B-SAHIC: A Blockchain based Secured and Automated Health Insurance Claim processing system

By

Mahafuja Khatun and ID: 012191020

Submitted in partial fulfilment of the requirements
of the degree of Master of Science in Computer Science and Engineering

January 16, 2022

Department of Computer Science and Engineering
United International University

Abstract

In recent years, Blockchain technology has been successfully used in many distributed environments where different stakeholders, who do not have any trust relationship among them, interact each other in a secured and transparent way. An insurance industry might be benefited through the use of an automated, transparent claim submission and processing system. 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 the Entity-Relationship (ER) diagram by identifying all the actors, their related data and the relationship among them. We also develop the business model and algorithms for all necessary steps. We implement the system in Hyperledger Fabric and deploy chaincode and smart contract to implement these algorithms. We introduce CouchDB to store the OffChain data where we store the up-to-date information. B-SAHIC provides a web-based portal for all the actors to interact with the Blockchain. Privacy of clients' claim related data is ensured by encrypting treatment related data with a key that is derived uniquely for each data. We have deployed Hyperledger Explorer, a user-friendly web application tool that facilitates us to explore, invoke, deploy or query the exiting blocks, transactions and associated data. Moreover, our performance study shows that B-SAHIC is scalable and fast, and consumes minimum resources. We believe that the scope of B-SAHIC will not only be limited to health insurance industry. The development process of this Blockchain based platform can be applicable for the automation of any insurance industry.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Insurance is an agreement between an insurance company (i.e., the insurer) and an individual client (i.e., the insured) is being signed, where the insurance company takes the hedge and provides financial protection or compensation against losses of assets [2]. 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 [3] that occur 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 [4].

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 [5]. It takes place between an insurance 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 [6].

Finally, for a valid claim, the insurance company pays for legal claims, requested by the healthcare or client according to the insurance policy.

1.1 Research Motivation and Objectives

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 and also have a direct monetary impact [7].

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) [8], 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 [9]. 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 [10]. 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 [11].

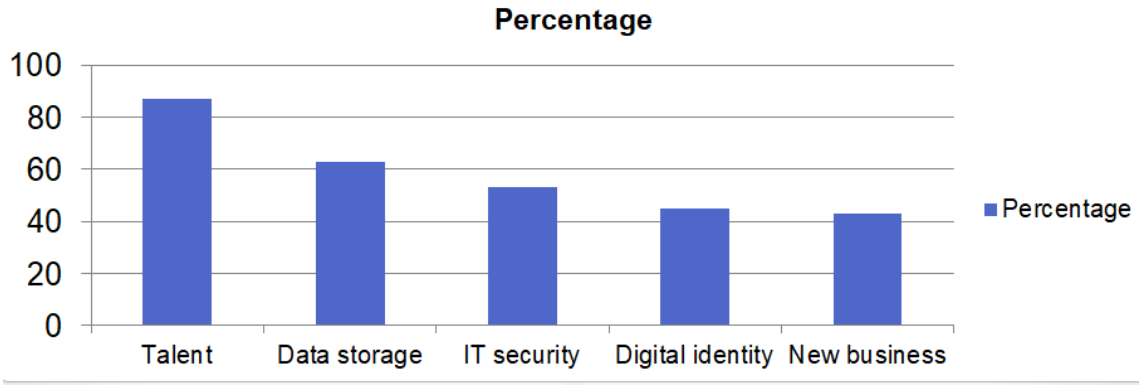


Figure 1.1: Challenges faced by the current insurance industries.

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 [12]. The significance of these challenges are shown in percentages in Figure 1.1. 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 [5].

Furthermore, both the healthcare and the insured person might be dishonest and can send claim for their personal financial benefits. They may send same claim request to multiple insurers [10] for expanding their insurance profits. In the same way, Healthcare can forward fraudulent claim request by simply changing the patients' admission and discharge date into and from the hospital. Traditional Insurance fraud detection techniques are very complex to understand [11] also consume cost, time and energy.

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.

Smart contract is another key element of Blockchain technology, which is written into code of lines [5]. Smart contracts are state-driven, self executable contracts that regulate assets [13]. All the business logic and pre-determined access policies with obligations could be implemented by deploying smart contract. According to Iansti and Lakkhani (2017) five basic principles will be ensured by adopting Blockchain technology.

i. Distributed database: All the participants on the network get access to the entire database and its complete transaction along with authenticating the records of its transaction partners directly. There is no central authentication needed i.e. no intermediary or third party is required for this purposes.

ii. Peer-to-peer transmission: Without any central dependency, communication occurs directly between peers connected on the Blockchain network. Each member stores and forwards transaction to all members presented on the system.

iii. Transparency with Pseudonymity: All the transactions and its associated value occur in the chain are visible to all participants with access to the system. Each user on the Blockchain network has a unique address, one can proof his identity by using this address.

iv. Ir-reversibility of records: Tempering data is quite impossible in Blockchain due to algorithm employed in Blockchain technology. Each record is connected with every other records in the network, which prevents altering data.

v. Computational logic: Computational logic can be deployed in Blockchain through programming. Members can format algorithms and rules that automate and trigger transactions between them.

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 [14].

1.2 Thesis Contributions

In this study, 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 benchmarking tool for Blockchain technology.
- Customers' data privacy has been maintained. Data has been stored in encrypted form using unique encryption key for each data.

1.3 Organization of the Report

The rest of this thesis book is organized as follows: Chapter 2 presents the key concepts relating to health insurance and Blockchain technology. Chapter 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. Chapter 4 describes the design architecture of the proposed health insurance claim processing system including the Entity-Relationship (ER) diagram. Chapter 5 discusses the implementation of the proposed Blockchain platform. It describes the majors steps, business model, algorithms to be implemented in chaincodes, 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 Chapter 6. A benchmark testing is performed and performance is measured using Hyperledger caliper in chapter 7. Finally, Chapter 8 concludes the thesis by identifying the contributions and the future plan.

Chapter 2

Background

This chapter 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 [4]. Following are different definitions and processes that are important to understand a insurance claim system.

Insurer is a person, company or a firm responsible for taking risks of some specific losses and pays for damages incurred by another individual or company through a contractual agreement. The Insurer is the Insurance company indeed, calculating the risk of damages, providing Insurance policies and paying out based on Claims.

Insured an entity i.e. a person, group or organization who makes agreement with Insurance company to receive compensate for the damage of his life or property is called an Insured or a policy holder. For health Insurance, the Insured is a person transferring the risk of his health damage to the Insurer.

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 [15].

Claim is a formal request sent by the client to the insurance company for a compensation when the client experiences loss for a covered policy. On verification of the claim, the insurance company issues payment to the client. Health insurance claims can be classified

into medical, surgical, pharmaceutical and sometimes dental services [6].

Anomalies arise in claim processing steps either due to unintentional error or intentional fraud and abuse [6].

- 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.

.

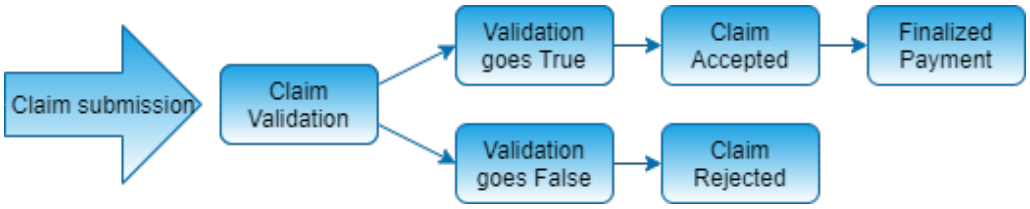


Figure 2.1: Claim processing pipeline

In case of health Insurance, the Policy holder receives health services from the networked healthcare and then the service provider file claims to the Insurer. Then the claim goes for automatic or manual verification (figure 2.1). In validation process pricing is applied [7], if the authentication is accurate the authority accepts the claim, discards otherwise. Finally, the claim get confirmed and payment is sent to the provider.

2.2 Blockchain technology

Blockchain (a "chain of blocks") was first introduced in 2008, during the Bitcoin initiative [16, 17] 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 [18].

Blockchain is a Distributed Ledger Technology (DLT), where records are stored in distributed ledgers in an immutable manner [14]. 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 [16]. 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 [19]. In Blockchain, the history of each transaction can be recreated at any time [1].

