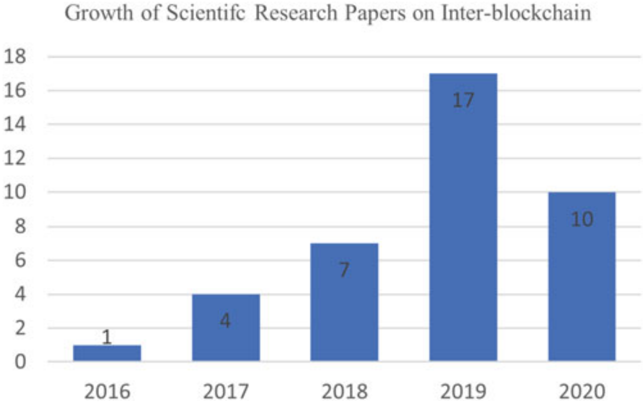


Fig. 6.5 Growth of scientific papers based on years

Table 6.12 QAR distribution among scientific papers

Result	No. of paper	Paper number
4	1	A39
5.5	1	A9
5.75	1	A12
6	2	A32, A36
6.75	1	A23
7	3	A34, A35, A22
7.25	1	A16
7.75	2	A27, A38
8.25	4	A3, A4, A19, A30
8.5	6	A1, A5, A6, A8, A18, A24, A31
8.75	2	A15, A29
9	4	A13, A17, A20, A25
9.25	1	A10
9.5	3	A7, A11, A28
9.75	4	A14, A21, A26, A37
10	2	A2, A33

Table 6.13 QAR distribution among projects

Result	No. of Projects	Paper number
4	1	P29
4.25	1	P37
4.5	1	P15
5	1	P22
5.75	1	P2
6.25	1	P23
6.5	1	P24
7	4	P4, P10, P13, P26
7.25	3	P20, P31, P35
7.5	1	P27
7.75	2	P21, P30
8.25	2	P18, P7
8.5	4	P9, P11, P19, P34
8.75	5	P3, P5, P6, P8, P33
9	4	P1, P14, P25, P32
9.25	4	P12, P16, P28, P36
9.5	1	P17

Table 6.14 Selected scientific papers and projects

ID	Title	Type	Year	Ref
A1	"Blockchain Router: A Cross-Chain Communication Protocol"	Journal	2017	[75]
A2	"A Multiple Blockchains Architecture on Inter-Blockchain Communication"	Conference	2018	[42]
A3	"InterChain: A Framework to Support Blockchain Interoperability"	Conference	2018	[23]
A4	"Inter-Blockchain Communication"	Conference	2017	[18]
A5	"Towards Scalable and Private Industrial Blockchains"	Conference	2018	[48]
A6	"Towards Secure Interoperability between Heterogeneous Blockchains using Smart Contracts"	Conference	2017	[20]
A7	"An Analysis of Atomic Swaps on and between Ethereum Blockchains using Smart Contracts"	Journal	2018	[7]
A8	"XCLAIM: Trustless, Interoperable, Cryptocurrency-Backed Assets"	Conference	2019	[84]
A9	"Toward an Interoperability Architecture for Blockchain Autonomous Systems"	Journal	2019	[34]
A10	"A Multi-blockchain Architecture Supporting Cross-Blockchain Communication"	Conference	2020	[79]
A11	"Towards a Novel Architecture for Enabling Interoperability among Multiple Blockchains"	Conference	2018	[41]
A12	"Inter-Bank Payment System on Enterprise Blockchain Platform"	Conference	2018	[76]
A13	"Multi-Blockchain Model For Central Bank Digital Currency"	Conference	2017	[71]
A14	"A New Consensus Protocol for Blockchain Interoperability Architecture"	Journal	2020	[57]
A15	"HyperService: Interoperability and Programmability Across Heterogeneous Blockchains"	Conference	2019	[50]
A16	"SoK: Communication Across Distributed Ledgers"	Journal	2019	[83]
A17	"An Electricity Cross-Chain Platform Based on Sidechain Relay"	Journal	2020	[77]
A18	"Atomic Cross-Chain Swaps: Development, Trajectory and Potential of Non-Monetary Digital Token Swap Facilities"	Journal	2019	[53]
A19	"Towards Blockchain Interoperability: Improving Video Games Data Exchange"	Conference	2019	[8]
A20	"Interoperability and Synchronization Management of Blockchain-Based Decentralized e-Health Systems"	Journal	2020	[8, 9]
A21	"Bifröst: a Modular Blockchain Interoperability API"	Conference	2019	[68]
A22	"Blockchain-Based Interoperable Electronic Health Record Sharing Framework"	Conference	2019	[16]
A23	"CEPS: A Cross-Blockchain based Electronic Health Records Privacy-Preserving Scheme"	Conference	2020	[15]
A24	"Reliable inter-blockchain communication framework for improving scalability"	Journal	2020	[49]
A25	"Towards Blockchain Interoperability"	Report	2020	[69]
A26	"Disincentivizing Double Spend Attacks Across Interoperable Blockchains"	Conference	2019	[66]

(continued)

Table 6.14 (continued)

ID	Title	Type	Year	Ref
A27	"Enabling Enterprise Blockchain Interoperability with Trusted Data Transfer (Industry Track)"	Conference	2019	[1]
A28	"Dextt: Deterministic Cross-Blockchain Token Transfers"	Journal	2019	[12]
A29	"Cross-chain interoperability among blockchain-based systems using transactions"	Journal	2020	[60]
A30	"Toward a Policy-based Blockchain Agnostic Framework"	Conference	2019	[67]
A31	"A Blueprint for Interoperable Blockchains"	Journal	2019	[24]
A32	"CVEM: A Cross-chain Value Exchange Mechanism"	Conference	2018	[82]
A33	"Enhanced Decentralized Management of Patient-Driven Interoperability Based on Blockchain"	Conference	2019	[43]
A34	"Strong Federations: An Interoperable Blockchain Solution to Centralized Third-Party Risks"	Journal	2016	[22]
A35	"A Workflow Interoperability Approach Based on Blockchain"	Conference	2020	[28]
A36	"Blockchain-Based Interoperable Electronic Health Record Sharing Framework"	Conference	2019	[16]
A37	"Blockchain-based Interoperable Healthcare using Zero-Knowledge Proofs and Proxy Re-Encryption"	Conference	2020	[70]
A38	"A Framework for Blockchain Interoperability and Runtime Selection"	Journal	2019	[29]
A39	"Demo Abstract: An Interoperable Avatar Framework Across Multiple Games and Blockchains"	Conference	2019	[14]
P1	"RSK"	Project	2015	[55]
P2	"Elements Alpha"	Project	2015	[25]
P3	"Plasma"	Project	2017	[63]
P4	"POA Network"	Project	2017	[78]
P5	"Anlink Network"	Project	2017	[3]
P6	"Mimblewimble"	Project	2016	[56]
P7	"Cosmos "	Project	2019	[46]
P8	"Polkadot"	Project	2020	[61]
P9	"ICON"	Project	2019	[37]
P10	"AION"	Project	2017	[32]
P11	"Wanchain"	Project	2017	[74]
P12	"Blocknet"	Project	2018	[35]
P13	"Interledger"	Project	2017	[72]
P14	"ARK"	Project	2019	[4]
P15	"Hyperledger Quilt"	Project	2019	[39]
P16	"Metronome"	Project	2018	[52]
P17	"Block Collider"	Project	2018	[10]
P18	"Liquid"	Project	2018	[54]
P19	"Loom network"	Project	–	[40]
P20	"Pantos"	Project	2018	[27]
P21	"Zendoo"	Project	2020	[36]
P22	"Testimonium "	Project	2020	[30]

(continued)

Table 6.14 (continued)

ID	Title	Type	Year	Ref
P23	"Peace Relay"	Project	–	[58]
P24	"BTC Relay "	Project	2018	[77]
P25	"Hyperledger Cactus"	Project	2020	[13]
P26	"chainX"	Project	2020	[17]
P27	"DeXTT"	Project	2019	[11]
P28	"Fusion"	Project	2017	[31]
P29	"Tokrex"	Project	–	[73]
P30	"Komodo"	Project	2019	[45]
P31	"COMIT"	Project	2020	[19]
P32	"chainlink"	Project	2019	[26]
P33	"HyperExchange"	Project	2018	[38]
P34	"PolyNetwork"	Project	2020	[62]
P35	"Ferrum Network"	Project	2019	[81]
P36	"ArcBlock"	Project	2017	[51]
P37	"GOST Protocol"	Project	–	[33]

Table 6.15 QAR results

Number	QAR 1	QAR 2	QAR 3	QAR 4	QAR 5	QAR 6	QAR 7	QAR 8	QAR 9	QAR 10	Total
A1	1	1	0.75	0.75	0.5	1	0.75	0.75	1	1	8.5
A2	1	1	1	1	1	1	1	1	1	1	10
A3	1	1	1	1	1	1	0.5	0.5	0.25	1	8.25
A4	1	1	0	0.75	1	1	1	0.75	0.75	1	8.25
A5	1	1	0.25	1	0.25	1	1	1	1	1	8.5
A6	1	1	1	1	1	1	0.5	0.75	0.25	1	8.5
A7	1	0.5	1	1	1	1	1	1	1	1	9.5
A8	1	0.75	0.5	1	0.25	1	1	1	1	1	8.5
A9	1	1	0.75	0.5	0.25	1	0	0	0	1	5.5
A10	1	1	0.5	1	1	1	1	0.75	1	1	9.25
A11	1	1	0.5	1	1	1	1	1	1	1	9.5
A12	1	1	1	0.75	0.25	0.75	0	0	0	1	5.75
A13	1	1	1	0.75	0.25	1	1	1	1	1	9
A14	1	1	1	1	0.75	1	1	1	1	1	9.75
A15	1	0.75	1	1	0.25	1	1	0.75	1	1	8.75
A16	1	0.75	0.25	0.25	1	0.75	0.25	1	1	1	7.25
A17	1	1	0.75	1	0.25	1	1	1	1	1	9
A18	1	1	1	1	1	1	0.25	1	0.25	1	8.5
A19	1	1	1	1	1	1	0.25	0.75	0.25	1	8.25
A20	1		1	1	1	1	1	1	1	1	9
A21	1	1	0.75	1	1	1	1	1	1	1	9.75
A22	1	1	1	0.25	1	0.25	0.25	0.5	0.75	1	7
A23	1	1	1	0	0	0.25	1	0.5	1	1	6.75
A24	1	0.75	0.5	0.75	1	0.5	1	1	1	1	8.5
A25	1	1	0.25	1	0.75	1	1	1	1	1	9
A26	1	1	0.75	1	1	1	1	1	1	1	9.75
A27	1	1	1	1	1	1	0	0.25	0.5	1	7.75
A28	1	1	0.5	1	1	1	1	1	1	1	9.5
A29	1	1	0.5	0.5	1	1	1	0.75	1	1	8.75
A30	1	1	1	0.75	1	1	0.25	1	0.25	1	8.25
A31	1	1	1	1	1	1	0	0.75	0.75	1	8.5
A32	1	1	0.25	1	0.25	1	0	0.25	0.25	1	6
A33	1	1	1	1	1	1	1	1	1	1	10
A34	1	1	1	1	0.25	1	0.25	0.25	0.25	1	7
A35	1	1	0.25	1	1	1	0.25	0.25	0.35	1	7.1
A36	1	1	1	0.5	0.5	1	0	0	0	1	6
A37	1	1	1	1	0.75	1	1	1	1	1	9.75
A38	1	1	0.25	1	1	1	0.5	0.5	0.5	1	7.75
A39	0.5	0.25	1	0.25	0.25	0.75	0	0	0	1	4

(continued)

Table 6.15 (continued)

Number	QAR 1	QAR 2	QAR 3	QAR 4	QAR 5	QAR 6	QAR 7	QAR 8	QAR 9	QAR 10	Total
P1	1	1	1	1	0	1	1	1	1	1	9
P2	1	0.5	1	1	0.25	1	0	0	0	1	5.75
P3	1	1	1	1	1	1	0.75	0.25	0.75	1	8.75
P4	1	0.75	1	0.75	0.75	0.25	1	0.25	0.25	1	7
P5	1	1	1	1	1	1	0.75	0	1	1	8.75
P6	1	1	1	1	1	0.75	1	0	1	1	8.75
P7	1	1	1	1	0.5	1	1	0	0.75	1	8.25
P8	1	1	1	1	0	1	1	0.75	1	1	8.75
P9	1	1	1	1	0.5	1	1	0.25	0.75	1	8.5
P10	1	1	0.5	1	0	1	0.75	0	0.75	1	7
P11	1	1	1	1	0.75	1	0.75	0.25	0.75	1	8.5
P12	1	1	1	1	1	1	1	0.25	1	1	9.25
P13	1	1	1	1	0.25	1	0.25	0.25	0.25	1	7
P14	1	0.5	1	1	0.75	1	1	0.75	1	1	9
P15	0.5	0	1	1	0	1	0	0	0	1	4.5
P16	1	1	1	1	0.25	1	1	1	1	1	9.25
P17	1	1	1	1	1	1	1	0.5	1	1	9.5
P18	1	1	1	1	0.5	1	1	0	0.75	1	8.25
P19	1	0.5	1	1	1	1	1	0	1	1	8.5
P20	1	1	0.75	1	0.75	1	0.25	0	0.5	1	7.25
P21	1	1	0.75	1	1	1	0.25	0.25	0.5	1	7.75
P22	1	0.5	0	0.5	0.5	1	0.25	0	0.25	1	5
P23	1	0.5	1	0.75	0.75	0.75	0.25	0	0.25	1	6.25
P24	1	1	0.75	1	0	1	0.25	0	0.5	1	6.5
P25	1	1	1	1	1	1	1	0	1	1	9
P26	1	1	0.75	1	0.25	1	0.25	0	0.75	1	7
P27	1	1	0	1	1	1	0.75	0	0.75	1	7.5
P28	1	1	1	1	1	1	1	0.25	1	1	9.25
P29	1	0.25	0.25	0.5	0.5	0.25	0.25	0	0	1	4
P30	1	0.5	0.75	0.75	0.75	1	0.75	0.5	0.75	1	7.75
P31	1	1	1	1	0.5	1	0.25	0	0.5	1	7.25
P32	1	1	1	1	0	1	1	1	1	1	9
P33	1	0.75	1	1	0	1	1	1	1	1	8.75
P34	1	1	1	1	0.25	1	1	0.25	1	1	8.5
P35	1	1	1	1	0.25	0.75	0.5	0	0.75	1	7.25
P36	1	0.75	1	1	0.75	1	1	0.75	1	1	9.25
P37	1	0.25	0.25	0.5	0.25	0.5	0.25	0	0.25	1	4.25

References

1. E. Abebe, D. Behl, C. Govindarajan, Y. Hu, D. Karunamoorthy, P. Novotny, V. Pandit, V. Ramakrishna, C. Vecchiola, Enabling enterprise blockchain interoperability with trusted data transfer (industry track), in *Proceedings of the 20th International Middleware Conference Industrial Track* (2019), pp. 29-35
2. B. Adam, C. Matt, D. Luke, F. Mark, M. Gregory, M. Andrew, P. Andrew, T. Jorge, W. Pieter, *Enabling Blockchain Innovations with Pegged Sidechains* (2014). https://blockstream.com/sidechains.pdf. http://www
3. Anlink Blockchain Network Whitepaper v 1.0. Technical report (2017)
4. Ark.io | the simplest way to blockchain 0. https://ark.io/
5. R. Barnes, *Factors in the Portability of Tokenized Assets on Distributed Ledgers* (2020), pp. 1–9
6. R. Belchior, A. Vasconcelos, S. Guerreiro, M. Correia, *A Survey on Blockchain Interoperability: Past, Present, and Future Trends* (2020)
7. P. Bennik, V.L. Gijtenberg, *An Analysis of Atomic Swaps on and between Ethereum Blockchains Using Smart Contracts* (2018)
8. L. Besancon, C.F. Da Silva, P. Ghodous, Towards blockchain interoperability: Improving video games data exchange, in *Proceedings of the IEEE International Conference on Blockchain and Cryptocurrency (ICBC 2019)* (2019), pp. 81–85
9. S. Biswas, K. Sharif, F. Li, Z. Latif, S.S. Kanhere, S.P. Mohanty, Interoperability and synchronization management of blockchain-based decentralized e-health systems. IEEE Trans. Eng. Manag. **67**(4), 1363–1376 (2020)
10. Block collider 0. https://www.blockcollider.org/whitepaper
11. M. Borkowski, M. Sigwart, P. Frauenthaler, T. Hukkinen, S. Schulte, Dextt: Deterministic cross-blockchain token transfers. IEEE Access **7**, 111030–111042 (2019)
12. M. Borkowski, M. Sigwart, P. Frauenthaler, T. Hukkinen, S. Schulte, Dextt: Deterministic cross-blockchain token transfers. IEEE Access **7**, 111030–111042 (2019)
13. Cactus Whitepaper MD. https://github.com/hyperledger/cactus/blob/master/whitepaper/whitepaper.md
14. W. Cai, X. Wu, Demo abstract: an interoperable avatar framework across multiple games and blockchains, in *Proceedings of the IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2019 (INFOCOM 2019)* (2019), pp. 967–968
15. S. Cao, J. Wang, X. Du, X. Zhang, X. Qin, CEPS: A cross-blockchain based electronic health records privacy-preserving scheme, in *Proceedings of the IEEE International Conference on Communications (ICC 2020-2020)* (IEEE, New York, 2020)
16. G. Carter, H. Shahriar, S. Sneha, Blockchain-based interoperable electronic health record sharing framework. Proc. Int. Comput. Softw. Appl. Conf. **2**, 452–457 (2019)
17. Chainx Whitepaper. Technical report (2019)
18. Z. Chen, Z. Yu, Z. Duan, K. Hu, Inter-blockchain Communication **(Cst)**, 448–454 (2017)
19. Comit network 0. https://comit.network/
20. G.G. Dagher, C.L. Adhikari, T. Enderson, Towards secure interoperability between heterogeneous blockchains using smart contracts, in *Future Technologies Conference (FTC) 2017* (2017), pp. 73–81
21. L. Deng, H. Chen, J. Zeng, L.J. Zhang, *Research on Cross-chain Technology Based on Sidechain and Hash-locking*, vol. 10973. LNCS (Springer, Berlin, 2018)
22. J. Dilley, A. Poelstra, J. Wilkins, M. Piekarska, B. Gorlick, M. Friedenbach, *Strong Federations: An Interoperable Blockchain Solution to Centralized Third-party Risks* (2016)
23. D. Ding, T. Duan, L. Jia, K. Li, Z. Li, Y. Sun, Interchain: a framework to support blockchain interoperability, in *Second Asia-Pacific Workshop on Networking* (2018)

24. T.T.A. Dinh, A. Datta, B.C. Ooi, A blueprint for interoperable blockchains, in *Proceedings of the 28th Modern Artificial Intelligence and Cognitive Science Conference (MAICS 2017)* (2017), pp. 189–190
25. Elements | elementsproject.org 0. https://elementsproject.org/
26. S. Ellis, A. Juels, S. Nazarov, Chainlink a Decentralized Oracle Network. Technical report (2017)
27. Pantos Team, Pantos Vision Paper. (2020). https://pantos.io/pdf/pantos-visionpaper.pdf
28. Y. Fang, X. Tang, M. Pan, Y. Yu, *A Workflow Interoperability Approach Based on Blockchain*, vol. 11894. LNCS (2020)
29. P. Frauenthaler, M. Borkowski, S. Schulte, *A Framework for Blockchain Interoperability and Runtime Selection (December 2017)* (2019)
30. P. Frauenthaler, M. Sigwart, C. Spanring, S. Schulte, *Testimonium: A Cost-efficient Blockchain Relay*, vol. 2 (2020)
31. Fusion whitepaper an inclusive cryptofinance platform based on blockchain fusion foundation *. Technical report (2017)
32. Get Started 0. https://docs-aion.theoan.com/docs
33. Gost protocol—cross chain transactions between Ethereum and GoChain—GoChain 0. https://gochain.io/gost-protocol-cross-chain-transactions/
34. T. Hardjono, A. Lipton, A. Pentland, Toward an interoperability architecture for blockchain autonomous systems. IEEE Trans. Eng. Manag. **67**(4), 1298–1309 (2019)
35. Home—blocknet documentation 0. https://docs.blocknet.co/
36. Horizen sidechain, 0. https://www.horizen.io/sidechain/
37. ICON, Icon, White Paper **2**, 1–38 (2017). http://docs.icon.foundation/ICON-Whitepaper-EN-Draft.pdf. https://icon.foundation/resources/whitepaper/ICON_Whitepaper_EN.pdf
38. Hyperexchange features | hyperexchange 0. http://docs.hx.cash/
39. Hyperledger quilt archives—hyperledger 0. https://www.hyperledger.org/category/hyperledger-quilt
40. Intro to loom network | loom SDK 0. https://loomx.io/developers/en/intro-to-loom.html
41. H. Jin, X. Dai, J. Xiao, Towards a novel architecture for enabling interoperability amongst multiple blockchains, in *Proceedings of the International Conference on Distributed Computing Systems* (2018), pp. 1203–1211
42. L. Kan, Y. Wei, A.H. Muhammad, W. Siyuan, G. Linchao, H. Kai, A multiple blockchains architecture on inter-blockchain communication, in *Proceedings of the 2018 IEEE 18th International Conference on Software Quality, Reliability, and Security Companion, QRS-C 2018* (2018), pp. 139–145
43. A.U. Khan, A. Shahid, F. Tariq, A. Ghaffar, A. Jamal, S. Abbas, N. Javaid, Enhanced decentralized management of patient-driven interoperability based on blockchain, in *Lecture Notes in Networks and Systems* (2020)
44. B. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering version 2.3. Engineering **45**(4ve), 1051 (2007)
45. Teams, Komodo, Komodo White Paper. (2018). https://komodoplatform.com/wp-content/uploads/2018/03/2018-03-12-Komodo-White-Paper-Full.pdf
46. J. Kwon, E. Buchman, Cosmos -A Network of Distributed Ledgers, p. 53. https://cosmos.network/cosmos-whitepaper.pdf
47. P. Lafourcade, M. Lombard-platet, *About Blockchain Interoperability* (2019), pp. 1–10
48. W. Li, A. Sforzin, S. Fedorov, G.O. Karame, *Towards Scalable and Private Industrial Blockchains* (2017)
49. K. Liu, Y. Ohsawa, Reliable *Inter-blockchain Communication Framework for Improving Scalability* (2020), pp. 1–11
50. Z. Liu, Y. Xiang, J. Shi, P. Gao, H. Wang, X. Xiao, B. Wen, Y.C. Hu, Hyperservice: interoperability and programmability across heterogeneous blockchains, in *Proceedings of the ACM Conference on Computer and Communications Security* (2019), pp. 549–566
51. F.C. Mao, Robert. Arcblock—technical white paper (2017)
52. Metronome: The built-to-last cryptocurrency 0. https://www.metronome.io/

53. M.H. Miraz, D.C. Donald, Atomic cross-chain swaps: Development, trajectory and potential of non-monetary digital token swap facilities. Ann. Emerg. Technol. Comput. **3**(1), 42–50 (2019)
54. J. Nick, A. Poelstra, G. Sanders, Liquid: A Bitcoin Sidechain. Technical report (2020)
55. (no title) 0. https://www.rsk.co/Whitepapers/RSK-White-Paper-Updated.pdf
56. (no title) 0. https://download.wpsoftware.net/bitcoin/wizardry/mimblewimble.txt.
57. Y. Pang, A new consensus protocol for blockchain interoperability architecture. IEEE Access **8**, 153719–153730 (2020)
58. Peace Relay 0. https://peacerelay.io/
59. B. Pillai, K. Biswas, V. Muthukkumarasamy, *Blockchain Interoperable Digital Objects*, vol. 11521. LNCS (Springer, Berlin, 2019)
60. B. Pillai, K. Biswas, V. Muthukkumarasamy, Cross-chain interoperability among blockchain-based systems using transactions. Knowl. Eng. Rev. **35**, 1–17 (2020)
61. G. Wood, Polkadot: Vision for a heterogeneous multi-chain framework. White paper, Draft 1, 1–21 (2017). https://github.com/ethereum/wiki/wiki/Chain-Fibers-Redux https://github.com/ethereum/wiki/wiki/Chain-Fibers-Redux%0A https://github.com/ethereum/wiki/wiki/Chain-Fibers-Redux%0A https://polkadot.network/PolkaDotPaper.pdf
62. Team, Poly, Polynetwork: An Interoperability Protocol for Heterogeneous Blockchains, (2020). https://www.poly.network/PolyNetwork-whitepaper.pdf
63. J. Poon, *Plasma: Scalable Autonomous Smart Contracts* (2017)
64. I.A. Qasse, M.A. Talib, Q. Nasir, *Inter Blockchain Communication: A Survey* (2019)
65. P. Robinson, *Consensus for Crosschain Communications* (2020), pp. 1–15
66. K. Sai, D. Tipper, *Disincentivizing Double Spend Attacks Across Interoperable Blockchains* (2019)
67. E. Scheid, B. Rodrigues, B. Stiller, Toward a policy-based blockchain agnostic framework, in *Proceedings of the 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM 2019)* (2019), pp. 609–613
68. E.J. Scheid, T. Hegnauer, B. Rodrigues, B. Stiller, Bifrost: a modular blockchain interoperability API, in *Proceedings of the Conference on Local Computer Networks, LCN, 2019-Octob* (2019), pp. 332–339
69. S. Schulte, M. Sigwart, P. Frauenthaler, M. Borkowski, *Towards Blockchain Interoperability* (2019)
70. B. Sharma, R. Halder, J. Singh, Blockchain-based interoperable healthcare using zero-knowledge proofs and proxy re-encryption, in *Proceedings of the 2020 International Conference on COMmunication Systems and NETworkS, COMSNETS 2020* (2020), pp. 1–6
71. H. Sun, H. Mao, X. Bai, Z. Chen, K. Hu, W. Yu, *Multi-blockchain Model for Central Bank Digital Currency* (2018)
72. S. Thomas, E. Schwartz, A Protocol for Interledger Payments. Interledger.Org, (2015), pp. 1–25. https://interledger.org/interledger.pdf
73. Tokrex Protocol 0. https://tokrex.org/
74. Wanchain—Introduction 0. https://www.explorewanchain.org/
75. H. Wang, Y. Cen, X. Li, *Blockchain Router: A Cross-chain Communication Protocol*, vol. Part F1282 (Association for Computing Machinery, New York, 2017), pp. 94–97
76. X. Wang, X. Xu, L. Feagan, S. Huang, L. Jiao, W. Zhao, *Inter-bank Payment System on Enterprise Blockchain Platform* (2018)
77. H. Wang, D. He, X. Wang, C. Xu, W. Qiu, Y, Yao, Q. Wang, An electricity cross-chain platform based on sidechain relay. J. Phys. Conf. Ser. **1631**(1), 012189 (2020)
78. Welcome to poa—poa 0. https://www.poa.network/
79. X. Xiao, Z. Yu, K. Xie, S. Guo, A. Xiong, Y. Yan, A multi-blockchain architecture supporting cross-blockchain communication, vol. 1253. CCIS (Springer, Deutschland, 2020), pp. 592–603
80. D. Yaga, P. Mell, N. Roby, K. Scarfone, Blockchain technology overview, in *National Institute of Standards and Technology* (2018)
81. N. Yaganeh, I.M. Friend, *Ferrum Network Whitepaper v0.5-work in Progress*. Technical report (2019)

82. S. Yang, W. Liu, H. Wang, X. Fu, W. Li, CVEM: A cross-chain value exchange mechanism, in *ACM International Conference Proceeding Series* (2018), pp. 80–85
83. A. Zamyatin, M. Al-Bassam, D. Zindros, E. Kokoris-Kogias, P. Moreno-Sanchez, A. Kiayias, W.J. Knottenbelt, Sok: communication across distributed ledgers, in *IACR Cryptology ePrint Archive*, vol. 2019/1128 (2019), pp. 1–23
84. A. Zamyatin, D. Harz, J. Lind, P. Panayiotou, A. Gervais, W. Knottenbelt, *Xclaim: Trustless, Interoperable, Cryptocurrency-backed Assets* (2019)

Chapter 7
Blockchain-based Incident Reporting System for Patient Safety and Quality in Healthcare

Dounia Marbouh, Mecit Can Emre Simsekler, Khaled Salah, Raja Jayaraman, and Samer Ellahham

7.1 Introduction

In the last two decades, healthcare organizations have devoted substantial efforts to improve patient safety because of the high rate of incidents harming thousands of people globally [1]. To prevent patient safety incidents and deliver high-quality services, hospitals and national health care services have started implementing incident reporting systems (IRSs). These systems can identify failures, risks, and threats and help investigate their root causes and their contributory factors to avoid them recurring in the future [2]. Although various tools and methods are used for risk identification [3], IRSs have a unique position with the opportunity of providing narratives for practitioners to learn from their mistakes and experiences [4], and for policy-makers to implement safer care policies [5]. However, despite their distinct features, earlier studies identified a range of obstacles that potentially prevent these reporting systems from contributing to the safety efforts [6, 7].

D. Marbouh · M. C. E. Simsekler (✉) · R. Jayaraman
Department of Industrial and Systems Engineering, Khalifa University of Science and Technology, Abu Dhabi, UAE
e-mail: 100050079@ku.ac.ae; emre.simsekler@ku.ac.ae; raja.jayaraman@ku.ac.ae

K. Salah
Department of Electrical Engineering and Computer Science, Khalifa University of Science and Technology, Abu Dhabi, UAE
e-mail: khaled.salah@ku.ac.ae

S. Ellahham
Heart and Vascular Institute and Quality and Patient Safety Institute, Cleveland Clinic Abu Dhabi, Abu Dhabi, UAE
e-mail: ellahas@clevelandclinicabudhabi.ae

M. H. ur Rehman et al. (eds.), *Trust Models for Next-Generation Blockchain Ecosystems*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-75107-4_7

Underreporting is among the significant concerns that create a reservoir of information [8]. Further, incomplete incident data, hindsight bias, unreliable classifications, confidentiality, blame culture, the time delay between reporting and investigation, fear of possible consequences, and lack of feedback are other frequent barriers encountered in incident reporting [9–11]. Further, IRSs have not been successfully connected across hospitals, even though each may identify different yet complementary patient safety issues to learn systematically [5]. As a result, current reporting systems are limited to collect comprehensive information on safety events, as incident data is scattered and fragmented across the system [12].

To address the limitations mentioned above in current reporting systems, blockchain technology may provide opportunities with its unique features, such as transparency, immutability, privacy, among others [13]. As an emerging technology, blockchain is expected to leverage the exchange of data among stakeholders in different domains and industries, including healthcare [13, 14]. This technology has the potential to place patients at the heart of the healthcare system and increase their privacy, security, and interoperability of their medical data. It could also make electronic health records (EHRs) more secure and efficient by transforming and providing a new model for health information exchange (HIE). Therefore, in this project, we aim to investigate the application of blockchain technology in IRS to identify potential improvement opportunities for patient safety and risk management. In addition to the opportunities, we will also address challenges in developing and implementing blockchain technology in this context.

In this particular patient safety context, healthcare providers in health services may benefit from the use of blockchain to report and investigate incidents in a secure, faster, and reliable manner. To explore this, we propose a blockchain-based solution that builds a reliable IRS. The fundamental contributions of this chapter are as follows: we first present a review of the current IRS as applied to patient safety in healthcare. Second, we develop a blockchain-based solution for reporting incidents using smart contracts. Then, we propose a framework along with algorithms that outline the mechanisms of the proposed solution and provide a detailed sequence diagram of the blockchain-based reporting system. Finally, we present a cost and security analysis of the proposed solution to show the feasibility of the implementation in healthcare.

The remainder of this chapter is organized as follows: the relevant literature on incident reporting and blockchain is presented in Sect. 7.2, followed by a description of the proposed blockchain-based solution in Sect. 7.3. Section 7.4 presents the implementation steps, along with testing scenarios. Discussion of the proposed approach, cost and security analysis, challenges, and future research directions are presented in Sect. 7.5. Finally, concluding remarks are presented in Sect. 7.6.

7.2 Related Work

In this section, we provide background information related to the IRSs. Further, we give an example of a sophisticated reporting system and explain the potential benefit of adopting blockchain technology.

7.2.1 Incident Reporting System (IRS)

Incident reporting systems (IRSs) are among the most important tools for risk management and improvement of patient safety worldwide. The primary purpose of incident reporting is to improve the safety of patients by learning from mistakes [15]. Many countries, such as UK, Sweden, Norway, Australia, have developed electronic reporting systems providing a platform for confidential and voluntary-based reporting. These systems are methods of reporting incident events or/and near misses enabling organizational improvements [12]. A patient safety incident is denoted as any accidental or unforeseen adverse event that may lead to the injury of one or more patients receiving a form of care [10]. They can also support hospitals in learning from adverse events and identifying patterns or tendencies related to patient safety [9]. Further, they present useful insights into why and how patients get hurt due to various contributory factors and hazards [6].

An ideal IRS is a multi-stage process; its main steps are described by Hewitt and colleagues in Fig. 7.1 [16].

In general, the information would be reported in the IRS at the detection stage. These incident reports should be then aggregated, examined, and analyzed at the analysis stage. The following stage is learning, in which the understanding of patterns and trends happens and where changing current practices takes place [10]. The final stage is the feedback that can happen at either the analysis or learning stage [17]. To create an effective reporting system, the World Health Organization (WHO) emphasizes the importance of both detection and feedback as key features [5, 18]. As discussed earlier, even though incident reporting is the vehicle to learn from past experiences in a patient safety context, underreporting is still the main challenge. Further, hindsight bias, partial data, the time lag between incident reporting and feedback, recall bias, and inaccurate classifications have also been identified as limitations in the literature [19]. Moreover, the following are significant concerns in IRSs that have been identified in recent studies [5]:

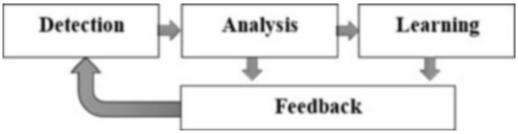


Fig. 7.1 Main stages in incident reporting systems

Change: The ability to measure changes and trends over time is limited.

Feedback: The reporter may fail to track the status of the event and timely feedback is not provided.

Complementary: IRSs identify different but complementary patient safety issues for several hospitals. Yet, these systems are not well connected.

Timely feedback: IRSs fail to communicate identified events with stakeholders promptly.

While individual hospitals may have their internal IRSs, called Local Reporting Management System (LMRS), some countries have also implemented national-level centralized reporting systems to share learning with all possible stakeholders. To understand the role of such aspects in IRSs, one example can be given from the British National Health Service (NHS): the National Reporting and Learning System (NRLS). The NRLS was established in 2003 as a platform that can aggregate national patient safety incidents and enable sharing and learning from these events to mitigate their recurrence. NHS Improvement is the organization responsible for managing the NRLS. This organization operates the NRLS and uses the incident data to provide guidance, insights, and advice for the national health services (NHS) organizations. At its inception, NRLS was fully voluntary. However, by 2010, reporting serious incidents and deaths due to an error became compulsory [20].

To foster reporting and encourage openness, reporting to NRLS is voluntary except for serious incidents [20]. As a result, NRLS data represent only a partial number of patient adverse events taking place in the NHS. In the case of patient adverse events that resulted in severe harm or death, they must be submitted to NRLS. They are separately assessed by NHS Improvement physicians to maximize the learning and act at a national level. Moreover, incidents reported to NRLS are also shared with several other organizations to identify the hazards and develop patient safety insights, solutions, and guidance. These organizations include Public Health England (PHE), NHS England, the Medicines and Healthcare products Regulatory Agency (MHRA), and Care Quality Commission (CQC) [21]. All these different entities and system layers show that many stakeholders and organizations are part of the system but are not connected. Therefore, if such a system is fragmented and if organizations' incident databases do not communicate with each other, this may hinder the collection and analysis of incidents in a systematic manner.

