

B-SAHIC: A blockchain based secured and automated health insurance claim processing system

Mahafuja Khatun^a, Ridwan Arefin Islam^b and Salekul Islam^{a,*}

^a*Department of Computer Science and Engineering, United International University, Dhaka, Bangladesh*

^b*Oxford International School, Dhaka, Bangladesh*

Abstract. In recent years, blockchain technology has been successfully used in many distributed environments where different stakeholders, who do not have any trust between them, interact with each other through a secured and transparent platform. The use of blockchain in healthcare insurance industry has not been studied methodically. In this study, we develop a blockchain based Secured and Automated Health Insurance Claim (B-SAHIC) processing system. First, we design an Entity-Relationship (ER) diagram identifying all actors with their respective data and relations between them. We also develop the business model and algorithms for all necessary steps. We implement the system in Hyperledger Fabric and deploy a smart contract to implement these algorithms. We introduce CouchDB to store the OffChain data where we store World-State Database. B-SAHIC provides a web-based portal for all actors who interact with the blockchain. Privacy of clients' claim-related data is ensured by encrypting treatment-related data with a new key that is derived uniquely for each new submission done of clients' personal documents. We have also deployed Hyperledger Explorer, a user-friendly web application tool for monitoring the health state of each node participating on the blockchain network. We have studied the performance of B-SAHIC for two to six peer nodes. Moreover, our performance study shows that B-SAHIC is fast and scalable. In our study, the average query latency is decreased from 0.07 second (for two peer nodes) to 0.02 second (for six peer nodes) in case of 1300 queries per second while the average transaction latency remains unchanged (around 3.5 seconds) for 300 transactions per second. Moreover, B-SAHIC consumes minimum resources, around 350MB only for two peer nodes. We believe that the development process of this blockchain based platform can be applicable for the automation of other insurance industry too.

Keywords: Health insurance, insurance claim processing, blockchain, hyperledger fabric, smart contract

1. Introduction

An insurance is an agreement between an insurance company and an individual client, where the insurance company takes the hedge and provides financial protection or compensation against losses of assets [1]. Similarly, a health insurance handovers the risk of health damage from an individual client to an insurance company. This type of insurance provides

coverage of medical expenses [2] that occurs due to sickness, and thus, protects an insured from potential financial losses. It compensates a wide range of healthcare related expenses including doctor consultation fees, hospitalisations cost, prescriptions drugs, fees incurred due to disability or injury from any accident, accidental death and sometimes dental expenses [3].

A health insurance claiming consists of the bills of a patient who took services from a healthcare. It is a crucial but complicated step of healthcare insurance system [4]. It takes place between an insurance

*Corresponding author. Salekul Islam, Department of Computer Science and Engineering, United International University, Dhaka, Bangladesh. E-mail: salekul@cse.uui.ac.bd.

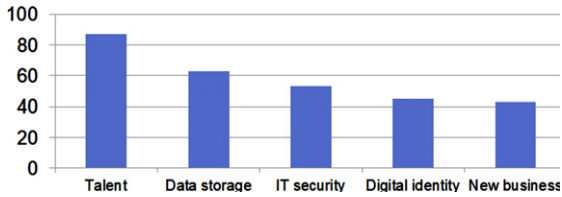


Fig. 1. Challenges faced by the current insurance industries.

company and a healthcare or a client. Generally, the claiming process begins with the client, who receives services from the healthcare service providers. Then, depending on the insurance model, the healthcare provider or the client submits a claim to the insurance company. Now, it goes for validation based on the policies including the patient's insurance plan and value [5].

Finally, for a valid claim, the insurance company pays for legal claims, requested by the healthcare or client according to the insurance policy. The increase of the health insurance cost badly effects the services offered by the insurance company. Poor information management and human error further increase the expenses of the insurance company. One major cause is payment error while dealing out claims, which results in further administrative works called rework for reprocessing a claim. For a typical health insurance company, it accounts up to 30% managerial staffs need to accomplish this rework [6].

Another challenge faced by the insurance company is information leakage that might includes fraudulent claims, abuse, waste and errors, which leads to the increase of the premiums by hundreds of dollars. According to Federal Bureau of Investigation (FBI) [7], about 80 Billion US dollar economic losses arise annually due to insurance fraud. In addition, 18% of individual injury and 21% of forcibly injury claims concluding with a full refund are fraudulent [8]. Note that, both the healthcare provider and the client might be dishonest and can send claims for their personal financial benefits. A client may send the same claim request to multiple insurance companies too [9]. Similarly, a healthcare provider can forward fraudulent claim request by simply changing the patients' admission and discharge dates. Traditional insurance fraud detection techniques are very complex to understand, require expert knowledge and notable inspections, and also consume cost, time and energy [10].

According to the PwC annual report of 2017, insurance sectors are facing some major challenges such as lack of talent, data storage and privacy, IT security,

digital identity and lack of new business models [11]. The significance of these challenges are shown in percentages in Fig. 1. All these challenges are related to technology except talent that requires human capital. Automated claim processing solutions may reduce the aforementioned problems to some extent but they might also introduce new issues including patients' and healthcare providers' information privacy [4]. Therefore, there is a pressing need to deploy a suitable technology that has the potential to conceive distributed, transparent and secured solutions to overcome those challenges along with personal data protection. One such solution can be developed using the permissioned blockchain where information are shared in a distributed common ledger and accessed in a collaborative manner. Moreover, access is limited to the selected and authenticated participants only.

Blockchain is a Distributed Ledger Technology (DLT), where records are stored in distributed ledgers in an immutable manner [12]. These records are shared among multiple participants connected through the blockchain network. Blockchain, in general, consists of a set of ordered blocks that hold secured time-stamp based transactions [13]. Each block is connected with its preceding block using a cryptographic hash. If any change occurs in one block that would alter hashes of all the subsequent blocks. Thus, it preserves integrity and security of transaction logs while eliminates the risk of hacking [14]. In blockchain, the history of each transaction can be recreated at any time [15]. Smart contract is another key element of blockchain technology, which is written into code of lines [4]. Smart contracts are state-driven, self executable contracts that regulate assets [16]. All the business logic and pre-determined access policies with obligations could be implemented by deploying smart contract.

In this research, we demonstrate that permissioned blockchain technology has the ability to provide traceability and transparency in a distributed health insurance claim system. Permissioned blockchain will allow to exchange information among multiple connected participants in a secured way without any central dependency while preserving privacy. This decentralized trust property of blockchain is skillfully suited for the insurance claim processing and provides scalability [12]. In this paper, we propose a blockchain based Secured and Automated Health Insurance Claim (B-SAHIC) processing system that removes the frauds and mitigate the risks during claim request submission and processing. Our proposal also furnishes clients' personal data privacy. Using

Hyperledger Fabric, we have developed a permissioned blockchain-based automated claim processing system. We have identified three main actors—the insurer, the healthcare and the client—who are connected through this system. In our system both the healthcare or the client can submit a claim with proper evidence of treatments. Our system can also detect duplicate treatment application records (if submitted) for the same client. We have made our model user-friendly and easily accessible for all the actors involved through designing different web portals for each category of user. Using this portal users can interact directly with the blockchain network. Our system provides privacy protection for clients' treatment related data that she submits during claim submission. Their data is stored in encrypted format where a data-specific new encryption key is derived run-time for each data. Finally, we have installed a user friendly web application, Hyperledger Explorer, to view blocks, transactions, chaincode, network information and any other relevant data stored in the ledger. The main contributions of this research work are summarized as follows:

- A blockchain based Secured and Automated Health Insurance Claim (B-SAHIC) processing system has been proposed that aims to solve the challenges faced by the existing solutions.
- All the involved actors, necessary workflows, components and their relationship have been identified. All the algorithms for the smart contracts have been designed, and been implemented through chaincode.
- A prototype has been implemented using popular, open-source blockchain solution, i.e., Hyperledger Fabric. A user-friendly, web-based GUI has been developed that provides different interfaces for all the actors to interact with the blockchain network. A performance study has been carried out using Hyperledger Caliper, a bench-marking tool for blockchain technology.
- Customers' data privacy has been maintained. Data has been stored in encrypted form using unique encryption key for each data.

The rest of the paper is organized as follows: section 2 presents the key concepts relating to health insurance and Blockchain technology. Section 3 summarizes the previous work based on blockchain technologies, and other technologies (e.g., data mining, big data and artificial intelligence) that are used to find fraudulent and re-work claims in the insurance systems. Section 4 describes the design architecture

of the proposed health insurance claim processing system including the Entity-Relationship (ER) diagram. Section 5 discusses the implementation of the proposed blockchain platform. It describes the majors steps, business model, algorithms to be implemented in chaincode, development environment, distributed ledger storage structure, and security and privacy services provided by our proposed system. The execution of the system through the GUI we have developed and the Hyperledger Explorer have been described in section 6. Section 7 presents the performance study using Hyperledger Caliper, which has been carried out based on latency and resource utilization. Section 8 compares B-SAHIC with the existing solutions and identifies the distinguishing features of it. Finally, section 9 concludes the paper by identifying the contributions and the future plan.

2. Background

This section provides background of the two key concepts—health insurance system and blockchain technology—that are necessary for our study.

2.1. Health insurance system

Health insurance is a type of insurance that typically provides coverage the risk of health of an individual. It protects one from sudden financial loss and compensates for the costs incurred due to medical treatments and prescribed drugs [3]. Following are different definitions and processes that are important to understand a insurance claim system.