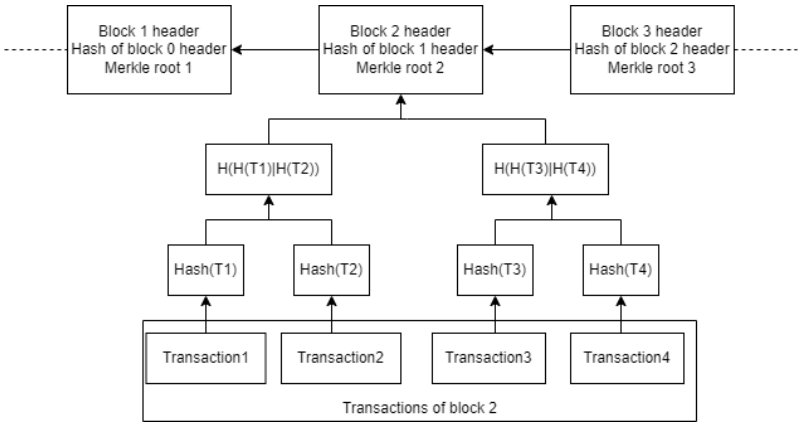


Figure 2.2: Generic structure of Blockchain

.

Figure 2.2 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 Blockchain Basics

Gatteschi et al. [18] have identified some fundamental concepts that we need to know about Blockchain to understand how this technology works.

Transactions: Any exchange of cryptocurrency from an entity to another (i.e. from A to B) is called a transaction [18], which are linked together and stored in each block those are presents in a Blockchain in a consecutive way [1]. Cryptocurrency is just the consequence of transactions that might occurred in past and due to this result, all the transactions from first to last are kept on Blockchain [18].

Block: Blockchain consists of a number of blocks, each of which includes transactions arisen in a particular time frame; also maintains a reference to the precedence block.

Nodes: The Blockchain which is a group of some ordered blocks is stored on each node connected on a network. Each transaction is verified and then added to the new blocks by this network of nodes.

Majority consensus: In a Blockchain network, nodes should not have to rely on central authority. Since, decisions on the network are taken based on majority consensus. For this purpose, to make it mirror the status of the majority, each node changes its local copy of the Blockchain [18].

Mining: Transactions are verified for the determination of authentication and stored them on the new blocks; this process is called mining [1]. Previous transactions occurred in the blocks are checked to verify during mining to determine whether an entity is eligible and having enough money to spend it. After solving a complex computational-intensive mathematical problem by the miners called proof-of-work, a block is inserted into the chain. This problem acts as hindrance to malicious entities for operating the Blockchain by counterfeiting the transactions. Inserting a new block or modifying an existing block would demand to agree on this purpose from majority of nodes; thus the probability of compromising network becomes tremendously low [18].

Wallet: By using wallet, cryptocurrencies are exchanged between peers connected on a Blockchain network. Wallet allows them to store and manage their ether or bitcoin; also called digital or cryptocurrency wallet [21]. Basically, cryptocurrency like bitcoin, ether etc. can't be stored on physical memory. Each wallet is related to one or more sole addresses and stores credentials only; if a user wants to spend money from his wallet she must have to mention the address of the recipient's wallet and expected amount along with validating transactions using his credentials which guarantees that he was the actual originator of the transaction [18].

2.2.2 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 [13]. 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.

2.2.3 Hyperledger Blockchain

Decentralized public Blockchain such as bitcoin [16] 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 [22], 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 [11]. Linux Foundation has

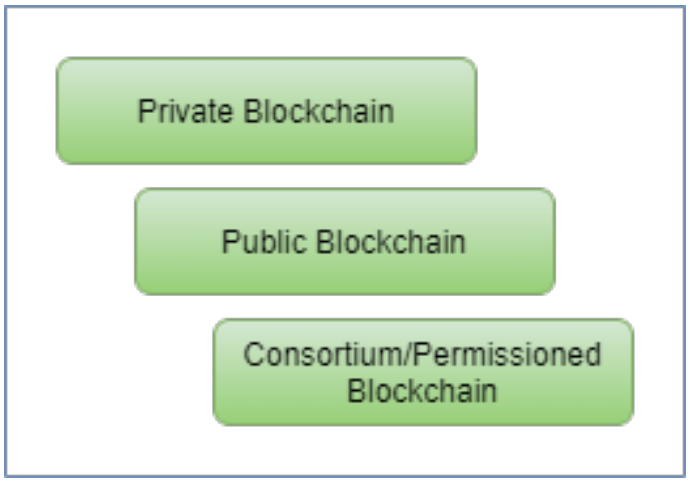


Figure 2.3: Different types of Blockchain Technology

made the Hyperledger Project [1] as an open source distributed ledger framework and code base [23]. 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 [22].

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 [2, 24]. Without depending on any central authority smart contract can be executed appropriately when all the predefined rules are met [25] and govern the assets of the Blockchain network [26].

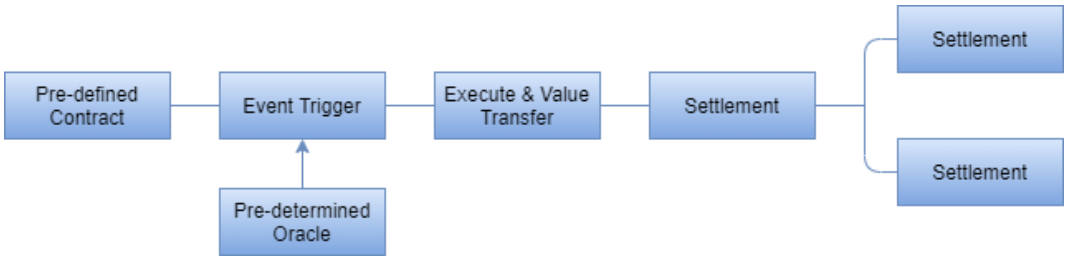


Figure 2.4: The execution of a smart contract.

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). Figure 2.4shows different steps of execution of 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

[1] www.hyperledger.org

are settled. For example, in case of an insurance claim, several claim related amounts and numbers will be recalculated.

2.2.4 The way Blockchain works

Alice wants to transfer a specific amount of cryptocurrency from her wallet (say, address x1y) to John's wallet (assume address v4y) (figure 2.5). Here, cryptocurrency could be any asset with a digital counterpart [1].

In order to make a transfer, Alice creates a statement consisting of the recipient's address along with the desired amount to be transferred and validates it with her credentials (by signing digitally with secret information stored on her wallet); thus ensuring that initiation of the transaction and no one else would be able to alter. Then she broadcast's the transaction to the network. The peer nodes connected to the network verify this transaction to make sure that Alice is the originator of this transaction (by confirming that the message was correctly validated by her credentials) and she is capable to spend the amount to be transferred. This could be done by checking their local copy of the Blockchain and analyzing all the previous transactions (both incoming and outgoing) occupied in Alice wallet's. If both are valid, i.e.Alice initiated the transaction and she is entitled to spend that amount, this transaction will be added to a new block by peer nodes together with other transactions occurring in the same time frame [18]. The newly created block contains a list of transactions and their summary called hash (a fixed size encrypted output calculated from a combination of letters and digits) in its header.

Now, the new block is needed to insert into the Blockchain. In order to do that, nodes participate in a competition to solve a complex mathematical problem called mining and find a random number called proof-of-work through mining process; this number is a combination of the hash of transactions and of the earlier block headers. If a node discovers a possible outcome, he sends the result to the peer nodes for consensus. When majority of consensus agree on it, the block is considered as valid and then added into the Blockchain. Each node also updates its local copy of blocks. The fastest node who finds that random number could also receive a monetary reward in the form of transaction fee [1]. Finally, after completing mining process John receives the money in his wallet sent from Alice.