7.2.2 Blockchain Technology and Potential Benefits

A blockchain is a distributed ledger recording transactions between multiple entities in a verifiable and efficient manner. Satoshi Nakamoto, the anonymous person or persons who wrote the bitcoin white paper in 2008, described blockchain as a technology that combines data records or blocks in a chain [22]. This technology guarantees immutability thanks to the distributed network without the need for any third-party authority [23]. Blockchain entails having records that are ordered and placed in a block structure. These data blocks comprise a hash (unique identifier), the hash of the preceding block, and timestamped sets of the latest transactions. This design ensures that blocks are linked chronologically and they build what is called a blockchain [14].

Blockchain technology has the potential of creating trustless and decentralized transaction environments [24] and that is why the healthcare industry is a prime candidate for it [25, 26]. Blockchain can address key problems, such as public health management, claims validation, and supply chain management [27]. Healthcare current methods involve data to be stored with third parties, which is a major concern for many stakeholders including patients. Therefore, blockchain technology can potentially elevate such concern and afford transparency to data management processes while diminishing the risks of data misuse or mishandling [28]. Blockchain technology also uses its intrinsic characteristics to ensure immutability, transparency, privacy, security, and accuracy of data collection, unlike a conventional database system. Further, in a conventional centralized database, it is impossible to reward a user for reporting an incident. However, a blockchain platform can make the incentivizing process easy. Thanks to blockchain technology's cryptocurrency properties, reporters can get a reward for reporting an incident to encourage reporting and enhance the culture. Overall, distinct advantages are available in the blockchain-based incident reporting system compared to the traditional database system, such as the NRLS. The comparison and features are summarized in Table 7.1 [29, 30].

In the context of reporting patient safety incidents, the IRS has several gaps that can be summarized as follows: lack of information dissemination in real time, absence of incentives, lack of security and privacy, fragmentation of adverse-event data across different organizations, and the inability to have constructive feedback on whether the incident report had led to an action [7]. To respond to the current needs of these reporting systems, the blockchain technology features can be exploited. The blockchain technology's key aspects, such as time sequence, data security, and privacy, decentralization, transparency, incentives, and traceability, can be useful for ensuring better reporting systems.

Moreover, performing data analytics in healthcare is possible when blockchain technology is combined with artificial intelligence (AI) technologies such as machine learning [31] that recently gained attention in the patient safety context [32, 33]. By combining blockchain technology and AI, the process of data analysis would be conducted automatically without the intervention or need of statisticians. This feature would allow the national health services to check reports on any adverse events in real time. Blockchain technology and AI can also automate the analysis stage by having a node for data analytics in the network, which would oversee data cleaning and anonymization. The latter is made possible because raw data would be fed into this analytics node, and data integrity would be ensured by blockchain technology [34].

While blockchain technology has merits to provide a secure incident reporting and sharing platform, its full capabilities have not been explored for incident reporting platform yet. Therefore, we intend to build upon this research to leverage the

Table 7.1 Comparison between using a centralized database and a blockchain platform

Feature	Conventional centralized database	Blockchain platform
Authority	Controlled by a central authority (administrator)	Authority is shared among stakeholders and is decentralized
Data integrity	Data might be revised and changed	Data cannot be changed or altered, in other words, it is immutable
Data handling	Can support only these functions: read, update, create, and delete	Read and write are the only available options
Data privacy	Chances of malicious cyberattacks are high	Transaction data is stored in blocks using cryptography technology
Data provenance	Databases cannot ensure that data has not been altered, forged, reproduced, or stolen	Users can trace and verify the provenance of all the previous transactions by accessing any node in the network
Transparency	Data transparency is not as high as in BC	Transaction data is stored in a distributed network and stakeholders have access to the same data
Quality assurance	Administrators are required for data authentication	The provenance of data can be known using cryptography mechanisms
Fault tolerance	Considerable threat of single point of failure (SPF)	The ledger is fault-tolerant
Incentive	There is no incentives mechanism in databases	Health professionals and patients can be incentivized for reporting promptly and accurately
Consensus	Databases do not have a consensus mechanism as they are centralized	The validation mechanism ensures the integrity of the data as much as possible
Cost	Easy implementation and maintenance as it is a conventional technology	Limited certainty in operation and maintenance costs
Performance	Fast (more transactions processed per second) and offer high scalability	Minimal transactions per second can be handled, and scalability is an issue since the technology is at its developing phase

benefits of blockchain technology and exploit the essential blockchain technology features that may help accelerate patient safety enhancement.

7.3 Proposed Blockchain-based Solution

7.3.1 System Overview

The existing national IRSs like the NRLS have several gaps, as discussed before. Blockchain technology, however, has the potential to fill in these gaps and to improve the current process. According to a study by Naome and colleagues, more than 59% of healthcare providers confirmed that knowing what, how, and who to report incidents to may improve adherence. Further, almost half of the respondents supported that offering rewards could encourage reporting, and 55.7% confirmed that providing feedback and corrective action plans of the reported incidents increased reporting. Also, 55.7% claimed that providing training to health practitioners to detect incidents inspired incident reporting [35]. Therefore, barriers to incident reporting can be summarized as: lack of knowledge and instructions, absence of reward and incentive, and absence of feedback and corrective actions. By removing these barriers, we can ensure better adherence to incident reporting.

Figure 7.2 shows our proposed system overview and how the stakeholders would interact with the blockchain platform to remove the barriers discussed. As a natural process in incident reporting, the patient or health practitioner would report an incident to the blockchain. Our system contains different stakeholders such as the Ministry of Health (MoH), FDA, pharmacies, healthcare practitioners, and patients. The party reporting the incident (patient or healthcare practitioner) would be able to upload the details of an incident that can be stored in the IPFS. For serious incidents, FDA and MoH would be required to develop an action plan to be shared among other stakeholders and that can be stored in the IPFS as well. One unique feature of our system is the ability to distribute some coins as a reward for the incident reporters. Adding a reward mechanism in our proposed solution would ensure healthcare providers' and patients' participation in incidents sharing. According to Kingston and colleagues, senior medical staff agreed on the lack of motivation to report an incident. They added that incident reporting is of little value and time waste, and

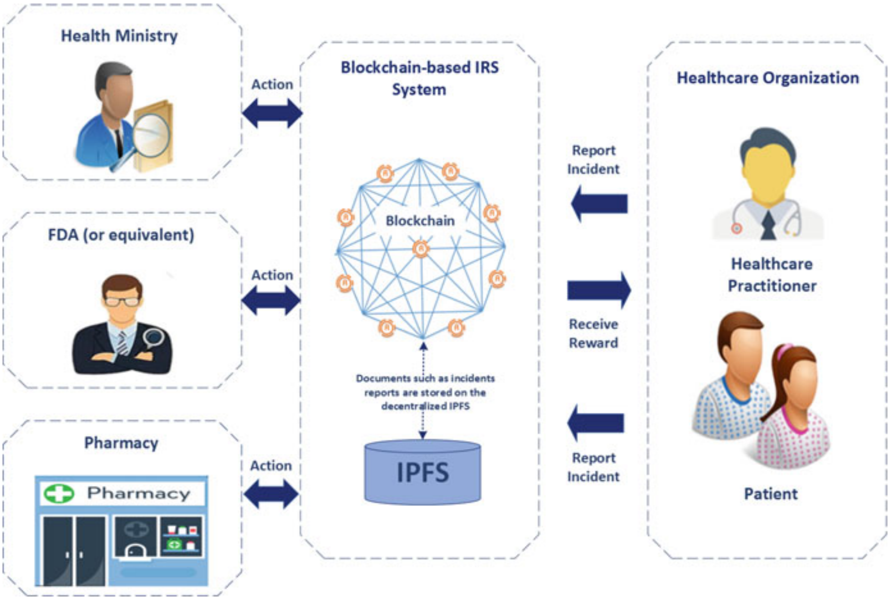


Fig. 7.2 System overview of the proposed solution

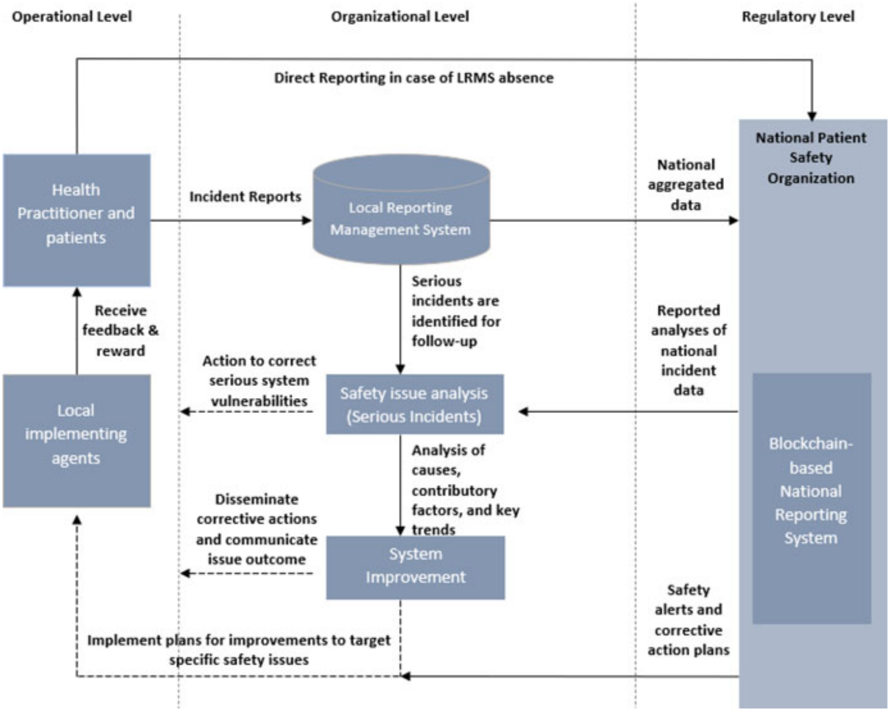

Operational Level
Organizational Level
Regulatory Level
Direct Reporting in case of LRMS absence
Health Practitioner and patients
Incident Reports
Local Reporting Management System
National aggregated data
National Patient Safety Organization
Serious incidents are identified for follow-up
Receive feedback & reward
Reported analyses of national incident data
Action to correct serious system vulnerabilities
Local implementing agents
Safety issue analysis (Serious Incidents)
Blockchain-based National Reporting System
Analysis of causes, contributory factors, and key trends
Disseminate corrective actions and communicate issue outcome
System Improvement
Safety alerts and corrective action plans
Implement plans for improvements to target specific safety issues

Fig. 7.3 Framework of incident reporting system

advised that financial incentives for generated reports might be a greater motivation for reporting [36].

In addition to a reward mechanism, having a feedback mechanism is crucial. Effective feedback on IRSs is vital to learn from the deficiency of care delivery. Feedback from adverse events should also involve corrective action plans and how to enhance safety and eliminate certain vulnerabilities in the system. In the case of failures elimination and prevention, the feedback loop would be closed [37]. Figure 7.3 describes the essential functional levels of learning from incident reporting, drawing upon the existing safety systems from the literature review, in combination with our proposed solution. Our proposed IRS operates on three distinct levels. The first level is operational, and this is where the reporter (health practitioner or patient) reports the incident. The second level is organizational, in which the responsible party (health practitioner) is required to provide detailed information about the incident into the LRMS or blockchain directly. In the case of any serious incident, a safety issue analysis should be run by the organization to identify the contributory factors and immediately improve the system. The third and last level is regulatory and involves parties such as MoH and FDA. At this level, the incident data are consolidated, aggregated, and analyzed for potential feedback. After the analysis of the aggregated data, corrective actions, results, and feedback are disseminated across the system and stored in the IPFS to correct vulnerabilities, and finally, the reporter gets awarded.

While Figs. 7.2 and 7.3 show the system overview and the proposed feedback framework, the following section will highlight the roles and responsibilities of stakeholders in the proposed solution.

7.3.2 Stakeholder Roles and Responsibilities

Various stakeholders play a role in a typical IRS, as discussed earlier. Our proposed blockchain-based incident reporting system will have a particular example of medication incidents, as one of the leading incident types in healthcare [38]. A medication incident takes place when a medication causes or leads to patient harm. Medication incidents may include the following particular examples: medicines given to the incorrect patient, incorrect medicine/dose/route being given, expired medicine, incorrect storage/labeling/packaging/naming of medicine, etc. [11]. The stakeholders involved in a medication incident reporting and sharing process are mainly the Food and Drug Administration (FDA) (or equivalent), pharmacies, MoH, healthcare practitioners, and patients. Table 7.2 encapsulates the roles and key responsibilities of these stakeholders.

7.4 Implementation

In this section, we introduce and examine the system architecture, message sequence diagram, and the used algorithms for the implementation of the proposed blockchain-based IRS. The smart contract has been written in Solidity, the used language for Ethereum smart contracts. The contract[1] is then executed with Remix IDE, a browser-based compiler that comprises an embedded debugger. The debugger is used for warning and alerting the user with error notifications and alarming accordingly [39].

7.4.1 System Architecture

In this section, we propose an Ethereum blockchain-based solution. In the proposed solution, blockchain is used to share the incident report data among stakeholders and prevent this data from being tampered with. Further, in our proposed solution,

[1] https://github.com/douniamarbouh/Incident-Reporting-System.git.

Table 7.2 Stakeholder roles and responsibilities

Stakeholders	Roles	Responsibilities
Patient	An individual who receives care in a healthcare setting	- Notifies, communicates, and provides the necessary incident-related information. - Submits an incident report.
Hospital practitioner	An individual who delivers care to a patient and who may witness, at any stage, an incident or near-miss.	- Notifies, communicates, and provides the necessary incident-related information. - Submits the incident report to the hospital's local risk management system. - Submits directly to the blockchain platform in case the hospital does not have a local risk management system. - Follows the corrective action plan once provided.
Pharmacy	Prepares medications by reviewing and interpreting physician orders.	- Adheres to the corrective action plan. - Adjusts the storage conditions, if applicable. - Withdraws the medication, if needed.
Health Ministry (MoH)	Reviews incidents, generates action plans, and develops guidance and advice.	- Reduces patient safety risk by sharing national learning guidance. - Uses data from the blockchain platform to develop guidance to minimize risks to patients.
FDA (and equivalent)	Monitors and prevents medication errors of regulated drugs and therapeutic biological products.	- Analyzes and monitors medication error reports. - Guides manufacturers about designing and naming the drug products. - Takes regulatory actions such as issuing a safety communication and revising the labeling.

the shared incident data can be traded, and the encrypted transaction information exists among the stakeholders to ensure its reliability and security.

The system architecture in Fig. 7.4 integrates the IPFS technology into the system for the storage of a compilation of hashed files. These files can be retrieved at any point in time once integrated within the blockchain. Incident reports collected and stored in the IPFS are assigned to a unique identifier or hash that can be used later in tracking the corresponding incident report. Therefore, this mechanism makes the IPFS an ideal environment for data storage, since reports are immutable, traceable, timestamped, and time-sequenced. Examples of certain vital files that can be potentially stored in the IPFS are healthcare professionals' e-forms, patients' e-forms, serious incidents (SIs), standard operating procedure (SOP) (detailed instructions on how to report an event), incident reporting protocol, FDA and MoH action plans, and their medical history. The system architecture would compromise four layers, as follows.

User Layer Stakeholders of the blockchain in the user layer have access to the same kind of information. Stakeholders can also use the blockchain to fulfill transactions that can be tracked and can further use it to inhibit the shared incident reports from being tampered with.

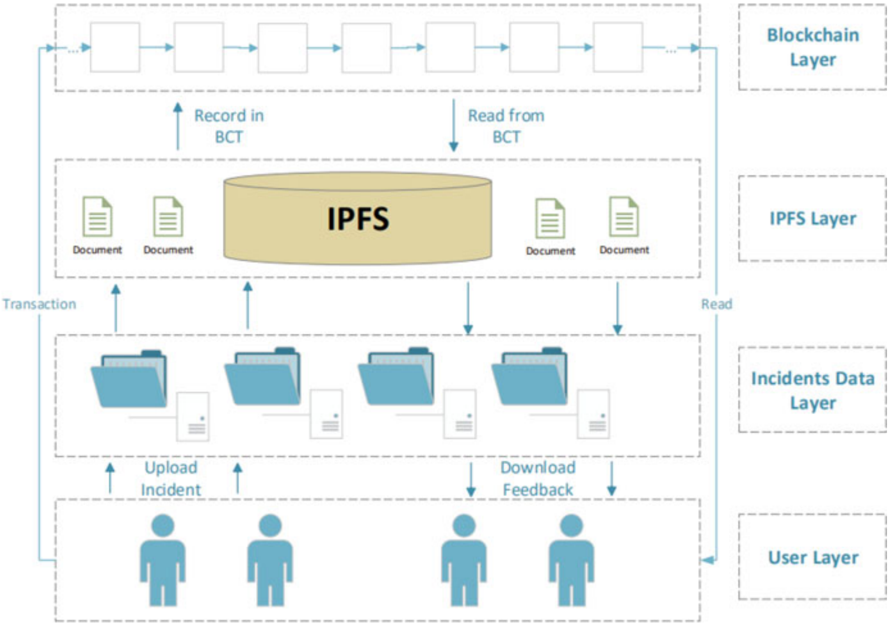


Fig. 7.4 System architecture of incidents reporting among stakeholders

Data Layer Data represents the incident reports and action plans that stakeholders, such as patients or healthcare practitioners, want to share and protect. Stakeholders in the data layer can collectively maintain the data. For the privacy of the uploaded data, the data is encrypted using cryptographic mechanisms. Afterward, the data of the encrypted incident is uploaded onto the IPFS for sharing.

IPFS Layer Interplanetary File System is a P2P network and protocol. It is a decentralized network where each file is stored and identified through its hash value. The encrypted data is uploaded by the owner using a symmetric key that is encrypted further with the data owner's public key. When data is required from the database (based on the hash of the file), it returns the proxy [13]. As explained before, the IPFS would contain important files such as the incident reports and FDA and MoH action plans.

Blockchain Layer Blockchain layer can permanently record all ownership details, transactions, restorations, and modifications of incidents data on the decentralized, distributed, tamper-proof, and transparent ledger. Thanks to the transparency of transactions, the trust of patients and healthcare organizations is strengthened. Blockchain technology also provides feedback to reporters and the history of the incident thanks to the immutability of data in a blockchain.

7.4.2 Entity-Relationship Diagram

The entity-relationship diagram in Fig. 7.5 describes the functions of our smart contract as well as its attributes. The diagram also depicts the relationship between the smart contract and the stakeholders. Capturing these metadata and relations is crucial in executing our smart contract. Moreover, our stakeholders are assumed to be single entities; thereby, the relationship between the contract and a stakeholder is one-to-one. Our stakeholders as described before are the FDA, pharmacies, MoH, patients, and healthcare practitioners.

7.4.3 Message Sequence Diagram

The message sequence diagram's main function is to capture the interactions between several entities while concurrently displaying the events triggered. Further, every participating entity in the network has an Ethereum Address (EA) that supports the interaction of one with another when calling a function. Figure 7.6 shows the sequence flow among several stakeholders from uploading an incident report to getting feedback.

Initially, the MoH would register authorized hospitals in the system via a function called *AuthorizeHospital()*. Then, healthcare practitioners and patients would be registered and assigned an Ethereum Address by executing the functions *AddDoctor()* and *AddPatient()*. By being registered in the system, the reporting party would be able to upload information about an incident. This incident claim is either reported in the Local Reporting Management System (LRMS), if the hospital has one, or is reported directly to the blockchain. This happens thanks to the execution of the function called *LocalIncidentReporting()*. If the authorized hospital does not have a LRMS, the reporting party can report directly into the blockchain platform by executing the following functions, *addIncidentDoctor()* and *addIncidentPatient()*. After the successful reporting of the incident, the event is broadcasted among all stakeholders by *NewIncidentReported()*. In the case of a serious incident, the stakeholders are requested to act. The MoH will have to review the incident and generate an action plan via function *MinistryReview()*.

Additionally, the FDA may need to review the incident *FDAReview()* and act if required by *FDAActions()*. If the MoH and the FDA confirm and approve the danger of the medication involved in the incident, the pharmacies are informed to withdraw the latter via function *PharmacyReview()*. At the end of the process, the reporter is rewarded by executing *RewardReporter()* for reporting the incident and encouraged to report on future occasions.

7.4.4 Implementation Framework

We now describe the algorithms that highlight the working principles of our proposed blockchain solution for IRS. Using the virtual test Ethereum network and

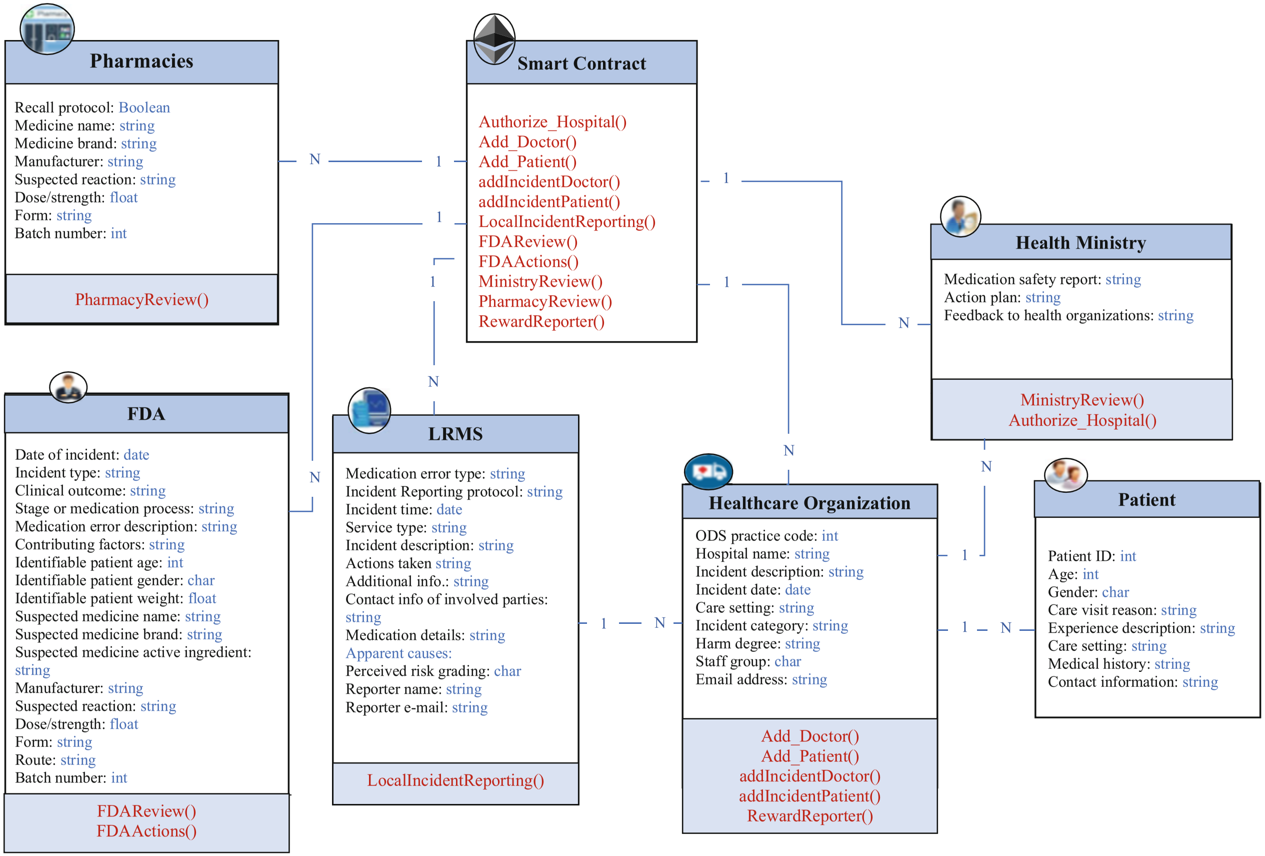


Fig. 7.5 Entity-relationship diagram between stakeholders and smart contract

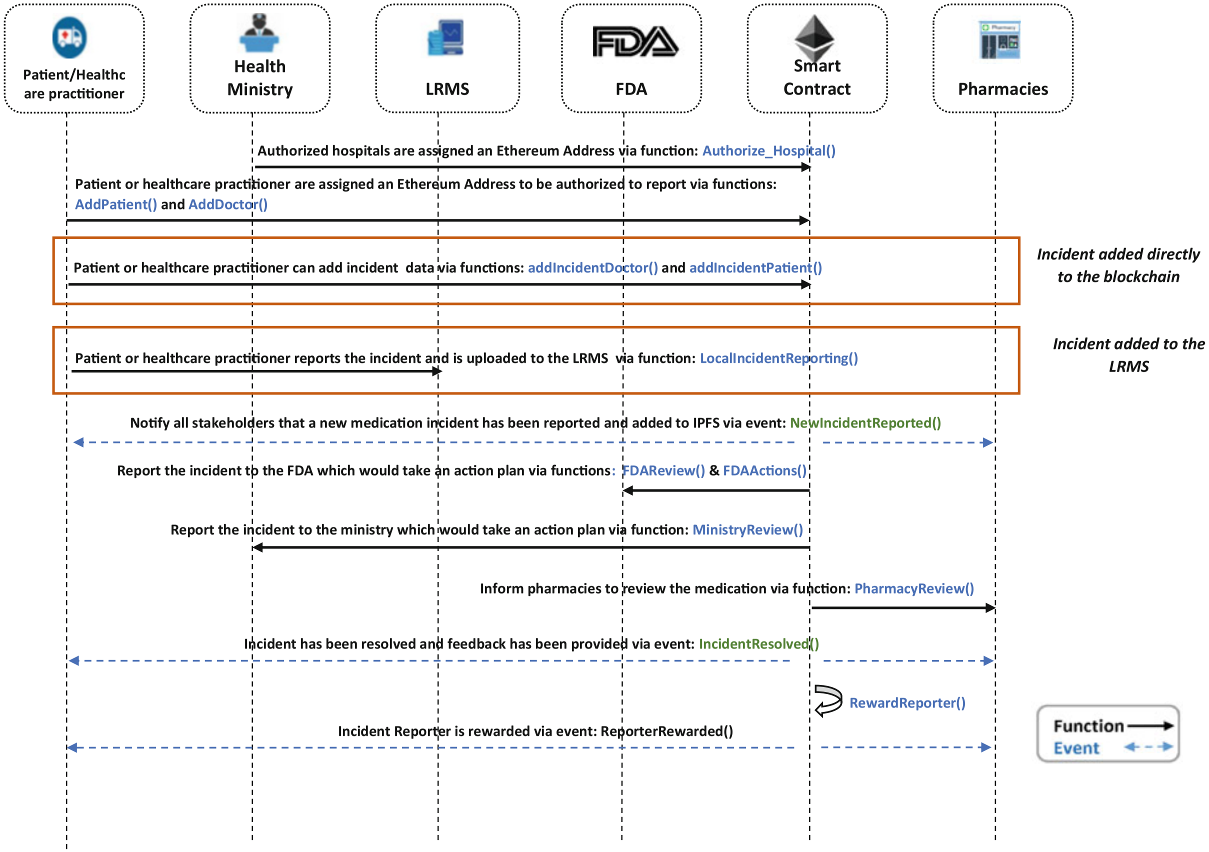
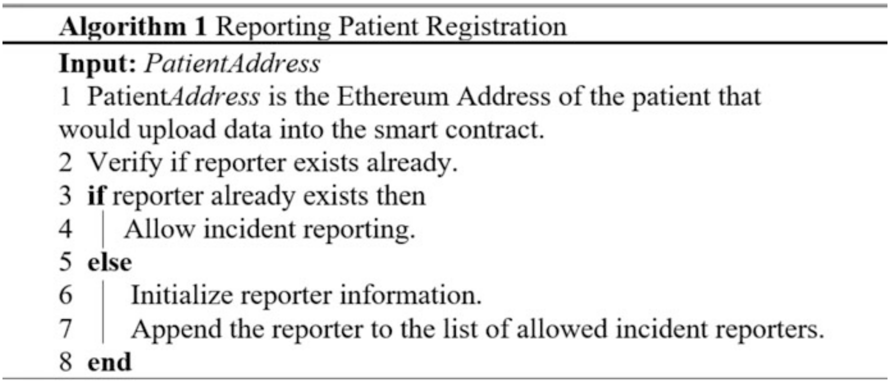

Patient/Healthc are practitioner
Health Ministry
LRMS
FDA
Smart Contract
Pharmacies
Authorized hospitals are assigned an Ethereum Address via function: Authorize_Hospital()
Patient or healthcare practitioner are assigned an Ethereum Address to be authorized to report via functions: AddPatient() and AddDoctor()
Patient or healthcare practitioner can add incident data via functions: addIncidentDoctor() and addIncidentPatient()
Incident added directly to the blockchain
Patient or healthcare practitioner reports the incident and is uploaded to the LRMS via function: LocalIncidentReporting()
Incident added to the LRMS
Notify all stakeholders that a new medication incident has been reported and added to IPFS via event: NewIncidentReported()
Report the incident to the FDA which would take an action plan via functions: FDAReview() & FDAActions()
Report the incident to the ministry which would take an action plan via function: MinistryReview()
Inform pharmacies to review the medication via function: PharmacyReview()
Incident has been resolved and feedback has been provided via event: IncidentResolved()
RewardReporter()
Incident Reporter is rewarded via event: ReporterRewarded()
Function
Event

Fig. 7.6 Message sequence diagram between different stakeholders and smart contract

Remix IDE, our proposed solution was executed and then tested. The smart contract code was also deployed and debugged. As per the functions, they can be checked in the console for verification of methods' output, functionality, and execution cost.

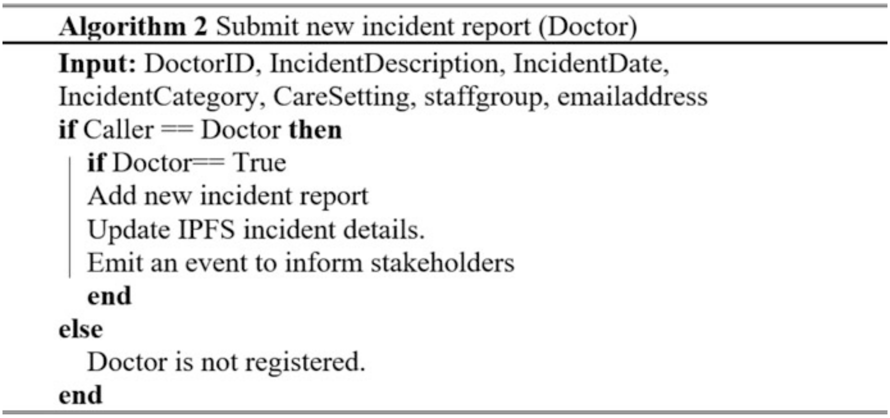
Algorithm 1 below depicts the initial steps that would be taken to register the incident reporting party (patient or healthcare practitioner). First, an Ethereum address (EA) is assigned to incident reporters to enable their interaction with the smart contracts. The algorithm also describes how incident reporters can be registered thanks to the functions AddDoctor() and AddPatient(). In case the reporter is not registered, these functions are used in registering the latter by adding its EA to the incident reporters list (e.g., doctors, patients, etc.).



Algorithm 1 Reporting Patient Registration

```
Input: PatientAddress
PatientAddress is the Ethereum Address of the patient that
would upload data into the smart contract.
Verify if reporter exists already.
if reporter already exists then
    Allow incident reporting.
else
    Initialize reporter information.
    Append the reporter to the list of allowed incident reporters.
end
```

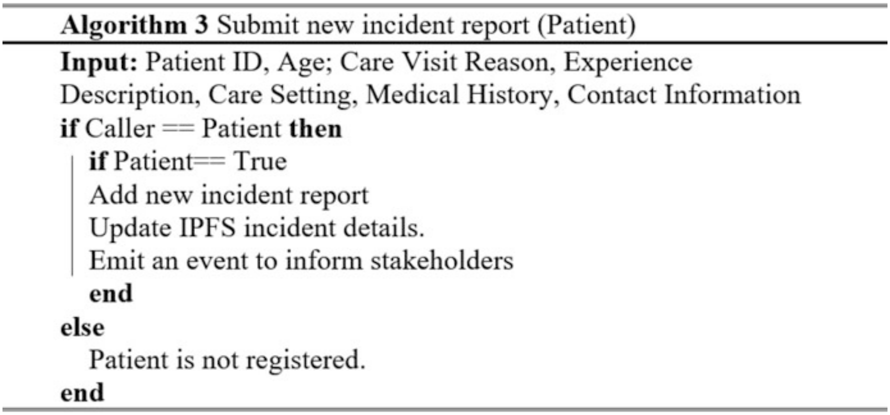
Algorithm 2 shows that doctors with a valid address are permitted to connect and interact with the smart contract and can report. It also shows the variables that are needed from the doctor when deciding to report an incident. These variables include Doctor ID, incident description, incident date, incident category, harm degree, etc. If the address of the patient is unauthorized or unregistered, the patient would not be able to successfully report the details of the incident as the smart contract would deny access to any unauthorized party.



Algorithm 2 Submit new incident report (Doctor)

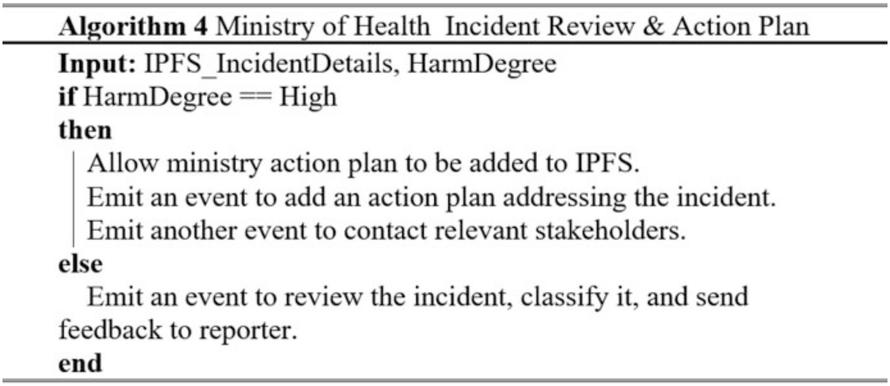
```
Input: DoctorID, IncidentDescription, IncidentDate,
IncidentCategory, CareSetting, staffgroup, emailaddress
if Caller == Doctor then
    if Doctor== True
    Add new incident report
    Update IPFS incident details.
    Emit an event to inform stakeholders
    end
else
    Doctor is not registered.
end
```

Algorithm 3 shows that only patients permitted to connect and interact with the smart contract can report at this stage. Allowed patients are those with an authorized Ethereum address. It also shows the variables that are needed from the patient when deciding to report an incident. These variables include ID, age, primary care visit reason, experience description, etc. If the address of the patient is authorized and recognized, the patient would be able to successfully report the details of the incident.