Policy is a contractual agreement (contract) including the conditions and circumstances, under which the insurance company compensates to the clients. The policy holder or the client has to pay an amount to the insurance company known as a premium in monthly, quarterly, half-yearly or yearly basis as per written on the contract agreement for covering the unexpected event financially, suffered by the client.

Insurance coverage, covered by the insurance policy, is the amount of risk or liability for the purpose of recovering the client economically. The most common types of insurance coverage include life insurance, health insurance, auto insurance and homeowner insurance [17].

Claim is a formal request sent by the healthcare (i.e. cashless) or by the client (i.e. reimbursement) to the insurance company for a compensation when the client experiences loss for a covered policy. On ver-

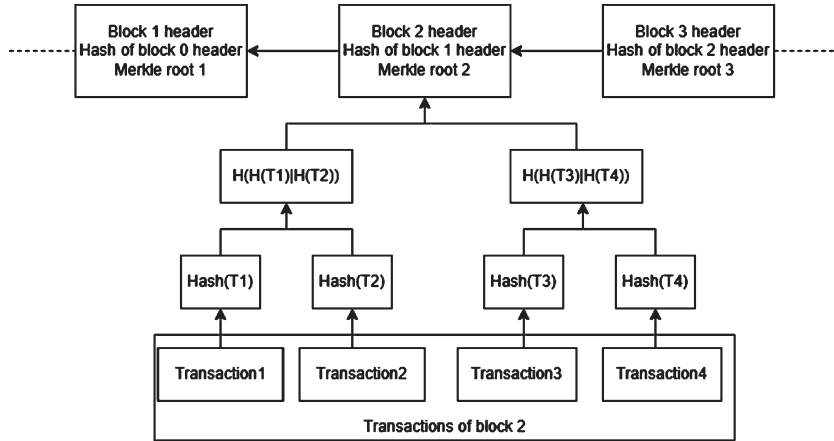


Fig. 2. Generic structure of blockchain.

ification of the claim, the insurance company issues payment to the healthcare or to the client. Health insurance claims can be classified into medical, surgical, pharmaceutical and sometimes dental services [5].

Anomalies arise in claim processing steps either due to unintentional error or intentional fraud and abuse [5].

- Fraud: Intentionally misleading or falsifying the transactions in order to secure benefits or get over payment.
- Abuse: Inappropriate utilization of services, for example prescribing a patient drugs or tests that are not medically required but covered by the policy.
- Error: Accidental faults happened in processing claims are errors. However, if someone makes the same mistake frequently then it might be considered as an abuse. If a claim needs reprocessing it is known as rework. This extra-administrative work could have direct monetary impact due to paying more or less money than what it should have.

2.2. Blockchain technology

Blockchain (a “chain of blocks”) was first introduced in 2008, during the Bitcoin initiative [13, 18] for the purpose of transferring online payments without depending on any central authority or intermediary. Here, blockchain is used as a ledger for recording bitcoin transactions, and due

to cryptographic operations, it provides guarantee of self-indulgence of payments [19].

Figure 2 shows the basic structure of a blockchain. Typically, a blockchain is a chain of multiple blocks containing cryptographic hash of the immediate earlier block, a timestamp and transaction data (represented as a Merkle tree). Transactions are grouped together into blocks and each block is numbered using ascending order starting with 0 means oldest block [20]. Due to this design, once recorded, transactions of any given block cannot be interfered without alteration of all subsequent blocks. If any temper occurs to a block, it will changes the cryptographic hash of each block on the network and thus, participants can detect the tempering.

2.2.1. Classification of blockchain

Blockchain has progressed a lot in the last decade. The first generation of blockchain is the public blockchain that started with bitcoin and much appreciated. Here, anyone can join the network for reading and can submit transactions for inclusion. It is a fully decentralized, transparent, secured and immutable ledger. Ethereum is another popular public blockchain platform.

On the contrary, a single highly trusted and authorized organization who owns the blockchain, wheels the private blockchain while ensuring privacy, faster output and power efficiency. Only authorized participants can join the network and the submission of transactions is limited to them [16]. The third category is hybrid, a combination of private and public blockchains. Only selected participants can verify,

read or write transactions in a hybrid network. Consequently, no single participant can alter the blockchain if he wants.

2.2.2. Hyperledger blockchain

Decentralized public blockchain such as bitcoin [13] allows everyone to participate on the blockchain network, i.e., read and create transactions along with verification and fasten verified transactions to the ledger. Permissioned blockchain such as Hyperledger [21], in contrast, is an open source permissioned Distributed Ledger Technology (DLT), where only selected and authenticated participants can partake in maintaining the shared ledger. Achieving consensus could take time in a distributed system where decision making process in a private blockchain is more centralized but much faster [10]. Linux Foundation has made the Hyperledger Project¹ as an open source distributed ledger framework and code base [22]. It can be customized with members of the network who can see and access the ledger. Its purpose is to advance the blockchain technology. Hence, it is chosen by a large number of developers and adopters [21].

2.2.3. Smart contract

Smart contracts are computer programs that can be defined as self-executing contracts, consisting of rules, regulations and terms of the contract between a buyer and a seller written into code of lines [1, 23]. Without depending on any central authority smart contract can be executed appropriately when all the predefined rules are met [24] and govern the assets of the blockchain network [25].

Two leading open source platforms for executing smart contracts are Ethereum and Hyperledger, where Ethereum, a public blockchain that runs on Ethereum Virtual Machine (EVM). In a smart contract, predefined contracts are written with code first, which can be triggered by a pre-determined oracle. Oracles collect important information from the outside world and feed into the blockchain. Smart contracts need these vital information to execute. At the execution of a smart contract values or assets are being transferred. In the settlement step, this transferred values are settled. For example, in case of an insurance claim, several claim related amounts and numbers will be recalculated.

3. Related works

We have found several efforts to automate the healthcare insurance claim processing system. We divide those works into traditional (without blockchain) and blockchain based solutions.

3.1. Traditional (without blockchain) solutions

Health insurance costs in private sectors increase due to hidden costs overrun on the back-end that might include fraud, errors, abuse and waste. Lu et al. [5] propose an embedding components technique to improve the anomaly detection in health insurance claim. For classifying the subsequent embedded sequences into classes they use a classifier which is a small, fully connected neural network. Srinivasan et al. [26], develop two unique applications in order to detect health insurance claim anomalies using big data analytic. The first application examines and identifies possible abnormalities while the second application produces alerts to reduce or reject payments submitted by the healthcare service providers in case of suspected frauds, abuses, errors and wastes. Park et al. [27] introduces an IT infrastructure, Health Insurance Review and Assessment (HIRA) service that focuses on the Payment Request (PARE) and Claim Review Support (CRS) systems. The healthcare providers can send claims through the PARE system and the CRS system processes those claims by automatic screening for errors, missing items, miscalculations and so on. An interactive prioritization component based claim processing system has been proposed in [28]. Here, the claims containing similar underlying reasons are grouped into clusters and then provided together to an auditor. The aim of this work is to train the auditor to become more efficient and effective and thus to reduce errors in claim processing. The same authors also propose another machine learning based approach [6] that attempts to reduce payment errors called rework during claim processing. This system generates explanations for assisting the auditors by predicting claims those are highly needed to rework. The system involves minimal machine learning skill to install and automate the selection of parameters. Based on the International Classification of Diseases (ICD)-10 Kimura et al. [29] develop a computer-aided, post-entry standardization method using a disease name dictionary and also an anonymous linkage system. A system for categorizing insurance claims relating to diseases, treatments and medicals has proposed by Popowich

¹ www.hyperledger.org

[30] where both Natural Language Processing (NLP) and text mining are used.

3.2. Blockchain-based solutions

We have found a number of blockchain based works that have been carried out for automated insurance claim. However, only a few of them have been developed for health insurance claims.

3.2.1. Blockchain for general insurance claim systems

A blockchain based solution for addressing the performance and security concerns associated in insurance processes is presented in [31]. Different insurance policies have different set of endorsers and thus smart contracts on Hyperledger fabric has been deployed that specifies the proper set of subscribers. Dhieb et al. [10] develops a Smart Insurance System based on Blockchain and Artificial Intelligence (SISBAR) to arrange rules of business, automatic processing of claims, reduce clients risk levels, detect fraudulent claims and reduce human interaction. They have used permissioned blockchain which requires authenticated members for entering into the network and limits the activities of the participants thus shielding sensitive and private information. Sheth et al. [32] study the effect of implementing smart contracts based applications on the traditional insurance markets. They also study how demand vs. supply of insurance market will be affected by the smart contracts. Similarly, Hans et al. [16] study how the insurance market will be affected by the disruptive technologies such as blockchain and smart contracts. Zhou et al. [33] propose a blockchain based Medical Insurance Storage (MISore) System that helps the insurance company to maintain the sum of the patients' medical spending records. Instead of storing a patient's data inside a single server, data are stored in distributed manner among different servers and thus the privacy is maintained. Ciocarlie et al. [22] introduce BlockCIS, a blockchain-based continuous monitoring and processing for cyber insurance system. Note that cyber insurance cover the business liabilities that is occurred due to any kind of breach of sensitive data. For automobile insurance industries, Oham et al. [12] develop a blockchain based Framework for Auto-Insurance Claim and Adjudication (B-FIFA), which is based on permissioned blockchain. Blockchain is being used successfully in storing and accessing Electronic Health Records

(EHR) while the security and privacy of patients' data is ensured [34].