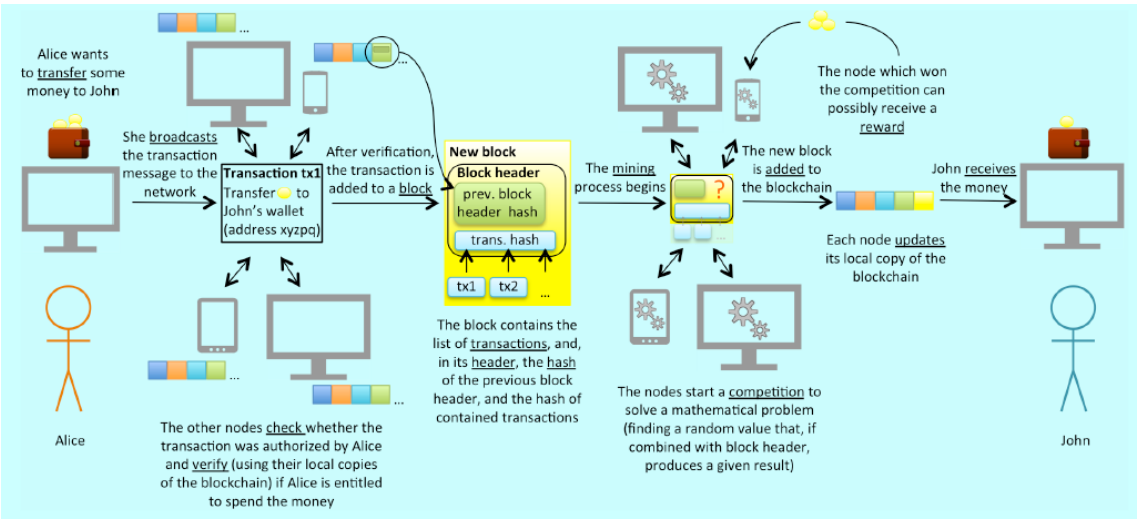


Figure 2.5: Performing transactions on the Blockchain[1].

Chapter 3

Related works

In this chapter, related research works for Insurance and other sectors using different technologies like Data mining, Big Data, Artificial Intelligence and so on are in section 3.1, research study using Blockchain technology is in section 3.2 and an analysis gap with problem definition is in section 3.3 are presented.

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. [6] 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. [27], 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. [28] 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 [29]. 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 [7] 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. [30] 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 [31] 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 [32]. 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. [11] 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. [33] 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. [13] study how the insurance market will be affected by the disruptive technologies such as Blockchain and smart contracts. Zhou et al. [34] propose a Blockchain based Medical Insurance Storage (MIStore) 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. [23] 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. [14] develop a Blockchain based Framework for Auto-Insurance Claim and Adjudication (B-FICA), 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 [35].

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 [36]. However, the authors have not actually implemented their idea and no detailed fraud detection algorithm has been presented.

Saldamli et al. [10] 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 [5]. 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.

3.3 Gap Analysis and Problem Definition

Blockchain and smart contract have been identified as a disruptive technology for the insurance industries [13]. Blockchain has been successfully used in other insurance claim processing, e.g., automobile insurance [14] and cyber insurance [23]. 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. But very limited studies have been carried out in health insurance systems.

In this study, we propose a Blockchain based platform for health insurance claim processing. Table 3.1 summarizes how we distinguish our work from the earlier efforts. Since there is no existing work that can be fully compared with our work, we have selected the existing work that addressed the same feature that we are comparing.

Table 3.1: 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 [32] 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 [36]. They have not actually implemented their idea and no detailed fraud detection algorithm has been presented.
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 [37], where all users have the same role and thus, no restriction is applied based on the user's role. Note that [37] 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 [5] 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 [23]) depend on a third party as they have access rights on transactions.

Chapter 4

A proposed B-SAHIC system

This chapter 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 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 Figure 4.1. 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.

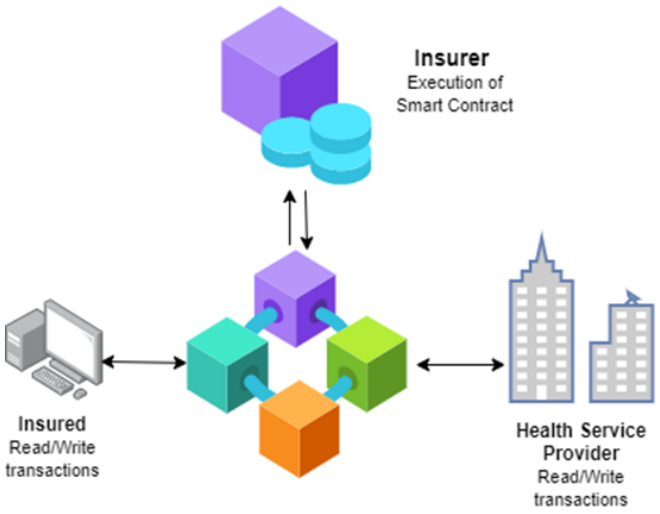


Figure 4.1: System model of B-SAHIC platform.

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 Figure 4.2. 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 Figure 4.2 and their related relationships and properties are not implemented in our prototype.

4.4 Access to information

In a permissioned Blockchain, each participant is connected on the network, have reading access of other members information as shown in Table 4.1, for example, a Client can read an Insurer's name, location and so on. When he holds a Policy, gets read/write access to Policy-details, Payment-details and Claim details. Similarly, a Health service provider have only reading access to some information's of his Patients' like Name, Policy-no etc. to verify whether the Insurance card provided by the Insured is valid or not. Other particulars such as Gender, Age, Medical_history is required to know for giving proper treatment to that patient. Insurer, in contrast, is responsible to execute the self-enforcing

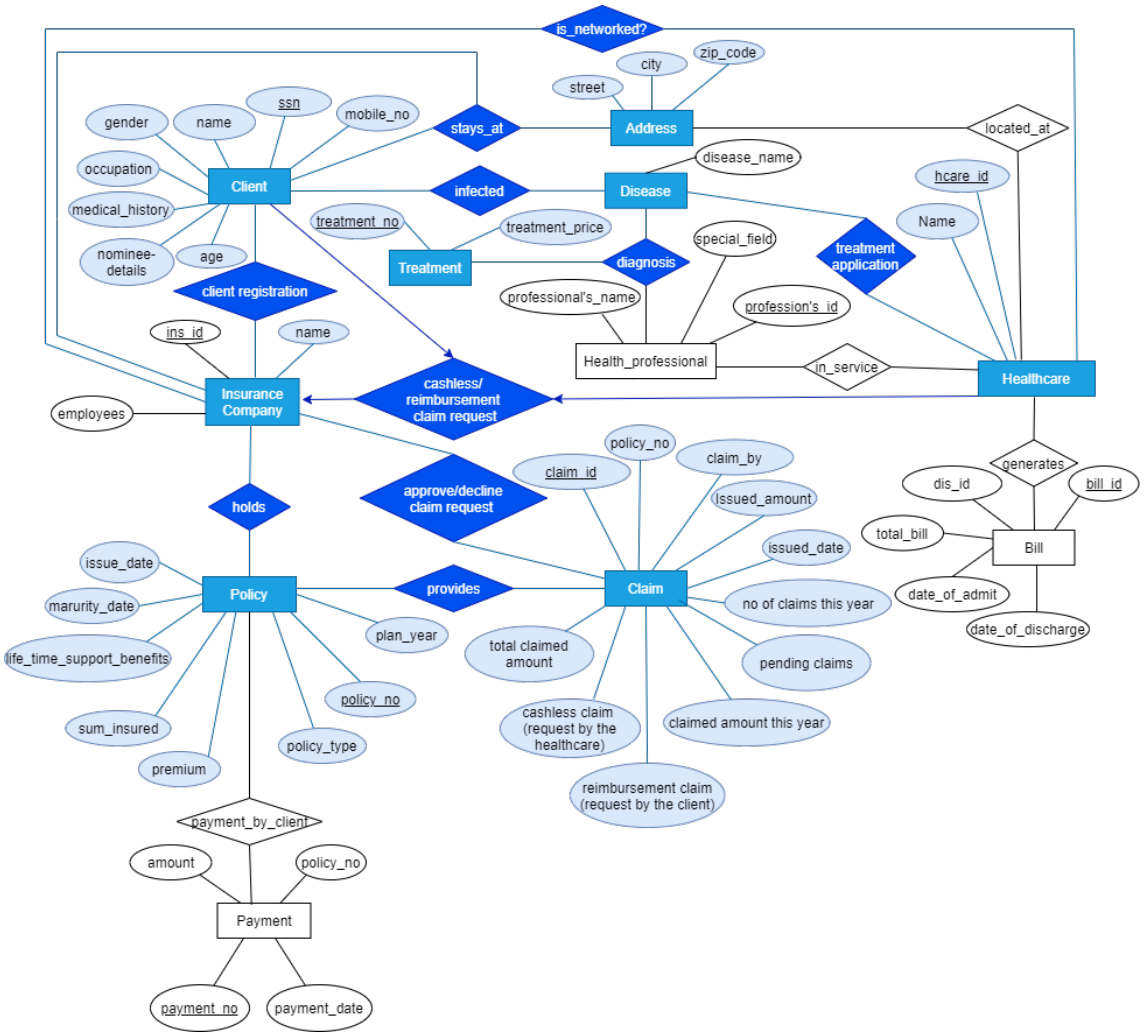


Figure 4.2: ER diagram of stakeholders associated with B-SAHIC

complex function i.e. smart contract in Blockchain network and thus they can read all the Clients and Service provider's information, payment details of premiums from the Insured and billing details of healthcare providers. Since, Blockchain is a temper-resistance ledger, no one else can change/modify the data.