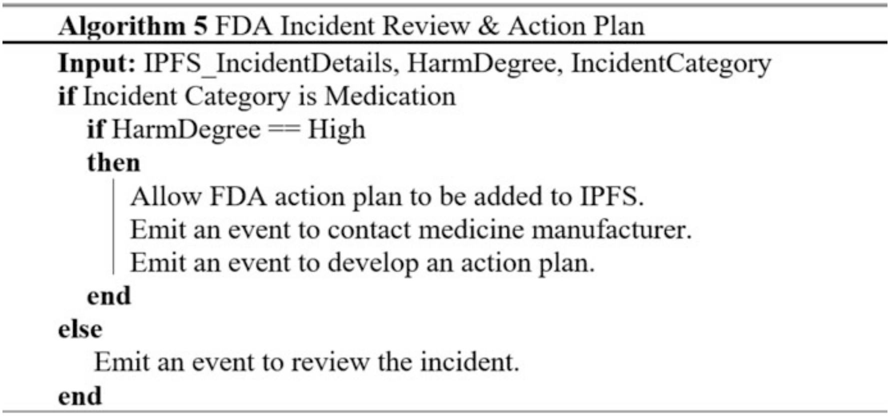
```
Algorithm 3 Submit new incident report (Patient)
Input: Patient ID, Age; Care Visit Reason, Experience
Description, Care Setting, Medical History, Contact Information
if Caller == Patient then
    if Patient== True
    Add new incident report
    Update IPFS incident details.
    Emit an event to inform stakeholders
    end
else
    Patient is not registered.
end
```

Algorithm 4 illustrates the actions undertaken by the MoH if the incident reported is serious (leading to a serious injury or death). In our proposed solution, we are focusing on the case of medication incidents and errors. Thereby, in the case of a serious incident, the MoH would review the latter and develop an action plan to be added to the IPFS. This action plan can be accessed by the different stakeholders (FDA, pharmacies, etc). Furthermore, the MoH would classify the incident and send any feedback when applicable as explained in Fig. 7.3.



```
Algorithm 4 Ministry of Health  Incident Review & Action Plan
Input: IPFS_IncidentDetails, HarmDegree
if HarmDegree == High
then
    Allow ministry action plan to be added to IPFS.
    Emit an event to add an action plan addressing the incident.
    Emit another event to contact relevant stakeholders.
else
    Emit an event to review the incident, classify it, and send
feedback to reporter.
end
```

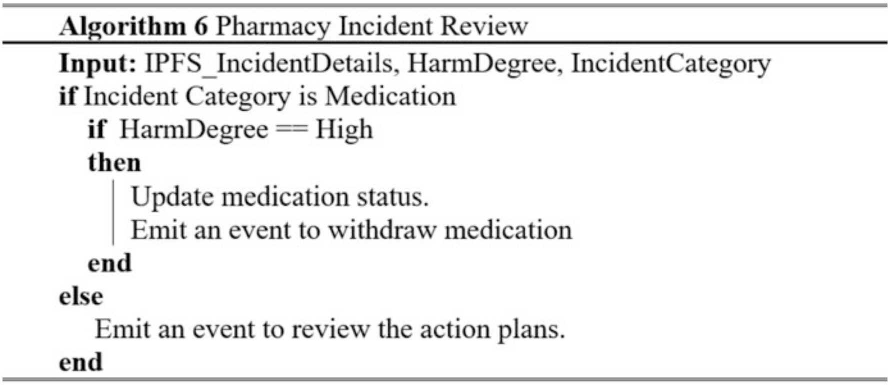
Algorithm 5 describes the steps taken by the FDA when a serious incident is reported. The FDA would review the incident and classify it to determine its type and root causes. The FDA would then determine and examine the information resulting in this medication error and contact the medication manufacturer to either review the labeling, product design, proprietary name, packaging, and/or stop manufacturing the medication by developing an action plan.



Algorithm 5 FDA Incident Review & Action Plan

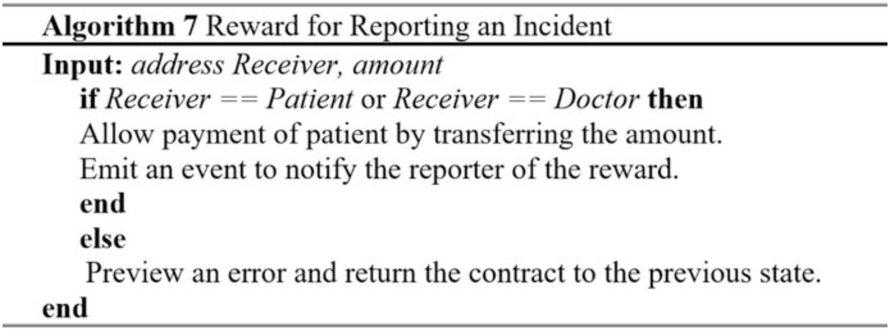
```
Input: IPFS_IncidentDetails, HarmDegree, IncidentCategory
if Incident Category is Medication
   if HarmDegree == High
   then
       Allow FDA action plan to be added to IPFS.
       Emit an event to contact medicine manufacturer.
       Emit an event to develop an action plan.
   end
else
    Emit an event to review the incident.
end
```

Algorithm 6 demonstrates the process undertaken by pharmacies once they receive a medication-related incident. The pharmacy would review the action plan developed by both the FDA and/or the MoH. The pharmacy would then update the status of the medication according to the action plan. If necessary, pharmacies would withdraw the medication. Algorithm 7 explains how a stakeholder can be rewarded for reporting an incident. Rewarding is an essential component of our proposed solution as it would give stakeholders more reasons to report. The receiver can be either the reporting patient or healthcare practitioner. If the Ethereum address does not correspond to an entity with a valid address and who recently reported a true incident event, the system would preview an error.



Algorithm 6 Pharmacy Incident Review

```
Input: IPFS_IncidentDetails, HarmDegree, IncidentCategory
if Incident Category is Medication
   if  HarmDegree == High
   then
       Update medication status.
       Emit an event to withdraw medication
   end
else
    Emit an event to review the action plans.
end
```



```
Algorithm 7 Reward for Reporting an Incident
Input: address Receiver, amount
    if Receiver == Patient or Receiver == Doctor then
    Allow payment of patient by transferring the amount.
    Emit an event to notify the reporter of the reward.
    end
    else
     Preview an error and return the contract to the previous state.
end
```

7.5 Discussion

Incident reporting aims to report incidents and share information about adverse events to ensure lessons are learned, and previous tragedies are not repeated. Since their inception, the reporting systems have used the patient safety incidents reported to identify risks and how they might be avoided. Annually, up to nine thousand people die in the United States alone as a result of a medication error [40]. Moreover, hundreds of thousands of other patients can experience a complication related to a medication or adverse reaction, but it is often not reported [3]. It has also been reported that looking after patients who suffer from medication-associated errors exceeds 40 billion dollars each year. A medication error does not only involve a financial cost but also includes the physical and psychological pain and distress resulting from the error to the patients [40]. Therefore, our proposed platform can contribute to mitigating these challenges while presenting a financially feasible solution as described in the following section.

7.5.1 Cost Analysis

Our proposed blockchain-based solution for incident reporting captures the primary operations required in the reporting process. In this section, we present the cost analysis of the proposed system. To ensure the successful execution of transactions, stakeholders need to pay a gas fee in the network. The unit utilized in measuring the computational effort necessary for transaction executions is Ethereum gas. During the execution of transactions in Ethereum, we incur two types of costs: *execution* and *transaction* costs. The execution cost involves the charges related to modifying states in the contract and internal storage. The *transaction cost* involves the *execution cost* along with the cost of data transfer [41]. Furthermore, we should note that the fee of appending transactions that were verified to each block rises when the gas price rises. Therefore, during high network traffic, this amount is likely

Table 7.3 Transaction Cost Incurred at an average Gas Price of 4 Gwei at an Exchange Rate of 1 ETH = 369 USD

Function name	Transaction gas	Execution gas	Average transaction fee (USD)
Deployment	14,42,069	10,46,869	4.86
AuthorizeHospital()	22,123	1253	0.038
AddPatient()	23,903	1223	0.04
AddDoctor()	24,123	1443	0.041
addIncidentDoctor()	28,484	2284	0.05
addIncidentPatient()	26,788	1996	0.046
LocalIncidentReporting()	25,091	1579	0.043
FDAReview()	22,542	1270	0.038
MinistryReview()	22,366	1094	0.038
PharmacyReview()	22,387	1115	0.038
RewardReporter()	23,988	1116	0.039

to grow because of the competitiveness between miners who try to earn transaction fees by appending transactions in the blocks.

Table 7.3 illustrates the execution and transaction gases as well as the corresponding transaction fees for executing functions and deploying the contract. According to ETH Gas Station, the average gas price is 4 Gwei, and this was obtained on the 15th of October 2020. We converted the obtained transaction fee into US Dollars using the ETH gas station at a rate of 1 ETH = 369 USD. As a result, we observe that the cost incurred when applying our proposed solution is slightly over 6 USD. This analysis demonstrates that applying the introduced solution is financially feasible to all stakeholders in the network and even promotes cost savings to them.

7.5.2 *Security Analysis*

In this section, we explore the properties of security of our proposed blockchain-based incident reporting system to address fundamental concerns related to accountability, integrity, availability, authorization, nonrepudiation, and resistance to malicious attacks such as distributed denial-of-service (DDoS) attack [42].

The code of this implementation is made publicly accessible in GitHub and has been validated via a security tool, called SmartCheck. This tool enables the evaluation and eradication of weaknesses in the code. Further, our proposed smart contract would be less vulnerable than other smart contracts as it has no fallback loops or functions. Moreover, by design, blockchain technology has built-in security features that enable building secure, resilient, and trusted networks and services. For example, security requirements such as authorization, nonrepudiation, integrity, privacy, and availability can be achieved easily using blockchain technology.

Integrity The participating stakeholders in the incident reporting can sign the transactions digitally to guarantee that incident data integrity will be well preserved. Moreover, once data about an incident is appended to the blockchain network, it becomes challenging to alter it thanks to the immutability feature, combination of cryptography, sequential hashing, and the decentralized structure, unlike a conventional database.

Availability The transaction history of entities involved in incident reporting is always available to trace the provenance of an incident. Moreover, duplicated incidents are collected on the nodes of the blockchain. As a result, the network becomes resilient and robust against threats such as a single point of failure.

Authorization The role of different stakeholders in incident reporting is altered as per Table 7.2. Through the authorization feature, only permitted stakeholders can perform a particular task. To ensure that only authorized users can participate and append incident data, we need to secure data access in blockchain networks. Furthermore, the nature of the blockchain and its infrastructure guarantees that every data block can only be added to the chain of existing blocks if it is fully encrypted. Therefore, in case an attacker gained access to the blockchain network, it does not imply that the latter can read the information or retrieve it because of the end-to-end encryption mechanisms.

Nonrepudiation All transactions of incident reporting are cryptographically and digitally signed. This feature implies that organizations can trace a particular incident back at any point in time. Therefore, the user behind any transaction can be identified using their public address. Thanks to this property, users can be reassured that their signature cannot be duplicated on a transaction not created by them. This feature is also enhancing the reliability of the system since the detection of fraudulent transactions becomes easier to notice.

Resistance to Cyberattacks Fraudulent and malicious attacks are increasingly more complicated because of the increase in the threats from cyber organizations and sophisticated malware. Valuable and sensitive data such as medical information, financial data, and intellectual property are incredibly attractive to several users and organizations, and they might attempt stealing them.

7.5.3 Challenges

Blockchain technology has many great features that make it ideal for many application cases. However, it also carries a set of challenges. These challenges contributed greatly to its slow-moving adoption, and these include:

Scalability As the number of transactions increases every day, the blockchain network traffic might become bulky. Nodes of the blockchain are expected to store all validated transactions, which is a hurdle since the block size is restricted and the time required to generate a new block is long. Processing a few transactions per second is a problem since millions of transactions are expected to be handled in real time. Consequently, because of the size limitation of the block, miners choose to validate transactions with higher transaction rates that cause some transactions to be delayed [43, 44].

Selfish Mining The blockchain network that depends on a consensus of the majority to validate transactions is prone to attackers if they could compromise a significantly large group of nodes. For example, malign actors can compromise a public blockchain network if they could manipulate at least 51% of the consensus and mining power. The same problem can also occur if several miners secretly join forces to create a majority and control the blockchain. "Selfish miners" follow a strategy of creating a private separate branch and mining without broadcasting. Then, the private chain is only published when the current public chain is shorter compared to it [39]. In other words, they mine the chain with no competitors, while "honest miners" spend time and resources on mining worthless branch [29].

Legal Challenges Until this date, smart contracts, and blockchain, in general, are highly de-regulated and standardized at the national and international levels. Due to having many stakeholders, data ownership and existing medical law of the traditional healthcare system are essential issues. Further, new regulations on health policy, digital inequality, policies of digital health services, data sharing, and digital connectivity should be addressed [45].

Privacy Concerns In blockchain technology, information such as the balance and details of public keys are visible to all network stakeholders; as a result, the system might become prone to privacy leakage. To solve the issue, some have proposed solutions such as *anonymous solution* and *mixing solution* to reach participants anonymity [43]. The mixing service involves sending funds to multiple output addresses from multiple input addresses. On the other hand, the anonymous solution unlinks the payment origins of a transaction that prevents the graph analysis of a transaction [46].

7.6 Conclusion

In this chapter, we proposed a new blockchain-based framework for incident reporting using Ethereum smart contracts. Our proposed blockchain-based solution supports traceability and transparency and aligns communication among stakeholders in the system. Moreover, it ensures data immutability and security while simultaneously encouraging the collection of incidents from various stakeholders. The smart contract code was used to capture interactions and share the process in the appropriate order. Further, we integrated the IPFS technology in our framework to store various files such as incident reports, corrective action plans, etc. that serves as a decentralized storage technology. The proposed solution can also ensure the reliability of reporting while maintaining high efficiency in the non-fully trusted environment. Further, the proposed blockchain-based reporting system encourages both healthcare practitioners and patients to actively participate in sharing and reporting incidents thanks to the incentivizing mechanism.

The system architecture, sequence diagram, and algorithms are not limited to medication errors but can be extended to include reporting several other types of errors and incidents (falls, operations, procedures, etc.). The functions developed were tested in the Remix environment, and the transaction costs incurred were computed by performing cost analysis. It revealed that a minimal cost of less than 1 USD is incurred when executing transactions, while deploying the contract was less than 5 USD. This shows that the proposed solution is viable as the stakeholders pay almost 6 USD amount compared to traditional incident reporting systems, requiring a partial payment to be made to third-party service providers. Future research can be extended to include the development of front-end DApps that patients and healthcare practitioners can easily use to report any incident. The smart contract can also be developed to capture further areas such as EHRs, to enhance learning from incidents and feedback mechanism.

Acknowledgments This work was supported by the Khalifa University of Science and Technology under Award CIRA-2019-001 and RCII-2019-002- Research Center for Digital Supply Chain and Operations Management.

References

1. M.A. Makary, M. Daniel, Medical error-the third leading cause of death in the US. BMJ **353**, i2139 (2016)
2. Y. Wang, E. Coiera, W. Runciman, F. Magrabi, Using multiclass classification to automate the identification of patient safety incident reports by type and severity. BMC Med. Inform. Decis. Mak. **17**(1), 1–12 (2017)
3. M.C.E. Simsekler, The link between healthcare risk identification and patient safety culture. Int. J. Health Care Qual. Assur. **32**(3), 574–587 (2019)
4. K.G. Shojania, E.J. Thomas, Trends in adverse events over time: why are we not improving? BMJ Qual. Saf. **22**(4), 273–277 (2013)
5. E. Ramírez, A. Martín, Y. Villán, M. Lorente, J. Ojeda, M. Moro, C. Vara, M. Avenza, M.J. Domingo, P. Alonso, M.J. Asensio, J.A. Blázquez, R. Hernández, J. Frías, A. Frank, Effectiveness and limitations of an incident-reporting system analyzed by local clinical safety leaders in a tertiary hospital: prospective evaluation through real-time observations of patient safety incidents. Medicine **97**(38), e12509 (2018)
6. O. Levtzion-Korach, A. Frankel, H. Alcalai, C. Keohane, J. Orav, E. Graydon-Baker, J. Barnes, K. Gordon, A.L. Puopolo, E.I. Tomov, L. Sato, D.W. Bates, Integrating Incident Data from Five Reporting Systems to Assess Patient Safety: Making Sense of the Elephant. Jt. Comm. J. Qual. Patient Saf. **36**(9), 402–AP18 (2010)
7. C. Macrae, The problem with incident reporting: Table 1. BMJ Qual. Saf. **25**(2), 71–75 (2016)
8. D.J. Noble, P.J. Pronovost, Underreporting of Patient Safety Incidents Reduces Health Care's Ability to Quantify and Accurately Measure Harm Reduction. J. Patient Saf. **6**(4), 247–250 (2010)

9. J.E. Anderson, N. Kodate, R. Walters, A. Dodds, Can incident reporting improve safety? Healthcare practitioners' views of the effectiveness of incident reporting. Int. J. Qual. Health Care **25**(2), 141–150 (2013)
10. G. Armitage, S. Moore, C. Reynolds, P.-A. Laloë, C. Coulson, R. McEachan, R. Lawton, I. Watt, J. Wright, J. O'Hara, Patient-reported safety incidents as a new source of patient safety data: an exploratory comparative study in an acute hospital in England. J. Health Serv. Res. Policy **23**(1), 36–43 (2018)
11. A. Tariq, A. Georgiou, J. Westbrook, Medication incident reporting in residential aged care facilities: Limitations and risks to residents' safety. BMC Geriatr. **12**(1), 1–16 (2012)
12. C. Stavropoulou, C. Doherty, P. Tosey, How Effective Are Incident-Reporting Systems for Improving Patient Safety? A Systematic Literature Review: Incident-Reporting Systems for Improving Patients' Safety. Milbank Q. **93**(4), 826–866 (2015)
13. I.A. Omar, R. Jayaraman, K. Salah, M.C.E. Simsekler, I. Yaqoob, S. Ellahham, Ensuring protocol compliance and data transparency in clinical trials using Blockchain smart contracts. BMC Med. Res. Methodol. **20**(1), 1–17 (2020)
14. I.A. Omar, R. Jayaraman, K. Salah, M.C.E. Simsekler, *Exploiting Ethereum Smart Contracts for Clinical Trial Management* (IEEE, New York, 2019), pp. 1–6
15. G. Hagley, P.D. Mills, B.V. Watts, A.W. Wu, Review of alternatives to root cause analysis: developing a robust system for incident report analysis. BMJ Open Quality **8**(3), e000646 (2019)
16. T. Hewitt, S. Chreim, A. Forster, Incident reporting systems: A comparative study of two hospital divisions. Arch. Public Health **74**(1), 34 (2016)
17. Y. Gong, H. Kang, X. Wu, L. Hua, Enhancing Patient Safety Event Reporting: A Systematic Review of System Design Features. Appl. Clin. Inform. **08**(03), 893–909 (2017)
18. World alliance for patient safety: WHO draft guidelines for adverse event reporting and learning systems: from information to action (2005)
19. M.C.E. Simsekler, A. Qazi, M.A. Alalami, S. Ellahham, A. Ozonoff, Evaluation of patient safety culture using a random forest algorithm. Reliab. Eng. Syst. Saf. **204**, 107186 (2020)
20. Learning from patient safety incidents (2018)
21. National patient safety incident reports. Technical report, NHS, UK (2019)
22. R. Cole, M. Stevenson, J. Aitken, Blockchain technology: implications for operations and supply chain management. Supply Chain Manag. Int. J. **24**(4), 469–483 (2019)
23. M. Atzori, Blockchain technology and decentralized governance: Is the state still necessary? Journal of Governance and Regulation **6**(1), 45–62 (2017)
24. I. Marco, K.R. Lakhani, The truth about blockchain, in *Harvard Business Review* (2017)
25. A.H. Mayer, C.A. da Costa, R. da Rosa Righi, Electronic health records in a Blockchain: A systematic review. Health Informatics J. **26**(2), 1273–1288 (2020)
26. P.P. Ray, D. Dash, K. Salah, N. Kumar, Blockchain for IoT-based healthcare: Background, consensus, platforms, and use cases. IEEE Syst. J. **15**(1), 1–10 (2020)
27. R. Jayaraman, F. AlHammadi, M.C.E. Simsekler, *Managing Product Recalls in Healthcare Supply Chain* (IEEE, New York, 2018), pp. 293–297
28. H.-J. Yoon, Blockchain Technology and Healthcare. Healthcare Inf. Res. **25**(2), 59 (2019)
29. M.A. Khan, K. Salah, IoT security: Review, blockchain solutions, and open challenges. Futur. Gener. Comput. Syst. **82**, 395–411 (2018)
30. S. Khezr, M. Moniruzzaman, A. Yassine, R. Benlamri, Blockchain Technology in Healthcare: A Comprehensive Review and Directions for Future Research. Appl. Sci. **9**(9), 1736 (2019)
31. P. Mamoshina, L. Ojomoko, Y. Yanovich, A. Ostrovski, A. Botezatu, P. Prikhodko, E. Izumchenko, A. Aliper, K. Romantsov, A. Zhebrak, I.O. Ogu, A. Zhavoronkov, Converging blockchain and next-generation artificial intelligence technologies to decentralize and accelerate biomedical research and healthcare. Oncotarget **9**(5), 5665–5690 (2018)
32. S. Ellahham, N. Ellahham, M.C.E. Simsekler, Application of Artificial Intelligence in the Health Care Safety Context: Opportunities and Challenges. Am. J. Med. Qual. **35**(4), 341–348 (2020)

33. M.C.E. Simsekler, A.P. Gurses, B.E. Smith, A. Ozonoff, Integration of multiple methods in identifying patient safety risks. Saf. Sci. **118**, 530–537 (2019)
34. Y. Wehbe, M.A. Alzaabi, D. Svetinovic, *Blockchain AI Framework for Healthcare Records Management: Constrained Goal Model* (2018) pp. 420–425
35. T. Naome, M. James, A. Christine, T.I. Mugisha, Practice, perceived barriers and motivating factors to medical-incident reporting: a cross-section survey of health care providers at Mbarara regional referral hospital, southwestern Uganda. BMC Health Serv. Res. **20**(1), 1–9 (2020)
36. M.J. Kingston, S.M. Evans, B.J. Smith, J.G. Berry, Attitudes of doctors and nurses towards incident reporting: a qualitative analysis. Med. J. Aust. **181**(1), 36–39 (2004)
37. J. Benn, M. Koutantji, L. Wallace, P. Spurgeon, M. Rejman, A. Healey, C. Vincent, Feedback from incident reporting: information and action to improve patient safety. Qual. Saf. Health Care **18**(1), 11–21 (2009)
38. M.R. Cohen, J.L. Smetzer, Preventing Mix-Ups between various formulations of Amphotericin B; Arixtra is not a Hemostat; measurement Mix-Up; drug names too close for comfort; New Vaccine errors reporting program. Hosp. Pharm. **48**(2), 95–98 (2013)
39. D. Marbouh, T. Abbasi, F. Maasmi, I.A. Omar, M.S. Debe, K. Salah, R. Jayaraman, S. Ellahham, Blockchain for COVID-19: Review, Opportunities, and a Trusted Tracking System. Arab. J. Sci. Eng. **45**(12), 9895–9911 (2020)
40. R.A. Tariq, R. Vashisht, A. Sinha, Y. Scherbak, Medication Dispensing Errors And Prevention, in *StatPearls* (StatPearls Publishing, Treasure Island, 2020)
41. A. Chaer, K. Salah, C. Lima, P.P. Ray, T. Sheltami, *Blockchain for 5G: Opportunities and Challenges* (IEEE, New York, 2019), pp. 1–6
42. H.R. Hasan, K. Salah, Combating Deepfake Videos Using Blockchain and Smart Contracts. IEEE Access **7**, 41596–41606 (2019)
43. Md.M.H. Onik, S. Aich, J. Yang, C.-S. Kim, H.-C. Kim, Blockchain in Healthcare: Challenges and Solutions, in *Big Data Analytics for Intelligent Healthcare Management* (Elsevier, Amsterdam, 2019), pp. 197–226
44. P. Zhang, D.C. Schmidt, J. White, G. Lenz, Blockchain Technology Use Cases in Healthcare, in *Advances in Computers*, vol. 111 (Elsevier, Amsterdam, 2018), pp. 1–41
45. E. Gökalp, M.O. Gökalp, S. Çoban, P.E. Eren, Analysing Opportunities and Challenges of Integrated Blockchain Technologies in Healthcare, in *Information Systems: Research, Development, Applications, Education*, ed. by S. Wrycza, J. Maślankowski, vol. 333 (Springer, Cham, 2018), pp. 174–183. Series Title: Lecture Notes in Business Information Processing
46. A. Dubovitskaya, Z. Xu, S. Ryu, M. Schumacher, F. Wang, Secure and trustable electronic medical records sharing using blockchain. AMIA Ann. Symp. Proc. AMIA Symp. **2017**, 08 (2017)

Chapter 8
Implementation of Hyperledger-Based Secure Firmware Update Delivery for IoT Devices

Meriem Bettayeb, Qassim Nasir, and Manar Abu Talib

8.1 Introduction

The Internet of Things (IoT) is the uprising technology that has become the main term to define the enormous number of devices that communicate via the internet. The IoT is widely applicable in wearable devices, applications of smart homes, and the control scheme of companies and hospital devices [51]. But, security problems are potentially the greatest obstruction to the IoT advancement, as security issues can compromise confidential data and threaten public safety. As a result, it is needed to authenticate the process of updating the software in the IoT devices to verify whether or not it is legitimate [21].

Firmware (FW) is embedded software that runs on embedded systems to enable its functionalities and control them [32]. Nowadays, many IoT devices have updatable firmware to fix threats found in the system throughout the device's life cycle and to add new functionality to the device, thereby enhancing reliability and security [52]. Firmware-driven devices are found in many domains such as home automation, networking, entertainment, and health care. Many attacks have been encountered against the firmware updates of the embedded systems, such as modification attacks [40].

M. Bettayeb (✉)
Department of Electrical and Computer Engineering, University of Sharjah, Sharjah, UAE
e-mail: u17105766@sharjah.ac.ae

Q. Nasir
Department of Electrical Engineering, University of Sharjah, Sharjah, UAE
e-mail: nasir@sharjah.ac.ae

M. A. Talib
Department of Computer Science, University of Sharjah, Sharjah, UAE
e-mail: mtalib@sharjah.ac.ae

M. H. ur Rehman et al. (eds.), *Trust Models for Next-Generation Blockchain Ecosystems*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-75107-4_8

One of the methods used to mitigate these attacks on firmware is to keep the IoT devices up to date [39]. To maintain the integrity of any IoT device, its firmware version needs to be checked and validated, and if necessary, a secure firmware update must be downloaded. Moreover, the UAE and many other countries are moving toward smart cities, resulting in a rise in the quantity of IoT devices. The need to update all these devices synchronously from a dedicated server will cause latency and network traffic in the vendor. The available server–client method for ensuring a secure firmware update process is therefore not appropriate for firmware distribution in the IoT environment [51].

In this chapter, a decentralized solution was designed and deployed to securely update the firmware in IoT devices using blockchain technology. This work procedure secures the IoT devices against receiving any malicious firmware files from fake vendors [10]. The literature review included in this chapter investigated existing assessments of IoT vulnerability to counterfeit firmware updates. A feasibility study was conducted to assess technical requirements and develop defensive measures. To the best of our knowledge, there is no standard for securing IoT firmware updates, even though the issue is a subject of much discussion by security experts and practitioners [44]. In this chapter, a secure framework for providing firmware updates for IoT devices by using blockchain technology was defined. Furthermore, this chapter is the first implementation of blockchain-based secure firmware updates for IoT devices. This chapter is a continuation and implementation of the idea proposed previously by the authors in [4].

The main contribution of this chapter is to design and implement a blockchain-based architecture for IoT that securely delivers lightweight distributed firmware updates and that performs the following functions:

- Secure download of the latest firmware for the IoT devices.
- Validate the integrity and correctness of the downloaded firmware file.
- Deploy the whole system.

The structure of this chapter is as follows: After providing the introduction to firmware/software security in Sect. 8.1, the background of Hyperledger blockchain and the characteristics of the blockchain network are explained in Sect. 8.2. Related work on the existing firmware update solutions using blockchain is reviewed in Sect. 8.3. In Sect. 8.4, the research objectives and methodology are discussed, along with the design of a decentralized firmware update system for IoT devices. Section 8.5 describes the experiment and lays out the methodology of the tests performed on our proposed system of secure firmware updates in Hyperledger blockchain. It also provides a security and performance analysis of the implementation of our proposed scheme. Finally, a summary and some directions for future work are presented in Sect. 8.6.

8.1.1 Motivation and Significance

Nowadays, many IoT devices have updatable firmware to fix bugs found in the system during its life cycle, increase reliability and security, and add new features to the device. Our architecture makes sure that the IoT devices have the latest firmware update coming from the right manufacturer and ensures that it has not been modified.

Many attacks can occur when downloading firmware updates from the vendor to the IoT device through a public network. These include modification attacks, man-in-the-middle attacks (MITM), denial-of-service attacks (DoS), impersonation attacks, etc. [38]. The proposed design tries to minimize these types of attacks by using blockchain to download the firmware update to the IoT devices. A blockchain network is a trusted and secure environment for firmware updates, as it provides tracking records for all updates for both the vendor and the end user [29]. Moreover, access to data is restricted to the members of the permissioned blockchain, i.e., Hyperledger Fabric. Hence, no one can read or write in this network except for authorized network members.

The increasing number of IoT devices in smart cities and update requests for each device connected to the internet will result in overhead on the vendor server. The use of blockchain will reduce network traffic in the vendor while also cutting down latency in comparison with the server–client technique of updating the firmware [33].

8.2 Preliminaries

Blockchain is the new driving technology in the future, and it has many applications in healthcare, cryptocurrency, supply chain, data management, and many other applications. This decentralized architecture delivers a trustful output without the requirement of trusting any entity within it [54].

8.2.1 Distributed Ledger

Blockchain is a decentralized list of all transactions in a peer-to-peer network [26]. This technique was originally described in 1991. It was adopted by "Satoshi Nakamoto" (pseudonym) in 2008 to create the digital cryptocurrency Bitcoin. It performs a trustful transaction between two entities without any third entity [14]. One great property of this decentralized ledger is that once data has been saved on a blockchain, it is hard to modify it. Blockchain characteristics are described in Fig. 8.1. The key advantages of blockchain technology are:

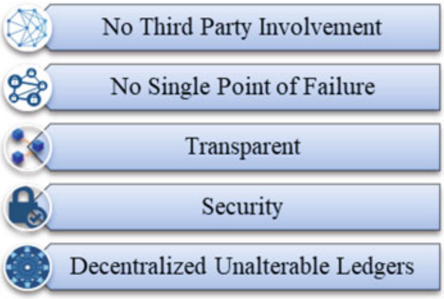

No Third Party Involvement
No Single Point of Failure
Transparent
Security
Decentralized Unalterable Ledgers

Fig. 8.1 Blockchain characteristics

- Decentralization: The architecture of the blockchain is decentralized, which lowers the cost of its operations and increases robustness, as it does not have any single point of failure [13].
- Integrity: The ledger is copied between the nodes in the network, and if any change occurs in the ledger, the change will be updated in all nodes, thereby providing immutability and transparency [13].
- Flexibility: Blockchain is open source and programmable, which provides an opportunity for other researchers to work with this technology [13].

8.2.2 Blockchain Types

Three types of blockchain have been developed based on the permission of each node to access a block in the blockchain network, as shown in Fig. 8.2:

- Public: Anyone can participate in the blockchain network, send transactions, create blocks, and have a copy of the ledger. This type values anonymity, immutability, and transparency over efficiency. It is also called permissionless ledger, as it requires no permissions to read and write in the ledger. Ethereum and Bitcoin are examples of public blockchains [9].
- Private: Participating in the network is restricted in a private blockchain. A private blockchain is operated by one entity. This type of blockchain values efficiency over anonymity, immutability, and transparency. The blockchain requires permission to read and write in the ledger. Ripple is an example of a private blockchain [9].
- Permissioned: A mix of public and private blockchain types. A permissioned blockchain is operated by known entities, such as the stakeholders of a given industry. It values immutability and efficiency over anonymity and transparency. Hyperledger Fabric and Corda are examples of permissioned blockchains [9].

The type of blockchain selected for any work depends on the requirements of the application. For this chapter, a permissioned blockchain called Hyperledger was chosen to prevent unauthorized users from gaining access to the blockchain network.

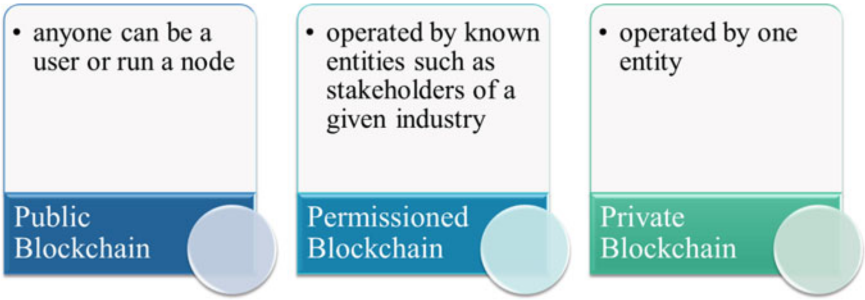

• anyone can be a user or run a node
Public Blockchain
• operated by known entities such as stakeholders of a given industry
Permissioned Blockchain
• operated by one entity
Private Blockchain

Fig. 8.2 Blockchain types

8.2.3 Hyperledger Project

Hyperledger is an enterprise-grade distributed ledger. It was built as a project of the Linux Foundation in December 2015. Hyperledger Fabric supports a form of smart contract logic called chaincode. It is a reframing of the smart contract but with more characteristics. A chaincode is a computer code that runs on a blockchain network after the agreement of the chaincode parties. It has predefined rules that need to be met to execute automatically. It is validated within the consensus process by the chain validators. Developers use chaincode to regulate decentralized applications, build business contacts, and asset definitions. Hyperledger Fabric uses the chaincode method to enforce the trust.

Hyperledger business blockchain frameworks benefit different organizations by establishing their enterprise blockchains. They are divergent from public ledgers such as Bitcoin and Ethereum. The Hyperledger frameworks consist of:

- An append-only distributed ledger
- A consensus algorithm that lowers the risk of fraudulent transactions by confirming any modifications in that
- Transactions' privacy by permissioned access
- Chaincode to handle transaction requests

The Hyperledger tools or modules are software deployed to maintain blockchain networks and auditing information on the ledgers [5]. Figure 8.3 shows different Hyperledger frameworks and tools. These tools are reusable across the various distributed ledger technology frameworks. The concentration of this chapter is focused on using Hyperledger Fabric and Composer tools.

Hyperledger Fabric was selected for our purposes because Smart Dubai used Hyperledger Fabric as the main blockchain platform in all their government projects [23]. The UAE Government launched an initiative called Emirates Blockchain Strategy 2021 intending to capitalize on blockchain technology to transfer 50% of all government transactions onto the blockchain by 2021. Dubai smart government selected Hyperledger Fabric as the main blockchain platform for government

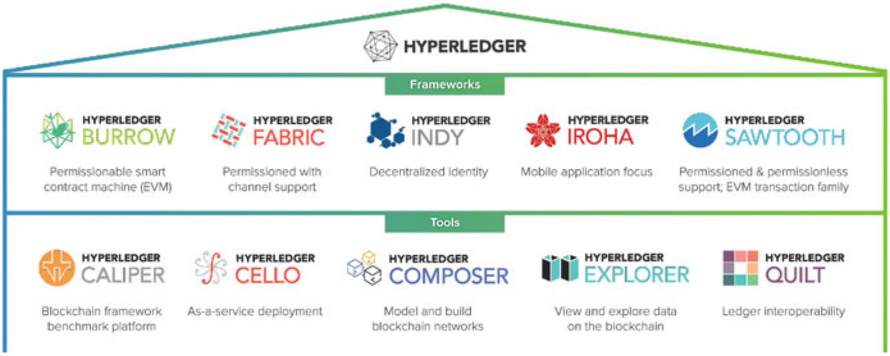


Fig. 8.3 Hyperledger project [41]

applications. Related government applications in different networks will need to interact with each other. However, these government applications will be isolated from each other, and the data from the blockchain applications cannot be shared. That is why the Hyperledger Fabric framework was selected from among other blockchain frameworks in this chapter.

Another reason is that Hyperledger Fabric ensures confidentiality, resiliency, flexibility, and scalability. It allows parties to establish confidential transactions without the need to pass the information through a central authority. This is accomplished because the Hyperledger Fabric has different channels in the network and because of the labor division between the different nodes in the blockchain network. Hyperledger Fabric has the following characteristics [9]:

- Permissioned network: provides access rights within the blockchain network. As a result, not anyone can be a member of the network.
- Confidential transactions: give the flexibility to select entities to visible the transactions in the blockchain network if they have the right encryption keys.
- No cryptocurrency: does not need mining and costly computations to ensure transactions.
- Programmable: leverages the embedded logic in chaincode to automate business processes across a given architecture.
- Private channels: The framework's distributed ledger and chaincode platform allow for private channels. It is possible to create channels and share private data between certain entities in a larger network. Moreover, not every transaction can be seen by every participant of the network. Hyperledger Fabric assures transparency by permitting private transactions (unlike Ethereum).

As the platform implements a permissioned ledger, it forms a security infrastructure for authentication and authorization. It provides transaction authorization by using public-key certificates, and confidentiality for chaincode accomplished by using in-band encryption [9].

More precisely, to connect to the network, every peer needs an enrollment certificate from an enrollment CA. Furthermore, to perform transactions, every peer must obtain a transaction certificate issued by a transaction CA and support pseudonymous authorization for peers, which makes the multiple transactions for the same peer not linked to each other.