3.2.2. Blockchain for health insurance claim systems

Thenmozhi et al. proposes a tamper free and secured health insurance claim processing system using blockchain framework, which is based on proof of work algorithm to mitigate the risk and reduce the fraud [35]. However, the authors have not actually implemented their idea and no detailed fraud detection algorithm has been presented.

Saldamli et al. [9] propose another blockchain based architecture to detect two kind of frauds: multiple claims submission against one treatment to the same healthcare or several healthcare providers. They use HIPAA (Health Insurance Portability and Accountability Act of 1996) validations that processes the data submitted for the Electronic Data Interchange (EDI) claims and also allows sharing patients' data across multiple healthcare providers.

To provide privacy to patients' He et al. propose a HIPAA compliance system in [4]. According to HIPAA, the clearing house acts as an intermediary between the healthcare and the insurance company to securely transfer the electronic claims while preserving patients' personal information. By introducing blockchain technology they propose a decentralized insurance claim system and thus, eliminate the role of the centralized clearing house. Note that HIPAA is only applicable within the United States.

Note that our proposed system has been compared with different existing solutions and the a summary has been presented later in section 8.

4. Proposed system

This section explains the architecture of B-SAHIC, the proposed health insurance claim processing system.

4.1. Types of health insurance claim

We have considered two types of health insurance claim settlement: cashless and reimbursement. A cashless claim is a direct settlement between the healthcare service provider and the insurance company.

Note that typically, the insurance company makes an agreement with a number of healthcare service

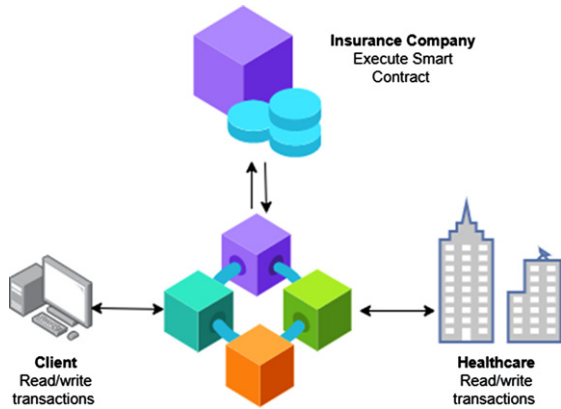


Fig. 3. System model of B-SAHIC platform.

providers (e.g., health professionals and hospitals) who are considered as 'within the network'. A cashless claim is submitted by a networked healthcare after providing health services to a client. Here, the client need not worry about the amount of money need to expense for taking health services from healthcare provider that she has not exceeded their allowable amount. That is why it is cashless. A reimbursement claim is another type of claim request where, the client has to pay first to the healthcare to receive services. Later, she can submit a claim request providing all relevant billing documents issued by the healthcare, to reimburse the amount (either full or a portion depending on the policy) she had spent.

4.2. System model

The high-level architecture of the B-SAHIC system is shown in Fig. 3. Our system considers three main actors: client, insurance company and healthcare. Each actor contains some attributes while holding relationship with other actors. All of these actors are connected through a permissioned and secured private blockchain and agreed on some pre-specified access policies and these policies will be executed by the smart contract. The insurance company is responsible to execute the smart contract and both the client and the healthcare can read or write transactions from or to the blockchain network.

4.3. Entity-relationship (ER) diagram

In developing a system, an Entity-Relationship (ER) diagram helps us to show the relationship among different entities in a visual way. We have developed a comprehensive ER diagram for our B-SAHIC

system, which is shown in Fig. 4. In this diagram, in addition to the three main actors, we have identified several entities including Policy, Payment, Health_professional, Bill, Claim, etc. All the entities are connected through different relationships, such as insured, claim_to, claim_decision, diagnosis, insured, infected, etc. Note that the properties or data belong to an entity are known as attributes in an ER diagram. For example, properties of the Policy entity are policy_no, premium, plan_year, issue_date, policy_type, etc.

In our implementation, we have implemented the major portion of this ER diagram. However, the entities (e.g., Bill, Payment and Health_Professional) shown without any color in Fig. 4 and their related relationships and properties are not implemented in our prototype.

5. Development of B-SAHIC

Health insurance company undoubtedly need to know their customers for efficient business strategy. Therefore, anonymous transaction can not be allowed in this business [36]. This requirements guide us to use permissioned private blockchain with high efficiency and fast transaction speed, where a single organization owns the entire network and access is limited to only predetermined entities.

As opposed to public blockchain networks like bitcoin or Ethereum, where everyone can take part, private blockchain like Hyperledger Fabric is a permissioned distributed ledger network and only accessible to the users who were explicitly invited to join or use the network.

In our study, we use Hyperledger Fabric for developing secured and automated health insurance claim processing platform, which is an open source collaborative effort hosted by the Linux Foundation [37].

5.1. Sequence of messages

We have identified all the major steps of the proposed B-SAHIC system, which are shown as sequence of messages in Fig. 5. In the following we have briefly described the steps:

1. Registration: The role of the insurance company is to register a client into the system when a client purchases a policy and the client then

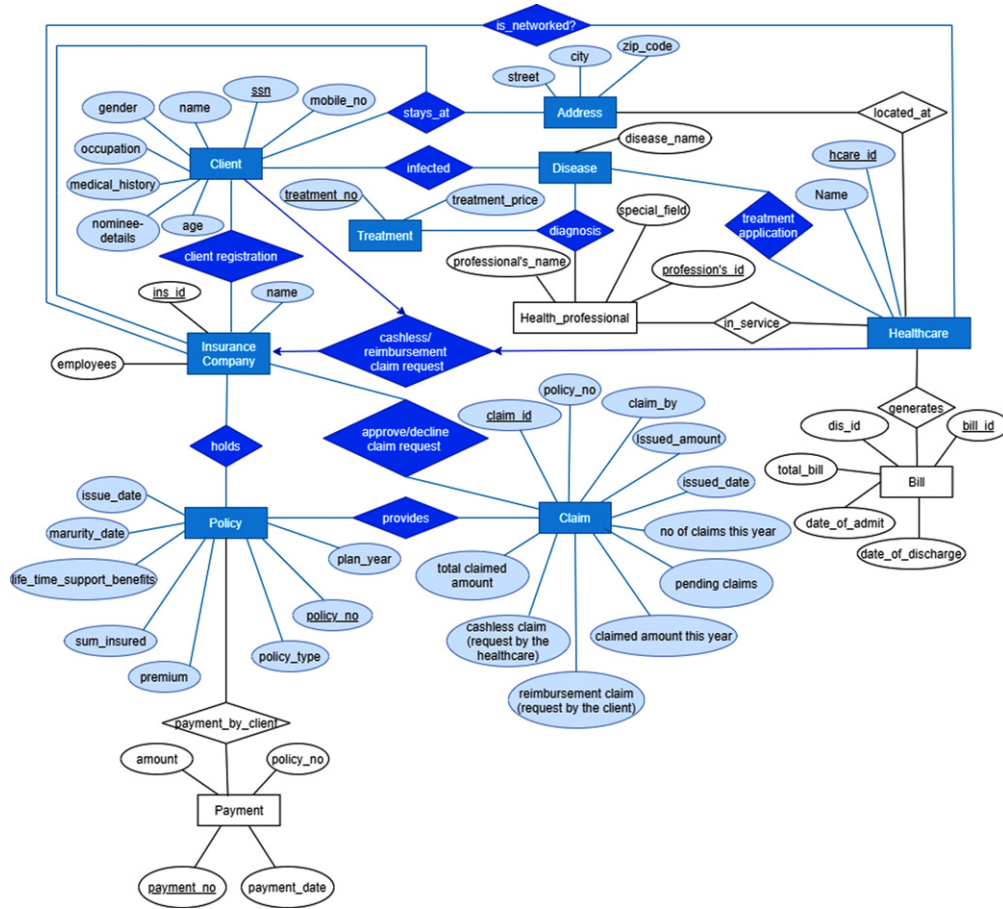


Fig. 4. ER diagram of stakeholders associated with B-SAHIC.

- becomes an insured with a valid policy number.
2. Issue a policy: On successful registration, the insurance company issues a valid policy number (i.e, a unique identification number) to the client.
3. Application for client treatment: A client has to apply for a treatment if she wants to get a desired health service. The client sends this application to the insurance company.
4. Query client data: A healthcare can query the insurance policy about the validity of a client using the client's policy number.
5. Provide health services: This step represents that the client has policy to receive the requested health services, and the service has been provided by the healthcare.
6. Claim request (reimbursement or cashless): Claim request is submitted by the client (reimbursement) or by the healthcare (cashless).

7. Approve or decline claim: The claim request is either approved or declined by the insurance company.

We have described the detailed of these steps in section 5.3 where we have presented the necessary algorithms. Note that the first two steps—registration and issue a policy—are completed first. Table 1 shows the sequence of steps followed in case of reimbursement and cashless claims.

5.2. Business model

Our proposed framework has the following three distinguishable units to represent the actors, information and actions performed by the actors. Figure 6 shows these three units of our B-SAHIC system.

Participant actively involves with the blockchain network. It might be a person or an organization.