To achieve a common goal, business services and policies could be distributed among many members participated in the network through smart contract [23].

4.5 Process of health Insurance Claim

At the beginning, when a customer decides to purchase a Policy, she must have to fill out an application form collected from the Insurer, containing details of the Client's information along with a fee and sends it to the Insurance company. When the company founds everything alright, they accept the application and issue an Insurance Policy to the Client specifying all the terms and conditions with amount to be paid as premiums in monthly, quarterly or yearly for the desired services. After purchasing Policy and paying specific

Table 4.1: Access to Information from entity to others

Entity	Other Entities	Access to Information
Client	Insurance_Company	name, location, policy_details, payment_details, claim_details.
	Healthcare	name, location, treatment, billing_details
Insurance_company	Client	name, ssn, contact, gender, occupation, age, medical_history, payment_details.
	Healthcare	hcare_id, name, location, name_prof, special_field, disease_name, dis_id, price, billing_details.
Healthcare	Client	name, policy_no, gender, age, medical_history, wallet_amount
	Insurance_Company	ins_id, name, location, claim_details,

amount of premiums, an Insured becomes eligible to take coverage of health damage form the Insurer. When the Insured gets infected from disease, she may go to the networked Health service provider for the purpose of cashless treatment and sends her Insurance card to them for verification. If the card is valid with a legal Policy no, the Client is considered as genuine and health services are provided,if not, she can take the services through cash payment. As opposed to, non cashless she can take the services from a disconnected Healthcare with cash Figure 4.3.

After providing required treatments to the Insured, the Provider generates bill and submits Claim to the relevant Insurer along with all the copy of bills, evidence of treatments and diagnosis. The Claim is then verify for validation by the Insurance provider. The results may approval, deduction or rejection of Claim. If it is an invalid Claim or need to made changes, the Insurer return it back to the Healthcare and ask for providing sufficient and authenticated documents. However, if they found the Claim is valid, the Claimed amount will be deducted from the Insured's life time support wallet as Insurance coverage and make payment to the service provider. The same happens for reimbursement Claim request, excepts it is submitted by the Insured herself.

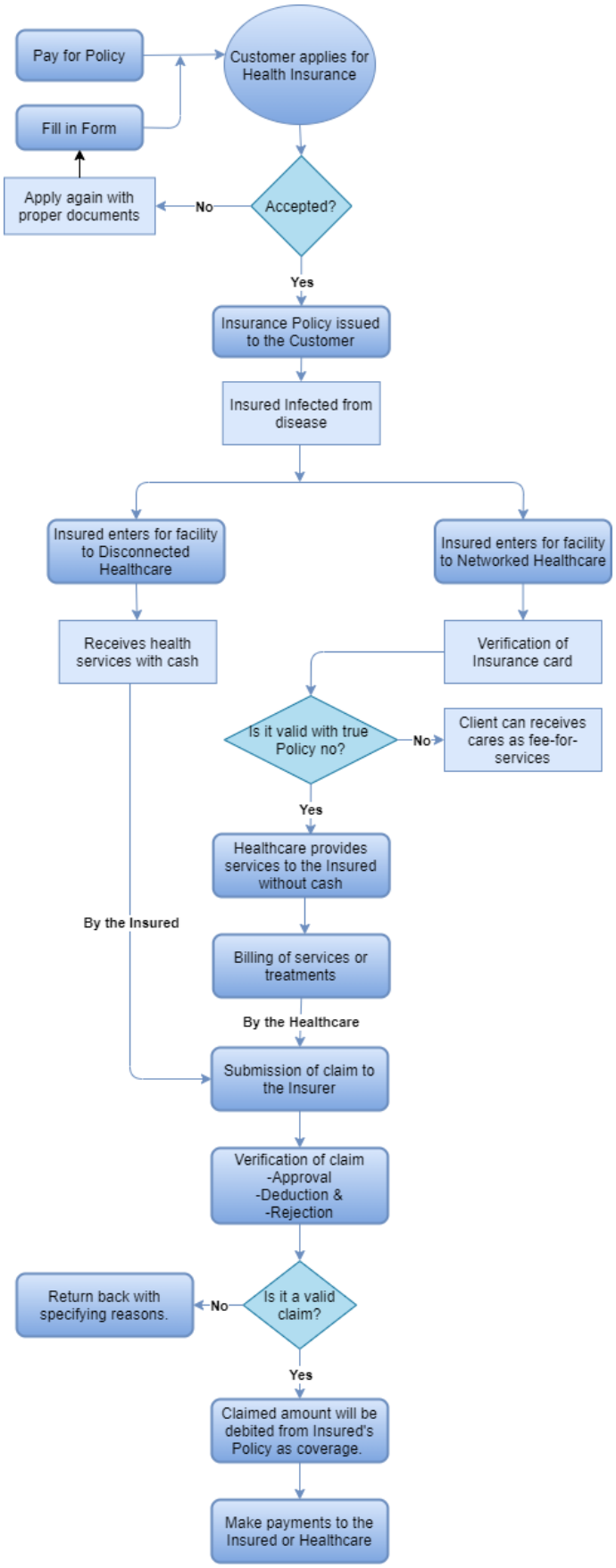


Figure 4.3: Flowchart of health Insurance Claim process.

Chapter 5

Development of B-SAHIC

Health insurance company undoubtedly need to know their customers for efficient business strategy. Therefore, anonymous transaction ca not be allowed in this business [38]. 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, 39].

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 Figure 5.1. 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 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.

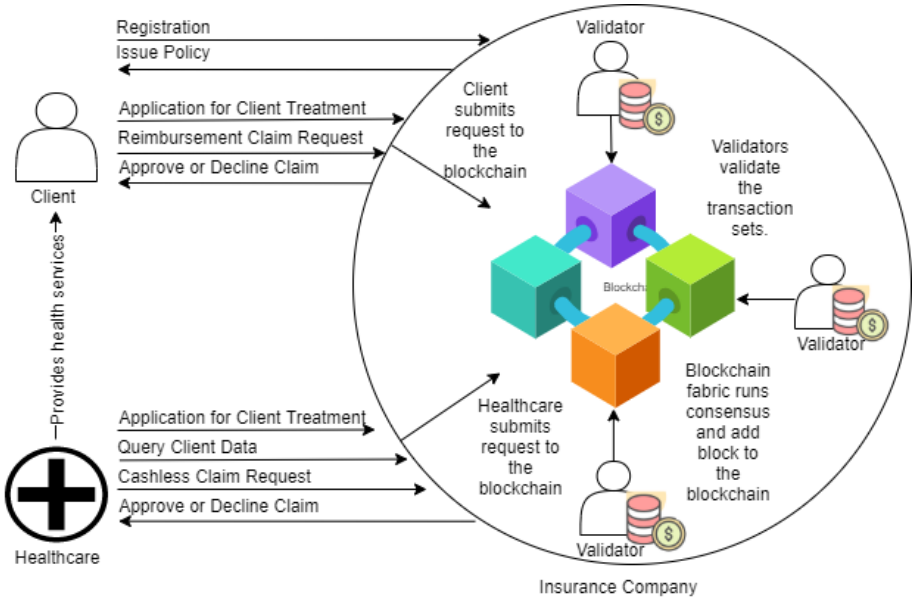


Figure 5.1: Major steps of B-SAHIC system

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 5.1 shows the sequence of steps followed in case of reimbursement and cashless claims.