Hyperledger Composer can be selected from Hyperledger project tools. Composer is an open development toolset that is used for creating and managing business network applications (BNAs) deployed on the Hyperledger platform using high-level composer language [2]. Furthermore, it can be used to create business applications on Hyperledger DLT frameworks (Hyperledger Fabric in our case). The primary goal is to accelerate the development of blockchain applications on Hyperledger. The benefits of using Composer are that it:

- Hides the complexity of the underlying infrastructure.
- Offers business modeling capability by way of modeling language that can easily be used by non-technical team members such as a business analyst.
- Chaincode or transactions can be written in JavaScript, which most developers are familiar with. As a result, it becomes easy to write and manage chaincode.

8.3 Literature Review

In this section, our research findings and the related work about firmware update security are discussed, analyzing different solutions for securing the firmware update process for IoT devices [20, 30]. Part of the survey conducted in this section has been presented in the authors' previous research [3]. In IoT devices, attackers try to modify the IoT's firmware file by injecting fraudulent lines of code in it. To solve this problem, some researchers started to merge blockchain technology with IoT to download the firmware updates securely. Authors in [18] introduced the idea of applying blockchain technology to assure the security of information transferred between participants of an IoT network.

Pillai et al. [35] described the advantage of using a blockchain framework for checking the integrity of the updates by smart contracts and consensus algorithms. Furthermore, Lee et al. [6] proposed a validation process for the correctness of the firmware update and obtained it by the IoT devices. In the proposed scheme, every IoT device is required to save the blockchain ledger in their local storage. However, the IoT devices have limited resources such as storage capacity and computations. So, this mechanism is difficult to be deployed in a real-life IoT environment. In the case of Boudguiga et al. [7], the authors added the ability of the checking potential bugs in the firmware image on the top of the validation process. Additionally, [19] used the ESP8266 board to simulate the update process from the blockchain network to the IoT device and validate its integrity. The firmware is processed in fragments as the device contains 50 KB RAM.

Moreover, Yohan et al. [49] defined direct and indirect update methodologies based on the entity responsible for creating the smart contract whether it is the manufacturer of the intermediate entity. The study concentrated on resource-constrained IoT devices such as the Wi-Fi smart plug and sensors. Similarly, Yohan et al. [51] defined the firmware update distribution process and smart contract creation process. This chapter was extended in [50] by defining a process in which the IoT devices would be able to request the firmware update from an immediate entity in the blockchain network.

Furthermore, Zhao et al. [53] proposed the usage of outsourced attribute-based signature as a privacy scheme. Leiba et al. [28] presented a digital currency reward process to the manufacture when a software update process happens. The problem with the proposed approaches is that they did not provide a comprehensive performance evaluation so as a result, they will have a limitation when the frameworks got deployed in real-life scenarios with real IoT devices. That is why this is still an open area of research that should be taken into consideration.

Recently, Anastasiou et al. [1] used LoRa technology to update the firmware in the blockchain network. The evaluation included trying different firmware sizes and network sizes. They concluded that more gateways are needed to perform a reliable firmware update. Dhaka et al. [12] proposed to continuously check the firmware if it has any bugs, but no details were presented about the methodology or the design. Islam et al. [24] programmed the smart contract in the blockchain-based firmware update architecture to perform different services for device registration, ownership information checking, device identity verification, ownership transfer, and firmware update. This chapter verifies the IoT device instead of the firmware update file. Furthermore, public blockchain was used, which added transaction cost each time the smart contract was executed. Choi et al. [11] proposed to save the firmware image in the distributed ledgers in the blockchain architecture. By this, even if the manufacturer left the blockchain network, the firmware is still available. For this proposed solution, each IoT needs to verify if it needs a firmware update. This requires a lot of time and energy consumption from the low-power embedded device. Tsai et al. [43] implemented a blockchain-based firmware update architecture using MQTT protocol. Multiple MQTT servers are connected as blockchain nodes for the real-time firmware update patches delivery and integrity check. Finally, Witanto et al. [47] introduced two types of blockchain-based Open Connectivity Foundation (OCF) firmware update that is direct and peer-to-peer schemes. The first one gets the firmware update from the manufacturer, while the second approach gets it from any nearby gateway. They concluded that the second method is faster than the first one based on evaluating the latency and throughput.

Table 8.1 presents the most related distributed solutions for secure firmware updates to our proposed design to show the contribution of our work. After investigating the available blockchain-based firmware update methods for IoT device frameworks, a gap was identified in the literature and a new treatment architecture was designed to contribute to the research goal. Our blockchain-based secure firmware update solution for IoT devices adds the following features to the existing solutions:

Table 8.1 Comparison between related work and this chapter

Prior work	Year	Disadvantages	This chapter contribution
Firmware verification of embedded devices based on a blockchain [27]	2016	• Used client–server model • Single point of failure, which causes many attacks such as MiTM, DOS attacks, and causes network traffic.	• Used peer-to-peer network model
Toward better availability and accountability for IoT updates by means of a blockchain [7]	2017	• Sends the firmware binary that increases the block size during firmware transaction from the manufacturer.	• Sends only the URL of the firmware binary that saves the block size during the transaction.
Blockchain-based secure firmware update for embedded devices in an Internet-of-Things environment [6]	2017	• Model does not check the integrity of the binary file before downloading it onto the embedded devices. • Provides the firmware update from only one manufacturer • Excessive resource consumption on the low-power IoT devices coming from applying the consensus algorithm that exhausts the IoT devices as they have limited resources. • Did not cover the following features: heterogeneity, authentication, and confidentiality	• The verification of firmware update is done by checking firmware file integrity and its hash to ensure binary file authenticity before downloading it to the IoT devices.
Blockchain-based firmware update framework for Internet-of-Things environment [51]	2018	• Uses public blockchain network Ethereum • Gateway hardware is not part of the blockchain network, rendering it vulnerable.	• Uses permissioned blockchain network Hyperledger to authorize the participants. • Gateway is a node in the blockchain network that connects the vendor to the IoT devices.
An over-the-blockchain firmware update framework for IoT devices [49]	2018	• Did not check if the firmware source is legitimate source or not. • If the gateway gets compromised, the update fails. • A smart contract needs to be sent whenever firmware is sent to the nodes, which causes excessive traffic. • Did not consider the following features: availability, authentication, and confidentiality.	• This chapter uses a permissioned blockchain network and uses signed certificates to check if the source is legitimate.

(continued)

Table 8.1 (continued)

Prior work	Year	Disadvantages	Our paper contribution
An architecture to enable secure firmware updates on a distributed trust IoT network using blockchain [16]	2019	• Did not provide algorithm, did not implement proposed solution. • The components of the block chain network are: firmware author, author application ser ver, vendor distribution server, and payload distribution server.	• Provided a detailed explanation of the solution structure and its implementation. • The blockchain compon ents are vendor, block chain admin, and gateway admin.
Securing over-the-air IoT firmware updates using Blockchain [19]	2019	• Did not cover availability and confidentiality • The firmware was processed in fragments.	• The firmware update link was sent through the blockchain network to save storage and enhance speed.
Private blockchain network for IoT device firmware integrity verification and update [12]	2019	• Did not provide any information about the process of checking the firmware.	• Provided a detailed explanation of the solution structure and its implementation.
Remote configuration of integrated circuit features and firmware management via smart contract [24]	2019	• Verifies the IoT device instead of the firmware update file. • Uses public blockchain that added transaction cost each time the smart contract was executed.	• Verifies the firmware update file and verifies the authenticity of the IoT/ RPi. • Uses permissioned block chain.
Blockchain-based distributed firmware update architecture for IoT devices [11]	2020	• Each IoT needs to verify if it needs a firmware update, which requires a lot of time and energy consumption.	• The comparison is done automatically in the bl ock chain network when a new update is generated from the vendor.

- This chapter solves some of the existing challenges in the four previous designs to enhance the security of the firmware download. One main challenge of the previous works was that the vendor was excluded from the blockchain network, as described in [27]. This challenge caused serious security issues in the connection between the vendor and the blockchain network. Another challenge was that the gateway, which is connected to the IoT devices, was excluded from the blockchain network, as described in [51]. This challenge caused serious security issues concerning the connection between the blockchain and the gateway. Our design architecture includes both the vendor and the gateway as nodes in the permissioned blockchain network.

- Our architecture is designed to securely check the firmware version and validate the authenticity of the firmware file to avoid any malware injection in the binary firmware file through the update. The validation of the firmware update file is done more securely than those proposed in the literature.
- This chapter is the first implementation of a secure firmware download system for IoT using a blockchain network, as all previous papers have been theoretical.

8.4 Methodology and System Architecture

The research methodology of this chapter is based on design science research methodology. Design science is defined by Wieringa as “The design and investigation of artifacts in context. The artifacts are studied and designed to interact with a problem context to improve something in that context” [46]. Design science is adopted to investigate the security of firmware updates in IoT devices through blockchain and to design improvements to current procedures. This chapter aims to solve a design problem, which is the question of how to implement a secure solution to protect the firmware update process. It is for this reason that the design cycle is followed in detail, with steps that are related to this research. Four steps are discussed in detail in this section: problem investigation, treatment design, treatment validation, and treatment implementation.

8.4.1 Problem Investigation

The problem defined in this research is the insecure firmware update process currently used in IoT devices. This process may have the negative effects of divulging private user information to the attacker or allowing the attacker to gain administrative privileges. According to the Open Web Application Security Project (OWASP), insecure firmware/software is reported as one of the top ten IoT vulnerabilities, as shown in Fig. 8.4 [42]. Insecure firmware update processes in IoT devices can have a variety of causes:

- Insecure communication used for downloading the firmware update
- Firmware update code modified by inserting a malware
- IoT not updated to the most recent version; the old version containing bugs and vulnerabilities

The goal is to provide a trusted network where it is safe to download the latest firmware version to the connected IoT devices without any modification. That is why blockchain technology was chosen as the secure network for firmware download on IoT devices.

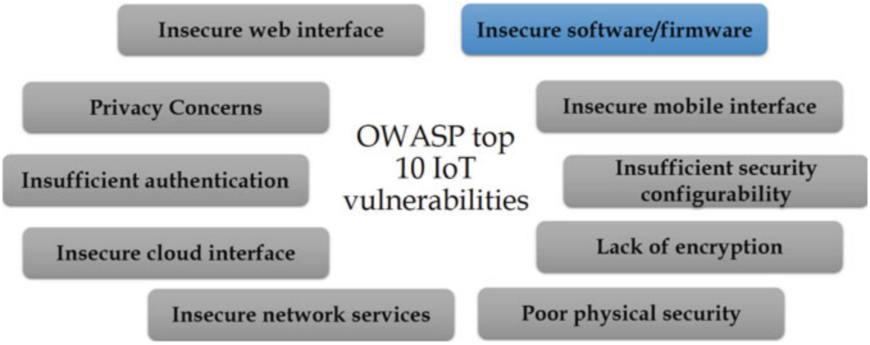

Insecure web interface
Insecure software/firmware
Privacy Concerns
Insecure mobile interface
OWASP top 10 IoT vulnerabilities
Insufficient authentication
Insufficient security configurability
Insecure cloud interface
Lack of encryption
Insecure network services
Poor physical security

Fig. 8.4 OWASP top 10 IoT vulnerabilities

8.4.2 Treatment Design

In this step, the available framework for the blockchain-based firmware update process for IoT devices is investigated to identify gaps in the literature. Moreover, a new architecture is developed that contributes to the goal of securing the update process. The investigation steps are done in the literature review section above. In the present section, the treatment design of this chapter will be explained. The roles of each entity in the architecture for the blockchain-based firmware update process for IoT devices are identified. Furthermore, the objectives and contributions of this chapter are clarified.

The proposed design contains a firmware provider that alerts the blockchain network nodes connected to the IoT devices whenever there is a new version of the firmware. The response received from the blockchain node indicates whether or not the firmware on the IoT device is up to date. If the firmware version is not the latest, the firmware provider securely downloads the new version of the firmware to the device. This is done using a peer-to-peer network, after verifying the authenticity of the firmware file to make sure that no malicious code has been added. Otherwise, if the firmware on the IoT device is already up to date, no action is taken.

8.4.2.1 Proposed Design Objectives

The objective of this research is to find defensive techniques for securing IoT devices. Specifically, the goal of this effort is to investigate common vulnerabilities in IoT devices to counterfeit firmware updates. The achievement of this goal delivers valuable information about the feasibility, technical requirements, and implementation characteristics of secure firmware updates in IoT devices. The main role of this chapter is to implement a blockchain-based architecture for IoT that delivers secure, lightweight, and decentralized firmware updates.

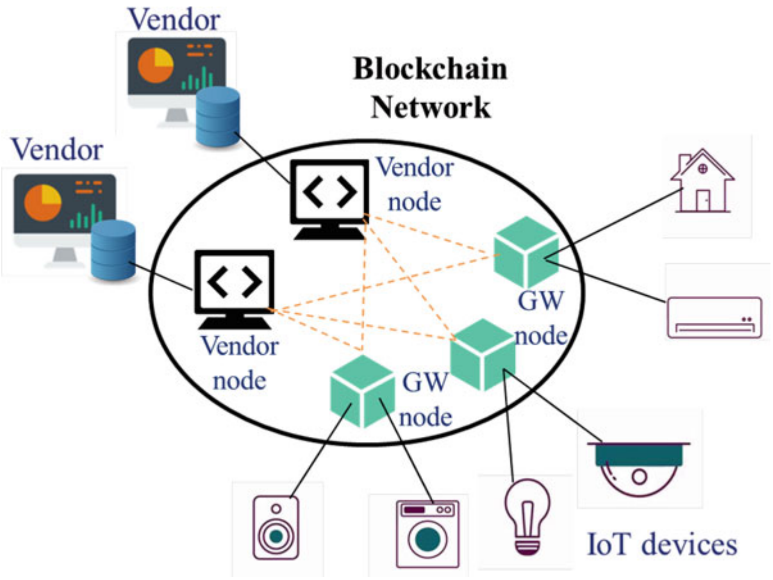


Fig. 8.5 The proposed blockchain-based secure firmware update for IoT devices

8.4.2.2 Proposed Methodology

In light of the proposed blockchain-based infrastructure to secure firmware updates for IoT devices, Fig. 8.5 demonstrates the blockchain node participants. The IoT devices are connected to the blockchain through an additional gateway, which is a node in the blockchain network. The vendor providing the firmware update is also a node in the blockchain. A blockchain administrator is needed to manage the IoT information.

The proposed design delivers an original and authentic firmware file. The encrypted firmware update link is sent in the blockchain network from the vendor repository to the gateway blockchain node, which connects to the IoT devices by performing a transaction in the blockchain network. In the design, the device manufacturer distributes information about the new version of the binary firmware file to the blockchain network. The proposed architecture has the following four entities (the first two entities are considered to be the same party):

- Manufacturer: A firmware repository owned by the vendor of the device. The manufacturer has the original binary firmware files of the IoT devices. It is responsible for sending information about the new version of the firmware update to the blockchain network.
- Vendor: A node in the blockchain network that is connected to the real manufacturer of the device. The vendor is responsible for sending transactions for the updated firmware binary information and the original file to other nodes in the network.

Table 8.2 Blockchain participants

Participant	Roles
Blockchain Administrator	1. Creates users (vendor, gateway, and IoTs) 2. Manages privacy policies
Blockchain Gateway Administrator	1. Checks for firmware updates 2. Checks firmware integrity 3. Downloads the latest version of firmware update 4. Verifies the data of connected IoTs (type, firmware version)
Vendor/Manufacturer	1. Interacts with blockchain network to secure firmware update 2. Provides updated firmware 3. Stores previous firmware as backup if the update fails

- Gateway: A node in the blockchain network responsible for connecting IoT devices to the blockchain network. The connected IoT devices are registered on the gateway blockchain node, which collects data about the registered IoT devices including device manufacturer, model, and current firmware version. The gateway node also receives notifications of new firmware update transactions and executes firmware updates from the vendor nodes.
- IoT: Embedded devices.

Some assumptions are made in the proposed architecture. Firstly, the manufacturer is assumed to be connected to the vendor node in a secure channel. Secondly, for each IoT device type, a corresponding vendor node is assumed to exist in the blockchain network, which stores firmware files and version information. Lastly, the connection between gateway blockchain node and IoT devices is assumed to use HTTPS. To securely deliver the firmware update file, the roles for each participant in the blockchain network are explained in Table 8.2.

The proposed method for the blockchain-based firmware update process for IoT devices supports confidentiality, integrity, and availability of the firmware update file. This is because the data is sent encrypted and signed by the private key of the authorized parties (the vendors), thereby ensuring the confidentiality of the data and protecting devices from impersonation attacks. Moreover, the firmware update link is sent through secure channels that are visible by selected entities. The consensus mechanism in the blockchain network guarantees the integrity of the data sent (i.e., the firmware update link), protecting it from modification attacks. Furthermore, the decentralized architecture of blockchain protects vendors from DOS attacks because the network is always available [34].

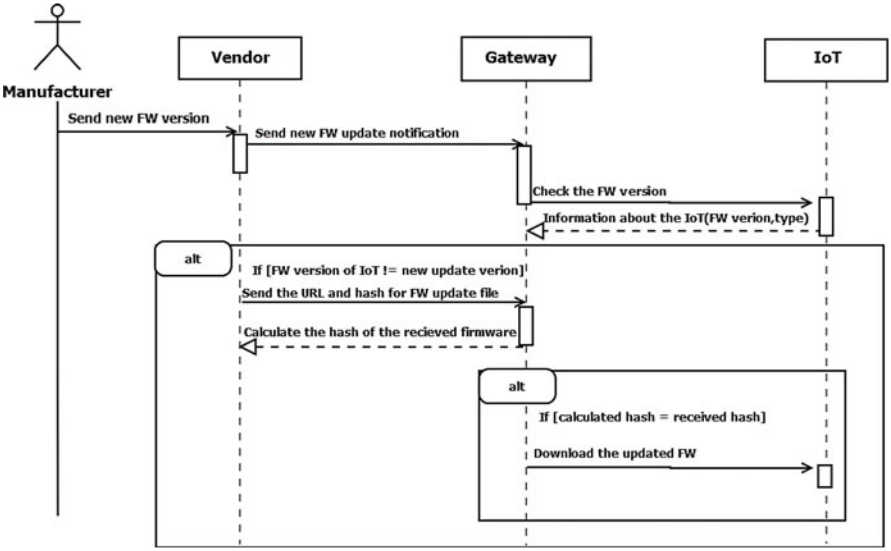

Manufacturer
Vendor
Gateway
IoT
Send new FW version
Send new FW update notification
Check the FW version
Information about the IoT(FW verion,type)
alt
If [FW version of IoT != new update verion]
Send the URL and hash for FW update file
Calculate the hash of the recieved firmware
alt
If [calculated hash = received hash]
Download the updated FW

Fig. 8.6 Sequence diagram of the firmware download

8.4.2.3 Proposed Design Procedure

This chapter examines Hyperledger blockchain technology and discusses how it can be utilized to implement protocols for secure firmware updates delivered to IoT devices using chaincode. The proposed solution consists of a chaincode that passes sensitive data from one blockchain node (vendor) to another blockchain node (a gateway that is connected to IoT devices). The firmware update process in the blockchain network is demonstrated in detail in the UML sequence diagram in Fig. 8.6.

The manufacturer and the vendors are the same, but the manufacturer is a physical entity, while the vendor is a node in the blockchain network. Communication between the vendor and the gateway takes place inside the blockchain network. Furthermore, all transactions in this communication link are encrypted and signed in the Hyperledger Fabric network. This encryption guarantees the confidentiality of the chaincode used to send firmware updates from a vendor to a gateway.

In Hyperledger Fabric, channels are used to privately share data with a subset of network members. Only members of the channel can store and have access to the data. Moreover, it is possible to encrypt the data on-chain such that the chaincode is still able to see the data and apply logic to it. The idea is to transmit the encryption key as part of a transient input parameter (which is not transmitted to the ordering service and another channel node). Therefore, only the endorsers and nodes that can receive the key can decrypt the data.

The data transmitted through Hyperledger Fabric can also be encrypted by Transport Layer Security (TLS) encryption in the REST server between the GUI application and the Hyperledger Fabric network. TLS is a cryptographic protocol that provides networks with secure end-to-end communications and is widely applied for online transactions. It is used to prevent eavesdropping, tampering, and message forgery. TLS is the successor protocol to Secure Sockets Layer (SSL). TLS is the newer version of SSL. It protects data transfer using encryption similar to the SSL. The application may encrypt the data before submitting the data to the blockchain [56]. Moreover, the communication between the gateway blockchain node and the IoT/ RPi uses HTTPS encryption in real-life scenarios, which makes this connection secure.

The proposed solution of the firmware update process for IoT devices (RPi in this chapter) through the blockchain network is as follows:

1. The manufacturer develops a new firmware binary file for a specific IoT device and sends it to the vendor node.
2. The vendor node in the Hyperledger blockchain stores the information of the new update file version and the hash value of the firmware file and deploys the firmware update file in the blockchain network with the hash value.
3. The vendor sends a notification about the newly released firmware update to the IoT gateway blockchain node by using a transaction.
4. The IoT gateway blockchain node checks the information of the newly released version in the notification. The IoT gateway blockchain node executes the chaincode that compares the new version with the firmware version of the registered IoT connected to the corresponding gateway blockchain node.
5. The gateway blockchain node receives the URL, hash, and requests for the firmware binary from the vendor repository. The gateway blockchain node compares the version of the updated firmware file with the version of the firmware file in the Raspberry Pi (IoT). If they are the same version, no changes are required. However, if the received version is newer than the one in the IoT, then the gateway blockchain node calculates the hash value of the received firmware binary file by using SHA-256. This is done to ensure the integrity of the firmware and to confirm that it has not been modified. If the calculated hash of the updated firmware binary is the same as the hash sent by the vendor, then the firmware file is correct and no modification has occurred during the delivery of the new firmware.
6. The firmware update file is downloaded from the gateway blockchain node to the IoT/RPi if the gateway blockchain node has the same authentication information (username, password) as the one in the IoT device/RPi. If the password is not the same, the firmware update file is not downloaded to the IoT/RPi.
7. A server on the RPi keeps listening for new firmware updates from the gateway blockchain node.

8.4.3 Treatment Validation

Treatment validation is an important step in design science methodology, as it shows whether or not the proposed design contributes to stakeholder goals and satisfies their requirements. Treatment validation can be done by one or both of the following two methods: validation by expert opinion or validation by comparison between the proposed architecture and the existing models.

To validate the obtained results, an expert opinion was asked of the Dubai Electronic Security Center (DESC) about the design of the blockchain model and the security requirements for the firmware update process in the blockchain. After discussing it in several meetings with DESC, they agreed to implement the proposed design and to deploy it in a scenario for smart government services. It was also compared with other related works to show the differences between the proposed architecture and the various designs presented in Sect. 8.2.

8.4.4 Treatment Implementation

In this section, the design of the blockchain-based firmware update process for IoT devices was evaluated to ensure compatibility between the IoT technology and the blockchain technology. Also, it was essential to ensure that the firmware file could be downloaded from the vendor to the gateway blockchain node through the blockchain, and then from the gateway blockchain node to the connected IoT device. Performing this setup configuration was the starting point for building the proposed architecture for a blockchain-based firmware update process for IoT devices.

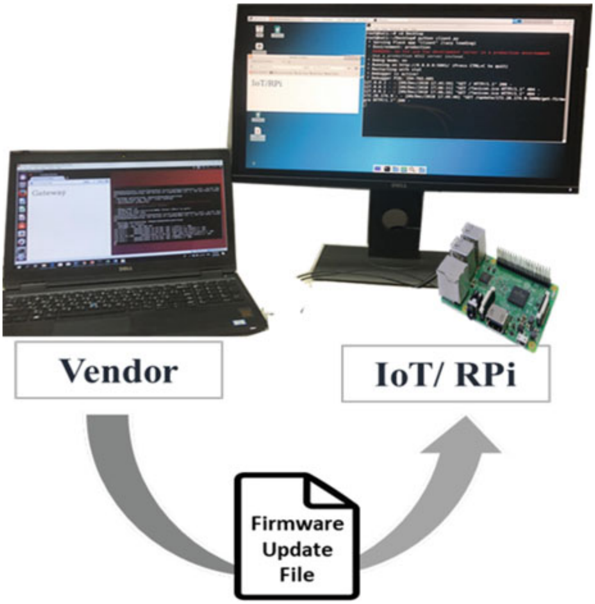
8.4.4.1 Hardware Setup

Both the vendor and the gateway are added as nodes in the blockchain network. The vendor is the server that stores the firmware images and runs on the Linux Ubuntu Operating System (OS). The IoT device is simulated by a Raspberry Pi. The blockchain network provides a secure environment for the firmware update. The firmware image is sent from the vendor to the gateway blockchain node and then delivered to the IoT through an HTTPS connection if needed. The final hardware structure of the proposed architecture is illustrated in Fig. 8.7.

8.4.4.2 Software Setup

The blockchain environment was built so that the bottom-most layer is composed of the set of runtimes supported by Hyperledger Fabric [37]. In the middle tier, there is a Composer REST server that connects to the backend of the supported

Fig. 8.7 Hardware structure



runtime by way of the Composer SDK. The presentation layer has an Angular application that connects to the Composer REST server by way of REST API calls [8]. A Hyperledger Fabric development environment can be set up in different ways depending on the operating system. In the implementation of the proposed work, the device used for our purposes ran on Ubuntu OS. As a result, certain requirements were downloaded to install Hyperledger Fabric development tools on a local Ubuntu machine. To run Hyperledger Composer and Hyperledger Fabric, it is recommended that the device has at least 4 Gb of memory. The following are prerequisites that are necessary to install the required development tools [36]:

- Operating Systems: Ubuntu Linux
- Docker Engine: Version 17.03 or higher
- Docker-Compose: Version 1.8 or higher
- Node: 8.9
- npm: v5.x
- git: 2.9.x or higher
- Python: 2.7.x
- A code editor, in this chapter, VSCode was used

8.4.4.3 Hyperledger Fabric Development Environment Setup

The development environment setup is configured for only one organization in this chapter, called org1. The docker-compose file defines multiple docker containers that make up the infrastructure components for org1 [25]. Four docker containers are initiated or created as part of the development environment:

- Certification Authority container CA-MSP ca.org1.example.com
- Orderer container orderer.example.com
- Peer container peer0.org1.example.com (this container depends on CouchDB for state data)
- CouchDB

The configuration for all these containers is managed in two folders, Crypto and Config, where Crypto has the credential data and Config has the configuration files for all of these containers. The development environment components are shown in Fig. 8.8. The peer is configured to use level DB for transaction data, and it is set up to use an instance of CouchDB for state management.

Multiple utility scripts are available for managing the development environment. Two scripts are used to start and stop the Fabric environment. As a developer of business network applications, interaction with the dev environment takes place through executing scripts. These scripts are available for launching, shutting down, or restarting the dev environment. The Composer CLI tool was used to create the business network archive files, and then the Composer tool was used to deploy the BNA to the Fabric runtime as shown in Fig. 8.9.

8.4.4.4 Composer Playground Testing

Composer Playground is a web application that allows us to deploy, edit, and test business network applications. It is available online at this URL (https://composer-playground.mybluemix.net), or it can be deployed locally on the machine or in a server-based environment. The user of Composer Playground can define the model using the UI. The created model is then stored in the local browser storage. Moreover, users can test the model that they are working on. The define tab contains the model, script, access control, and query files. In the test tab, the user can create different types of entities that are defined in the model, such as a new participant. Furthermore, the user can see all transactions in the log and submit new transactions.

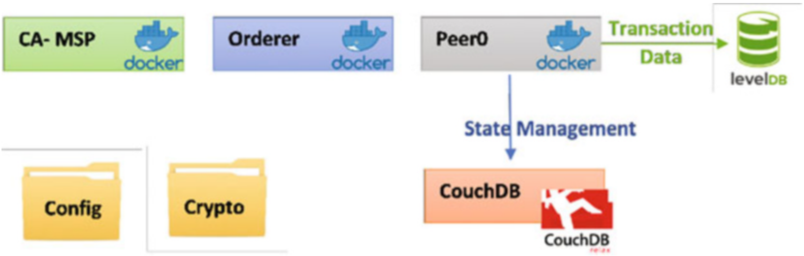


Fig. 8.8 Development environment

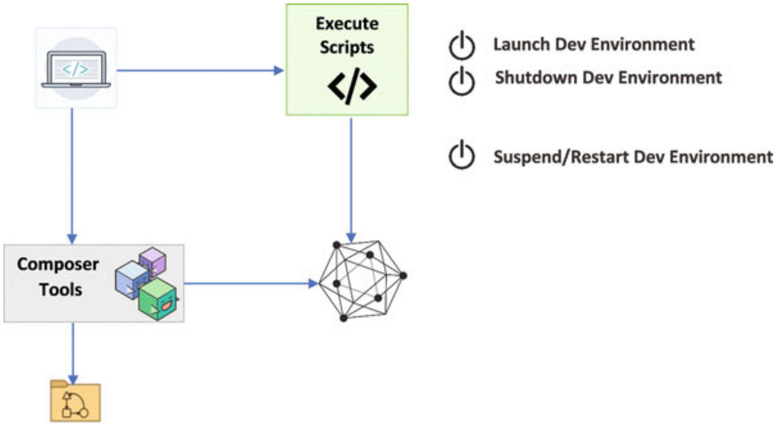


Fig. 8.9 Development environment scripts

Composer Playground also validates all transactions submitted and applies all rules that are specified in the network definition.

The playground can also assist the developer with the coding part of the model. The developer can code the transactions and update the model using the Playground UI. Furthermore, developers can download the model itself (business network application archive) to their local storage by using the export feature of Composer Playground. This archive can be stored in the source code control system. Developers can make changes to the model and the code and then use the deploy feature to put it back on Composer Playground for testing using the import feature.

In this chapter, Composer Playground was not used to define the built business network application; instead, the application was developed locally in JavaScript using Visual Studio Code, as it is an easier environment in which to work and has more features. Composer Playground was subsequently used for simulated testing. The procedure followed in Composer Playground was as follows: Firstly, visual code was used to code the model and the transaction logic. Next, the tools available in the Composer CLI were used to create the application archive. The model and the deployed code were subsequently tested on the Composer Playground by dropping the business network application archive and then using Composer Playground features for testing.

After explaining the environment used to write the chaincode of the BNA, the construction and coding of the BNA and its components are explained in the next subsection. The BNA cards used to connect and access our application are also discussed.

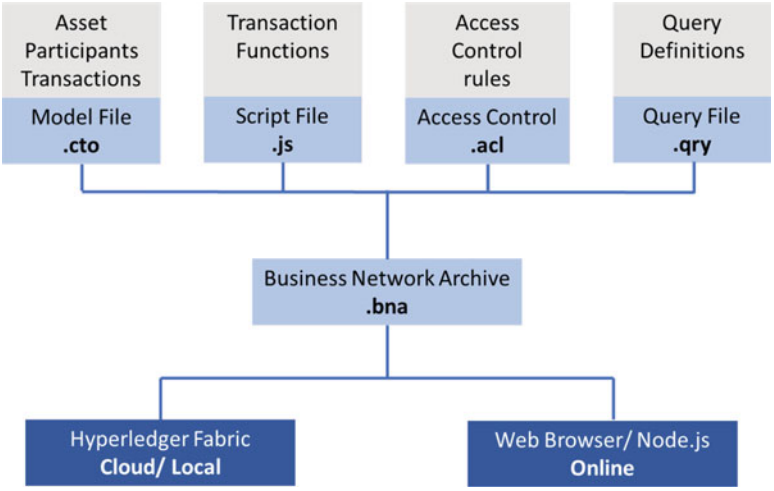


Fig. 8.10 Hyperledger composer structure

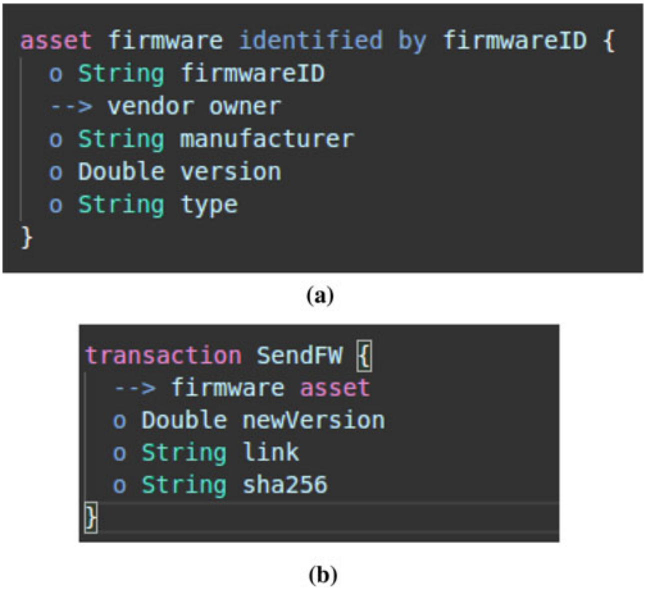
8.4.4.5 Business Network Application Development and Administration

The development workflow of the chaincode is illustrated in Fig. 8.10. The developer writes the chaincode in Java, Golang, or NodeJS and then uses common software practices to iteratively compile and test the code until they are satisfied that the code is working in the way they want. In the next step, the developer deploys the chaincode to the peer using a deployment transaction. The developer can also put together the endorsement policy for the chaincode in the deployment process. The deployment transaction is then propagated to the network. Once the deployment transaction is successful, the transaction log and state data are updated. Participants in the network can use applications to invoke the chaincode, which is recorded in the transaction log. All state changes are recorded in the state database.

A NodeJS environment was used to create the business network application. All chaincodes that were developed in this chapter were written in JavaScript. The organization namespace is defined in the (.cto) file, and all resources created are items in this file. Hyperledger Composer resources consist of assets, participants, and transactions [45].

The scripts may contain transaction processor functions that process the transactions defined in the Business Network Definition model files. Transaction processor functions are automatically called by the runtime environment when transactions are submitted using the Business Network Connection API.

Hyperledger Composer includes an access control list (ACL) that provides the rules that specify which participants are allowed to create, read, update, or delete components in a domain model. There are two parts of the chaincode: assets and transactions. An asset is a digital representation of the asset, where the record of ownership is saved in the blockchain network. Our asset is composed of firmware



```
asset firmware identified by firmwareID {
  o String firmwareID
  --> vendor owner
  o String manufacturer
  o Double version
  o String type
}
```

(a)

```
transaction SendFW {
  --> firmware asset
  o Double newVersion
  o String link
  o String sha256
}
```

(b)

Fig. 8.11 Chaincode components. (**a**) Asset definitions. (**b**) Transaction definitions | implementation

that has the five attributes shown in Fig. 8.11a. Transactions create the asset and manage the state of the assets. The code for managing the state of the firmware is written in JavaScript, as illustrated in Fig. 8.11b. The transaction code implements the business logic and carries out typical operations on various assets defined in the business network application. These operations are create, retrieve, update, and delete.

The confidentiality of the chaincode used to send firmware updates and verify the state of the firmware is ensured by symmetric-key encryption of transactions and states. This function is enabled by a blockchain-specific key with an enrollment certificate accessible to all peers in the blockchain network. As a result, the encrypted firmware update is sent through the blockchain system.

Users of the Hyperledger Fabric network can take actions based on their roles in the business network application (BNA). The business network cards contain the configuration information needed by the tools and applications to connect to the business network applications and the Hyperledger Fabric infrastructure components. There are two administration roles for the BNAs:

- Peer Administrator: This user is responsible for activities at the infrastructure level or the node level. The peer administrator is created as part of the environment setup.
- Network Administrator: This user is responsible for activities at the application level. A network administrator is created by the peer administrator.

For this to work, the application needs to have access to what is known as the business network card. The card contains credentials, keys, and certificates, as well as a connection profile. A user can have multiple cards set up on their machine to connect to different business networks. The connection profile holds the information needed to reach the CA, the peers, and the orderers (via a URL for each of these components).