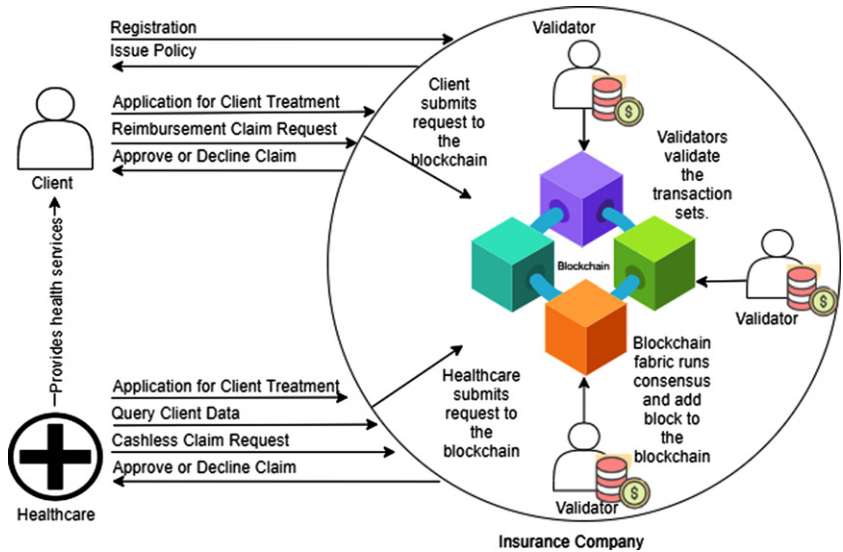


Fig. 5. Major steps of B-SAHIC system.

Table 1
Sequence of steps for reimbursement and cashless claims

Reimbursement claim steps	Cashless claim steps
1. Application for client treatment	1. Application for client treatment
2. Provide health services	2. Query client data
3. Reimbursement claim request	3. Provide health services request
4. Approve or decline claim	4. Cashless claim request
	5. Approve or decline claim.

In our system, the three actors—client, the insurance company and the healthcare—are the participants. Note that they have permission to read/write into the blockchain (client and healthcare) and executing smart contracts (the insurance company)

Asset is something that have either virtual or physical values, e.g., policy details, client details, treatment records, billing details, claim details, payment details, etc. in our system. Note that all the attributes in our ER diagram (shown in Fig. 4) can be considered as asset.

Transaction is an operation performed by a participant or member of the network, to transfer or modify the amount of assets. It could be recording of an event that modifies the blockchain network. In our system it includes client registration, treatment application, reimbursement claim request, cashless claim request, approve claim request and decline claim request. Note that these are the major steps we have identified and shown in Fig. 5.

Table 2
Transaction and Event definition of the proposed blockchain platform

Component	Type	Role
Client Registration	Transaction	Insurance company registers new client into the blockchain system.
	Event	Notifies that the client has been registered with a valid policy number.
Treatment Application	Transaction	Client updates ledger while receiving a treatment.
	Event	Notifies that the client has been applied for a treatment.
Cashless Claim Request	Transaction	Healthcare submits a claim request after giving health services.
	Event	Notifies that the healthcare has submitted a claim request.
Reimbursement Claim Request	Transaction	Client submits a claim request after receiving health services.
	Event	Notifies that the client has submitted a claim request.
Approve Claim Request	Transaction	The insurance company approves a claim request.
	Event	Notifies that a claim request has been approved by the insurance company.
Decline Claim Request	Transaction	The insurance company rejects a claim request.
	Event	Notifies that a claim has been rejected by the insurance company.
Query Client data	Event	Healthcare checks the validity of a client before providing health services.

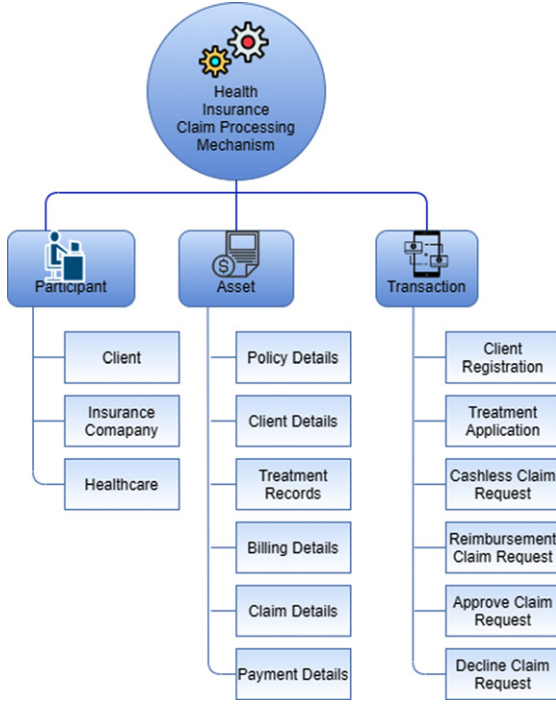


Fig. 6. Business model of the proposed blockchain platform.

Table 3
Notations used in our Algorithms

Notation	Description
TN	Treatment number applying for.
WA	Amount of total life time support money.
CL	Yearly limits of claim numbers.
ML	Yearly limits of claim amount on money.
CTY	Total number of claims made this year.
CMY	Total amount of money claimed this year.
PY	Plan year of health insurance policy.
Client[]	Array of client details.
PC[]	Array of pending claims.
TA[]	Array of treatments applied by a client.
TP[]	Array of treatments price.
TC[]	Array of percentage as treatments coverage.
ClaimID	Consisting of the treatment number applied for and the claim request submitting time in micro second.

Additionally, each transaction consists of one or more events that notify the participants, if a particular transaction is triggered by a contributor of a blockchain network. Table 2 lists the major steps, related transactions and events.

5.3. Algorithms for the smart contracts

We have developed algorithms for the transactions' logic or smart contracts presented in the

business model in Fig. 6. Later, these smart contracts have been implemented through chaincode in the Hyperledger Fabric. Note that programming language, NodeJS has been used to write the smart contract which is the chaincode for Hyperledger Fabric. Notations used in our Algorithms are described in Table 8.

The insurance company registers a client into the blockchain system using Algorithm 1. At the end of the registration process a unique policy number will be generated and client's details will be recorded into the system. We have considered policy number creation time in microseconds and concatenate it with a random number from 1000 to 9999. The resultant unique number is the desired policy number and it will be added to the new client's database. This policy number will be used as the identity of the client in this system.

Algorithm 1 Registration of a Client

```

1: read SSN, Name, Age, Gender,
   Contact_no, Address, Medical_history, Occupation,
   Issue_date, Nominee_details, Maturity_date, PY
2: microtime = [microtime]
3: random = select a random number from 1000 to 9999
4: Policy_no = microtime||random
5: return Policy_no
6: add Policy_no to Client[ ] array

```

A registered client needs to apply for a treatment before taking healthcare services using Algorithm 2. The treatment will be registered into the database only if all the conditions are true: 1) the client is a valid client, 2) the treatment is covered in their policy, 3) she did not apply for the same treatment earlier, and 4) the client has enough money left in their lifetime support wallet. Then the registered treatment will be notified to all the participants connected to the Fabric network informing that the client is going to receive a specific healthcare service.

The healthcare service provider uses Algorithm 3 to query the insurance company to determine whether a client is valid or not. If the client exists the algorithm shows the corresponding client's details, otherwise returns an error.

We have developed Algorithm 4 to implement submission of both cashless and reimbursement claim requests to the Fabric network. The insurance com-

Algorithm 2 Treatment Application for Clients

```

1: read Policy_no, TN, WA
2: read array elements of TP[ ]
3: read array elements of TC[ ]
4: if Policy_no and valid TN exist then
5:   Go to step 9
6: else
7:   print Failed to submit transaction, either
     Policy_no doesn't exist or, invalid Treatment
     number, TN has been chosen.
8: end if
9: if TN exists in array TA[ ] == false then //
     Client did not apply for this treatment before.
10:  go to step 14
11: else
12:  print Client has already applied for this treat-
     ment and return.
13: end if
14: if WA  $\geq$  TP[TN] * TC[TN] / 100 then // Client
     has enough money left in their lifetime support
     wallet.
15:  add TN to the array TA[ ]
16:  print Registration for this treatment has been
     successfully completed.
17: else
18:  print User is registered, but either does not
     have enough money or invalid treatment type
     chosen. Nothing has been updated !
19: end if

```

Algorithm 3 Query Client Data

```

1: read Policy_no
2: search Policy_no
3: if Policy_no exists then
4:  print Client [Name, Policy_no,
     Gender, Age, Medical_history,
     Life_time_support]
5: else
6:  print There is no Client registered with this
     Policy_no.
7: end if

```

pany can implement their own business policy using this Algorithm. The present business policy implemented through this algorithm makes sure that a claim request will be successfully submitted only if all of the following conditions are satisfied: 1) the client is a valid client, 2) the type of treatment is covered in their policy, 3) she did not apply for the same treatment earlier, 4) their total number of claim requests

Algorithm 4 Cashless or Reimbursement Claim Request

```

1: read Poilcy_no, TN, CL, ML, CTY,
   CMY, TA[ ], ClaimID
2: read array elements of TP[ ]
3: read array elements of TC[ ]
4: if Poilcy_no and valid TN exists then
5:  go to step 9
6: else
7:  print Failed to submit transaction, either
     Policy_no doesn't exist or, invalid Treatment
     number, TN has been chosen.
8: end if
9: if CTY  $\geq$  CL then
10:  print Client has already reached their yearly
     quota.
11:  break
12: else
13:  go to step 15
14: end if
15: if CMY  $\geq$  ML then
16:  print Client has reached yearly quota on
     money.
17:  break
18:  if TN exists in array TA[ ] == true then
19:    add ClaimID to array PC[ ] // Adding
     (ClaimID) as a pending claim.
20:    print Claim request has been success-
     fully placed, and Insurance company will be
     notified.
21:  else
22:    print User is registered, but treatment
     record does not exist, Claim unsuccessful! Noth-
     ing has been updated!
23:  end if
24: end if

```

submitted this year (CTY) is less than their yearly allowable number of claims (CL), and 5) their total amount of claim (CMY) submitted this year is less than their yearly allowable amount of claim (ML). Note that, during the claim submission, related proof of taking healthcare services might be submitted for assessment.