Table 5.1: 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
4. Approve or decline claim	4. Cashless claim request
	5. Approve or decline claim.

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 5.2 shows these three units of our B-SAHIC system.

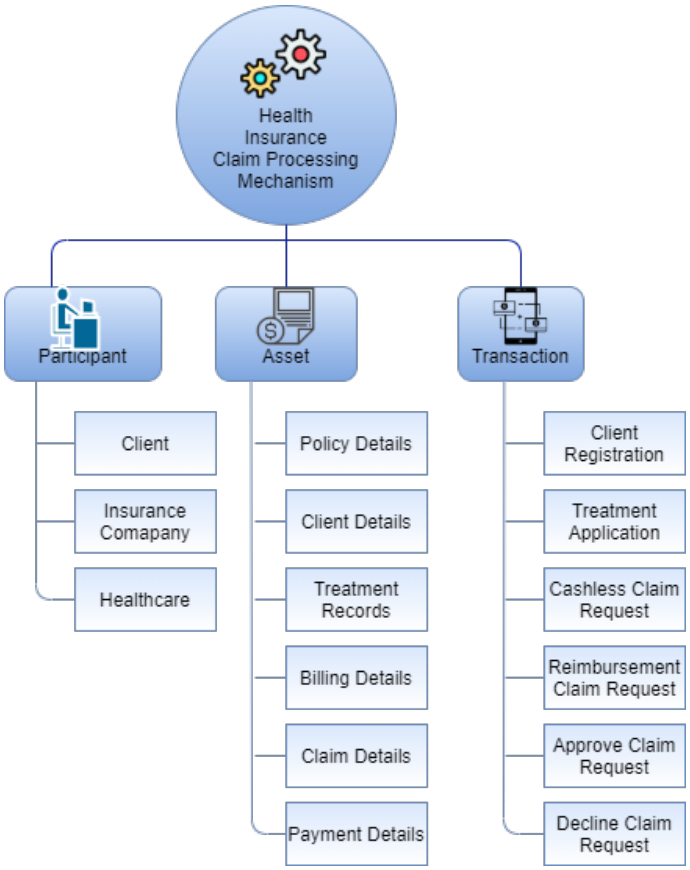


Figure 5.2: Business model of the proposed Blockchain platform.

Participant actively takes part into the Blockchain network. It might be a person or an organization. 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 Figure 4.2) 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 Figure 5.1.

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 5.2 lists the major steps, related transactions and events.

Table 5.2: 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.

5.3 Algorithms for the smart contracts

We have developed algorithms for the transactions' logic or smart contracts presented in the business model in Figure 5.2. 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 5.3.

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.

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

Table 5.3: 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.

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: $microtime = \lfloor microtime \rfloor$
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

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 company 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 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

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 treatment 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**

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 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.

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 successfully placed, and Insurance company will be notified.
21: **else**
22: **print** User is registered, but treatment record does not exist, Claim unsuccessful! Nothing has been updated!
23: **end if**
24: **end if**

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 5.4 and 5.5 summarize the back-end and front-end development environments. At the back-end, the Blockchain platform is developed using Hyperledger Fabric, docker engine and docker composer. Docker image and docker container are built using docker engine. NodeJS has been used for programming language. This network is deployed on Ubuntu

```
async claimrequest(ctx, policy_no, treatmentapplyingfor, claimid) { // add treatment record to a user
    var treatementprice = [200, 300, 400, 500, 600, 200000];
    var availpercentage = [10, 20, 30, 40, 50, 5];
    treatmentapplyingfor = treatmentapplyingfor - 3011;
    //check if user is in the network
    const carAsBytes = await ctx.stub.getState(policy_no);
    if (!carAsBytes || carAsBytes.length === 0) {
        return (`Failed to submit transaction! Client ID: ${policy_no} does not exist`); //Client is not in the network
        //After throw the function terminates
    }
    var result = carAsBytes.toString();
    result = JSON.parse(result);
    const client = JSON.parse(carAsBytes.toString());
    //check if client treatment record exists
    if (client.thisyear >= 3) {
        return (`Client ID: ${policy_no} has already reached yearly quota.`);}
    if (client.thisyearmoney >= 2000) {
        return (`Client ID: ${policy_no} has already reached yearly quota on money.`);}
    if (client.currentoperations.includes(treatmentapplyingfor) == true) {
        //Client treatment record exists
        console.log("loading claim...\n");
        client.pendingclaims.push(claimid);
        var i = 0;
        while (i < client.currentoperations.length) {
            if (client.currentoperations[i] === treatmentapplyingfor) {
                client.currentoperations.splice(i, 1);}
            else {
                ++i;}}
        await ctx.stub.putState(policy_no, (JSON.stringify(client)));
        return ("Claim request has been successfully placed, and insurance company will be notified.");}
    else {
        return ("User is registered, but treatment record does not exist, claim unsuccessful! So no changes made!");
    }
}
```

Figure 5.3: Chaincode for cashless/reimbursement claim request

```
async claimapprove(ctx, policy_no, claimid, resulta) { // add treatment record to a user
    var treatementprice = [200, 300, 400, 500, 600, 200000];
    var availpercentage = [10, 20, 30, 40, 50, 5];
    var claimidstr = claimid.split("-");
    var treatmentapplyingfor = claimidstr[0] - 3011;
    const carAsBytes = await ctx.stub.getState(policy_no);
    if (!carAsBytes || carAsBytes.length === 0) {
        return (`Failed to submit transaction! Client ID: ${policy_no} does not exist`); //Client is not in the network
    }
    var result = carAsBytes.toString();
    result = JSON.parse(result);
    const client = JSON.parse(carAsBytes.toString());
    if (client.pendingclaims.includes(claimid) == true) { //Client claimid exists
        if (resulta == 1) {
            var amountgranted = ((treatementprice[treatmentapplyingfor]) * availpercentage[treatmentapplyingfor] / 100);
            if (amountgranted > (2000 - client.thisyearmoney)) { //Partial claim will be granted, since otherwise yearly limit will be crossed.
                amountgranted = 2000 - client.thisyearmoney;}
            client.thisyear = client.thisyear + 1;
            client.lifetimesupport = ((client.lifetimesupport) - amountgranted);
            client.thisyearmoney = client.thisyearmoney + amountgranted}
        var i = 0;
        while (i < client.pendingclaims.length) {
            if (client.pendingclaims[i] === claimid) {
                client.pendingclaims.splice(i, 1);
            } else {
                ++i}}
        await ctx.stub.putState(policy_no, (JSON.stringify(client)));
        if (resulta == 0) {
            return ("User is registered, and treatment record exists, but claim has been declined!");}
        return ("Claim has been granted to the healthcare."); }
    else {
        return ("User is registered, but treatment record does not exist, claim unsuccessful! So no changes made!");}
}
```

Figure 5.4: Chaincode for claim request approval

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, an open-source high-performance HTTP web server, 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