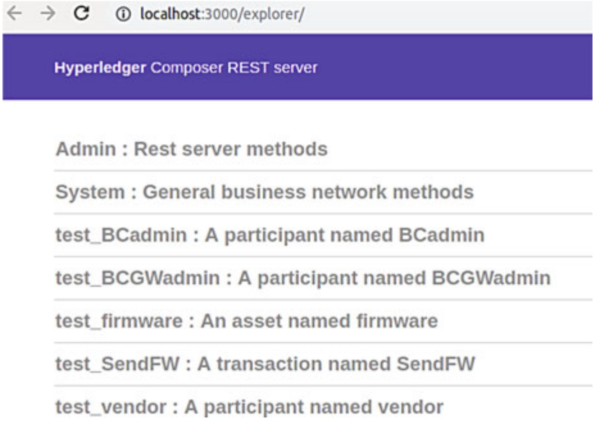
8.4.4.6 REST Server Setup

The RESTful API is a set of constraints designed to use specific HTTP verbs for operating on resources, depending on the type of the operation [22]. These REST APIs are GET, PUT, POST, and DELETE. POST is used to create a new resource. GET is used to read the resource state used by the business domain model. PUT is used to update the state of the resource, while DELETE is used to delete a resource. Together these operations are referred to as the CRUD operations.

The REST server process takes place before a transaction reaches the Fabric network application. The connection configuration is part of the REST server setup that enables the REST server to connect with the BNA. Developers can request the URL exposed by the REST server to get information about the API and also to try out the API and test it. Applications can connect directly to the REST server instead of connecting to Hyperledger Fabric. The benefit of that is that the application developer will not need to depend on any specific Hyperledger Fabric library, making the code simpler and more maintainable. Therefore, the application can invoke current operations and model transactions through the API with this architecture.

Before launching the Composer REST server using the composer-rest-server command, Hyperledger Fabric needs to be running and the BNA should be available to the REST server. After ensuring the availability of the Hyperledger network and BNA, the REST server can be launched by providing the card name and namespaces, refusing any other parameters. The Composer REST server listens on port 3000 by default. After the REST server is launched, it will become available as shown in Fig. 8.12. The sample model has multiple resources: sample participants, sample assets, and sample transactions. In this chapter, the model contains the participants: blockchain admin “BCadmin,” blockchain gateway admin “BCGWadmin,” and vendor. The asset is the firmware update file, while the transaction is the send firmware “sendFW” function as shown in Fig. 8.12. Under each entity are multiple API endpoints, as well as the HTTP verbs that apply to those URLs. Essentially, these are the CRUD operations that can be carried out on the asset firmware in our work. For example, the POST API can be used to create a firmware asset after defining the asset values. At this point, the REST server is connected to the network application. The response body will indicate whether or not the POST was successful (response code=200). Additionally, users can submit transactions through the POST API, which will change the asset value.

The main advantages of using the REST server are [36]:



localhost:3000/explorer/

Hyperledger Composer REST server

Admin : Rest server methods

System : General business network methods

test_BCadmin : A participant named BCadmin

test_BCGWadmin : A participant named BCGWadmin

test_firmware : An asset named firmware

test_SendFW : A transaction named SendFW

test_vendor : A participant named vendor

Fig. 8.12 Rest server

- Easy access to network resource and transaction specifications.
- The developer can use the REST server interface for:
  - Exploration
  - Testing
  - Observing transactions executed against the business network application
- Client application code will be simpler to write and maintain as the application will only have to invoke the REST API exposed by the REST server instead of using libraries specific to Hyperledger Fabric.

In summary, the REST server exposes business network apps as RESTful APIs, enabling developers to carry out CRUD operations on these APIs and giving them the ability to execute transactions [55]. The REST server is set up with a common card that is used to connect to the Hyperledger Fabric backend. A single card representing a single identity is used by the REST server to execute all transactions.

8.4.4.7 Developing GUI in Angular Application

Angular is a typical solution architecture that was suggested by the developers of the Composer framework. The Yeoman Hyperledger Composer generator is used to create a skeleton Angular application that developers can customize for their own needs. The Composer development team proposes the adoption of a three-year solution architecture as shown in Fig. 8.13.

In this typical solution architecture, the bottom-most layer is the set of supported runtimes. In the mid-tier, there is a Composer REST server that connects to the backend of the supported runtime by way of the Composer SDK. The presentation layer has an Angular application that connects to the Composer REST server by way of REST API calls [9].

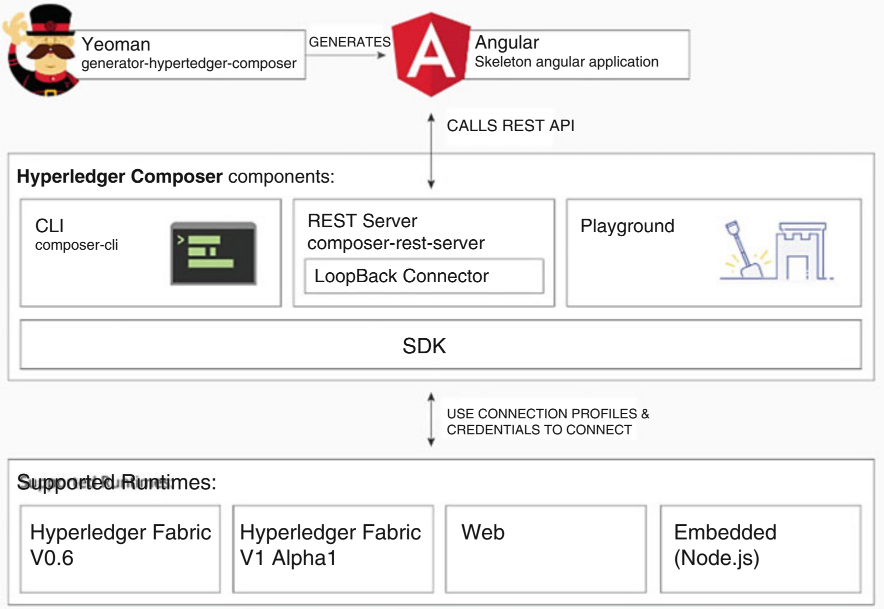


Fig. 8.13 Typical Hyperledger solution architecture [15]

There are multiple alternative implementation options for implementing the frontend of the Hyperledger Fabric application:

- Use any frontend JavaScript framework.
- Create a desktop application that uses Composer SDK to connect to the REST server or directly to the Fabric backend, eliminating the mid-tier for security reasons.
- Integrate with an existing application by way of the REST server or by connecting directly to the Fabric runtime.

For this chapter, it is necessary to decide whether to adopt the typical architecture solution or not. To help make that decision, the pros and cons of using the typical solution architecture are listed in Table 8.3. Due to the great advantages of using the Yeoman generator for the skeleton application, it is used in this chapter as the frontend application.

8.5 Experimental Results

This section describes the experimental testing of this chapter, from the reception of a new firmware update from the manufacturer to the delivery of the updated firmware file to the IoT device, trying both authentic and modified files and

Table 8.3 Yeoman generator skeleton application's pros and cons

Pros	Cons
• Quickly set up a frontend application using the Yeoman generator. • Consistent code standardization. Developers will be able to understand each other's code quickly because it is based on that same baseline implementation. • Testing. Yeoman can be used to test the code without the need to build an application on top of the skeleton app.	• Ensuring the security of the Composer REST server is not a trivial task.

monitoring the response of the developed chaincode. The design of the blockchain-based firmware update process for IoT devices is evaluated to ensure compatibility of the IoT device and the blockchain. We also had to verify that the firmware file could be downloaded from the vendor to the gateway blockchain node through blockchain, and from the gateway blockchain node to the connected IoT device.

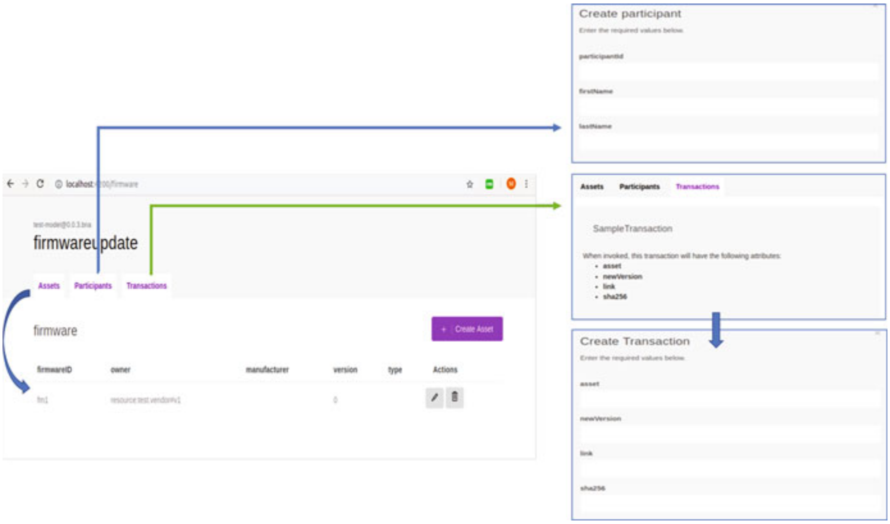
Furthermore, this section discusses the security and privacy of the blockchain-based secure firmware update process for IoT devices. The effectiveness of our solution is analyzed to prevent several critical security attacks that are particularly relevant for our work such as man-in-the-middle attacks, impersonation attacks, and denial-of-service attacks. Moreover, the performance evaluation of the proposed scheme is explained at the end of this section.

8.5.1 System Evaluation

Figure 8.14 shows the GUI of the firmware update application in port 4200. The process for creating participants such as blockchain admin, gateway admin, and vendor, and for creating and invoking transactions to send firmware updates from the vendor to the RPi/IoT device through the blockchain network is also demonstrated in Fig. 8.14. The GUI is produced as a result of performing the following steps:

- Execute shellcode to launch the Fabric Runtime environment.
- Deploy the business network application test model and the admin card generated as part of the deployment process.
- Launch the Composer REST server with the correct admin card.
- Test using Composer Playground.
- Execute this command every time the Angular application is opened.

The next step is to test the connection between the blockchain network and the IoT device. The firmware file is downloaded from the gateway blockchain node to the IoT device (Raspberry Pi). A server on the RPi is constantly listening for a new firmware update from the gateway blockchain node. If the version sent from the vendor to the gateway blockchain node is newer than the one currently used by the



firmwareID	owner	manufacturer	version	type	Actions
fm1	resource:test.vendor#v1		0		

Enter the required values below.

participantId

firstName

lastName

When invoked, this transaction will have the following attributes:

- asset
- newVersion
- link
- sha256

Enter the required values below.

asset

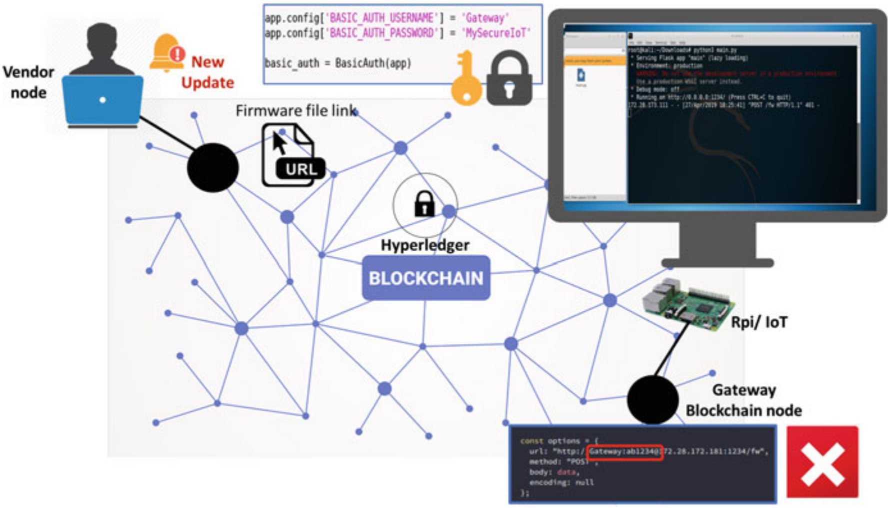
newVersion

link

sha256

Fig. 8.14 GUI of the firmware update application

IoT, the gateway blockchain node ensures the integrity of the firmware and confirms that it has not been modified by calculating the hash (SHA-256). If the calculated hash of the updated firmware binary is the same as the hash sent by the vendor, then the firmware file is correct and no modification has occurred during the delivery of the new firmware. If all conditions are satisfied and the gateway blockchain node shares the same password as the IoT device, the firmware binary is sent directly from the gateway blockchain node to the IoT/RPi. Figure 8.15 illustrates that the firmware file has not been sent from the gateway blockchain node to the IoT/ RPi because the gateway blockchain node password does not match the password of the IoT device. However, if the passwords match, that indicates that the gateway sending the firmware update file is authentic and legal. As a result, it is downloaded to the IoT/RPi. Firmware update files are sent in an isolated way from the gateway blockchain node to the IoT/RPi in the local network. Furthermore, HTTP basic authentication is used to prevent attackers from impersonating the gateway and sending fake firmware files to IoT devices to expose privacy information. In real-life scenarios, the connection uses HTTPS. As the firmware update file and credential information that is sent will be encrypted using HTTPS, the IoT devices will be protected from impersonation attacks and man-in-the-middle attacks.



```
app.config['BASIC_AUTH_USERNAME'] = 'Gateway'
app.config['BASIC_AUTH_PASSWORD'] = 'MySecureIoT'

basic_auth = BasicAuth(app)
```

```
root@kali:~/Downloads# python3 main.py
 * Serving Flask app "main" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:1234/ (Press CTRL+C to quit)
172.28.173.111 - - [27/Apr/2019 18:25:41] "POST /fw HTTP/1.1" 401 -
```

```
const options = {
  url: "http://Gateway:ab1234@172.28.172.181:1234/fw",
  method: "POST",
  body: data,
  encoding: null
};
```

Fig. 8.15 Sending firmware file from gateway blockchain node to IoT/RPi

8.5.2 Security and Performance Analysis

Any secure architecture needs to meet three main security requirements that are confidentiality, integrity, and availability. This is known as the CIA Triad, which is a distinguished model that defines the policies of information security [44]:

- Data confidentiality ensures the privacy of user information by using various procedures such as data encryption. This prevents the improper disclosure of information to any unauthorized entity. In the IoT architecture, confidentiality ensures that the sensors protect the collected data from other nodes and do not reveal information to unauthorized readers [17].
- Data integrity ensures that no information modification can occur during communication between parties. It, therefore, ensures the accuracy and trustworthiness of data by using various mechanisms such as the error detection algorithms Checksum and Cyclic Redundancy Check (CRC). The continuous syncing of data for backup purposes like version control keeps a record of any changes that occur to a given file in the system. In the IoT architecture, version control ensures the integrity of information in IoT devices [17].
- Data availability ensures the availability of data to authorized users whenever they want to access it. It is important to place a firewall in the network to avoid certain attacks such as DOS attacks, which make data unavailable to the end user. Redundancy and failover backup methods provide duplication of information in the system in the case of system failure or conflict. As a result, the reliability and availability of required information are ensured [17].

As a permissioned blockchain network was used, all vendors were added manually and assumed to be non-malicious. The firmware download link, hash, and metadata were sent through the blockchain, thereby guaranteeing their validity [48]. The only way for a firmware update to become invalid was for some kind of MITM attack. However, since the link and hash received remain valid, any modification to the firmware is easily detected and modified firmware can be rejected simply by comparing hashes. Even if the download takes place through HTTP, this method detects attacks. By using the properties of the blockchain, a secure transfer of firmware files can be guaranteed even on vulnerable channels and with computationally economical methods.

Our proposed method for a blockchain-based firmware update process for IoT devices supports the confidentiality, integrity, and availability of the firmware update files, as the data are sent encrypted and signed by the private key of the authorized parties (the vendors), ensuring that the shared data is confidential and protecting it from impersonation attacks. Moreover, the firmware update link is sent through secure channels that are visible by selected entities. The consensus mechanism in the blockchain network ensures the integrity of the transmitted data (the firmware update link) and protects it from modification attacks. Furthermore, the decentralized architecture ensures continuous availability, protecting the vendors from DOS attacks.

The performance of Hyperledger Fabric depends on the number of peers in the network and the number of transactions. The more peers using the platform, the worse its performance is. For the proposed design, only one peer was used. Hyperledger requires all peers to exchange messages, thereby limiting the maximum number of peers in the network. Hyperledger Fabric can have a maximum of 26 nodes due to communication overhead. As the number of transactions increases, the performance and throughput of the platform will decrease, while latency and execution time will increase. For a single peer, the Hyperledger Fabric platform can handle up to 20,000 simultaneous transactions. For multiple peers, the number of concurrent transactions that the platform can handle depends on the number of nodes in the network. Hyperledger Fabric cannot handle 10,000 concurrent transactions when there are more than six nodes in the network [31].

8.6 Conclusion and Future Work

In this chapter, our objective is to explore how Hyperledger blockchain technology can be utilized to implement protocols enabling secure firmware updates to be delivered to IoT devices using chaincode. Our solution consists of a chaincode that transmits sensitive data from one blockchain node (vendor) to another blockchain node (gateway connected to IoT devices). Hyperledger Fabric is incorporated because it can be deployed as a permissioned network to prevent unauthorized users from gaining access to the network. The main contribution of this chapter is to implement a blockchain-based architecture for IoT devices that delivers lightweight and decentralized firmware updates by allowing the platform to securely download the latest firmware for the IoT devices and to validate the integrity and authenticity of the downloaded firmware files. Our ongoing work explores enhancements to the process discussed in the paper, including the use of different IoT devices such as an IP camera and a smart bulb. In this ongoing work, firmware updates are delivered to these IoT devices in real-life scenarios. Moreover, the firmware will be downloaded to the IoT devices securely using an over-the-air (OTA) update in real-life scenarios.

Acknowledgments Thanks to the University of Sharjah, Dubai Electronic Security Center (DESC), and OpenUAE Research and Development Group for supporting this chapter.

References

1. A. Anastasiou, P. Christodoulou, K. Christodoulou, V. Vassiliou, Z. Zinonos, IoT device firmware update over LoRA: The blockchain solution, in *Proceedings of the 2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS)* (IEEE, New York, 2020), pp. 404–411
2. E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, et al. Hyperledger Fabric: a distributed operating system for permissioned blockchains, in *Proceedings of the Thirteenth EuroSys Conference* (2018), pp. 1–15
3. M. Bettayeb, Q. Nasir, M.A. Talib, Firmware update attacks and security for iot devices: Survey, *In: Proceedings of the ArabWIC 6th Annual International Conference Research Track*, (2019), pp. 1–6
4. M. Bettayeb, Q. Nasir, M.A. Talib, Hyperledger-based secure firmware update delivery for IoT devices, in *ArabWIC 2021: The 7th Annual International Conference on Arab Women in Computing in Conjunction with the 2nd Forum of Women in Research, August 25–26, 2021* (ACM, Sharjah, 2021), 8p. https://doi.org/10.1145/3485557.3485567
5. T.Q. Ban, B.N. Anh, N.T. Son, T.V. Dinh, Survey of Hyperledger blockchain frameworks: Case study in FPT University's cryptocurrency wallets, in *Proceedings of the 2019 8th International Conference on Software and Computer Applications* (2019), pp. 472–480
6. L. Boohyung, L. Jong-hyouk, Blockchain-based secure firmware update for embedded devices in an Internet of Things environment. J. Supercomput. **73**(3), 1152–1167 (2017)
7. A. Boudguiga, N. Bouzerna, L. Granboulan, A. Olivereau, F. Quesnel, A. Roger, R. Sirdey, Towards better availability and accountability for IoT updates by means of a blockchain, in *Proceedings of the 2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)* (IEEE, New York, 2017), pp. 50–58
8. C. Cachin, Architecture of the Hyperledger Blockchain Fabric, in *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, vol. 310 (2016), p. 4
9. C. Cachin et al., Architecture of the Hyperledger blockchain fabric, in *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, vol. 310 (2016)
10. C.-L. Chen, T.-T. Yang, C.-L. Fan, K.-H. Wang, A Secure and with a low cost updated information system in VANET, in *Proceedings of the International Conference on Applied System Innovation (ICASI)*, vol. 1 (IEEE, New York, 2016), pp. 1–5
11. S. Choi, J.-H. Lee, Blockchain-based distributed firmware update architecture for IoT devices. IEEE Access **8**, 37518–37525 (2020)
12. S. Dhakal, F. Jaafar, P. Zavarsky, Private blockchain network for IoT device firmware integrity verification and update, in *Proceedings of the 2019 IEEE 19th International Symposium on High Assurance Systems Engineering (HASE)* (IEEE, New York, 2019), pp. 164–170

13. A. Dorri, S.S. Kanhere, R. Jurdak, P. Gauravaram, Blockchain for IoT security and privacy: the case study of a smart home, in *Proceedings of the 2017 IEEE international conference on pervasive computing and communications workshops (PerCom workshops)* (IEEE, New York, 2017), pp. 618–623
14. A. Dorri, S.S. Kanhere, R. Jurdak, P. Gauravaram, LSB: a lightweight scalable blockchain for IoT security and anonymity. J. Parallel Distrib. Comput. **134**, 180–197 (2019)
15. J. Forcier, *Fabric Documentation* (2011)
16. G. Gabriel, R. Roy, S.B. Ramesh Kumar, An architecture to enable secure firmware updates on a distributed-trust IoT network using blockchain, in *Proceedings of the International Conference on Computer Networks and Communication Technologies (to appear)*, vol. 15 (Springer, Singapore, 2019), pp. 671–679
17. B.P. Gopularam, N. Nalini, Mechanism for secure content publishing for reporting platform hosted on public could infrastructure, in *Proceedings of International Conference on Advances in Computing* (Springer, Berlin, 2013), pp. 407–414
18. Y. Gupta, R. Shorey, D. Kulkarni, J. Tew, The applicability of blockchain in the Internet of Things, in *Proceedings of the 10th International Conference on Communication Systems and Networks (COMSNETS) IEEE* (IEEE, New York, 2018), pp. 561–564
19. X. He, S. Alqahtani, R. Gamble, M. Papa, Securing over-the-air IoT firmware updates using blockchain, in *Proceedings of the International Conference on Omni-Layer Intelligent Systems* (2019), pp. 164–171
20. J.L. Hernández-Ramos, G. Baldini, S.N. Matheu, A. Skarmeta, Updating IoT devices: challenges and potential approaches, in *Proceedings of the 2020 Global Internet of Things Summit (GIoTS)* (IEEE, New York, 2020), pp. 1–5
21. S.G. Hong, N.S. Kim, T. Heo, A smartphone connected software updating framework for IoT devices, in *Proceedings of the 2015 International Symposium on Consumer Electronics (ISCE)* (IEEE, New York, 2015), pp. 1–2
22. Hyperledger Composer Introduction. https://hyperledger.github.io/composer/v0.19/introduction/introduction.html
23. Hyperleger Fabric in Smart Cities in Dubai. https://www.ibm.com/blogs/blockchain/2017/04/blockchain-in-dubai-smart-citiesfrom-concept-to-reality/
24. M.N. Islam, S. Kundu, Remote configuration of integrated circuit features and firmware management via smart contract, in *Proceedings of the 2019 IEEE International Conference on Blockchain (Blockchain)* (IEEE, New York, 2019), pp. 325–331
25. A. Kosba, A. Miller, E. Shi, Z. Wen, C. Papamanthou, Hawk: The Blockchain Model of Cryptography and Privacy-preserving Smart Contracts, in *Proceedings of the 2016 IEEE symposium on security and privacy (SP)* (IEEE, New York, 2016), pp. 839–858
26. A. Kuzmin, Blockchain-based structures for a secure and operate IoT, in *Proceedings of the 2017 Internet of Things Business Models, Users, and Networks* (IEEE, New York, 2017), pp. 1–7
27. B. Lee, S. Malik, S. Wi, J.-H. Lee, Firmware verification of embedded devices based on a blockchain, in *Proceedings of the International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness* (Springer, Berlin, 2016), pp. 52–61
28. O. Leiba, Y. Yitzchak, R. Bitton, A. Nadler, A. Shabtai, Incentivized delivery network of IoT software updates based on trustless proof-of-distribution, in *Proceedings of the 2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)* (IEEE, New York, 2018), pp. 29–39
29. X. Li, P. Jiang, T. Chen, X. Luo, Q. Wen, A survey on the security of blockchain systems. Futur. Gener. Comput. Syst. **107**, 841–853 (2020)
30. N.S. Mtetwa, P. Tarwireyi, A.M. Abu-Mahfouz, M.O. Adigun, Secure firmware updates in the Internet of Things: A survey, in *Proceedings of the 2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC)* (IEEE, New York, 2019), pp. 1–7
31. Q. Nasir, I.A. Qasse, M.A. Talib, A.B. Nassif, Performance analysis of Hyperledger fabric platforms. Secur. Commun. Netw. **2018**, 6–11 (2018)

32. J. Ni, S. Member, K. Zhang, X. Lin, X.S. Shen, Securing Fog computing for internet of things applications: challenges and solutions. IEEE Commun. Surv. \ Tutorials **20**(1), 601–628 (2017)
33. J. Ni, K. Zhang, X. Lin, X.S. Shen, Securing fog computing for Internet of Things applications: challenges and solutions. IEEE Commun. Surv. Tutorials **20**(1), 601–628 (2017)
34. J. Park, J. Park, Blockchain security in cloud computing: use cases, challenges, and solutions. Symmetry **9**(8), 164 (2017)
35. A. Pillai, M. Sindhu, K.V. Lakshmy, Securing firmware in Internet of Things using blockchain, in *Proceedings of the 2019 5th International Conference on Advanced Computing and Communication Systems (ICACCS)* (IEEE, New York, 2019), pp. 329–334
36. R. Sakhuja, *Blockchain Development on Hyperledger Fabric*
37. J. Sousa, A. Bessani, M. Vukolic, A Byzantine Fault-tolerant Ordering Service for the Hyperledger Fabric Blockchain Platform, in *Proceedings of the 2018 48th annual IEEE/IFIP international conference on dependable systems and networks (DSN)* (IEEE, New York, 2018), pp. 51–58
38. I.A. Sumra, H.B. Hasbullah, J.B. AbManan, Attacks on security goals (confidentiality, integrity, availability) in VANET: a survey, in *Vehicular Ad-Hoc Networks for Smart Cities* (Springer, Berlin, 2015), pp. 51–61
39. T. Thanh, T.H. Vu, N.V. Cuong, P.N. Nam, A protocol for secure remote update of run-time partially reconfigurable systems based on FPGA, in *Proceedings of the 2013 International Conference on Control, Automation and Information Sciences (ICCAIS)* (IEEE, New York, 2013), pp. 295–299
40. T. Thanh, T.H. Vu, N.V. Cuong, P.N. Nam, A protocol for secure remote update of run-time partially reconfigurable systems based on FPGA, in *Proceedings of the 2013 International Conference on Control, Automation and Information Sciences (ICCAIS)* (IEEE, New York, 2013), pp. 295–299
41. The Hyperledger Greenhouse (2019). https://www.hyperledger.org/
42. Top 10 Vulnerabilities in IoT. https://www.owasp.org/index.php/Top_10_2014-I9_Insecure_Software/Firmware_Updates
43. M.-H. Tsai, Y.-C. Hsu, N.-W. Lo, An efficient blockchain-based firmware update framework for IoT environment, in *Proceedings of the 2020 15th Asia Joint Conference on Information Security (AsiaJCIS)* (IEEE, New York, 2020), pp. 121–127
44. A. Upadhyaya, M. Bansal, Deployment of secure sharing: authenticity and authorization using cryptography in cloud environment, in *International Conference on Advances in Computer Engineering and Applications* (IEEE, New York, 2015), pp. 852–855
45. N. van de Luijtgaarden, Optimizing the core business processes of financial asset management companies using blockchain technology. Stud. Undergraduate Res. E-J. **3**, 144–147 (2017)
46. R.J. Wieringa, What is design science?, in *Design Science Methodology for Information Systems and Software Engineering* (Springer, Berlin, 2014), pp. 3–11
47. E.N. Witanto, Y.E. Oktian, S.-G. Lee, J.-H. Lee, A blockchain-based OCF firmware update for IoT devices. Appl. Sci. **10**(19), 6744 (2020)
48. X. Xu, I. Weber, M. Staples, L. Zhu, J. Bosch, L. Bass, C. Pautasso, P. Rimba, A taxonomy of blockchain-based systems for architecture design, in *Proceedings of the IEEE International Conference on Software Architecture (ICSA)* (IEEE, New York, 2017), pp. 243–252
49. A. Yohan, N.-W. Lo, An over-the-blockchain firmware update framework for IoT devices, in *Proceedings of the 2018 IEEE Conference on Dependable and Secure Computing (DSC)* (IEEE, New York, 2018), pp. 1–8
50. A. Yohan, N.-W. Lo, FOTB: a secure blockchain-based firmware update framework for IoT environment. Int. J. Inf. Secur. (2019), pp. 1–22
51. A. Yohan, N.-W. Lo, S. Achawapong, Blockchain-based firmware update framework for Internet-of-Things Environment, in *Proceedings of the International Conference on Information and Knowledge Engineering (IKE)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp) (2018), pp. 151–155

52. P.G. Zaware, S.V. Shinde, Wireless monitoring, controlling and firmware upgradation of embedded devices using Wi-Fi, in *Proceedings of the International Conference on Advances in Communication and Computing Technologies (ICACACT 2014)* (IEEE, New York, 2014), pp. 1–6
53. Y. Zhao, Y. Liu, A. Tian, Y. Yu, X. Du, Blockchain based privacy-preserving software updates with proof-of-delivery for Internet of Things. J. Parallel Distrib. Comput. **132**, 141–149 (2019)
54. Z. Zheng, S. Xie, H. Dai, X. Chen, H. Wang, An overview of blockchain technology: Architecture, consensus, and future trends, in *Proceedings of the 2017 IEEE international congress on big data (BigData congress)* (IEEE, New York, 2017), pp. 557–564
55. Z. Zheng, S. Xie, H.-N. Dai, X. Chen, H. Wang, Blockchain challenges and opportunities: a survey. Int. J. Web Grid Serv. **14**(4), 352–375 (2018)
56. G. Zyskind, O. Nathan, et al., Decentralizing privacy: using blockchain to protect personal data, in *Proceedings of the 2015 IEEE Security and Privacy Workshops* (IEEE, New York, 2015), pp. 180–184

Chapter 9
Blockchain Security Framework for Government Private Blockchain Consortium

Ahmed Alketbi, Manar Abu Talib, and Qassim Nasir

9.1 Introduction

The adoption of new and emerging technologies is increasing in many sectors. Blockchain is a good example of a disruptive technology that is being explored and adopted in many industries. The adoption of blockchain technology is driven by the need to transform services, the creation of new business models, efficiency increase, and cost optimization [1]. Blockchain technology allows direct and nonreversible transactions, and it promises to improve applications' decentralized data management in a trustless, transparent, and immutable manner [2]. Bitcoin is one of and the most famous application of blockchain technology. Bitcoin was first introduced in 2008 [3]. Since 2008, blockchain technology and its applications have gained the interest of many industries, and that is evident in the number of applications developed using blockchain technology [4]. For instance, the Dubai Government is aiming to become paperless by adopting blockchain technology for all transactions. The strategic objectives behind the Dubai Government's choice to use blockchain technology are to increase government efficiency, create new specialized sectors (industry creation), and achieve global leadership. Blockchain has the potential to

A. Alketbi (✉)
Industrial Engineering and Engineering Management, University of Sharjah, Sharjah, UAE
e-mail: u00045915@sharjah.ac.ae

M. Abu Talib
Computer Science, University of Sharjah, Sharjah, UAE
e-mail: mtalib@sharjah.ac.ae

Q. Nasir
Electrical Engineering Department, University of Sharjah, Sharjah, UAE
e-mail: nasir@sharjah.ac.ae

M. H. ur Rehman et al. (eds.), *Trust Models for Next-Generation Blockchain Ecosystems*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-75107-4_9

improve many applications because of its immutable, transparency, and trustless decentralized data management.

Blockchain differs from a relational database in the way the records are stored. Instead of storing the transaction records in a table, Blockchain groups the records into a block, where new blocks are chained to a previous block using a cryptographic hash. The blockchain ledger is always verified and validated, and it is accessible by the blockchain nodes in the network. There are different types of blockchain: public, private, permissioned, and permissionless. These types are developed on the same main concepts of blockchain technology but differ in the implementation and the functionalities. The authors in [5, 6] studied the adoption of blockchain in governments using a systematic literature review. The studies resulted in identifying the challenges and the need for a proper design of blockchain for the government at the architecture level. The information security aspect is another vital aspect to consider for the design of blockchain technology. The authors in [7] analyzed the security benefits and challenges. The study shows that there is a lack of research examining information security threats and risks and potential mitigation for blockchain technology in the context of government services. Furthermore, the absence of a security control framework for blockchain is another challenge faced by governments that wish to adopt best security practices for blockchain.

In this chapter, we first explain background information about blockchain technology in Sect. 9.2. Then, we explain the blockchain security threat modeling in Sect. 9.3. Section 9.4 presents the developed blockchain security control framework, and the framework validation is explained in Sect. 9.5. Section 9.6 includes the conclusion and future work.

9.2 Blockchain Technology Background

There are two types of blockchain implementations: permissionless and permissioned blockchain. In permissionless blockchain, any participant in the blockchain network can access the network to create a transaction, validate transaction, or view previous transactions. Bitcoin is permissionless, and it is also public as it is not owned by an entity, where any participant can be part of Bitcoin network to create and validate bitcoin transactions. The permissions and access rights are not enforced for participation in the permissionless/public blockchain [8, 9]. On the other hand, in the permissioned blockchain networks, the access restricted to an identified participant who are known and can be trusted, and it is not open to the public. In general, the permissioned blockchain network is owned by one or more owners. The simplicity of the permissioned blockchain helps in reducing the time required for operations, and it offers more flexibility and increases efficiency compared to public permissionless blockchain [10]. The decentralized structure of blockchain holds many promises for this technology, mainly with relatively reduced operations cost and improved efficiency as the network cannot be affected by outages [1]. Another benefit of the decentralized structure of blockchain technology is the absence of single point of failure, the durability, reliability, and longevity. In addition, the blockchain technology provides ledger immutability, integrity, and transparency as the ledger is shared with all participants in the blockchain network [4]. The blockchain ledger is high quality, consistent, accurate, and complete [1]. Furthermore, blockchain allows disintermediation and trustless transactions between participants to eliminate counterparty risks and to provide integrity. Thus, blockchain users are in control of their data and empowered to grant access of information to third party. Blockchain technology has key properties regardless of the blockchain use case implementation or blockchain type. One of the key properties is the distributed ledger that is maintained by the network participants. Transactions are validated and added to all ledgers' copies maintaining identical copies [9]. Another key property of blockchain is the cryptography and digital signatures. Blockchain smart contracts are executable software installed and instantiated on the blockchain and executed when predefined rules are met [11]. Blockchain uses cryptography and digital signatures methods to maintain immutability of blockchain disrupted ledgers. Establishing identities and authenticity in blockchain is achieved by the use of cryptographic methods [12]. Moreover, blockchain uses consensus mechanisms to maintain identical copies of blockchain ledger across the network participants by defining the transaction validation procedure. The blockchain consensus mechanisms allow network participants to create transactions in a peer-to-peer without a centralized third party [13]. The benefits of implementing blockchain include, but not limited to, automation of transactions creation and disintermediation, streamlining processes, reduced transaction processing time, reduced cost, and establishing trust in trustless environment [2]. The use of blockchain is not restricted to specific industries. There are many promising use cases identified in the literature for various industries such as asset management, finance, construction, data storage, crowdfunding, prediction, supply chain management, and education and medical services [4, 10, 14–17].