When a claim is submitted by a client (reimbursement) or a healthcare (cashless), it will be accepted or declined by the insurance company following Algorithm 5. If all the conditions mentioned below are fulfilled, the claimed amount will be granted fully or partially as per the policy for that particular treatment: 1) the client is a valid client, 2) the type of treatment

671
672
673
674
675
676
677
678
679
680
681
682
683
684

Algorithm 5 Approve or Decline Claim Request

```

1: read Policy_number, TN, ClaimID,
   Accepted, WA, CTY, CMY, PC[ ],
   ML
2: read array elements of TP[ ]
3: read array elements of TC[ ]
4: if Policy_no and valid TN exists then
5:   go to step 10
6: else
7:   print Failed to submit transaction, either
     Policy_no doesn't exist or, invalid Treatment
     number, TN has been chosen.
8:   break
9: end if
10: if ClaimID exists in array PC[ ] == true then
11:   if (Accepted == 1) then //Manual
     authentication of clients' personal treatment
     records goes true.
12:     AmountGranted = TP[TN] *
     TC[TN] / 100
13:     if AmountGranted > (ML-CMY)
     then
14:       AmountGranted = ML - CMY
     // If the amount available in their wallet is less
     than the claimed amount, claimed amount will
     be granted partially.
15:     end if
16:     CTY = CTY + 1
17:     WA = WA - AmountGranted
18:     CMY = CMY + AmountGranted
19:   end if
20:   i ← 0
21:   while (i < PC[ ].length) do
22:     if PC [i] == TN then
23:       replace array PC [i] with 1
24:     else
25:       i = i + 1
26:     end if
27:   end while
28:   if (Accepted == 0) then
29:     print User is registered, but manual
     authentication of clients' personal treatment
     record goes false; Claim has been declined!
30:   end if
31:   print Claim has been granted.
32: else
33:   print User is registered, but ClaimID[]
     doesn't exists in array PC[], So no changes
     made!
34: end if

```

Table 4
Back-end Development Environment

Tools	Description
Hyperledger Fabric	v2.2.2
Docker Engine	Version 19.03.1 with API Version 1.40
Docker Composer	Version 1.25.5
Operating System	Ubuntu Linux 18.04.1 LTS
Programming Language	NodeJS
External Libraries	NodeJS Fabric-contract-api
Hyperledger Explorer	v1.1.4

is covered in her policy, 3) the claimed amount or the number of claims do not exceed the yearly limits, 4) client or healthcare did not send the same claim request earlier, 5) the claimed amount submitted by the client or the healthcare, is available in the client's life time support wallet WA, (6) manual authentication of clients' personal treatment documents goes true (Accepted=1), those were attached during claim request submission for assessment.

Note that each ClaimID is a unique number, which is the treatment number applied for that specific treatment concatenated with the time of submission of the claim request in microseconds. This ClaimID will be generated automatically and stored in the blockchain when a cashless or reimbursement claim request is submitted. This transaction will be visible to everyone connected to the network. Consequently, the submitter (client or healthcare) will not be able to submit the same claim request to multiple insurance companies at the same time. Finally, the accepted amount of claim request will be deducted from clients' life time support wallet amount (WA).

5.4. Development environment

For the development of the system we have used a number of open-source solutions. Tables 4 and 5 summarize the back-end and front-end development environments. At the back-end, the blockchain platform is developed using Hyperledger Fabric [37], docker engine and docker composer [38]. Docker image and docker container are built using docker engine. NodeJS [39] has been used for programming language. This network is deployed on Ubuntu 18.04.1 LTS on 4 virtual CPUs with 4GB memory, along with 20GB storage. Moreover, Hyperledger Explorer v1.1.4 is installed and integrated with our fabric network to view or query blocks.

For our front-end we have used Nginx [40], an open-source high-performance HTTP web server,

Table 5
Front-end Development Environment

Tools	Description
Programming Language	HTML, CSS, PHP
Browser	Google Chrome, Firefox, Microsoft Edge
Web Server	Nginx version 1.14.0

which is used as a web portal for the participants to interact with each other. We have used PHP, a general purpose scripting language that could be embedded into HTML and functionality could be added in a easier way without calling external files. This makes our web sites dynamic and interactive.

Additionally, we have used an advanced form of block cipher encryption AES-128-CBC algorithm for encryption and decryption of patients' personal treatment documents submitted for insurance claim.

5.5. Distributed ledger storage structure

This section explains the structure of the ledger from the proposed blockchain platform. In Hyperledger Fabric, a ledger can be split into two distinct but related parts: (1) blockchain and (2) world state. The blockchain part holds the entire transaction history, while the world state part holds the current values of a set of ledger states and make it easy to find the present state rather than traversing through all of the transaction logs.

Typically, there are two different layers of transaction flow in a blockchain platform. The first one is OnChain transaction, where the transactions are recorded into the distributed ledger. The second one is OffChain transaction, where the transactions are recorded outside the blockchain network. Transactions are stored into a regular database such as CouchDB and StateDB [41]. Note that OnChain transactions take longer time to traverse and need more expense to store as well. Note that blockchains are recorded in OnChain while the world states are recorded in OffChain.

In our project, we have integrated Apache CouchDB database with Hyperledger Fabric to implement the OffChain transaction details. Figure 7 shows our blockchain and OffChain CouchDB storage systems. Apache CouchDB is an open source document oriented state database for deploying index with chaincode. This index is used to query the current state of the read only data stored on blockchain ledger using the JavaScript Object Notation (JSON) queries. CouchDB helps to meet the auditing and

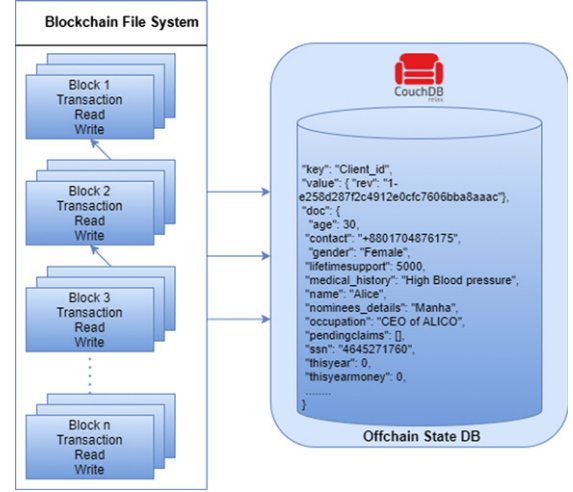


Fig. 7. Distributed Ledger Storage Structure of the proposed blockchain platform.

reporting requirements, also provides efficient and flexible way to query a large dataset from a chaincode.

5.6. Security and privacy of client information

In our proposed framework, we have used AES-128-CBC, a well deployed standard. AES-128-CBC has been used for both encryption and decryption of patients' personal treatment documents submitted for insurance claim. AES is a symmetric key cipher encryption process, where ten transformation rounds are needed to convert a plaintext to a ciphertext [42]. The symmetric key for encryption and decryption is being derived on-the-fly using the following equation. Due to the inclusion of the Current-Time every time a unique key will be derived. Thus, each document will be encrypted with a unique key, where

$$key = hash(convertToString(ClaimID) || convertToString(CurrentTimeinMicrosecond))$$

When an insurance company accepts a claim request from a client, an encrypted copy of their confidential treatment documents will be archived. Note that encrypted documents will be archived in directories which are created automatically according to a client's policy number. Other than the authorized person of the insurance company, no one else is allowed to read or write from/to that directory. Thus, the secrecy and privacy of those treatment records are preserved. Note that AES encryption protects con-

Fig. 8. Insurance company login form.

fidential documents even from brute force attacks [43].

6. Interaction with the system through the graphical user interface (GUI)

We have installed a user friendly web interface for Hyperledger Fabric, i.e., Hyperledger Explorer Client Version:1.1.4. The dashboard view of Hyperledger Explorer shows all kinds of pertinent information stored in the ledger. Explorer dashboard provides a summary section as well as details of blocks, transactions and associated data mined into the fabric network. Besides, it also shows the network information including the name, status, list of nodes etc. along with the chaincodes and transaction details.

As we have mentioned earlier our system consists of three types of users. We have developed different web-based interfaces for all of them.

Login of users

Each user has different login page along with distinct, though valid, username and password to interact with the system. Insurance company's login form is shown in Fig. 8. Similarly two other login pages exist for other two participants (i.e., the client and the healthcare) to connect with the fabric network.

Registration of a new client

An insurance company registers a new client into the blockchain system using the web form shown in Fig. 9. On successful registration, the system will return a unique id number, i.e., the Policy number. Moreover, this event will be recorded to a new block as well as a new transaction, and the client becomes as insured by the insurance company.