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
```

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.

Table 5.4: 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

Table 5.5: Front-end Development Environment

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

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 [40]. 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 5.5 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 reporting requirements, also provides efficient and flexible way to query a large dataset from a chaincode.

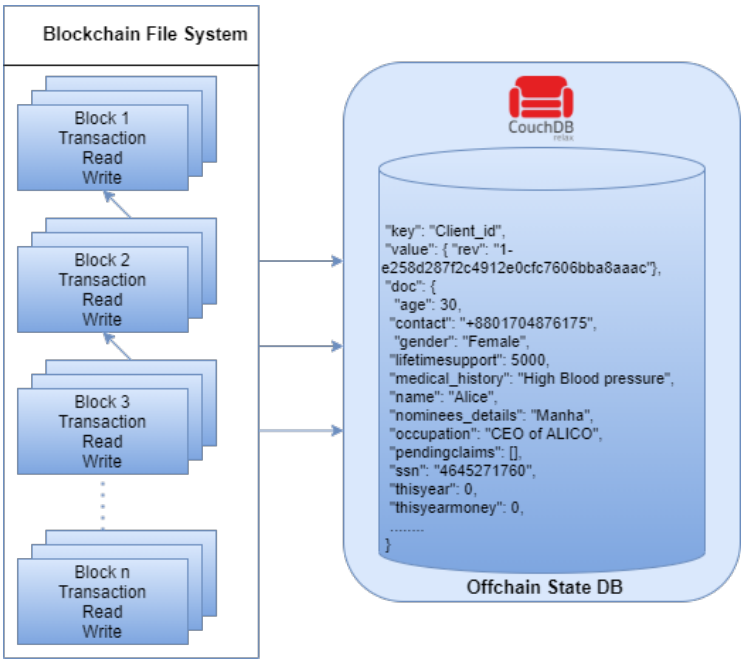


Figure 5.5: Distributed Ledger Storage Structure of the proposed Blockchain platform

5.6 Security and Privacy

In B-SAHIC, we have used Hyperledger Fabric–a permissioned Blockchain, where access is limited only to the known users, and thus their activities can be monitored and managed by access control lists. Therefore, 51% attacks and network partitioning attacks are not notable in threats [41]. We would like to mention that in addition to the security services provided by Hyperledger Fabric we have not deployed or implemented any security features. Thus the security services mentioned in the following section are integral part of Hyperledger Fabric. However, we have implemented the data privacy service presented in section 5.6.2.

5.6.1 Security Analysis

Due to the use of Hyperledger Fabric we can assure that our B-SAHIC system is secured from the following attacks.

Related-key attack In B-SAHIC, the key pair generation and sign of transactions are accomplished by using public key cryptography that uses elliptic curve encryption algorithm when making SSL connections between clients and nodes of the network. As a result, it becomes computationally infeasible for the adversaries to derive the private key. Besides, in each session agreement, a temporary (one time) private key is generated between nodes, which becomes unusable when the session is terminated. Moreover, the session key of an already established session is unlikely to be compromised.

Impersonation attack If the adversaries could breach the private key than this attack is only possible to occur, but since B-SAHIC generates temporary and separate private

key for each session agreement, this attack is not succeed.

Replay attack The replay attack can not be succeeded since the temporary private keys which are generated between nodes are bounded for lifetime.

Message modification attack This attack is possible to reduce since the attackers can not obtain the private keys.

False data injection attack B-SAHIC mitigate the risk of 'false data injection attack' by executing consensus algorithm before validating the records.

Tampering of data attack In Blockchain, 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, if any temper happen in B-SAHIC it could be detected easily. In the following we have summarized the other types of potential attacks that we have considered and how our system prevents those attacks:

Man-in-the-middle (MITM): MITM attacks are also possible to mitigate, as B-SAHIC uses distinct and temporary private key during each session and ensures mutual authentication between the client application and the endorsing nodes, which helps to ensure that Secure Socket Layer (SSL) cannot be manipulated into SSL-stripping or SSL-downgrade attacks in attempts to force the real users into using older, insecure versions of SSL or plaintext connections.

Sybil attack: There are various ways to reduce the impact of this attack on B-SAHIC, with one being through enabling of two factor authentication mechanism, and another be through collecting the IP and MAC addresses of unethical identities that could be identified as participants.

User credentials theft: For preventing user login passwords from getting stolen by potential hackers or from getting leaked accidentally by administrators, the passwords themselves be leaked to others, this is prevented by not storing the plaintext of the passwords in the database used for storing login credentials. Instead, the hash of plaintext is stored in the database and is used for comparing against the hash provided by the user application through when the user enters the password on their device, since hashes e.g. SHA256 are one-way hashes and cannot be reversed back to their plaintext original form. This hash is generated in the client application side which means that the original password can never be retrieved by a hacker on network or in control of the server.