9.3 Blockchain Security Threat Modeling

The threat modeling is the process of identifying the security threats to evaluate the risks to information asset and to develop risk mitigation. There are various threat modeling methodologies that can be used to identify the threats in blockchain technology, such as the Process for Attack Simulation and Threat Analysis (PASTA) and Hybrid Threat Modeling Method (hTMM). In this chapter, the STRIDE threat modeling methodology is used to identify and classify applicable threats to blockchain technology. Information security threats are events or conditions that have the potential to cause a negative impact on information assets. Threat modeling is the process of identifying, analyzing, and developing mitigation strategies against potential threats to information assets. The government blockchain proposed in [18] is used to analyze these threats. This threat modeling is used to define appropriate security mechanisms for blockchain technology and to develop the blockchain

Table 9.1 STRIDE threat definition and property violated

Threat	Threat definition	Property violated
Spoofing	Pretending to be something or someone other than yourself	Authenticity
Tampering	Unauthorized modification of data on transit or at rest	Integrity
Repudiation	Claiming that you did not do something (honestly or deceptively)	Non-Repudiation
Information disclosure	Gaining access and exposing information to unauthorized persons	Confidentiality
Denial of service	Preventing system from providing service by absorbing resources needed to provide service	Availability
Elevation of privilege	Allowing a program or user to do something they are not allowed to do	Authorization

security control framework. In order to perform holistic threat modeling for the proposed government blockchain reference model, the STRIDE threat modeling methodology is used to walk through the attacks and identify potential threats to blockchain technology. STRIDE stands for Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege, as explained in Table 9.1 [19].

Figure 9.1 shows the sequence for each step of creating a transaction in the proposed government blockchain reference model proposed in [18]. The threat modeling takes into consideration the actor, data flow, and blockchain component in each step:

- Step 1: The end user requests a new government service transaction by accessing the government blockchain application. Usually, the end user credentials are stored in the user wallet.
- Step 2: The request is then routed through the edge services to the blockchain application hosted by the Government Service Owner or the Government Service Consumer.
- Step 3: The blockchain application of the Government Service Owner or Government Service Consumer authenticates and authorizes the user.
- Step 4: The blockchain application of the Government Service Owner routes user requests that invoke blockchain smart contracts to the transaction manager through the Government Service Owner API gateway.
- Step 5: The transaction manager signs the transaction and invokes smart contracts on one or more blockchain nodes depending on the government service required.
- Step 6: The transaction manager writes and reads the data to and from the off-chain data store when the transaction is associated with off-chain data.
- Step 7: The government blockchain network processes the blockchain transaction according to the smart contract business rules and the consensus process in which endorsement from the required government service partners is obtained.

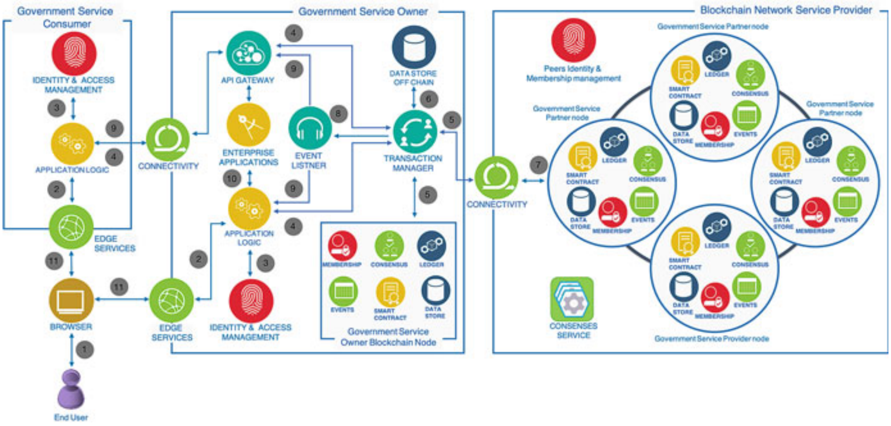


Fig. 9.1 Creating transaction in the government blockchain reference model [18]

- Step 8: The blockchain orders the transactions and forms blocks, sending events to the event listener through the transaction manager.
- Step 9: The blockchain application receives the events from the event listener, and the Government Service Consumer blockchain application receives the events through the API integration from the event listener.
- Step 10: The blockchain application may trigger actions in the enterprise system if required.
- Step 11: The blockchain application sends a message back to the user through the API gateway for transactions received from the Government Service Consumer.

The threat modeling results are explained for each threat category as per STRIDE, highlighting the number of affected steps in the government blockchain reference model use case.

9.3.1 Spoofing

Spoofing attacks are when hackers pretend to be something or someone other than themselves [20]. Spoofing can take various forms, such as spoofing a process on a machine, spoofing a file, spoofing a machine, spoofing a person, or spoofing a role. In all cases, spoofing attacks violate the authenticity property of security. Identified blockchain spoofing threats and proposed mitigations are laid out in Table 9.2. The spoofing threats identified in Table 9.2 are also applicable to the spoofing of an administrator or an operator user in the blockchain network.

Table 9.2 Spoofing threats and mitigation mechanisms

Identified threat	Use case step	Vulnerability	Proposed threat mitigation
T1: Spoofing by obtaining existing credentials stored on client	Step 1	Storing authentication credentials.	– Asymmetric authentication. – Access control. – Multi-factor authentication
T2: Spoofing by obtaining existing credentials on transit (Session Sidejacking)	Step 2	Authentication channels are not encrypted.	– Strong encryption and authentication such as SSL or IPsec.
T3: Spoofing due to insufficient authentication with weak credentials	Step 3	System configuration with default or weak credentials that can be guessed.	– Hardening and security baseline. – Log and audit authentications.
T4: Spoofing endpoint/peers by stealing keys	Step 1	Information disclosure where the data contains crypto keys	– Use of OS tools for secure key storage Use of hardware security modules (HSM)
T5: Spoofing endpoint/peer by forging keys	Step 3	System generates weak keys that can be forged by an attacker	– Strong mechanisms to generate keys
T6: Spoofing peer by compromising API keys	Step 4	Insufficient API authentication or authorization	– Strong authentication mechanisms for API

9.3.2 *Tampering*

Tampering threats involve the unauthorized modification of data on transit or at rest [21]. Tampering violates the integrity of a system and can include tampering with data in a file such as a configuration file or database, tampering with network packets, or tampering with memory. Although blockchain ledger technology is tamper-proof by design, other potential tampering threats may target blockchain infrastructure, as shown in Table 9.3.

9.3.3 *Repudiation*

Repudiation is the act of claiming you did not do something, either honestly or deceptively [22]. Repudiation violates the non-repudiation property of information security, and it may impact the business logic of a system. Repudiation can be performed by repudiating actions or by attacking the logs. The availability of logs, remediation, and auditing are important mitigation measures to repudiation

Table 9.3 Tampering threats and mitigation mechanisms

Identified threat	Use case step	Vulnerability	Proposed threat mitigation
T7: Process tampering to corrupt state	Step 2	Inputs are not appropriately validated.	– Input validation for all inputs
T8: Data flow tampering with a message/transaction	Step 4 Step 5 Step 6 Step 7 Step 8 Step 9 Step 11	Absence of message integrity protection	– Use integrity controls for transactions or message – Integrity controls for transactions or message
T9: Data flow tampering	Step 4 Step 5 Step 6 Step 7 Step 8 Step 9 Step 11	Absence of message encryption protection	– Strong authentication – Network channel encryption or tunneling
T10: Tampering with off-chain data store	Step 6	Misconfiguration or ineffectiveness of access lists (ACLs) permissions	– Configuration baseline and data access permissions.

threats. Table 9.4 shows the identified blockchain repudiation threats and mitigation mechanisms.

9.3.4 *Information Disclosure*

Information disclosure threats are related to unauthorized persons gaining access to information [23]. These threats violate the confidentiality property of information security. Information disclosure threats can be performed against process, data store, or data flow. Table 9.5 shows the information disclosure threats identified for blockchain.

9.3.5 *Denial of Service*

Denial of service (DoS) involves preventing the system from providing service by absorbing the resources needed to provide the service [24]. This attack violates the availability property of information security. It can be performed against process, data store, or data flow. Although the distributed architecture of blockchain technology helps to minimize the effect of DoS threats, DoS can still affect

Table 9.4 Repudiation threats and mitigation mechanisms

Identified threat	Use case step	Vulnerability	Proposed threat mitigation
T11: Repudiation by account takeover	Step 4 Step 5 Step 7	Weak authentication mechanisms	– Stronger authentication mechanisms
T12: Repudiation due to absent or unverified identities	Step 4 Step 5 Step 7 Step 7	Absent or insufficient digital signature verification	– Implement digital signature verification
T13: Deny sending	Step 5 Step 7	Malicious node claim transaction was not sent	– Digital signature – Transactions events logging
T14: Deny receipt	Step 5 Step 7	Malicious node claim transaction was not received	– Transaction events logging
T15: Transaction repudiation	Step 8	Decentralized logging of events	– Log consolidation and security monitoring
T16: Transaction repudiation	Step 8	Logging of events has different times	– Log events with unified time zone

Table 9.5 Information disclosure threats and mitigation mechanisms

Identified threat	Use case step	Vulnerability	Proposed threat mitigation
T17: Information disclosure via access to logs or events	Step 8	Logs and events containing important data about the process	– Design logging with information disclosure in mind – Access permissions
T18: Information disclosure by observing a message	Step 7	Absent or weak confidentiality	– Cryptographic keys – Permissions for on-system flows – Private blockchain transactions – Channel encryption
T19: Information disclosure due to poor coding practice	Step 2 Step 5 Step 7 Step 11	Poor security development—leakage of secrets/credentials/tokens	– Secure development practices.

Table 9.6 Denial-of-service threats and mitigation mechanisms

Identified threat	Use case step	Vulnerability	Proposed threat mitigation
T20: DoS application or API service	Step 2 Step 10	Consuming application and API service resources	– Allocate resources dynamically for critical components – Use DoS mitigation controls
T21: Input validation failures	Step 2 Step 10	Use of inputs that result in crashing the application	– Input validation
T22: Deny access to data store	Step 6 Step 10	Insufficient access control allows adding ACL or locking the data store	– Enforce access permissions for data store Limit configuration changes via rule-based access control

Table 9.7 Elevation of privilege threats and mitigation mechanisms

Identified threat	Use case step	Vulnerability	Proposed threat mitigation
T23: Elevation of privilege	Step 7	Absence of authorization checks	Strong authentication mechanisms
T24: Exploit code vulnerability	Step 5 Step 7	Smart contract code vulnerability	Source code review
T25: Kernel access	Step 4	Poor authorization controls at application and API level	API authorization practices

some components in the blockchain ecosystem, such as web applications and API services, as shown in Table 9.6.

9.3.6 Elevation of Privilege

Elevation of privilege threats are performed by a program or user to perform actions that do not fall under the granted privileges [19]. This threat violates the authorization property of information security and can be done by corrupting a process or bypassing protection mechanisms. Blockchain elevation of privilege threats are laid out in Table 9.7.

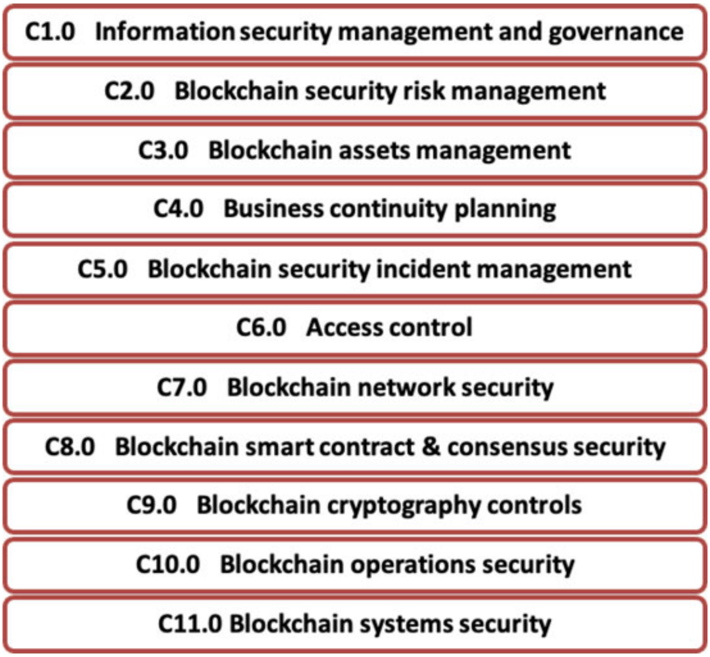

C1.0 Information security management and governance
C2.0 Blockchain security risk management
C3.0 Blockchain assets management
C4.0 Business continuity planning
C5.0 Blockchain security incident management
C6.0 Access control
C7.0 Blockchain network security
C8.0 Blockchain smart contract & consensus security
C9.0 Blockchain cryptography controls
C10.0 Blockchain operations security
C11.0 Blockchain systems security

Fig. 9.2 Blockchain security control framework domains

9.4 Blockchain Security Control Framework

Government entities willing to adopt blockchain technology must be aware of the various blockchain security threats and should develop robust security programs and implement blockchain security controls to mitigate security risks. In this section, the answer to research question 4 is presented with a blockchain security control framework based on the analysis of blockchain threats and risks. Figure 9.2 depicts a representation of the multilayered blockchain security control framework, which encompasses several information security domains and controls to guide government entities to address risks related to blockchain. The information security domains are proposed to have the control framework aligned with the main domains of international security standards such as ISO/IEC 27001—Information Security Management, and the main domains of Dubai Electronic Security Centre Information Security Regulation.

The purpose of the blockchain security control framework is to provide government entities with a framework to ensure an appropriate level of confidentiality, integrity, and availability for blockchain information and to minimize information security risks. The blockchain security control framework is a neutral framework and is not designed for a specific blockchain platform. Government entities should therefore take the technical aspects of specific blockchain platforms into considera-

Table 9.8 Mapping of the security control domains to the identified threats

Identified threat	C1.0	C2.0	C3.0	C4.0	C5.0	C6.0	C7.0	C8.0	C9.0	C10.0	C11.0
T1	✓	✓		✓	✓	✓			✓		
T2	✓	✓		✓	✓		✓				
T3	✓	✓		✓	✓	✓					
T4	✓	✓	✓	✓	✓	✓			✓	✓	
T5	✓	✓	✓	✓	✓				✓	✓	
T6	✓	✓	✓	✓	✓	✓			✓	✓	
T7	✓	✓		✓	✓			✓			✓
T8	✓	✓	✓	✓	✓		✓	✓	✓		✓
T9	✓	✓		✓	✓		✓				✓
T10	✓	✓	✓	✓	✓	✓				✓	✓
T11	✓	✓		✓	✓	✓			✓		
T12	✓	✓		✓	✓	✓		✓	✓	✓	✓
T13	✓	✓		✓	✓	✓			✓		
T14	✓	✓		✓	✓	✓			✓		
T15	✓	✓		✓	✓					✓	✓
T16	✓	✓		✓	✓					✓	✓
T17	✓	✓		✓	✓	✓		✓		✓	
T18	✓	✓		✓	✓		✓	✓			
T19	✓	✓		✓	✓			✓			✓
T20	✓	✓		✓	✓		✓	✓			
T21	✓	✓		✓	✓			✓			✓
T22	✓	✓	✓	✓	✓	✓				✓	✓
T23	✓	✓		✓	✓	✓					
T24	✓	✓		✓	✓			✓		✓	✓
T25	✓	✓		✓	✓	✓		✓		✓	✓

tion when adopting this framework. The blockchain information security control framework is designed to include administrative and technical security controls. These controls address threats identified in the government blockchain reference model threat modeling. The mapping of blockchain security control domains to the addressed threats is shown in Table 9.8. The table shows that the administrative controls in C1.0, C2.0, C4.0, and C5.0 contribute to addressing all identified threats.

9.4.1 C1.0 Blockchain Security Management and Governance

The proposed blockchain security management and governance controls emphasize the importance of information security governance and the need for blockchain network participants to be aligned on security policies and procedures. The blockchain security management and governance controls are laid out below:

- C1.1 Blockchain information security steering committee:
  In order to ensure the governance of the blockchain network and the implementation of blockchain security controls, blockchain consortium should establish blockchain information security steering committee. The blockchain information security steering committee should include heads from each group of blockchain network participants.
- C1.2 Blockchain information security policy:
  The blockchain network requires developing and maintaining a blockchain security policy that is aligned with the government blockchain strategy. The policy is vital to address the basic principles of how to protect blockchain information assets. The policy should be observed and adhered to be all blockchain network participants.
- C1.3 Blockchain operational policy:
  Besides the blockchain information security policy, the blockchain network should also maintain an operational blockchain policy that covers the required security protection measures and controls for blockchain implementation. The blockchain operational policy should be developed and regularly updated based on the results of risk assessments conducted by network participants. The blockchain operational policy must define the responsibility of all blockchain participants and stakeholders to safeguard blockchain-related information assets.
- C1.4 Blockchain security procedures:
  The blockchain operational policy should be supplemented with technical security procedures and guidelines covering the implementation details of the policies. These procedures should take into consideration the different environments supported by all blockchain network participants.
- C1.5 Blockchain information exchange policy:
  In order to govern the information exchange between the blockchain network participants, an information exchange policy and the required procedures should be developed and maintained.
- C1.6 Blockchain information non-disclosure agreement:
  Considering the nature of information exchange between the blockchain network participants, special non-disclosure agreement should be developed and signed by all concerned blockchain network participants and third parties. The agreement should address the need to protect the participants' information in the blockchain ledger from being leaked internally or externally and should also emphasize the "need to know" concept.
- C1.7 External parties service agreement:
  The blockchain information security requirements should be addressed in all service agreements with blockchain vendors and blockchain service providers contracted by any blockchain network participant.
- C1.8 Blockchain acceptable use policy:
  The acceptable use of blockchain information and the assets in blockchain participants' network environment needs to be governed by maintaining an acceptable use policy adhered to by network participants and personal.
- C1.9 Compliance audit:

The compliance audits with the blockchain information security policy requirements should be conducted by independent party covering all network participants. The audits should also ensure compliance with the government legal requirements and privacy requirements.

- C1.10 Blockchain security technical training:
  Blockchain network participants should provide periodic and adequate information security training for employees taking part in operating or securing blockchain systems.
- C1.11 Blockchain security awareness:
  Each network participant should educate the entity's personnel and make them aware of the confidentiality of the information stored in blockchain ledger, including the information owned by other blockchain participants.
- C1.12 Blockchain security KPIs:
  In order to facilitate decision-making, improve performance, and improve information security effectiveness for all blockchain network participants, the participants should develop, agree, and implement a set of information security Key Performance Indicators (KPIs).

9.4.2 C2.0 Blockchain Security Risk Management

The objective of blockchain security risk management controls is to identify, analyze, treat, and review risks associated with blockchain information and assets as described below:

- C2.1 Third-party risk management:
  The risk management of each blockchain network participant should include determining and assessing the information security risks associated with external blockchain participants, blockchain technology vendors, and blockchain service providers. Accordingly, the participants should select and apply appropriate controls for identified risks.
- C2.2 Blockchain risk assessment:
  Blockchain network participants should conduct detailed risk assessment for blockchain implementations in accordance with the adopted methodology. Identified security risks should be communicated with the other affected blockchain network participants for further assessment.
- C2.3 Blockchain risk analysis:
  The analysis of blockchain risks should be used to prioritize the risks based on importance in order to set out treatment plans and controls.
- C2.4 Blockchain risk treatment:
  Blockchain network participants should identify blockchain risk and determine proper risk treatment and appropriate operational security controls. Implementation of risk treatment should be performed in coordination with the concerned blockchain network participants if applicable.

- C2.5 Blockchain risk review:
  Blockchain network participants should review and monitor the effectiveness of implemented blockchain risk mitigation controls.

9.4.3 C3.0 Blockchain Assets Management

The blockchain assets management controls identify and classify blockchain assets and define measures for the proper handling, labeling, and disposal of these assets as described below:

- C3.1 Blockchain assets inventory:
  Each blockchain network participant is responsible to ensure all blockchain assets including blockchain processing facilities and components are identified, documented, and maintained in the asset register.
- C3.2 Blockchain assets ownership:
  The owner and custodian of blockchain assets should be identified. This must include the information owned by external blockchain participants.
- C3.3 Blockchain assets classification policy:
  Blockchain network should have unified blockchain assets classification policy to be implemented and adhered to by all network participants. The policy should define and implement a classification scheme/process for blockchain assets.
- C3.4 Blockchain assets classification:
  Blockchain network participants should ensure that blockchain assets are classified by the assets' owners based on information sensitivity, and in accordance to the blockchain assets classification policy.
- C3.5 Blockchain assets access control:
  Proper access control for blockchain assets has to be defined by the asset owner based on the assigned classification level.
- C3.6 Blockchain assets labeling:
  Adequate labeling and handling controls for the blockchain information assets should be implemented by network participants according to the requirements of each classification level.
- C3.7 Blockchain assets disposal:
  Blockchain network participants should develop and maintain a clear procedure for the authorization of disposal of blockchain information and information assets. The procedure should include the endorsements required by concerned parties in the network.

9.4.4 C4.0 Business Continuity Planning

The business continuity planning controls aim to minimize the impact of a disruption or failure of blockchain services on government services by ensuring the resiliency of the blockchain infrastructure. The business continuity planning controls are listed below.

- C4.1 Business impact analysis and continuity plan:
  Business impact analysis for blockchain systems should be conducted for the network and each participant. In addition, the entity's business continuity plan must be inclusive of the blockchain if it impacts critical business processes.
- C4.2 Blockchain recovery plan:
  Proper recovery plan for blockchain systems should be in place, if the blockchain system is identified as a critical business system as specified by the entity's operations.
- C4.3 Blockchain recovery plan drills:
  The blockchain network participants should exercise and periodically test the recovery plan covering various drill scenarios.
- C4.4 Blockchain backup policy and procedure:
  Blockchain network participants should develop and maintain a documented backup, storage, and retention policy in coordination with other blockchain network participants. The procedure should include the storage specifications.

9.4.5 C5.0 Blockchain Security Incident Management

The objective of blockchain security incident management controls is to minimize the impact of security incidents in the blockchain network by establishing measures to detect and respond to security incidents. The blockchain security incident management controls are listed below.

- C5.1 Blockchain incident management policy:
  Due to the distributed architecture of the blockchain network, and the involvement of external blockchain participants, blockchain network participants should develop and maintain a blockchain incident management policy that addresses the management of blockchain information security incidents including blockchain block collisions.
- C5.2 Blockchain incident response readiness:
  Each blockchain network participant must establish the required capabilities, such as the technical know-how and technologies, to respond to and handle blockchain information security incidents and maintain a knowledge base of blockchain security incidents.
- C5.3 Blockchain incident reporting:
  An agreement with all blockchain network participants should be established to report promptly any observed or suspected information security incidents or weaknesses in blockchain network components or services to the responsible team.
- C5.4 Blockchain incident handling:

The incident handling process should be implemented to handle blockchain information security incidents with the concerned blockchain network participants and engage external authorities for further investigation if needed.

- C5.5 Blockchain incident triage:
  Implement a process to perform incident triage and to gather and retain evidence related to blockchain security incidents with other blockchain network participants.
- C5.6 Blockchain event logging:
  Blockchain systems/applications should be enabled with audit logging for blockchain systems/applications and user access. That should include the transactions logging for valid and invalid transaction.
- C5.7 Blockchain log retention:
  Periodically review the logs and ensure adequate retention measures are applied.
- C5.8 Blockchain security monitoring:
  Centralized and decentralized monitoring are required for blockchain systems/applications, including transaction auditing and monitoring that include adequate log analysis mechanisms.

9.4.6 C6.0 Access Control

The objective of access control is to protect access to blockchain assets and information by implementing identification, authentication, and authorization controls as shown below.

- C6.1 Blockchain access control and authentication policy:
  The blockchain access control policy and an authentication policy should address security requirements for the implementation of effective access control for blockchain network participants. The policy should cover details about accessible blockchain networks and services, the authorization process for granting network access, etc.
- C6.2 Blockchain participants and user access management:
  The process for blockchain participant and user registration, de-registration, and access privilege modification, disabling or removal, etc. should be defined and implemented.
- C6.3 Blockchain user access control:
  Unified user ID that is standard across all blockchain network participants is required, and each user should be provided with a unique identifier.
- C6.4: Use access authentication
  Strong authentication strategy should be implanted to validate claimed user identities, such as two-factor authentication or multi-factor authentication mechanisms.
- C6.5 Blockchain role based access control:

Identification of the categories of users requiring access to blockchain components and the special privileges based on the roles and responsibilities of blockchain network participants.

- C6.6 Access restrictions:
  The creation of new assets, signing transaction inputs, and confirming transactions should be restricted.
- C6.7 Blockchain network access control:
  The authorization, activation, and termination of any blockchain network access and on-boarding/off-boarding network participants.
  Implement proper segregation controls on blockchain network components and other types of networks.
- C6.8 Blockchain systems access control:
  Manage and control access to blockchain system components through secure log-on processes and enforce controls to manage the use of utility programs; implement session time-out controls to prevent unauthorized access; implement lock-out policy; and restrict connection times.
  Implement restrictive controls on the installation of software in the operational blockchain environments.
  Implement proper access control procedures on smart contract source codes.
- C6.9 Blockchain ledger access control:
  Adequate security controls and access rights should be in place to restrict the access blockchain ledger information in alignment with the business rules.
- C6.10 Blockchain access review:
  Blockchain network participants should implement audit trails and log, maintain, and periodically review logical and physical access control lists.

9.4.7 C7.0 Blockchain Network Security

The blockchain network security controls aim to implement measures to secure network connections, routing, and communication between blockchain participant nodes. The blockchain network security controls are described below.

- C7.1 Blockchain connections security:
  The governance of the connections between blockchain peers and off-chain systems outside the blockchain network boundaries.
- C7.2 Network clock synchronization:
  Enable clock synchronization on all networking devices managed by all blockchain network participants, with agreed reference such as Universal Coordinated Time (UTC) to facilitate forensic analysis; continuously monitor its accuracy.
- C7.3 Secure network routing and monitoring:

Secure network routing controls should be implemented, which include the monitoring of network connections continuously and enforcement of security requirements.

- C7.4 Secure network transit:
  Adequate protection for confidentiality, integrity, and availability to prevent unauthorized access to information or data in transit must be implemented by all network participants.
- C7.5 Denial-of-service mitigation and protection:
  Blockchain components should be protected with denial-of-service mitigation and protection mechanisms.

9.4.8 C8.0 Blockchain Smart Contract and Consensus Security

The objective of the blockchain smart contract and consensus security controls is to protect blockchain networks from unauthorized modification or misuse through the integration of controls into the development life cycle of smart contracts and consensus algorithms. The blockchain smart contract and consensus security controls are explained below.

- C8.1 Blockchain applications and smart contract secure development:
  The procedures for secure deployment, distribution, provisioning, and decommissioning of blockchain smart contracts, applications, and application interfaces (APIs) through regular updates and checks to ensure adequate level of security:
- C8.2 Blockchain application and smart contract security requirements:
  Identifying and documenting the security and privacy requirements for blockchain systems, consensus algorithms, and smart contracts.
- C8.3 Blockchain consensus security requirements:
  Identifying the consensus requirements and implementing adequate controls to protect it.
- C8.4 Blockchain consensus security validation:
  Conducting proper testing to validate integrity of data output from blockchain consensus.
- C8.5 Blockchain source code change management:
  Implementing proper change management controls on smart contract source code development processes, whether performed in-house or outsourced. Implementing proper controls to limit the risk of changes to blockchain smart contract packages, such as validating the authenticity of source code before deployment.
- C8.6 Blockchain smart contract testing:
  Testing and verifying the operational status of all blockchain smart contracts after implementing any change.
- C8.7 Blockchain consensus security testing:
  Performing tests for the consensus algorithm to ensure protection.
- C8.8 Blockchain smart contract and application source code review:

Conducting periodic code reviews on blockchain source code, consensus code, and smart contracts developed in-house or by an external party.

9.4.9 C9.0 Blockchain Cryptography Controls

The blockchain cryptography controls aim to implement proper security measures for the cryptographic keys used to sign transactions and authenticate users in the blockchain network.

- C9.1 Cryptographic keys protection:
  Implementing proper protection and security controls on all cryptographic keys used on the blockchain.
- C9.2 Key lifecycle management:
  Employing full key lifecycle management practices, including generation and protection of keys using hardware security module (HSM), key rotation, secure key distribution, and secure destruction of keys at the end of their lifecycle.
- C9.3 Blockchain transactions co-signing:
  Implementing transaction signing and a process for multi-signature and co-signing of transactions and smart contract source code packages with blockchain network participants.

9.4.10 C10.0 Blockchain Operations Security

The blockchain operations security controls are related to securing blockchain from risks associated with the daily operations of blockchain components and infrastructure. The details of the blockchain operations security controls are described below.

- C10.1 Blockchain operating procedure:
  The operating procedure documentation for blockchain systems detailing inputs, outputs, and dependencies should be developed and maintained.
- C10.2 Blockchain security baseline:
  Blockchain network participants should document, maintain, and enforce up-to-date baseline configuration manuals of all blockchain systems and components.
- C10.3 Blockchain change management policy:
  Blockchain network participants should develop and maintain a documented change management policy that defines the blockchain change management process employed by any of the network participants, outlining roles and responsibilities of the participants.
- C10.4 Blockchain change management procedure:
  Supplementing the change management policy with a detailed procedure to facilitate the implementation of the change and configuration management process and provide guidelines for all blockchain network participants.

- C10.5 Segregation of duties:
  Segregating the duties and responsibilities by distributing tasks among multiple blockchain network participants so as to reduce errors, fraud and unauthorized modification, or misuse of the blockchain network.
- C10.6 Blockchain environment separation:
  Separating the blockchain development, testing, and production environments to mitigate the risk to all systems from unauthorized intentional or unintentional access or change.
- C10.7 Blockchain maintenance procedure:
  The development procedure for the upgrade of blockchain technology platform components, addressing the government blockchain ecosystem requirements and the government entity's requirement for ensuring proper security control implementation and baseline configuration requirements prior to acceptance or deployment.
- C10.8 Blockchain security acceptance criteria:
  Proper acceptance criteria for new blockchain systems or upgrades to existing blockchain systems should be in place.
- C10.9 Blockchain security testing:
  Blockchain network participants should carry out suitable security testing of blockchain systems prior to acceptance and deployment and ensure periodic testing.

9.4.11 C11.0 Blockchain Systems Security

The blockchain systems security controls secure blockchain components and infrastructure by conducting security tests and implementing protection controls as described below.

- C11.1 Blockchain security validation:
  In order to validate the integrity of data input controls on blockchain systems, proper testing should be conducted with validation checks integration into blockchain processing to detect any issues in processed information, including transaction validation.
- C11.2 Malware protection:
  Malware protection mechanisms should be implemented to detect and eradicate malware and malicious code for all blockchain systems.
- C11.3 Vulnerability assessment:
  Technical security reviews and vulnerability assessments must be performed by blockchain network participants in order to periodically assess technical infrastructure and applications against the latest blockchain threats and vulnerabilities.
- C11.4 Blockchain secure deployment:
  The deployment of blockchain systems and smart contracts into production environment should be after implementing the appropriate controls (including

virtualization security if applicable) and after completing tests and fixing identified defects.

- C11.5 Participant security sign-off process:
  The security sign-off process with all blockchain participants is required to confirm proper implementation of security controls on all blockchain components owned by participants prior to deployment.
- C11.6 Blockchain ledger confidentiality:
  Blockchain network participants should identify and protect the confidentiality of data processed in the blockchain using private data mechanisms.

9.5 Validation of the Blockchain Security Control Framework

The proposed blockchain security control framework is compared to Dubai Electronic Security Centre (DESC) Dubai Information Security Regulation (ISR) version 2 controls for validation to ensure that the proposed framework is comprehensive and that it covers all applicable information security domains. In Table 9.9, the ISR control domains covered in the proposed framework are marked. The comparison shows that the proposed blockchain security control framework is closely aligned with ISR control objectives, which have been drafted and revised to reflect various globally recognized standards, regulations, and frameworks related to information security, such as ISO/IEC standards (International Organization for Standardization/International Electrotechnical Commission), BSI standards (British Standards Institute), PCI (Payment Card Industry) council standards, and the COBIT (Control Objectives for Information and Related Technology) framework.

As shown in Table 9.9, some ISR control domains are associated with multiple factors in the proposed blockchain security control framework. This is because ISR controls are broken down into thirteen domains based on three major information security classes (governance, operations, and assurance), while the proposed blockchain security control framework is structured according to the layers presented in Fig. 9.2. The ISR control domains that are not covered in the proposed blockchain security control framework are shown in Table 9.9. The ISR Environmental and Physical Security domain is related to the development of physical access policies and procedures, which include the establishment of physical access perimeters for all physical accesses and points of entry. This domain is required to ensure protection of organization premises, information processing facilities, and resources from physical or environmental damages. These domain controls are not considered in the blockchain security control framework, as blockchain technology does not require unique physical and environmental security controls.

The domain of Roles and Responsibilities of Human Resources controls is related to human resources' roles and responsibilities prior to and during employment, as

Table 9.9 Blockchain security control framework and ISR control domains

ISR v2 Controls Domain	C1.0	C2.0	C3.0	C4.0	C5.0	C6.0	C7.0	C8.0	C9.0	C10.0	C11.0
Information Security Management and Governance	✓	✓			✓						
Information and Information Assets Management	✓		✓								
Information Security Risk Management		✓									
Incident and Problem Management					✓						
Access Control			✓		✓	✓					
Operation, Systems and Communication Management	✓				✓	✓	✓		✓	✓	✓
Business Continuity Planning				✓							
Information System Acquisition, Development and Management						✓		✓	✓		
Environmental and Physical Security											
Roles and Responsibilities of Human Resources											
Compliance and Audit	✓										
Information Security Assurance and Performance Assessment	✓										
Cloud Security											

well as after termination/change of employment. This domain is generic in nature, and no specific control is required in addition for blockchain technology. The ISR Cloud Security controls are for mitigating risks associated with cloud computing and usage of cloud services. The same controls would apply to blockchain nodes or components hosted in cloud environments, but no unique control for blockchain technology is required in this context; hence, the control is not included in the blockchain security control framework.

9.6 Conclusion and Future Work

In conclusion, the use of blockchain technology for government services introduces new capabilities that can mitigate information security risks. Due to the distributed and decentralized architecture of blockchain technology, the impact of information security attacks is minimized, as the system is not vulnerable to a single point of failure. Threats and attacks that may target participants of the blockchain network will not have an impact on the identical ledgers stored in other participants' nodes. The validation mechanisms enforced in permissioned blockchain consensus guarantee the integrity of the ledger blocks. Based on the consensus rules, validation ensures that only valid blocks are appended to the ledger and the integrity of the ledger is continuously maintained. Unlike tampering centralized databases or data stores, ledger tampering requires the attacker to compromise multiple nodes in order to bypass the blockchain consensus validations controls. The use of cryptography in blockchain helps to prevent these attacks. Cryptography is used in blockchain for user access as well as block validation and generation using digital signatures and hashing. Blockchain technology therefore requires strong key management to maintain encryption keys. However, blockchain technology may also introduce new information security risks when used for government services. Government entities should develop robust security programs and implement blockchain security controls to mitigate security risks. Based on the results of the threat modeling, the top three risks associated with blockchain are identified as follows:

- Compromised cryptographic keys and identities:
  Compromised cryptographic keys are one of the top security risks in permissioned blockchain. As shown in the threat modeling results, many threats can result if cryptographic keys are compromised. This risk is associated with the use of weak keys, incorrect or repeated use of keys, lack of key rotation, inappropriate storage of keys, or inadequate protection of keys. Robust key management is therefore essential to mitigate the risks associated with compromised cryptographic keys.
- Exploitation of vulnerable software or protocol:
  As is the case with any other software or protocol, blockchain is subject to errors or vulnerabilities in the code and protocols. This is also applicable to coding errors in off-chain applications that interact with the blockchain. Smart contract code may also include defects that can be exploited.
- Off-chain data source risks:
  The permissioned blockchain interfaces with external off-chain data sources and data stores to ingest data, as well as to reference data that is stored externally. This data is not subject to blockchain consensus validation mechanisms and security controls. Data corruption and tampering of off-chain data therefore introduce risks to permissioned blockchain security.

Blockchain technology immutability is one of the main characteristics and a key feature of blockchain. However, if a malicious transaction is added to a blockchain ledger, blockchain immutability restricts recovery from this malicious transaction. Therefore, government blockchain participants should establish a process to detect malicious transactions in the ledger and establish recovery options such as hard-fork. The distributed architecture of blockchain creates resiliency in the overall blockchain network. Nevertheless, the challenge lies in maintaining a strong information security posture across all blockchain network participants, as security compromise or breach may impact all other participants of the blockchain network.