Treatment application submission

Treatment application form, shown in Fig. 10, is used by the client to apply for health services and it will be recorded into the ledger for further claim processing. Besides, if duplicate treatment application record exists in the transaction block for the same client, no changes will be occurred until the claim request for the earlier application is processed. In this case, the client will be notified through the system that another application for this treatment already exists in the chain. Otherwise, application will be submitted successfully.

Query Client Data

When a client approaches to a healthcare for a service, the healthcare checks if the client is holding a valid policy number by querying the client data into the ledger. The client query message, shown in Fig. 11 also returns the rest of the amount available in client's account. It confirms that she is eligible to bear the expense of the treatment cost. Moreover, healthcare is entitled to view only name, age, gender, medical history and available amount of the client. Healthcare will not be able to retrieve other personal information.

Claim Request submission

The healthcare submits a claim request through the web portal by using the Cashless Claim Request form shown in Fig. 12. This transaction will be committed into the distributed ledger of Fabric network, thus, considered as an on-chain transaction. The healthcare can also attach all the required documents as evidence of providing the health services to a client. Similarly, a client submits a claim request through the Reimbursement Claim Request form shown in Fig. 13. The client can also submit attachment with this submission form.

Client Registration Form

Name: Yasmin_Elora

Social Security Number: 7293287637

Age: 50

Gender: Female

Mobile no.: +8801918544788

Medical History: High_blood_pressure

Occupation: Service

Nominees Details: Zannah_Rawza

Issue Date: 07/11/2021

Maturity Date: 06/11/2051

Plan Year: 30

Address: Dhaka, Bangladesh

Create Account

Fig. 9. Client registration form.

Treatment Application Form

Policy-No: 16406148514724

Treatment Applying for: 3013

Apply for Treatment

Registration for this treatment has been successfully completed.

Fig. 10. Treatment application form.

View Client Information

Client ID: 16406148514724

Submit

name:	Yasmin_Elora
age:	50
medical_history:	High_blood_pressure
gender:	Female
lifesupport:	5000

Fig. 11. Query Client Data into the Ledger.

Claim approval

The insurance company will find all the submitted claim requests in their Claim Approval Panel as

shown in Fig. 14. All of those transactions will be stored in a list serially along with the request sending date and time. Here, the insurance company can view all the attached documents sent from the claimer and may accept or decline a request after manual verification process. Finally, this accepted amount of money will be deducted from the client's wallet amount as per the insurance policy.

Fig. 12. Cashless Claim.

Fig. 13. Reimbursement Claim.

7. Performance study

We have used Hyperledger Caliper—an open source blockchain benchmark tool, developed by the Linux Foundation to measure the performance of B-SAHIC. Caliper uses a set of predefined use cases to evaluate the performance of a system by producing an HTML report consisting of a number of performance metrics

Fig. 14. Claim approval panel.

Table 6
Development Environment of Hyperledger Caliper benchmark tool

Tools	Version
Caliper	0.4.2
Hyperledger Fabric SDK	2.2.3
Docker Engine	20.10.8
Docker-Composer	1.25.5
Node	v10.24.1
Npm	6.14.2

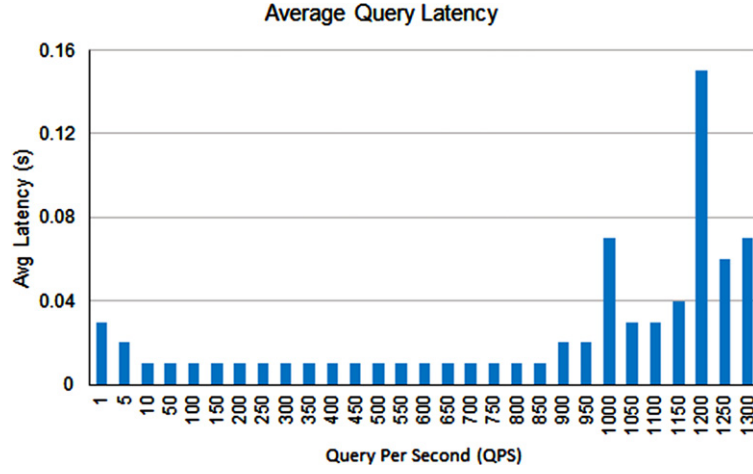
[44], e.g., transaction/read throughput, transaction success rate, transaction/read latency (minimum, average, maximum), resource consumption (CPU, memory, network I/O etc.). The environmental setup of Hyperledger Caliper for our experiment is depicted in Table 6.

Two key indicators for analyzing the performance of a blockchain based implementation are throughput and latency. In this study, we have examined latency, which can be further classified into query latency and transaction latency.

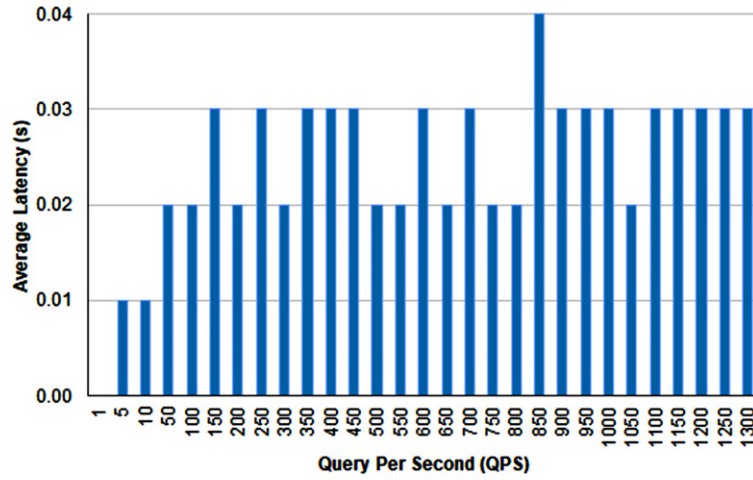
In our performance study we have observed average query and transaction latency in terms of Query Per Second (QPS) and Transaction Per Second (TPS). We have investigated the performance of our proposed B-SAHIC platform by considering different user groups. We have varied the number of peer nodes from two (one client and one healthcare provider) to six (three clients and three healthcare providers), while one order node is always needed.

7.1. Measuring average query latency

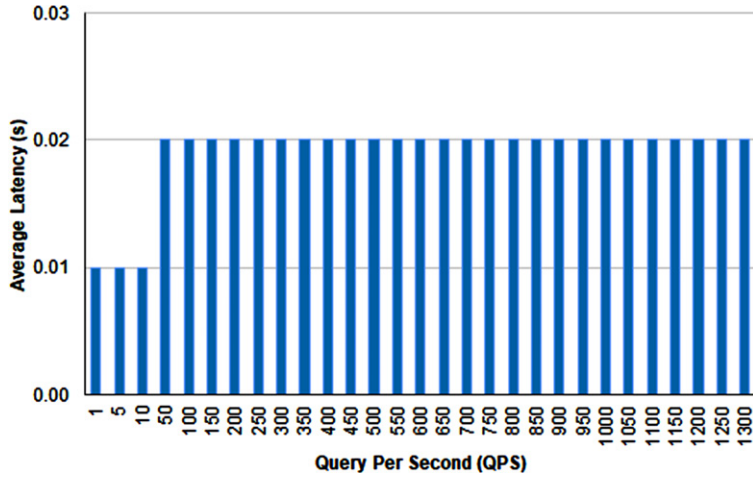
Query latency is the difference between the times when a query request is submitted and when the first