5.6.2 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 `CurrentTime` every time a unique key will be derived. Thus, each document will be encrypted with a unique key.

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

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 confidential documents even from brute force attacks [43].

Chapter 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. Figure 6.1 shows the dashboard view of Hyperledger Explorer for our project. It 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.

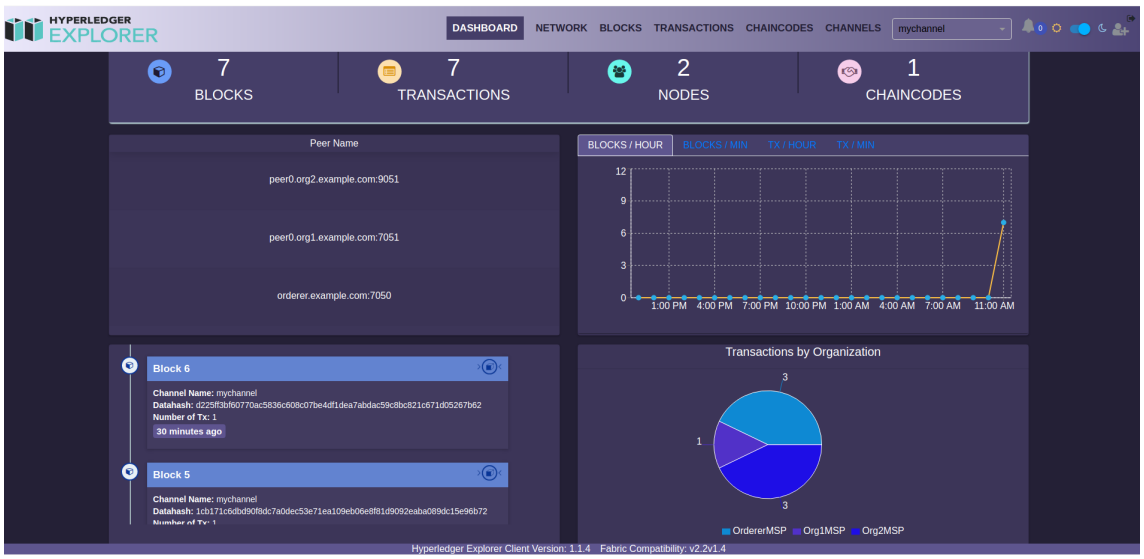


Figure 6.1: Hyperledger Explorer Dashboard

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

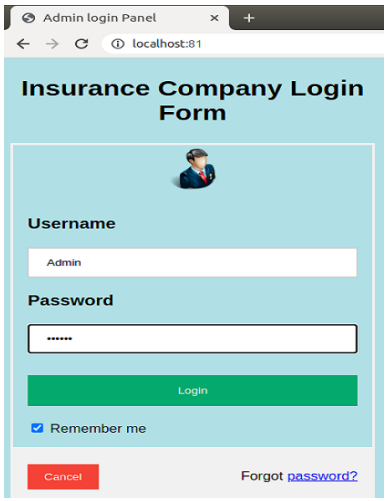


Figure 6.2: Insurance company login form.

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 Figure 6.2. 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 Figure 6.3. On successful registration, the system will returns an unique id number, i.e., the `Policy number` and a notification *User has been successfully added, and is registered with Client ID: 1640614851472*4, will be sent to the insurance company. 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 Figure 6.5, 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.

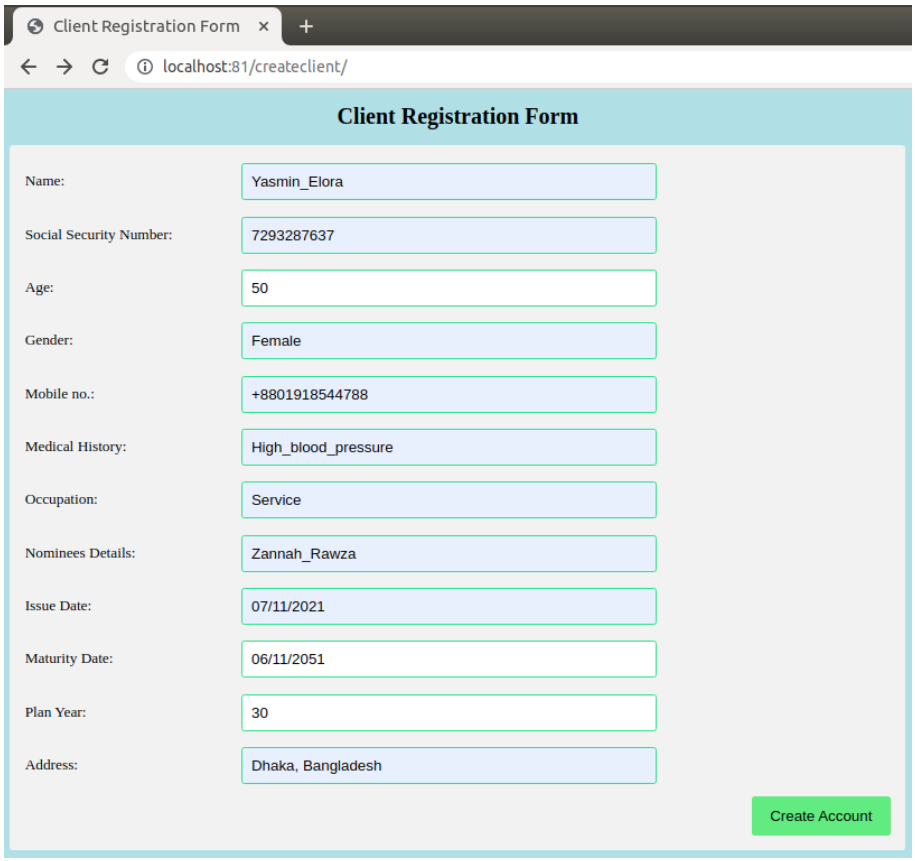


Figure 6.3: Client registration form.

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 Figure 6.6 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 Figure 6.7 (a). 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 Figure 6.7 (b). The client can also submit attachment with this submission form.

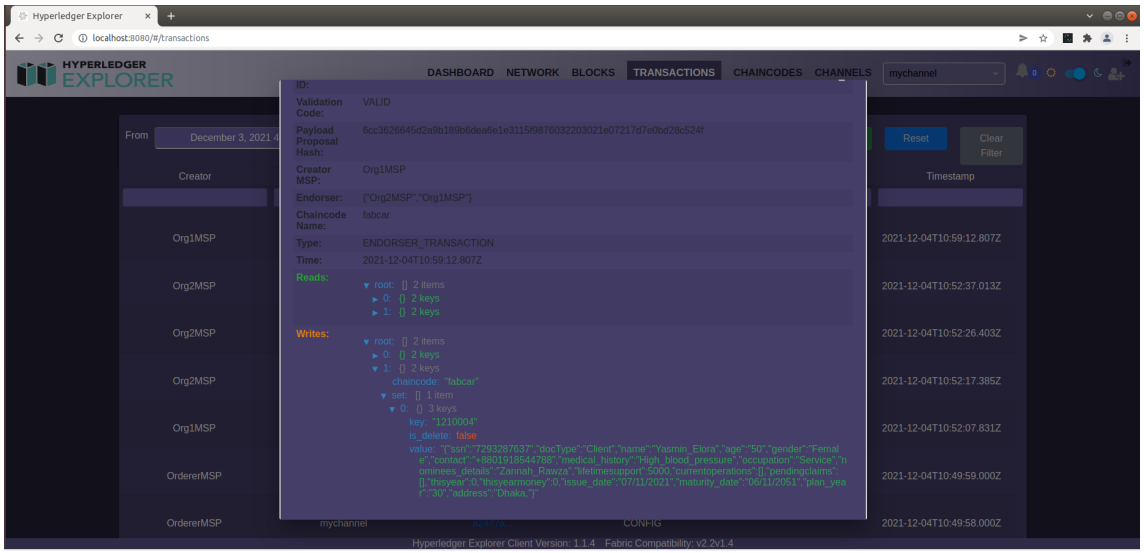


Figure 6.4: Explorer view: Successfully registered a new client into the Blockchain system.

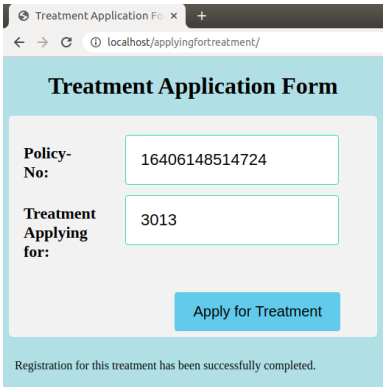


Figure 6.5: Treatment application form.

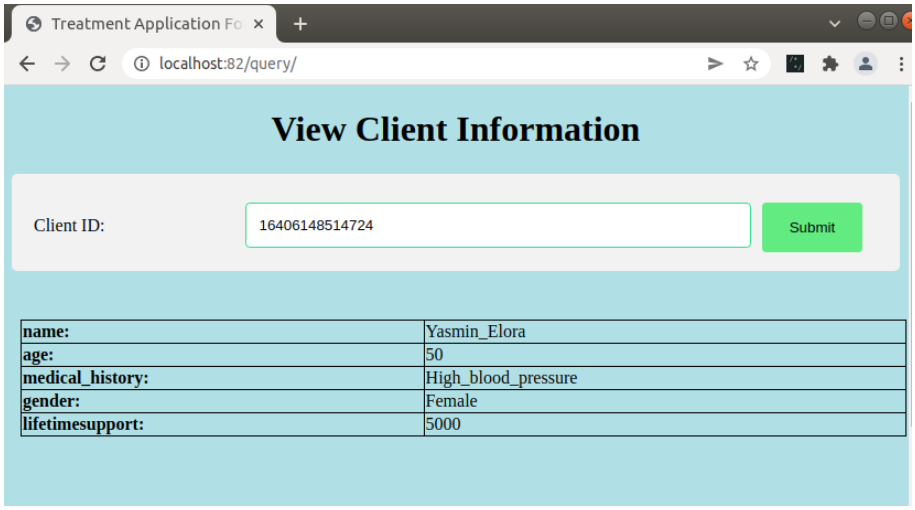
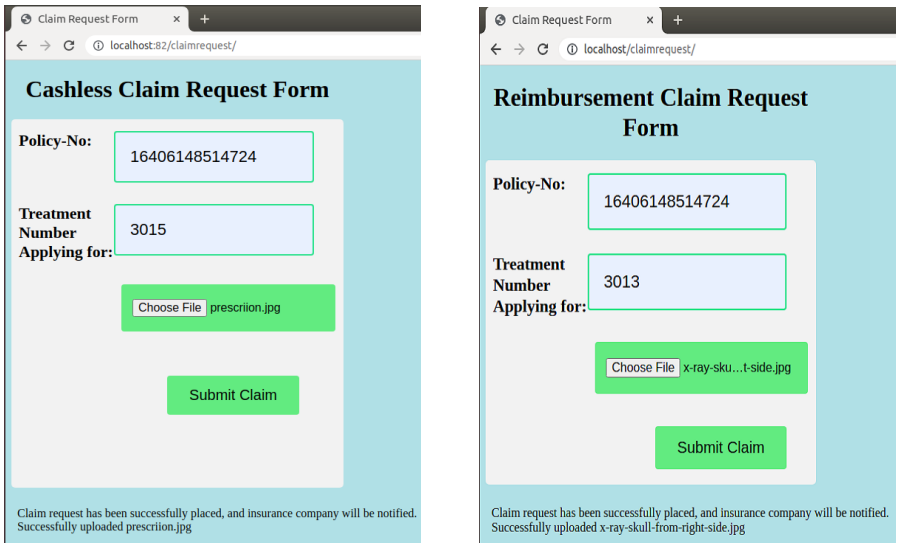


Figure 6.6: Query Client Data into the Ledger



(a) Cashless Claim (b) Reimbursement Claim

Figure 6.7: Claim Request submission

Claim approval

The insurance company will find all the submitted claim requests intheir Claim approval panel as shown in Figure 6.8. 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 verification. Finally, this accepted amount of money will be deducted from the client's wallet amount as per the insurance policy.

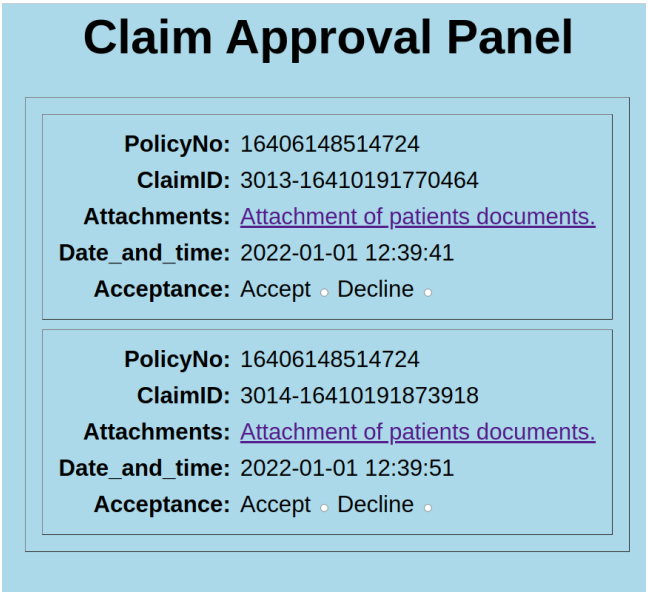


Figure 6.8: Claim approval panel

Chapter 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 [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 7.1.

Table 7.1: 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

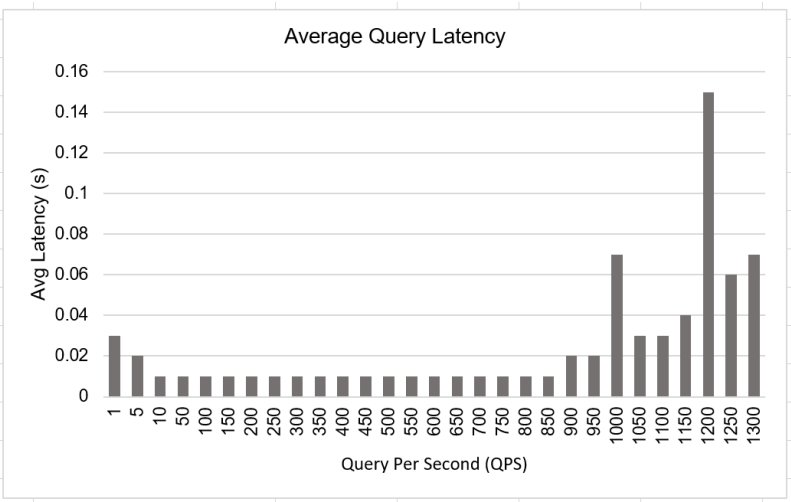


Figure 7.1: Average query latency in B-SAHIC

Two key indicators for analyzing the performance of an implemented Blockchain based

system are throughput and latency. In this study, we have examined latency, which can be further into classified into query latency and transaction latency. Query latency is the difference between the times when a query request is submitted and when the first query response is received. Equation 7.1 represents how the query latency is being calculated in our system.

$$Query\ Latency = Query\ Response\ Time\ -\ Query\ Request\ Time. \tag{7.1}$$

Transaction latency, on the other hand, is the difference between times when a transaction request is submitted and when it's effect is widely observed across the network. This measurement comprises any settling time including the propagation time needed for the consensus mechanism to take place. Equation 7.2 explains how we have calculated the transaction latency by subtracting the transaction submit time from the transaction confirmation time.

$$Transaction\ Latency = Transaction\ Confirmation\ Time\ - Transaction\ Submit\ Time. \tag{7.2}$$

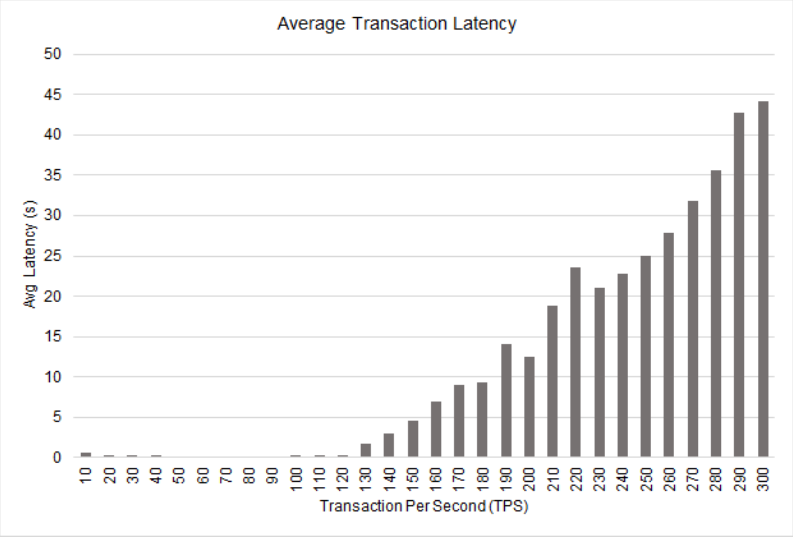


Figure 7.2: Average transaction latency in B-SAHIC

We have investigated the performance of our proposed B-SAHIC platform by considering different user groups. In Figure 7.1, we have examined the average latency needed to query for 1 to 1300 QPS (Query Per Second). It can be seen from the figure that the average latency is in acceptable range which is only 0.01 second for 10 to 850 QPS. If we increase the number of QPS, this amount of latency will increase slightly and reach to its peak at 0.15 second for 1200 QPS. However, for 1300 QPS there is a further drop and the average latency is measured as 0.07 second. Please note that in case of a query the system performs 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. 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 is occupied and the average latency has sharply increased in 1200 QPS.