Therefore, it is necessary to adopt a consistent information security framework and to align the security programs of each network security participant. Hosting blockchain nodes within BNSP infrastructure reduces the risk and simplifies the challenge, as most of the responsibility for security control implementation will be with the BNSP, and comprehensive network security controls can be enforced. BNSP also enforces security standard compliance on the blockchain network participants. It is clear that defining the roles and responsibilities of each blockchain network participant is vital to maintaining the security of the network. In future work, the proposed blockchain security framework will be applied to other implementations with different operating models to be tested for effectiveness.

Acknowledgments We would like to thank Graduate Studies office and OpenUAE Research and Development Group at University of Sharjah for supporting this research. We are also grateful to Dubai Electronic Security Center for supporting this research.

References

1. M. Swan, *Blockchain: Blueprint for a New Economy* (O'Reilly Media, New York, 2015)
2. L. Hughes, Y.K. Dwivedi, S.K. Misra, N.P. Rana, V. Raghavan, V. Akella, Blockchain research, practice and policy: applications, benefits, limitations, emerging research themes and research agenda. Int. J. Inf. Manage. **49**, 114–129 (2019)
3. S. Nakamoto, *Bitcoin: A Peer-to-peer Electronic Cash System* (2008)
4. L. Yang, The blockchain: State-of-the-art and research challenges. J. Ind. Inf. Integr. **15**, 80–90 (2019)
5. E.A. Franciscon, M.P. Nascimento, J. Granatyr, M.R. Weffort, O.R. Lessing, E.E. Scalabrin, A systematic literature review of blockchain architectures applied to public services, in *Proceedings of the 2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design* (Porto, Portugal, 2019)
6. F.R. Batubara, J. Ubacht, M. Janssen, Challenges of blockchain technology adoption for e-government, in *Proceedings of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age* (Delft, The Netherlands, 2018)
7. A. Alketbi, Q. Nasir, M.A. Talib, Blockchain for government services—use cases, security benefits and challenges, in *Proceedigs of the 15th Learning and Technology Conference* (IEEE, Jeddah, 2018)
8. K.H. Kwak, J.T. Kong, S.I. Cho, H.T. Phuong, A study on the design of efficient private blockchain, in *Computational Science/Intelligence and Applied Informatics, Studies in Computational Intelligence* (2019), pp. 93–121
9. M. Pilkington, Blockchain technology: Principles and applications, in *Handbook of Research on Digital Transformations'* ed. by F. Xavier Olleros, M. Zhegu (2016)
10. F. Casino, T.K. Dasaklis, C. Patsakis, A systematic literature review of blockchain-based applications: current status, classification and open issues. Telematics Inf. **36**, 55–81 (2019)
11. N.O. Nawari, S. Ravindran, Blockchain and the built environment: Potentials and limitations. J. Build. Eng. **25**, 55–81 (2019)
12. M.C. Nachiappan, P. Pattanayak, S. Verma, V. Kalyanaraman, Blockchain technology: beyond bitcoin. Appl. Innovation Rev. **2**(2), 6–19 (2016)
13. J. Mattila, The blockchain phenomenon—the disruptive potential of distributed consensus architectures, in *ETLA Working Papers* (2016)

14. C.T. Gibson, T. Kirk, Blockchain 101 for asset managers. The Investment Lawyer **23**, 7–14 (2016)
15. J. Li, D. Greenwood, M. Kassem, Blockchain in the built environment and construction industry: a systematic review, conceptual models and practical use cases. Autom. Constr. **102**, 288–307 (2019)
16. Y. Yuan, F. Wang, Towards blockchain-based intelligent transportation systems, in *Proceedings of the IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)* (Rio de Janeiro, Brazil, 2016)
17. P.T. Duy, D.T.T. Hien, D.H. Hien, V.-H. Pham, A survey on opportunities and challenges of blockchain technology adoption for revolutionary innovation, in *Proceedings of the Ninth International Symposium on Information and Communication Technology, Danang City, Viet Nam* (2018)
18. A. Alketbi, Q. Nasir, M. Abu Talib, Novel blockchain reference model for government services: Dubai government case study. Int. J. Syst. Assur. Eng. Manage. **11**(5), 1170–1191 (2020)
19. A. Shostack, *Threat Modeling: Designing for Security* (Wiley, New York, 2014)
20. K. Saatkamp, C. Krieger, F. Leymann, J. Sudendorf, M. Wurster, Application threat modeling and automated VNF selection for mitigation using TOSCA. J. Commun. Inf. Networks **4**(2), (2019)
21. X. Liang, S. Shetty, D. Tosh, Exploring the attack surfaces in blockchain enabled smart cities, in *Proceedings of the IEEE International Smart Cities Conference (ISC2)* (IEEE, Kansas, 2018)
22. M. Haider, S. Bin Saleem, J. Rafaqat, N. Sabahat, Threat modeling of wireless attacks on advanced metering infrastructure, in *Proceedings of the 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS), Karachi, Pakistan* (IEEE, New York, 2019)
23. M. Hagan, F. Siddiqui, S. Sezer, ABC: a cryptocurrency-focused threat modeling framework, in *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Paris, France* (IEEE, New York, 2019)
24. I. Homoliak, S. Venugopalan, Q. Hum, P. Szalachowski, A security reference architecture for blockchains, in *Proceedings of the IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA* (IEEE, New York, 2019)

Chapter 10
Toward Inter-Blockchain Communication Between Hyperledger Fabric Platforms

Ilham Qasse, Manar Abu Talib, and Qassim Nasir

10.1 Introduction

With the rapid evolution of blockchain technology, the applications number associated with blockchain has expanded enormously in numerous different fields such as healthcare, finance, asset and identity management, supply chain, etc. [1]. However, the adoption of this technology in the industry is limited due to challenges in the available blockchain platforms such as connectivity, privacy, and scalability [1]. Blockchain scalability is calculated as a function of the number of nodes the blockchain network can handle without failure. Blockchain interoperability can scientifically improve blockchain network scalability and increase the technology adoption by users and industry. Interoperable blockchain architecture is defined by the National Institute of Standards and Technology (NIST) [2] as a group of blockchain platforms, either homogeneous (same blockchain platform) or heterogonous (different blockchain platforms), where each has its own distributed ledger. Multiple platforms can participate in the execution of a transaction. The data recorded in one ledger or platform is verifiable and can be accessed semantically by an external transaction. Transferring and exchanging data between networks is possible in interoperable blockchain systems. However, the ability to interact with the transferred data is based on the level of interoperability whether it is foundational, structural, or semantic [3].

I. Qasse (✉) · M. A. Talib
Department of Computer Science, University of Sharjah, Sharjah, UAE
e-mail: iqasse@sharjah.ac.ae; mtalib@sharjah.ac.ae

Q. Nasir
Department of Computer and Electrical Engineering, University of Sharjah, Sharjah, UAE
e-mail: nasir@sharjah.ac.ae

M. H. ur Rehman et al. (eds.), *Trust Models for Next-Generation Blockchain Ecosystems*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-75107-4_10

Data exchange between systems is permitted at the foundational level, but the interpretation of the shared data is not allowed. The transferred data may be viewed at the structural level; however, you cannot interact with the data nor use it. On the other hand, the shared data between networks can be used at the semantic level [3].

The majority of blockchain networks operate in environments that are disconnected from each other. This raises the scalability and connectivity challenges while also restricting the technology adoption in some industry ecosystems. Currently, there is no shared trust in various blockchain networks, and there is no connection between these networks. Blockchain interoperability will improve the scalability and connectivity of the blockchain platforms [4–6]. Given the potential of cross-blockchain communication and blockchain interoperability, the number of the proposed solutions in the blockchain interoperability field is increasing in the last few years. However, most of the proposed solutions focused on the crypto ecosystems and did not address the second blockchain generation.

This chapter focuses on designing and implementing an inter-blockchain communication scheme between Hyperledger Fabric networks. The proposed scheme used smart contracts to achieve inter-blockchain communication between Hyperledger Fabric networks. We also evaluate the performance of the proposed scheme in terms of the execution time, latency, and throughput.

The main contributions of this chapter are:

- We designed and implemented an inter-blockchain communication system between Hyperledger Fabric networks. This is the first-ever implemented inter-blockchain communication in the context of permissioned and private blockchain networks.
- We evaluated the performance of the proposed scheme, which resulted in the first-ever performance evaluation for inter-blockchain solutions.

Following this section, Sect. 10.2 introduces blockchain technology and gives an overview of Hyperledger Fabric platform. Section 10.3 presents the proposed methodology to achieve the goal of interoperability. The implementation and performance evaluation results are discussed in Sects. 10.5 and 10.4, respectively. Section 10.6 discusses the related work. Finally, a conclusion and future research directions are presented in Sect. 10.7.

10.2 Background

10.2.1 Blockchain Technology

Blockchain is a distributed ledger of transactions in a peer-to-peer network [7]. Transactions are processed and stored in a series of linked blocks [7]. Blockchain was first introduced in 2008 when Nakamoto [8] proposed the digital currency bitcoin. It is now considered to be the backbone of all existing cryptocurrencies.

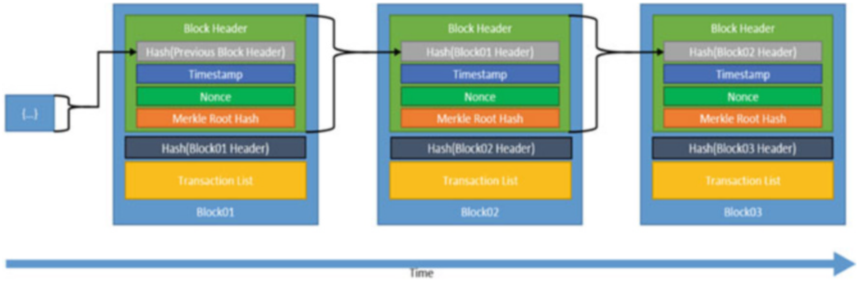


Fig. 10.1 Block structure [2]

The main feature of this technology is that the blockchain is a distributed public ledger where no one holds the ledger, and there is an equivalent copy of the ledger for any node in the network.

A blockchain consists of several blocks that are linked together. As shown in Fig. 10.1, each block contains the data, the previous block hash, and the hash of the block itself [2]. The information stored within the block is based on the blockchain type. Per block also has a hash that uniquely identifies the block and all of its contents.

10.2.2 Smart Contract

A smart contract is a computer program that is stored inside a blockchain. Like the blockchain itself, smart contracts are immutable and can never be changed or tampered with. Smart contracts have some general vital properties. Firstly, every smart contract is autonomous. Once the developer creates the logic of the smart contract and deploys it on the blockchain, the developer does not need to participate in any of the smart contract processes. Smart contracts are also self-sufficient, which means they can collect currency, conduct purchases, administer services, and issue and expend funds to allow storage and processing capacity to be more extensive. Lastly, smart contracts are decentralized. They are distributed by various network nodes instead of being focused on one single central source.

Smart contracts are mostly developed in network-specific scripting languages like Solidity, Viper, Rust, and Ivy-Lang. However, some blockchain platforms allow developers to write smart contracts in languages that they are already familiar with, such as Python and JavaScript. The main advantage of smart contracts is that they are secure, so that the data cannot be altered. Moreover, because smart contracts are entirely autonomous, they accurately conduct tasks without human errors.

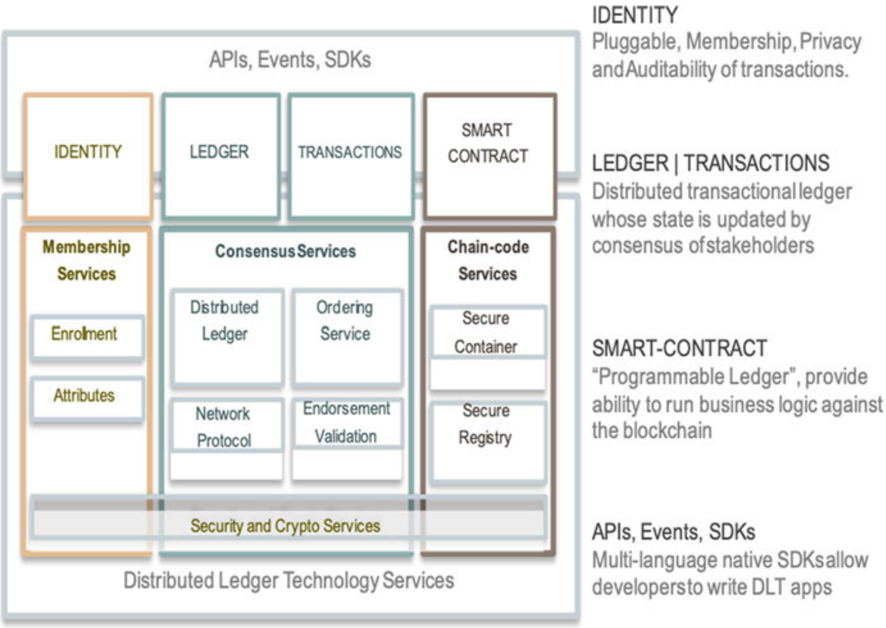

APIs, Events, SDKs
IDENTITY
LEDGER
TRANSACTIONS
SMART CONTRACT
Membership Services
Enrolment
Attributes
Consensus Services
Distributed Ledger
Ordering Service
Network Protocol
Endorsement Validation
Chain-code Services
Secure Container
Secure Registry
Security and Crypto Services
Distributed Ledger Technology Services
IDENTITY
Pluggable, Membership, Privacy and Auditability of transactions.
LEDGER | TRANSACTIONS
Distributed transactional ledger whose state is updated by consensus of stakeholders
SMART-CONTRACT
"Programmable Ledger", provide ability to run business logic against the blockchain
APIs, Events, SDKs
Multi-language native SDKs allow developers to write DLT apps

Fig. 10.2 Hyperledger Fabric architecture [11]

10.2.3 *Hyperledger Fabric*

Hyperledger Fabric is an open-source permissioned blockchain platform that is developed and funded by IBM and the Linux Foundation. In Hyperledger Fabric, access to the network is only limited to network participants. The consensus algorithm applied to verify the transactions and append new blocks in Hyperledger Fabric is PBFT [9]. In Hyperledger Fabric, the transactions are managed using chaincode, which is an alternative term of the smart contract. Chaincode is a program code that defines a set of rules to interact with the blockchain network [10] (Fig. 10.2).

10.3 Methodology

10.3.1 *Solution Design*

Based on the architecture of Hyperledger Fabric, to achieve cross-blockchain communication, it must be achieved in three different stages. Figure 10.3 shows the different stages that are required to achieve Hyperledger Fabric inter-blockchain communication.

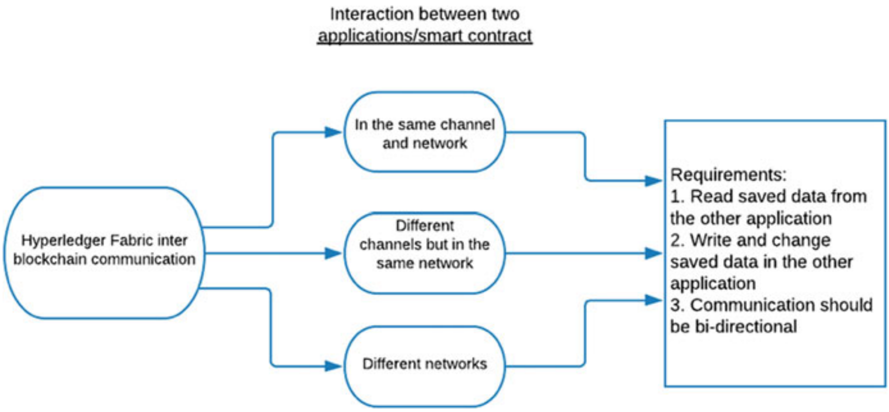

Interaction between two applications/smart contract
Hyperledger Fabric inter blockchain communication
In the same channel and network
Different channels but in the same network
Different networks
Requirements:
1. Read saved data from the other application
2. Write and change saved data in the other application
3. Communication should be bi-directional

Fig. 10.3 Hyperledger Fabric inter-blockchain communication design

10.3.1.1 Interaction Between Two Applications/Smart Contracts in the Same Channel and Network

In Hyperledger Fabric architecture, the channel is an isolation mechanism where the nodes outside the channel cannot access the transactions (txn) formed inside the channel. Multiple different smart contracts can be installed on the same channel. The first step to reaching interoperable Hyperledger Fabric networks is to achieve cross-communication between different smart contracts in the same channel and network (Fig. 10.4).

10.3.1.2 Interaction Between Two Applications/Smart Contracts in Different Channels but the Same Network

After achieving cross-communication between two smart contracts in the same channel, the next step is to investigate the interaction between two applications in different channels but in the same network (as shown in Fig. 10.5). The smart contracts in different channels are generally isolated, where the smart contract in one channel cannot be accessed from another channel. However, the structure of the network is known (number of peers, number of channels, etc.).

10.3.1.3 Interaction Between Two Applications/Smart Contracts in Different Networks

A network in Hyperledger Fabric cannot see the structure of another network (peers, channels, smart contracts). In this step, inter-blockchain communication is

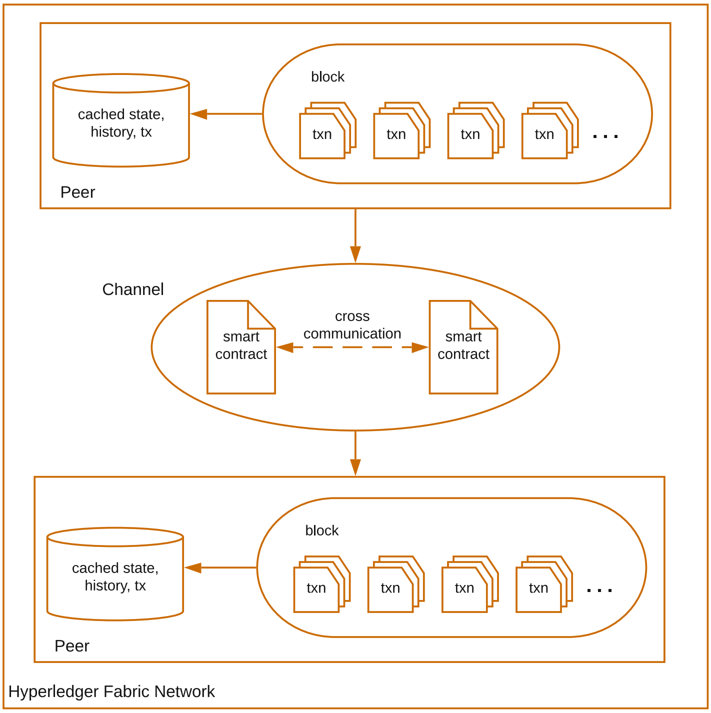


Fig. 10.4 Interaction between two applications/smart contracts in the same channel and network

enabled between two Hyperledger networks. The goal is to make two applications on different networks interact and share data (as illustrated in Fig. 10.6).

10.3.2 Inter-Blockchain Communication Solution

To achieve connectivity between two applications, whether they are on the same channel, different channels, or different networks, we propose the use of the Node-RED framework. We have chosen Node-RED software because it is the only software that can connect different APIs and because it supports the Hyperledger Fabric API. Node-RED [12] is a graphical editor that helps programmers to easily integrate most APIs such as physical I/O, databases, and cloud-based applications in any combination. The proposed solution will run the Node-RED tool to act as the mediator between different blockchain applications. Node-RED defines the behavior of an application as a group of connected black boxes known as nodes. In every blockchain application, we have attached a Node-RED flow that will act as a sender or receiver. Figure 10.7 demonstrates how inter-blockchain communication is achieved with Node-RED.

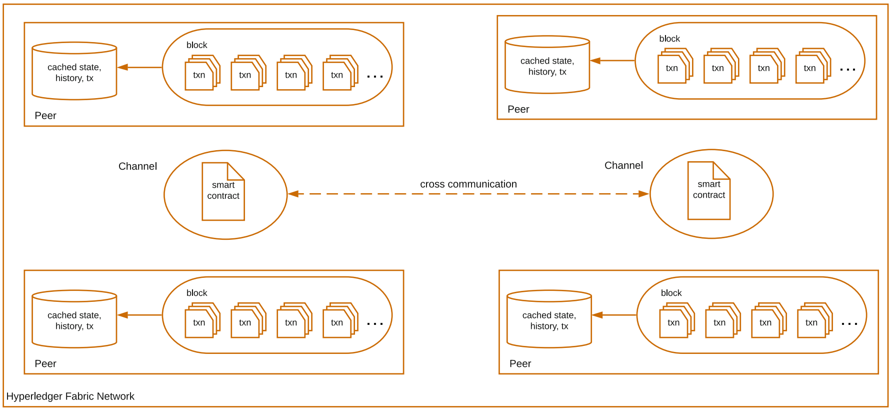


Fig. 10.5 Interaction between Two Applications/Smart contracts in different channels but in the same network

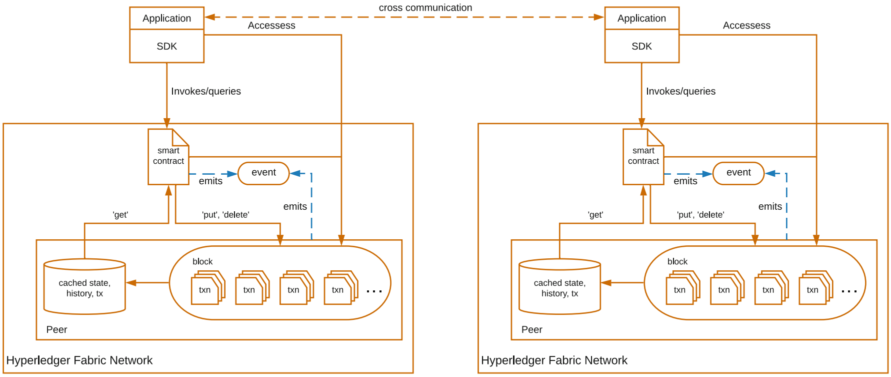


Fig. 10.6 Interaction between two applications/smart contracts in different networks

10.3.2.1 Node-RED Sender Flow

The sender flow monitors network transactions and filters any cross-communication requests. Such requests are then forwarded to the requested destination if the destination address is valid. The process of the sender flow is shown in Fig. 10.8.

10.3.2.2 Node-RED Receiver Flow

The Node-RED receiver flow, which is shown in Fig. 10.9, listens for any external transactions submitted by other applications. Once a request transaction is received, the flow will check to see if the submitted transaction is valid and whether the syntax of the request is correct. Once the transaction data is validated, the flow will submit

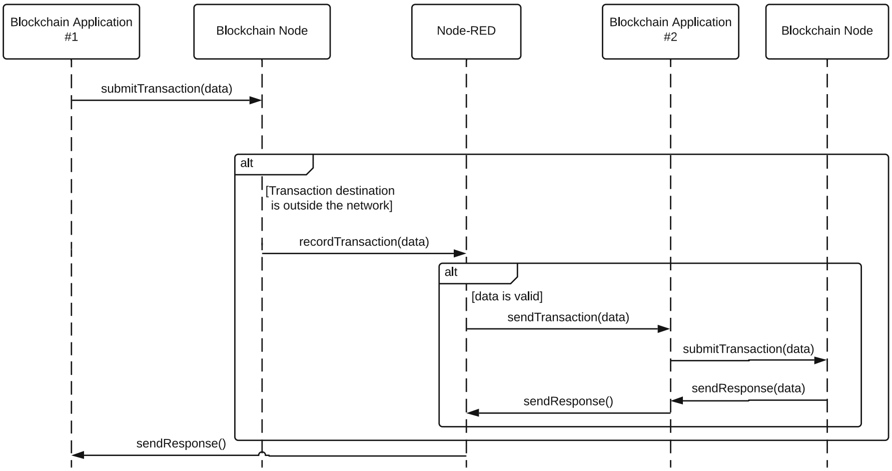


Fig. 10.7 Inter-blockchain communication solution process

the transaction in the application and send a response back to the sender application. Otherwise, the receiver flow will respond with an error message to the sender.

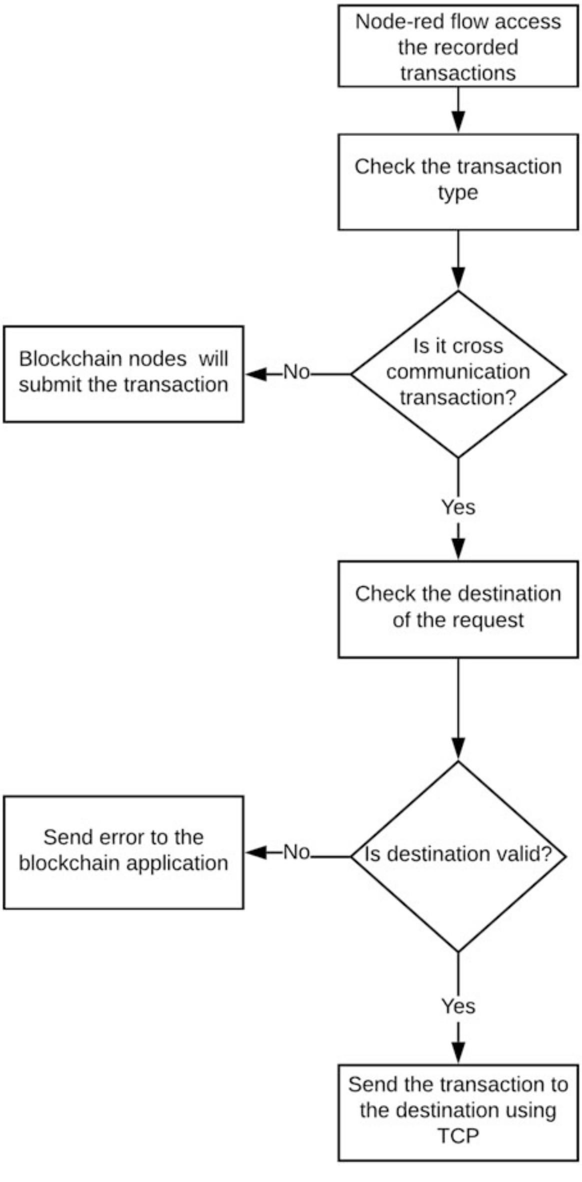
10.4 Implementation

In this section, the implementation details for an inter-blockchain communication system between different Hyperledger Fabric networks are discussed. The main components that are covered are: deploying the Fabric network, developing the blockchain application, and inter-blockchain communication solution.

10.4.1 Fabric Network Deployment

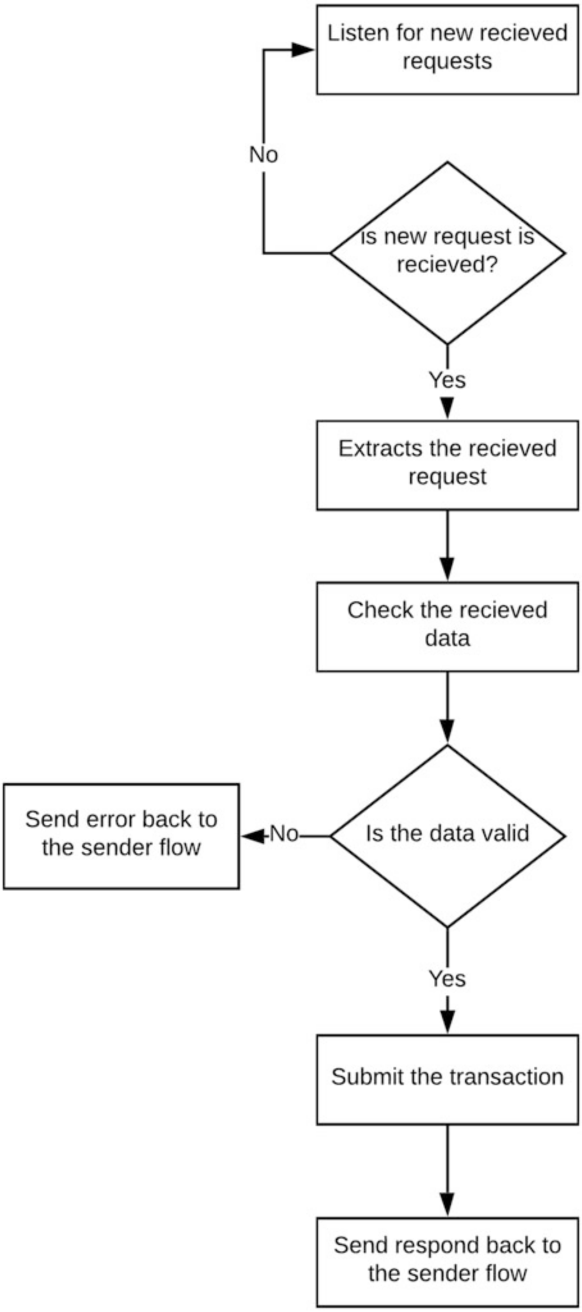
The first step in the implementation process is to deploy a Fabric network. The Fabric network consists of the orderer, peer, channel, and ledger. Based on the proposed methodology in Sect. 10.3, we have built three different Fabric network structures. In the first structure, we have one network and one channel. The Fabric network has a single peer and a solo orderer node as shown in Fig. 10.10. Figure 10.11 demonstrates the second structure, where the Fabric network contains multiple peers, two channels, and one orderer node. Finally, in the third structure, we have deployed two Fabric networks. Each Fabric network consists of a single peer, one channel, and one orderer node. Each network in this structure is the same as structure one. To create the Fabric networks, we used Docker and Docker-compose tools [13]. Docker is a tool for running applications in an isolated environment

Fig. 10.8 Sender flow process


Node-red flow access the recorded transactions
Check the transaction type
Is it cross communication transaction?
No
Blockchain nodes will submit the transaction
Yes
Check the destination of the request
Is destination valid?
No
Send error to the blockchain application
Yes
Send the transaction to the destination using TCP

known as a container. In Hyperledger Fabric, Docker is used to generate the different Fabric components as isolated containers. The Docker-compose file defines multiple Docker containers that make up the infrastructure components for the Fabric network. Setting up the Fabric network consists of three steps. The first step is to generate the required cryptography materials for the network component's identity. The generated cryptography materials include the certificates and public

Fig. 10.9 Receiver flow process


Listen for new recieved requests
No
is new request is recieved?
Yes
Extracts the recieved request
Check the recieved data
Send error back to the sender flow
No
Is the data valid
Yes
Submit the transaction
Send respond back to the sender flow

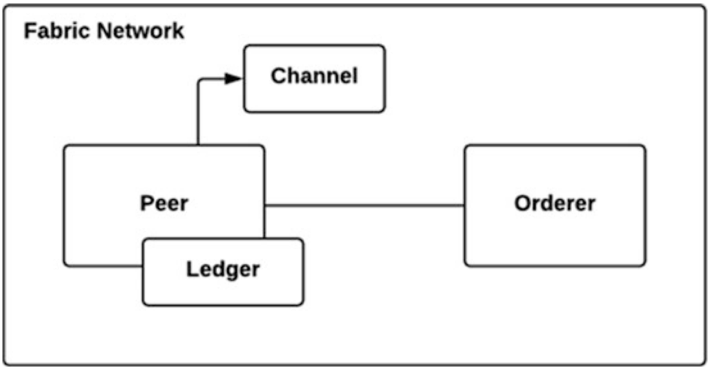


Fig. 10.10 Fabric network structure 1

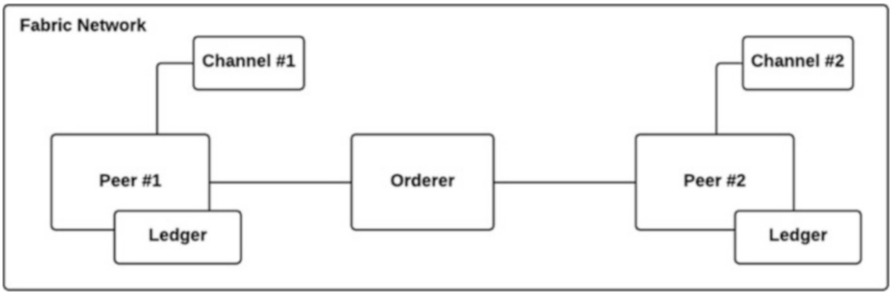


Fig. 10.11 Fabric network structure 2

and private keys for each Fabric component. The second step is to generate the genesis block and to create the channel artifacts. The final step is to use Docker-compose to create and run Docker containers for the Fabric components using the generated materials in the first two steps.

10.4.2 Blockchain Application

We have built a blockchain application that targets commodity trading. The asset in this application is the commodity, where it can be any physical commodities such as gold, oil, raw commodities, etc. Each asset is owned by a trader that is the main participant in the business network model. The main functions of the created business network model are creating new Commodity and Trader or querying the details of created data. The write function represents creating a new Trader, while the query request is the read function.

10.4.3 Solution Deployment

As mentioned in Sect. 10.3, Node-Red flow is used as an interconnector between different blockchain applications. Two nodejs applications were built to listen and record all the transactions in the network. The Node-Red application will access the recorded transactions and check the destination of the transactions. The Node-Red flows are presented by the different types of nodes that are connected. Each node in the flow has a clear functionality and purpose. The inter-blockchain communication flow contains two flows, sender flow, and receiver flow.

10.4.3.1 Sender Flow

The sender flow, as shown in Fig. 10.12, contains the multiple connected nodes, which are:

- Injection node: accesses the transactions record
- Data node: checks the transaction request data and its type
- Submit transaction node: filters cross-communication transactions
- Event listener node: captures the inter-blockchain communication requests
- Prepare transaction node: defines the transaction destination and checks if it is valid
- Submit request: sends the transaction to the requested application destination by using TCP node

10.4.3.2 Receiver Flow

Figure 10.13 demonstrates the receiver flow in Node-Red, which contains the following nodes:

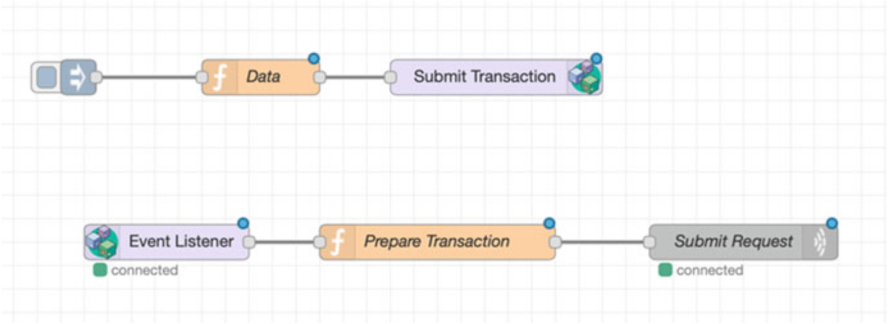


Fig. 10.12 Node-Red sender flow

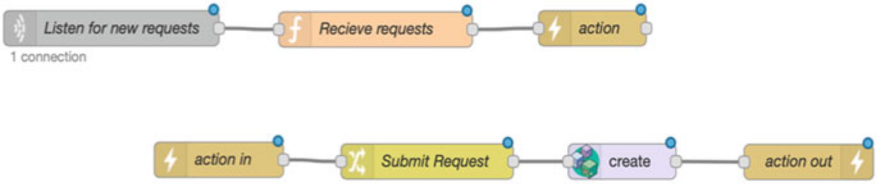


Fig. 10.13 Node-Red receiver flow

- TCP listen node: listens for new received requests
- Receive requests function: extracts the submitted request
- Action node: starts a certain flow once the data is received and extracted
- Action in: defines the start of the action flow
- Submit request: checks if the extracted transaction data is valid
- Create node: submits the transaction to the blockchain application if the data is valid
- Action out: defines the end of the action flow
- Response node: sends the response back to the sender application

10.5 Performance Evaluation

The adequate performance of a blockchain platform is an essential requirement and one of the main challenges in blockchain technology [14]. Overall, the performance of Hyperledger Fabric is high compared to Ethereum and bitcoin [14–16]. The proposed inter-blockchain solution must not affect the performance of the Hyperledger Fabric platform significantly. We conduct performance evaluations on the proposed solutions in terms of throughput, execution time, and latency to validate and evaluate our design. The time taken for a platform to effectively add and perform a transaction is the execution time. Throughput is described as the number of active transactions per second that a network can manage. [16]. Lastly, blockchain latency can be calculated as the time taken by a given network to respond to each transaction. The results of the performance evaluation should be compared with the performance of a conventional system with no inter-blockchain communication solution. The performance is analyzed by varying the set of transactions sent at the same time by the peers. In this assessment, up to 1000 transactions will be sent at once. The transactions can be requests either to read or to write. The results of the performance metrics are averaged over 20 independent runs and then aligned with the experiments conducted in [14, 16]. Figure 10.14 demonstrates the test workflow required to evaluate the performance.