(a) Two peer nodes: one client and one healthcare provider

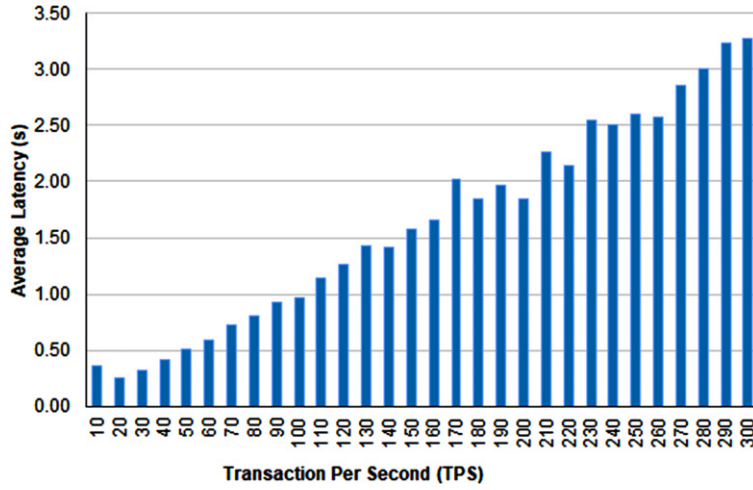


(b) Four peer nodes: two clients and two healthcare providers

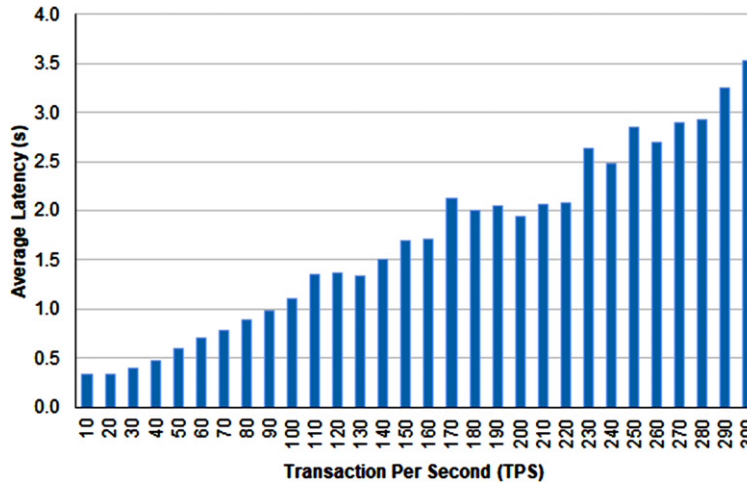


(c) Six peer nodes: three clients and three healthcare providers

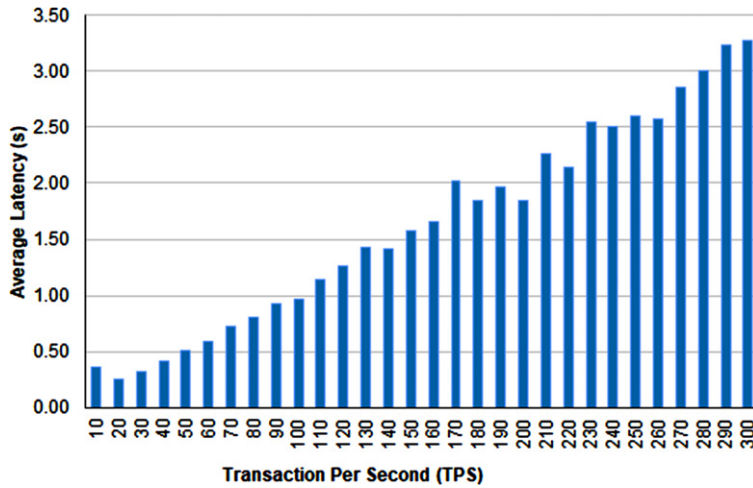
Fig. 15. Average query latency in B-SAHIC.



(a) Two peer nodes: one client and one healthcare provider



(b) Four peer nodes: two clients and two healthcare providers



(c) Six peer nodes: three clients and three healthcare providers

Fig. 16. Average transaction latency in B-SAHIC.

Table 7
Resource allocation analysis for B-SAHIC

Name of peer_container	CPU% (max)	CPU% (avg)	Memory (max) [MB]	Memory (avg) [MB]	Traffic In [KB]	Traffic Out [KB]
peer0.org1.com (Peer node of org1)	9.85	1.92	89.9	89.6	321	337
peer0.org2.com (Peer node of org2)	8.17	1.92	97.5	97.3	185	128
orderer.com (Orderer node)	3.78	0.30	19.4	19.1	95.5	167
couchdb0 (World State database of org1)	8.36	0.88	57.6	57.2	25.0	35.5
couchdb1 (World State database of org2)	10.20	1.01	57.3	56.9	10.0	17.9
ca_org1 (CA for org1)	0.00	0.00	10.9	10.9	0.07	0.00
ca_org2 (CA for org2)	0.00	0.00	10.3	10.3	0.00	0.00
ca_orderer (CA for orderer)	0.00	0.00	10.0	10.0	0.00	0.00

query response is received. The following equation represents how query latency is being calculated in our system:

$$\text{Query Latency} = \text{Query Response Time} - \text{Query Request Time}.$$

In Figs. 15a-15c, for two to six peer nodes, we have examined the average query latency needed for 1 to 1300 QPS. Please note that in case of a query the system performs a read operation into the blockchain, which can be performed concurrently. As a result, the latency does not increase significantly with the increase of the QPS.

It can be seen from Fig. 15a that for two peer nodes the average latency is in acceptable range which is 0.07 second for 1300 QPS. Here, we can ignore the exceptional peak at 0.15 second for 1200 QPS. Since our experiment is running in a desktop computer, the processor is being shared among many processes. Due to some background noise (e.g., other higher priority processes) the processor might be occupied and the average latency has sharply increased in 1200 QPS in Fig. 15a. When we increase the number of nodes to four and six, the average query latency decreases to 0.03 and 0.02 seconds, respectively. As we have mentioned several query/read operations can be performed concurrently. As a result, when we increase the number of queries, more query operations are executed in parallel and the average latency has dropped down, which clearly indicates the scalability of our system.

7.2. Measuring average transaction latency

Transaction latency, on the other hand, is the difference between the times when a transaction request is submitted and when its effect is widely observed across the network. This measurement comprises any settling time including the propagation time needed for the consensus mechanism taking place. The fol-

lowing equation explains how we have calculated the transaction latency by subtracting the transaction submit time from the transaction confirmation time:

$$\text{Transaction Latency} = \text{Transaction Confirmation Time} - \text{Transaction Submit Time}.$$

In Figs. 16a-16c, for two to six peer nodes, we have examined the average transaction latency needed for 1 to 300 TPS. Note that a transaction operation performs write operation into the blockchain and thus takes significantly longer time than query operation. Moreover, transactions cannot be performed concurrently. Therefore, the average latency increases steadily with the increase of the TPS.

It can be observed that the average transaction latency for two peers (shown in Fig. 16a) calculated for 10 to 120 TPS is just over 0.2 second. After that if we increase the number of TPS, the average latency is also increased. For 300 TPS it is increased to 3.25 second. The same pattern is observed for four and six peers also. However, the system demonstrates scalability by not increasing the transaction latency significantly with the increase of number of peer nodes. Results obtained in our experiment fully conform with the results reported in [45], where the key findings were the query and transaction latency do not increase significantly with the increase of peer nodes and Hyperledger Fabric demonstrates scalability only for small and medium enterprises.

7.3. Analysis of resource allocation

A further resource allocation analysis is performed by integrating the Hyperledger Caliper with B-SAHIC. The resource utilization result is measured in terms of the maximum and average usage of CPU, memory, incoming traffic and outgoing traffic, which is shown in Table 7. Here, two peer nodes (peer0.org1.com and peer0.org2.com) are representing the roles of the client and health-

Table 8
Comparison between B-SAHIC and existing works

Features	B-SAHIC	Existing works
Detailed framework	All the involved actors, necessary workflows, components and their relationship, have been identified.	Although a framework has been proposed in [31], no detailed description about the actors, their roles including the relationship between them have been identified.
Implementation of claim requests algorithms	Users are allowed to submit both cashless and reimbursement claim requests based on their roles using B-SAHIC. Algorithms that can detect multiple submissions have been designed and implemented in a proper way through smart contract chaincodes.	Only cashless claim request has been studied in [35]. They have not actually implemented their idea and no detailed fraud detection algorithm has been presented.
Performance study and scalability	Through a detailed performance study B-SAHIC has demonstrated its low and transaction latency, and scalability.	Although [46] and [12] have carried our performance study and showed scalability, none of them address the health insurance system. Note that our scalability results are similar to these two studies.
Design and development of user interface	A user-friendly, web-based interface has been designed and implemented through which users with all types of roles can easily interact with the blockchain system depending on their respective roles and permissions.	A simpler web-based interface has been developed in [46], where all users have the same role and thus, no restriction is applied based on the user's role. Note that [46] is not a solution for an insurance claim processing system too.
Privacy of users' treatment data	B-SAHIC offers a secured claiming processing where all the users' treatment related data are stored in encrypted form. For encryption a unique encryption key and AES-CBC-128 algorithm have been used which ensures privacy of these records with low operational cost.	An attribute-based access control (ABAC) has been used in [4] to implement access control and privacy. Although they have addressed privacy among different insurance companies, nothing has been mentioned about privacy of stored users' data within an insurance company system.
Third party interventions	Third party interventions are completely eliminated while strong trust relationship between the peers are formed.	Some solutions (e.g., BlockCIS [22]) depend on a third party as they have access rights on transactions.

care provider, respectively. One orderer node (orderer.com) controls the transaction flow and the order in which transactions are being written into the blockchain. Two identical copies of the world state database are maintained inside couchdb0 and couchdb1. Three Certificate Authorities (CA) (ca_org1, ca_org2 and ca_orderer) provide certificates for their respective organizations as a Membership Service Provider (MSP) so that they can register and add more peer nodes for their own organization. From Table 7 it is observed that the peer nodes, couchdb database servers and order nodes consume major portion of the resources. Note that overall the average resource utilization is very low.

8. Comparison with existing solutions

Blockchain and smart contract have been identified as a disruptive technology for the insurance industries [16]. Blockchain has been successfully used in other insurance claim processing, e.g., automobile insurance [12] and cyber insurance [22]. Moreover,

researchers have attempted to introduce machine learning and advanced cryptographic primitives to achieve efficient claim processing, fraud detection, users' privacy intelligence for other insurance systems. However, very limited studies have been carried out in health insurance systems.

In this study, we propose B-SAHIC, a blockchain based platform for health insurance claim processing. We have developed the prototype of B-SAHIC and also studied the performance of this system. Table 8 summarizes how we distinguish our work from the existing solutions. Since there is no existing work that can be fully compared with our work, first we have identified the salient features of B-SAHIC. Then we have selected the existing work that addresses the same feature that we are comparing. From Table 8 it can be observed that there exist a very few efforts to automate the health insurance claim processing with the advent of blockchain technology. B-SAHIC is a comprehensive solution that addresses all the necessary features including a prototype implementation, web-based user interface and performance study.