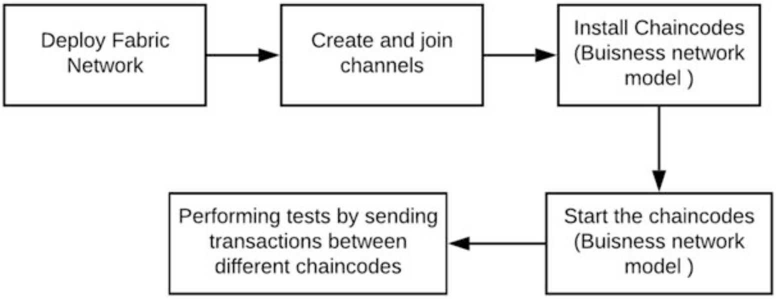

Deploy Fabric Network
Create and join channels
Install Chaincodes (Buisness network model)
Start the chaincodes (Buisness network model)
Performing tests by sending transactions between different chaincodes

Fig. 10.14 Performance evaluation test workflow

10.5.1 Average Execution Time

We evaluated and calculated the execution time for the different proposed stages by varying the workload and analyzing the execution time for the read and write functions.

10.5.1.1 Average Execution Time for Read Requests

In general, the average execution time for read requests increases when the number of transactions is increased. As shown in Fig. 10.15, the execution times in each stage are very similar to those of the control system. This is because a read request does not need to change the state of the ledger and the assists, as it is merely querying saved data. However, the execution time recorded when testing communication between two applications on different networks is the highest, compared to the other scenarios.

10.5.1.2 Average Execution Time for Write Requests

Similar to the execution time for read requests, the average execution time for write requests increases as the number of transactions increases. Execution time results were very similar in the cases of communication between two applications in one channel and communication between two channels. This is because the channel is only a virtual isolation mechanism where all the network members are still within the same network and not a physical separation. Execution time is the fastest in the scenario involving a normal system with a single application and is the slowest for the scenario involving communication between two networks. Figure 10.16 shows the execution time for write requests on different test cases.

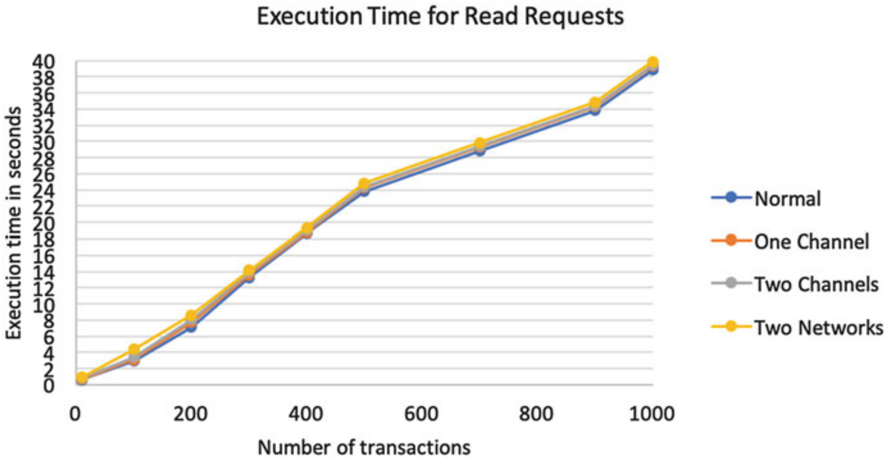


Fig. 10.15 Average execution time for read requests

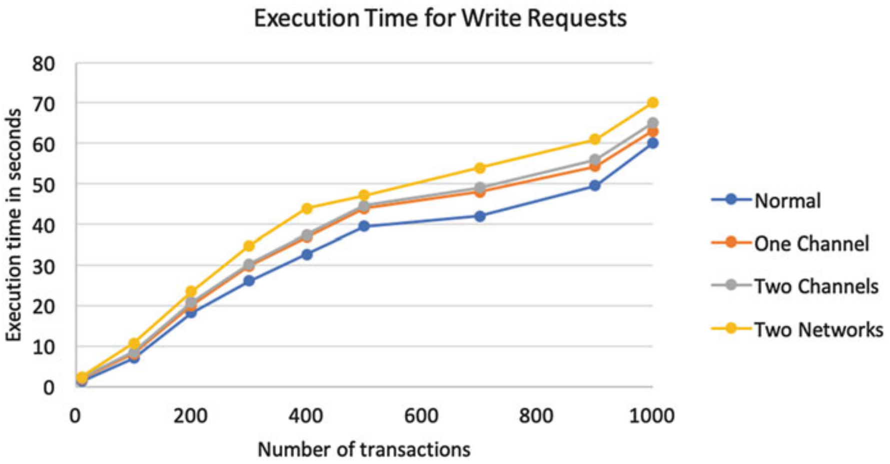


Fig. 10.16 Average execution time for write request

10.5.2 Average Latency

We evaluated the latency for the different proposed scenarios by gradually increasing the transaction numbers up to 1000 transactions in our testing scenarios. We also analyzed latency for the read and write functions.

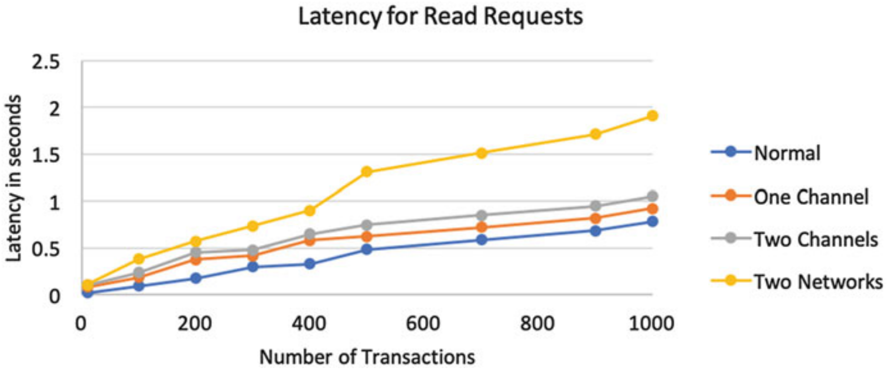


Fig. 10.17 Average latency for reading requests

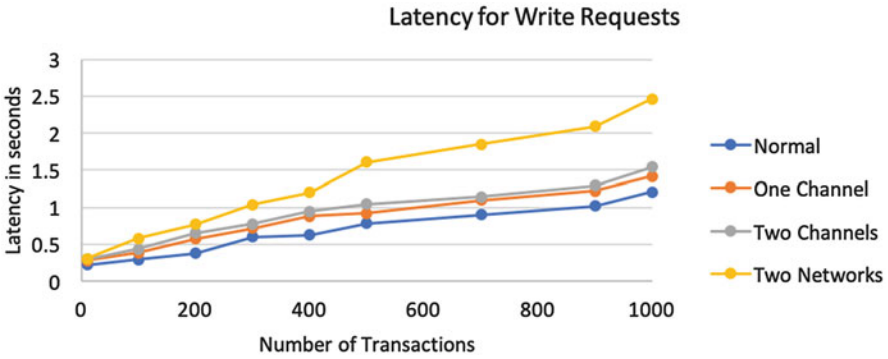


Fig. 10.18 Average latency for write requests

10.5.2.1 Average Latency for Read Requests

Figure 10.17 demonstrates the average latency for different scenarios when the number of transactions is varied. Latency is highest in the case of achieving communication between two applications on different networks. As the number of transactions increases, the difference in latency between the proposed scenarios increases.

10.5.2.2 Average Latency for Write Requests

The average latency for write requests is greater than the average latency for read requests. However, the two functions are similar in all other aspects. The average latency is at its maximum when measured for communication between two applications running on two isolated networks. Figure 10.18 demonstrates the average latency of write requests.

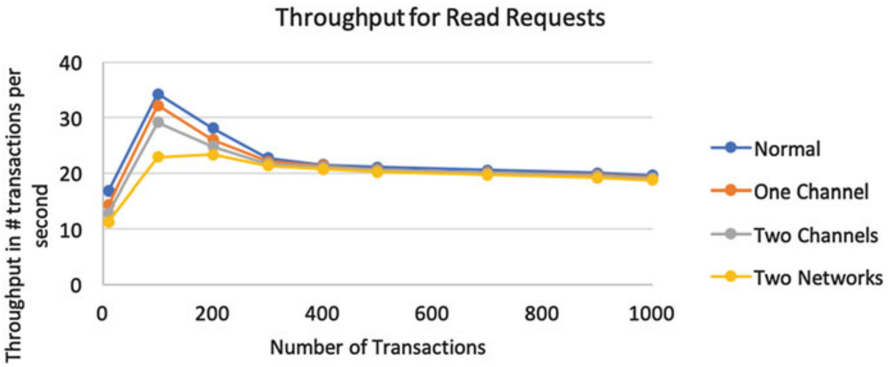


Fig. 10.19 Average throughput for reading requests

10.5.3 Average Throughput

We evaluated the average throughput for the different proposed scenarios by changing the workload up to 1000 transactions. We also analyzed the average throughput for the read and write functions.

10.5.3.1 Average Throughput for Read Requests

Figure 10.19 illustrates the average throughput for read requests. The average throughput is highest in the scenario involving communication in the normal system with one application. It is lowest for the case of communication between two networks. The maximum average throughput is 100 transactions in all different scenarios. As the number of transactions increases beyond 100, the average throughput decreases. Throughput results become more stable when the systems evaluated are given a large number of transactions to process. Moreover, the average throughput is very similar in the different cases when the workload is big.

10.5.3.2 Average Throughput for Write Requests

The result characteristics of average throughput for write requests are similar to those of average throughput in read requests. However, the average throughput for write requests is close to half, or even less than half of the average throughput for read requests. As shown in Fig. 10.20, the maximum average throughput is achieved when the requested workload is 100 transactions. This is because the default block size in Hyperledger Fabric fits 100 transactions.

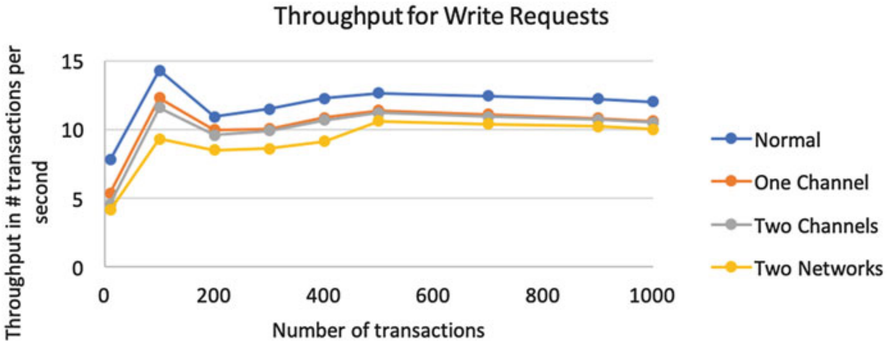


Fig. 10.20 Average throughput for write requests

10.5.4 Results and Discussion

In general, the performance evaluation showed that when the number of transactions is increased, the performance of the proposed solution decreases in all evaluation terms.

The main purpose of conducting the performance evaluation was to check whether the proposed solution significantly affects the overall performance. The results demonstrate that the proposed solution performance reduces the overall performance of Hyperledger Fabric in all metrics assessed. However, the overall performance of the proposed solution is acceptable in light of the interoperability and connectivity achieved by this solution. Table 10.1 shows the difference in performance between the proposed solution and the Hyperledger Fabric.

The differences in performance between the various network structures and the Hyperledger Fabric are very small for the read requests. This shows that the performance of the system was slightly affected by the proposed solution. On the other hand, in write requests, the effect on performance is noticeable, especially in the case of communication between two different networks. Differences in performance are similar between the read and write requests in average latency and throughput results. However, the effect of these operations on performance is clear in the execution time results.

10.6 Related Work

We previously conducted a comprehensive survey in the field of blockchain interoperability [17] in which we classified all available inter-blockchain communication solutions into four groups: sidechain solutions, blockchain routers, smart contracts, and industrial solutions. The sidechain approach is when a separate blockchain is connected to the main blockchain in a scenario where all transactions flow

Table 10.1 Difference in performance between the proposed solution and the Hyperledger Fabric

	One channel	Two channels in the same network	Two different networks
Average execution time for read requests	0.31 s	0.54 s	0.97 s
Average latency for read requests	0.14 s	0.21 s	0.43 s
Average throughput for read requests	1.3 transactions per second	2.39 transactions per second	4.12 transactions per second
Average execution time for write requests	2.6 s	3.19 s	6.28 s
Average latency for write requests	0.16 s	0.25 s	0.48 s
Average throughput for write requests	1.58 transactions per second	1.97 transactions per second	3.18 transactions per second

between the two blockchains [18]. Some examples of projects that used sidechain approach are RSK [19], Elements Alpha [20], Plasma [21], POA network [22], and Mimblewimble [23]. Blockchain nodes serve as routers for transmitting requests between various blockchain networks in the blockchain router method [17]. While many studies have been conducted involving this concept, including Wang et al. [24], Kan et al. [25], Anlink Blockchain [26], Ding et al. [27], and Chen et al. [28], none of them have yet implemented a blockchain router solution in a practical case study. Many industrial solutions verify transactions and nodes through a trusted validator or collection of trusted entities (validators) [17]. Examples of industrial solutions include Cosmos [29], Polkadot [30], ICON [31], AION [32], and Wanchain [33]. In the smart contract approach, smart contracts are used to create interoperable protocols between different blockchain networks [17]. Papers by Li et al. [34], Dagher et al. [35], Bennink et al. [36], and Scheid et al. [37] discuss the only available solutions in the smart contract category. In this research, we used smart contracts to achieve interoperability. Table 10.2 highlights the weaknesses of the available schemes and compares them to our scheme.

10.7 Conclusion and Future Work

Blockchain interoperability is one of the main issues limiting blockchain adoption in the industry ecosystem. Achieving blockchain interoperability between major blockchain platforms will enable various blockchain applications including data storage, finance, and smart contracts. In this chapter, we designed and implemented inter-blockchain communication between Hyperledger Fabric networks. Moreover, this chapter explored the available blockchain interoperability solutions and com-

Table 10.2 Comparison between the current available solutions and our scheme

Current available solutions	Our scheme
Few papers investigated the use of smart contracts to connect different blockchain platforms and networks.	Investigate the application of a smart contract among homogeneous blockchain (Hyperledger Fabric Platform) networks to build interoperable protocols.
Sharing applications and smart contracts between separated blockchain networks are not available.	Design and implement smart contracts (chaincodes) that interact between multiple Hyperledger Fabric networks.
Use a validator to validate and guarantee the node state and its trustworthiness.	Use smart contracts to validate access roles and to validate and guarantee the node state and its trustworthiness.
The performance of the schemes was not evaluated.	Performance evaluation for the scheme in terms of throughput, execution time, and latency.

pared them to the proposed architectures. According to the literature review, there is no complete inter-blockchain communication solution that can address industry ecosystem requirements.

We conducted a performance analysis of the proposed solution under three scenarios and compared the results with a normal blockchain system where only one application interacts with the blockchain.

In general, the performance evaluation showed that increasing the number of transactions affected the performance of the proposed solution, with recorded increases in terms of execution time and latency, and decreases in throughput. The maximum average throughput is achieved when the request workload is measured at 100 transactions for both read and write functions. Comparing between our various scenarios, the evaluated metrics performed with the least efficiency during the interaction between two blockchain applications on different networks.

Future research directions in the field of blockchain interoperability must focus on addressing the current challenges of the available solutions. Researchers need to ensure the reliability, performance, and security of their proposals when developing and designing cross-blockchain communication solutions. Moreover, creating interoperable protocols between heterogeneous blockchain networks will result in a massive paradigm shift for blockchain technology. Another potential research direction is to study the use of smart contracts to connect between different blockchain platforms and networks. A good case study in that regard might involve investigating how to share and connect applications between separated blockchain networks.

Unfortunately, connecting different heterogeneous blockchain platforms is a complicated process that requires time and resources to implement. The main challenge is that each heterogeneous platform has a different architecture, consensus algorithm, and protocol. Most available industrial solutions use validators to check and validate node honesty and node state. The shortcoming of this approach is that there is a single point of failure. A possible future research direction is to apply other implementation strategies, such as routers and sidechains, to Hyperledger Fabric and then compare the resulting architecture to the scheme proposed in this chapter.

References

1. M. Swan, *Blockchain: Blueprint for a New Economy* (OReilly Media Inc., New York, 2015)
2. D. Yaga et al., *“Blockchain Technology Overview*. NIST Draft NISTIR 8202 (2018). https://csrc.nist.gov
3. A.P. Sheth, Changing focus on interoperability in information systems: from system, syntax, structure to semantics, in *Interoperating Geographic Information Systems* (Springer, Berlin, 1999), pp. 5–29
4. S. Schulte, M. Sigwart, P. Frauenthaler, M. Borkowski, Towards blockchain interoperability, in *Proceedings of the International Conference on Business Process Management* (Springer, New York, 2019), pp. 3–10
5. P. Lafourcade, M. Lombard-Platet, About blockchain interoperability. Inf. Process. Lett. **161**, 105976 (2020)
6. Monika, R. Bhatia, Interoperability solutions for blockchain, in *Proceedings of the 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)* (2020), pp. 381–385
7. M. Crosby et al., *BlockChain Technology—UC Berkeley Sutardja Center 2015* (2015) [Online]. http://scet.berkeley.edu/wp-content/uploads/BlockchainPaper.pdf. [Accessed: 05-Dec-2019]
8. S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*. www.Bitcoin.Org (2008), p. 9
9. M. Castro, B. Liskov, Practical byzantine fault tolerance and proactive recovery. ACM Trans. Comput. Syst. (TOCS) **20**(4), 398–461 (2002)
10. *Hyperledger Fabric Project* [Online]. https://hyperledger-fabric.readthedocs.io/en/release/ [Accessed: 05-Aug-2017]
11. C. Cachin, Architecture of the Hyperledger blockchain fabric, in *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, vol. 310 (2016)
12. *Node-RED* [Online]. http://nodered.org/ [Accessed: 24-Jan-2019]
13. *Docker* [Online]. https://www.docker.com [Accessed: 24-Jan-2019]
14. Q. Nasir, I.A. Qasse, M.A. Talib, A.B. Nassif, Performance analysis of Hyperledger Fabric platforms. Secur. Commun. Netw. **2018** (2018)
15. T.T.A. Dinh, J. Wang, G. Chen, R. Liu, B.C. Ooi, K.-L. Tan, BLOCKBENCH: a framework for analyzing private blockchains, in *Proceedings of the 2017 ACM International Conference on Management of Data* (ACM, New York, 2017), pp. 1085–1100
16. S. Pongnumkul, C. Siripanpornchana, S. Thajchayapong, Performance analysis of private blockchain platforms in varying workloads, in *Proceedings of the 2017 26th International Conference on Computer Communication and Networks (ICCCN)* (IEEE, New York, 2017), pp. 1–6
17. I. Qasse, M.A. Talib, Q. Nasir, International blockchain communication: a survey, in *Proceedings of ACM ArabWIC 6th Annual International Conference Research Track Proceedings* (2019)
18. A. Singh, K. Click, R.M. Parizi, Q. Zhang, A. Dehghantanha, K.-K.R. Choo, Sidechain technologies in blockchain networks: An examination and state-of-the-art review. J. Network Comput. Appl. **149**, 102471 (2020)
19. B. Sergio, D. Lerner, *RSK* [White paper] (2015). https://docs.rsk.co/
20. S. Shapiro, et al. *Alpha Networks* [White paper] (2018). https://elementsproject.org/
21. J. Poon, V. Buterin, *Plasma: Scalable Autonomous Smart Contracts Scalable Multi-Party Computation* (2017), pp. 1–47
22. *POA Network* [White paper]. https://poa.network [Accessed: 08-Sep-2018]

23. A. Poelstra, *Mimblewimble*, [white paper], vol. 06 (2016), pp. 1–19
24. H. Wang, Y. Cen,, X. Li, Blockchain router: a cross-chain communication protocol, in *Proceedings of the 6th International Conference Informatics, Environment Energy Application (IEEA 17)* (2017), pp. 94–97
25. L. Kan, Y. Wei, A. Hafiz Muhammad, W. Siyuan, G. Linchao, H. Kai, A multiple blockchains architecture on inter-blockchain communication, in *Proceedings of the 2018 IEEE 18th International Conference Software Quality Reliability Security Companion (QRS-C 2018)* (2018), pp. 139–145
26. Z. Tech, *Anlink Blockchain Network* [White paper] (2017). https://alicliimg.clewm.net/049/389/1389049.pdf
27. D. Ding, *InterChain: A Framework to Support Blockchain Interoperability* (2018)
28. Z. Chen et al., Inter-blockchain communication, in *DEStech Transactions on Computer Science and Engineering*, vol. 0(cst) (2017), pp. 448–454
29. *Cosmos* [White paper]. https://cosmos.network/docs/resources/whitepaper.html [Accessed: 10-Aug-2018]
30. W. Gavin, *Polkadot: Vision for a Heterogeneous Multi-chain Framework* [White paper], pp. 1–21
31. *ICON* [White paper]. https://www.iconpartners.io [Accessed: 10-Aug-2018]
32. *AION* [White paper]. https://aion.network [Accessed: 10-Aug-2018]
33. *Wanchain* [White paper]. https://wanchain.org [Accessed: 10-Aug-2018]
34. W. Li et al., Towards scalable and private industrial blockchains, in *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts (BCC)*, vol. 17 (2017)
35. G. Dagher, et al., Towards secure interoperability between heterogeneous blockchains using smart contracts, in *Future Technologies Conference* (2017)
36. P. Bennink, et al., An Analysis of Atomic Swaps on and between Ethereum, in *Blockchains using Smart Contracts* (2018)
37. E.J. Scheid T. Hegnauer, B. Rodrigues, B. Stiller, Bifröst: a modular blockchain interoperability API, in *Proceedings of the 2019 IEEE 44th Conference on Local Computer Networks (LCN)* (2019), pp. 332–339

Index

A
Absolute finality, 28
Access control list (ACL), 213
Algorithmic governance
  acceptance of, 73
  code and regulations, 68
  data collection, 68
  governance and regulation, 78
  group refused algorithmic governance, 78
  rejection of, 73
  support and rejection of, 72–75
  support for, 73, 74
  trust in, 79
Amazon, 88
Angular application, 216–217
ArcBlock project, 147
Artificial intelligence (AI), 88
Assets management controls, 240
Auditing processes, 38

B
Big Block method, 111
Bitcoin, 2, 109, 111, 228
Blockchain application, 263
Blockchain-based e-voting use case, 120
Blockchain integration, 40
Blockchain interoperability, 253, 271, 272
Blockchain networks, 254
Blockchain scalability
  approaches, 115
  Big Block method, 111
  Bitcoin, 109, 111
  blockchain-based e-voting use case, 120
  byzantine fault tolerance (BFT), 110, 111
  case 01 evaluation and analysis
    graphical analysis, 123–126
    initialization, 120
    initial parameter values, 121
    scalability test, 126–127
    specification, 120
    voting asset, 122
    voting transactions, 122, 123
  case 02 evaluation and analysis
    graphical analysis, 128–130
    initial parameter values, 127
    scalability test, 130
    specification, 128
    voting asset, 128
    voting transaction, 128
  comparative analysis, 131
  dimensions
    client scalability, 113
    horizontal scalability, 112
    node scalability, 113
    vertical scalability (*See* Vertical scalability)
  off-chain solution
    hardware-assisted approach, 117
    parallel mining approach, 117
    payment channel, 116
    redesigning, 117
    sharding, 116–117
  on-chain solutions
    adjustment in block size, 116
    delivery network, 116

M. H. ur Rehman et al. (eds.), *Trust Models for Next-Generation Blockchain Ecosystems*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-75107-4

Blockchain scalability (*cont.*)
- pipelining, 115–116
- open issues and challenges, 132–133
- Proof of Work (PoW), 109, 110, 112
- state of the art, 110–111
- system implementation, 118–119
- testbed, 119
- with VISA, 109

Blockchain-tailored programs, 55

Blockchain technology, 172–174, 253–255
- access controls, 63
- accessibility, 95–96, 98–100
- background, 228–229
- characteristics and integration
  - business processes, 41–42
  - corporate governance, 42–46
  - decentralization, 41
  - immutability, 41
  - regulatory compliance, 46–47
  - transparency, 41
- characteristics of, 89
- confidential data, 101
- credential management, 51–52
- data minimization, 95
- data validation requirements, 102
- direct and nonreversible transactions, 227
- fine-grained access control, 94
- hierarchicality, 96–97
- human resources, 54–55
- information consistency, 49–50
- information privacy, 94
- inputs and interactions, 62–63
- interpretation, 94
- intragovernment communication, 94
- Lithopy sandbox, 69–70
- for medical data, 93
- modification and permanent verifiability, 39
- network complexity, 50–51
- practical recommendations, 94
- privacy-preserving techniques in
  - Blockchain 1.0, 90–91
  - Blockchain 2.0, 91–92
- privacy properties of, 88
- privacy requirements, 89
- private blockchains, 40
- public, private, hybrid, and side-chain models, 39
- realization prospects, 47–48
- record validation and authentication, 40
- regulatory oversight, 53–54
- “Smart Corpus” project, 92
- software interoperability, 52–53
- standards and legislative requirements
  - Control Objectives for Information Technology (COBIT), 58–61
  - ISO/IEC 27001, 61–62
  - Sarbanes Oxley Act (SOX), 55–58
- storage and retrieval, 63
- taxonomies of, 89
- validator-driven requirements, 102
- verifiability, 97–98
- verifiability requirements, 101

Blockchain types, 196–197

Block generation rate (BGR), 113, 114, 116, 132

Business continuity planning controls, 240–241

Business network applications (BNAs), 199

Business network connection API, 213

Business network definition model, 213

Business process management (BPM), 5

Business process model and notation (BPMN), 6

Business-to-business transactions, 93

Byzantine Fault Tolerance (BFT), 110, 111

C

Campaign backers, 3

Centralized database *vs.* blockchain platform, 174

Collaborative business processes, 5, 7–8

Commercial surveillance, 88

Composer Playground testing, 211–213

Control Objectives for Information Technology (COBIT), 58–61
- consensus protocols, 60–61
- development lifecycle, 59–60
- governance, 59
- governance primitives, 59
- IT controls, 59
- pre-implementation preparations, 59
- risk management, 59
- security controls, 60
- transactions and management, 60

Corporate governance, 38
- auditing and assurance, 43
- auditing processes, 43
- Corporate Information Base, 42–43
- risk assessment, 44–46
- service assurance, 44

Cosmos project, 147

Cryptography controls, 245

Cyclic Redundancy Check (CRC), 220

D

Data availability, 220
Data-centric business models, 1
Data confidentiality, 220
Data dependency, 18
 graph, 17
Data exchange, 254
Datafication, 88
Data integrity, 220
Decentralized applications (dApps), 1
 adoption and success, 2
 business process models, 5–7
 collaborative business processes, 5, 7–8
 distributed ledgers, 8
  decentralized reputation systems, 25–26
  immutable process event log, 23
  process components, 22
  smart contract activities, 24–25
  smart contract business process engines, 23–24
  tamper-proof hashed data storage, 22
  trust concerns, 21
  trust-enhancing method, 21
  uncertainty roots, 21
 lock-in effects, 4
 on-chain *vs.* off-chain data storage, 2
 process dependency analysis, 17
 relevancy analysis, 19–20
 software surveillance, 2
 stakeholder-driven platforms, 2
 trust-aware dApp, 8–11
  finality, 27–28
  governance, 28–29
  privacy, 30–31
 trust-by-design approach, 3
 trust concerns in, 4
 uncertainty aggregation, 18–19
 uncertainty identification
  activity-related uncertainty, 13–14
  event-related uncertainty, 14–15
  gateway-related uncertainty, 15–16
  process flow-related uncertainty, 16–17
 usiness process model, 3
Decentralized crowdfunding, 6
Decentralized crypto-ledgers, 38
Denial-of-service attacks (DoS), 195, 234, 235
Dependency graph, 17
Distributed Autonomous Organizations (DAO), 47
Distributed hash table (DHT), 1
Distributed ledger technologies (DLTs), 8
Docker, 260–263
Docker-compose tools, 260–263
Dubai Electronic Security Center (DESC), 209

E

eBay, 88
Economic finality, 28
Electronic health records (EHRs), 170
Ethereum blockchain-based solution, 177
European Union (EU), 53

F

Fabric network deployment, 260–263
FDA incident review & action plan, 185
Firmware (FW), 193

G

General Data Protection and Regulation standards (GDPR), 53–54
Global auditing firms, 38
Government blockchain. *See* Security threat modeling

H

Hardware-assisted approach, 117
Hashed Time-Lock Contracts (HTLCs), 146
Healthcare data, 93
Health information exchange (HIE), 170
Hyperledger-based secure firmware
 blockchain framework, advantage of, 199
 direct and indirect methodologies, 200
 LoRa technology, 200
 methodology and system architecture, 203
  angular application, developing GUI in, 216–217
  business network application development and administration, 213–215
  Composer Playground testing, 211–213
  design procedure, 207–208
  hardware setup, 209
  Hyperledger Fabric development environment setup, 210–211
  problem investigation, 203, 204
  proposed design objectives, 204
  proposed methodology, 205–207
  REST server setup, 215–216
  software setup, 209–210
  treatment validation, 209
 motivation and significance, 195
 outsourced attribute-based signature, 200
 preliminaries, 195
  blockchain types, 196–197
  distributed ledger, 195–196
  Hyperledger, 197–199

Hyperledger-based secure firmware (*cont.*)
- security and performance analysis, 220–221
- system evaluation, 218–220
- work and disadvantages, 201–202

Hyperledger Composer, 199
Hyperledger Fabric networks, 254, 256–258
Hyperledger project, 197–199

I

ICON project, 146
Incident reporting systems (IRSs)
- blockchain technology, 172–174
- centralized database *vs.* blockchain platform, 174
- complementary, 172
- cost analysis, 186–187
- entity-relationship diagram, 180, 181
- implementation framework
  - FDA incident review & action plan, 185
  - incident report (doctor), 183
  - incident report (patient), 184
  - MoH incident review & action plan, 184
  - pharmacy incident review, 185
  - reporting patient registeration, 183
- legal challenges, 189
- message sequence diagram, 180, 182
- multi-stage process, 171
- privacy concerns, 189
- proposed solution
  - framework, 176
  - stakeholders roles and responsibilities, 177, 178
  - system overview, 174, 175
- scalability, 188–189
- security analysis, 187–188
- selfish mining, 189
- system architecture
  - blockchain layer, 179
  - data layer, 179
  - IPFS layer, 179
  - stakeholders, 179
  - user layer, 178
- timely feedback, 172

Information privacy, 88
Information system (IS) auditors, 37
Integrating blockchain technology, 38
Inter-blockchain communication
- blockchain technology, 254–255
- Hyperledger Fabric, 256
- implementation
  - blockchain application, 263
  - Fabric network deployment, 260–263
  - solution deployment, 264–265
- methodology
  - node-RED receiver flow, 259–260
  - node-RED sender flow, 259
  - solution design, 256–258
- network structures and Hyperledger Fabric, 270
- performance evaluation, 265, 270
  - average execution time, 266
  - average latency, 267–269
  - average throughput, 269, 270
  - test workflow, 266
- sidechain approach, 270
- smart contract, 255
- solutions and scheme, 271, 272

Internal corporate auditing functions, 40
International Standards on Auditing (ISA), 37
Internet of Things (IoT), 88, 92, 193, 199
Interoperable blockchain
- applications and use cases
  - projects, 152, 154
  - scientific papers, 152, 153
- blockchain router solution, 138
- design and implementation methods
  - ArcBlock project, 147
  - Cosmos project, 147
  - ICON project, 146
  - scientific papers, 144–146
- future directions
  - projects, 154, 155
  - scientific papers, 153, 155
- industrial solution, 138
- limitation review, 155–156
- literature review, 139–140
- performance and evaluation metrics
  - projects, 151, 152
  - scientific papers, 151
- QAR distribution, 159
- QAR results, 163–165
- scientific papers and projects, 160–162
- sidechain/relay chain solution, 138
- smart contract, 138
- strengths and limitations
  - projects, 149–150
  - scientific papers, 148–149
- systematic literature review (SLR)
  - applied research methodologies, 141
  - data extraction strategy, 143
  - quality assessment rules (QARs), 142–143
  - research questions, 140
  - search strategy, 142
  - study selection, 142
  - synthesis of extracted data, 144
- using NIST, 137

K
Kickstarter, 4

L
Local reporting management system (LMRS), 172
Lock-in effects, 4
LoRa technology, 200

M
Man-in-the-middle attacks (MITM), 195
Medicines and healthcare products regulatory agency (MHRA), 172
Merkelized Abstract Syntax Tree (MAST), 111
Message dependency, 18
Miners, 107
Model-driven software engineering (MDSE), 5
MoH incident review & action plan, 184
MQTT protocol, 200

N
National Institute of Standards and Technology (NIST), 253
National reporting and learning system (NRLS), 172, 173
Network security controls, 243–244
NodeJS environment, 213
Node-RED receiver flow, 259–260
Node-RED sender flow, 259
Non-repudiation, 16, 188

O
Open Connectivity Foundation (OCF) firmware, 200
Open Web Application Security Project (OWASP), 203, 204
Operations security controls, 245–246

P
Parallel mining approach, 117
Private blockchains, 40
Probabilistic finality, 27
Proof of Work (PoW), 109, 110, 112
Public blockchains, 40
Public key infrastructure (PKI), 51

Q
Quality assessment rules (QARs), 142–143

R
Receiver flow, 264–265
RegTech policy, 70, 79–82
Regulatory compliance, 54
Risk management control, 239–240

S
Sarbanes Oxley Act (SOX)
  altering documents, mergers, and acquisitions:, 56
  executive privilege and disclosure, 56–57
  financial accuracy, 56
  IT security controls, 57–58
Scientific papers
  atomic swaps, 146
  blockchain router, 146
  methodologies, 145
  sidechains, 146
  solution type, 145
Secure Sockets Layer (SSL), 207
Security access control, 242–243
Security control framework
  access control, 242–243
  assets management controls, 240
  business continuity planning controls, 240–241
  cryptography controls, 245
  domains, 236
  and ISR control domains, 248
  management and governance controls, 237–239
  mapping, 237
  network security controls, 243–244
  operations security controls, 245–246
  purpose of, 236
  risk management controls, 239–240
  security incident management controls, 241–242
  smart contract and consensus security controls, 244–245
  systems security controls, 246–247
  validation, 247–248
Security incident management controls, 241–242
Security threat modeling
  denial of service (DoS), 234, 235
  elevation of privilege, 235
  information disclosure, 233, 234
  repudiation, 233–235
  spoofing attacks, 231, 232
  STRIDE, 230
  tampering, 232, 233
  transaction creation, 230–231

Self-assessment, 72
Sender flow, 264
SmartCheck, 187
Smart contract and consensus security controls, 244–245
Smart contract business process engines (SCBPEs), 24
Spoofing attacks, 231, 232
STRIDE, 230
Systematic literature review (SLR), Interoperable blockchain
- applied research methodologies, 141
- data extraction strategy, 143
- quality assessment rules (QARs), 142–143
- research questions, 140
- search strategy, 142
- study selection, 142
- synthesis of extracted data, 144

Systems security controls, 246–247

T

Transaction per second (TPS), 113
Transport Layer Security (TLS), 207
Trusted Execution Environments (TEEs), 31
Trust-enhancing Technologies and Techniques (TeTTs), 10
Trust mining method, 10
Trust Studio tool, 11

V

Vertical scalability
- block generation rate (BGR), 113
- block size, 114
- chain size, 114
- latency, 113–114
- storage scalability, 114
- traditional blockchain, 113
- transaction per second (TPS), 113

Y

Yeoman Hyperledger Composer generator, 216, 218