9. Conclusion

In this study, we have presented B-SAHIC: a secured and automated health insurance claim processing system using blockchain technology. The system uses Hyperledger Fabric, a permissioned blockchain, where, information are shared among participants in a distributed ledger. This Fabric architecture provides modular, scalable, trusted and secured foundation for the B-SAHIC platform. We have deployed smart contract chaincode at the back-end of each peer node for furnishing insurance claim process. Clients and healthcare of this network can smoothly interact with the insurance company's server through the web portal we have developed. Our performance study shows that with the increase of the peer nodes from two to six, the average query latency decreases from 0.07 to 0.2 second and the average transaction latency remains unchanged (3.5 seconds). Thus, it can be claimed that B-SAHIC provides a fast and scalable solution. Moreover, the memory consumption (350MB for two peer nodes) is very low as well.

Finally, note that the proposed system is easily extensible in the future version. Any ML or big-data analysis based intelligent method for fraud detection, or advanced cryptographic technique for document verification can be integrated. Moreover, this study is easily applicable for implementing automated claim processing for other insurance industries too.

References

- [1] J. Kagan, Insurance, <https://www.investopedia.com/terms/i/insurance.asp>, 2020. [Online; accessed 18 March 2021].
- [2] What is health insurance? URL: <https://www.icicprulife.com/health-insurance/what-is-health-insurance.html> (accessed: 16.09.2021), 2021.
- [3] J. Kagan, Health Insurance, <https://www.investopedia.com/terms/h/healthinsurance.asp>, 2021. [Online; accessed 18 March 2021].
- [4] X. He, S. Alqahtani and R. Gamble, Toward privacy-assured health insurance claims, in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, IEEE, 2018, pp. 1634–1641.
- [5] J. Lu, B.C. Fung and W.K. Cheung, Embedding for anomaly detection on health insurance claims, in *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, IEEE, 2020, pp. 459–468.
- [6] M. Kumar, R. Ghani and Z.-S. Mei, Data mining to predict and prevent errors in health insurance claims processing, in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010, pp. 65–74.
- [7] N. Smaili, J. Le Maux and W.B. Amar, Corporate fraud: Avenues for future research, in *Corporate Fraud Exposed*, Emerald Publishing Limited, 2020.
- [8] D. Corum, Insurance research council finds that fraud and buildup add up to \$7.7 billion in excess payments for auto injury claims, *Insurance Research Council, Tech Rep*, 2015.
- [9] G. Saldamli, V. Reddy, K.S. Bojja, M.K. Gururaja, Y. Doddaveerappa and L. Tawalbeh, Health care insurance fraud detection using blockchain, in *2020 Seventh International Conference on Software Defined Systems (SDS)*, IEEE, 2020, pp. 145–152.
- [10] N. Dhieb, H. Ghazzai, H. Besbes and Y. Massoud, A secure ai-driven architecture for automated insurance systems: Fraud detection and risk measurement, *IEEE Access* **8** (2020), 58546–58558.
- [11] P. Tasca, Insurance under the blockchain paradigm, in *Business Transformation Through Blockchain*, Springer, 2019, pp. 273–285.
- [12] C. Oham, R. Jurdak, S.S. Kanhere, A. Dorri and S. Jha, Bfica: Blockchain based framework for auto-insurance claim and adjudication, in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, IEEE, 2018, pp. 1171–1180.
- [13] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system. bitcoin. org, URL: <https://bitcoin.org/bitcoin.pdf> (accessed: 24.02. 2020), 2008.
- [14] A.A. Vazirani, O. O'Donoghue, D. Brindley and E. Meinert, Blockchain vehicles for efficient medical record management, *NPJ Digital Medicine* **3**(1) (2020), 1–5.
- [15] F. Lamberti, V. Gatteschi, C. Demartini, C. Pranteda and V. Santamaria, Blockchain or not blockchain, that is the question of the insurance and other sectors, *IT Professional* (2017), 1–1.
- [16] R. Hans, H. Zuber, A. Rizk and R. Steinmetz, Blockchain and smart contracts: Disruptive technologies for the insurance market, 2017.
- [17] J. Chen, Insurance Coverage, <https://www.investopedia.com/terms/i/insurance-coverage.asp>, 2020. [Online; accessed 18 March 2021].
- [18] M. Lischke and B. Fabian, Analyzing the bitcoin network: The first four years, *Future Internet* **8**(1) (2016), 7.
- [19] V. Gatteschi, F. Lamberti, C. Demartini, C. Pranteda and V. Santamaria, Blockchain and smart contracts for insurance: Is the technology mature enough? *Future Internet* **10**(2) (2018), 20.
- [20] I. Nath, Data exchange platform to fight insurance fraud on blockchain, in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, IEEE Computer Society, 2016, pp. 821–825.
- [21] A Blockchain Platform for the Enterprise, <https://hyperledger-fabric.readthedocs.io/en/release-2.2/>, 2020. [Online; accessed 18 March 2021].
- [22] G. Ciocarlie, K. Eldefrawy and T. Lepoint, Blockcis—a blockchain-based cyber insurance system, in *Proceedings of the 2018 IEEE International Conference on Cloud Engineering (IC2E)*, Orlando, FL, USA, 2018, pp. 17–20.
- [23] J. Frankenfield, Smart Contracts, <https://www.investopedia.com/terms/s/smart-contracts.asp>, 2021. [Online; accessed 18 March 2021].
- [24] N. Atzei, M. Bartoletti and T. Cimoli, A survey of attacks on ethereum smart contracts (sok), in *International Confer-*

- ence on *Principles of Security and Trust*, Springer, 2017, pp. 164–186.
- [25] M. Abadi, A. Chu, I. Goodfellow, H.B. McMahan, I. Mironov, K. Talwar and L. Zhang, Deep learning with differential privacy, in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 308–318.
- [26] U. Srinivasan and B. Arunasalam, Leveraging big data analytics to reduce healthcare costs, *IT professional* **15**(6) (2013), 21–28.
- [27] Y.-T. Park, J.-S. Yoon, S.M. Speedie, H. Yoon and J. Lee, Health insurance claim review using information technologies, *Healthcare Informatics Research* **18**(3) (2012), 215.
- [28] R. Ghani and M. Kumar, Interactive learning for efficiently detecting errors in insurance claims, in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2011, pp. 325–333.
- [29] S. Kimura, T. Sato, S. Ikeda, M. Noda and T. Nakayama, Development of a database of health insurance claims: Standardization of disease classifications and anonymous record linkage, *Journal of Epidemiology* **20**(5) (2010), 413–419.
- [30] F. Popowich, Using text mining and natural language processing for health care claims processing, *ACM SIGKDD Explorations Newsletter* **7**(1) (2005), 59–66.
- [31] M. Raikwar, S. Mazumdar, S. Ruj, S.S. Gupta, A. Chattopadhyay and K.-Y. Lam, A blockchain framework for insurance processes, in *9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, IEEE, 2018, pp. 1–4.
- [32] A. Sheth and H. Subramanian, Blockchain and contract theory: Modeling smart contracts using insurance markets, *Managerial Finance* (2019).
- [33] L. Zhou, L. Wang and Y. Sun, Mistore: A blockchain-based medical insurance storage system, *Journal of Medical Systems* **42**(8) (2018), 1–17.
- [34] A. Shahnaz, U. Qamar and A. Khalid, Using blockchain for electronic health records, *IEEE Access* **7** (2019), 147782–147795.
- [35] M. Thenmozhi, R. Dhanalakshmi, S. Geetha and R. Valli, Implementing blockchain technologies for health insurance claim processing in hospitals, *Materials Today: Proceedings* (2021).
- [36] R. Patsch, Survey on Hyperledger Fabric and comparison of Blockchain as a Service Providers. PhD thesis, University of Applied Sciences, 2020.
- [37] Hyperledger – open source blockchain technologies, <https://www.hyperledger.org/>. [Online; accessed 21 November 2022].
- [38] Docker: Accelerated, containerized application development, <https://www.docker.com>. [Online; accessed 21 November 2022].
- [39] Node.js, <https://nodejs.org>. [Online; accessed 21 November 2022].
- [40] Nginx: Advanced load balancer, web server, & reverse proxy, <https://www.nginx.com>. [Online; accessed 21 November 2022].
- [41] D. Pattnaik, Offchain storage in hyperledger fabric, <https://deeptiman.medium.com/offchain-storage-in-hyperledger-fabric-77e28bd99e0c/>, 2020. [Online; accessed 03 November 2021].
- [42] D. Crawford, How does AES Encryption work, <https://proprivacy.com/guides/aes-encryption>, 2019. [Online; accessed 18 September 2021].
- [43] M. Arora, How secure is AES against brute force attacks, <https://www.eetimes.com/how-secure-is-aes-against-brute-force-attacks/>, 2012. [Online; accessed 18 September 2021].
- [44] Hyperledger, Measuring Blockchain Performance with Hyperledger Caliper, <https://www.hyperledger.org/blog/2018/03/19/measuring-blockchain-performance-with-hyperledger-caliper>, 2018. [Online; accessed 10th December 2021].
- [45] D. Khan, L.T. Jung, M.A. Hashmani and M.K. Cheong, Empirical performance analysis of hyperledger Ibs for small and medium enterprises, *Sensors* **22**(3) (2022).
- [46] F. Jamil, O. Cheikhrouhou, H. Jamil, A. Koubaa, A. Derhab and M.A. Ferrag, Petroblock: A blockchain-based payment mechanism for fueling smart vehicles, *Applied Sciences* **11**(7) (2021), 3055.