We have conducted another experiment to calculate the average transaction latency in B-SAHIC, which is demonstrated in Figure 7.2. It can be observed that average transaction latency calculated for 10 to 120 Transaction Per Second (TPS) is just over 0.2 second. After that if we increase the number of TPS, the average latency is also increased. Note that a transaction operation performs write operation into the Blockchain and thus takes significantly longer time than query operation. Moreover, transactions can not be performed concurrently. Therefore, the average latency increases steadily with the increase of the TPS.

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 7.2. Here, two peer nodes (`peer0.org1.com` and `peer0.org2.com`) are representing the roles of the client and healthcare 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 7.2 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.

Table 7.2: 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

Chapter 8

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. Some user-friendly, web-based portals have been developed that provides different interfaces for all the actors through which both the client and the healthcare of this network can smoothly interact with the insurance company's server. Customers' data privacy has been maintained in B-SAHIC. Clients' personal treatment records has been stored in an encrypted form using unique encryption key for each record. In our proposed system, an unique claim id will be generated automatically for each cashless or reimbursement claim request consisting of treatment number applied for and claim request submission time in micro second. It will be recorded into Blockchain ledger thus, no one else would be able to send the same claim request to multiple insurance company. Again, we have installed Hyperledger Explorer–A user friendly web interface and integrated it with Hyperledger Fabric that provides a dashboard of summary section and details of that summary (blocks, transaction logs etc.) including network information,

According to the Hyperledger Fabric network structure, all the peer nodes in a channel must record all the transactions that happen in the channel. In our proposed system, we have considered one insurer and one healthcare in a single channel. This method allows less flexibility on what data an entity wants to store on the Blockchain, since this channel has only one chaincode.

In future, we want to develop multi-channel Blockchain network where each channel having the insurance company itself and the healthcare. This method will allow greater flexibility. In this case, all treatment payments of the clients will be visible to only the hospital and the insurance company from which the patient registers. Finally, note that the proposed system is easily extensible in the future version. Moreover, this study is easily applicable for implementing automated claim processing for other insurance industries

too.

References

[1] Fabrizio Lamberti, Valentina Gatteschi, Claudio Demartini, Chiara Pranteda, and Victor Santamaria. Blockchain or not blockchain, that is the question of the insurance and other sectors. *IT Professional*, pages 1–1, 2017.

[2] Julia Kagan. Insurance. `https://www.investopedia.com/terms/i/insurance.asp`, 2020. [Online; accessed 18 March 2021].

[3] What is health insurance? *URL: https://www.iciciprulife.com/health-insurance/what-is-health-insurance.html (accessed: 16.09.2021)*, 2021.

[4] Julia Kagan. Health Insurance. `https://www.investopedia.com/terms/h/healthinsurance.asp`, 2021. [Online; accessed 18 March 2021].

[5] Xinchi He, Sarra Alqahtani, and Rose 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)*, pages 1634–1641. IEEE, 2018.

[6] Jiaqi Lu, Benjamin CM Fung, and William K Cheung. Embedding for anomaly detection on health insurance claims. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 459–468. IEEE, 2020.

[7] Mohit Kumar, Rayid Ghani, and Zhu-Song 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*, pages 65–74, 2010.

[8] Nadia Smaili, Julien Le Maux, and Walid Ben Amar. Corporate fraud: Avenues for future research. In *Corporate Fraud Exposed*. Emerald Publishing Limited, 2020.

[9] 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.

[10] Gokay Saldamli, Vamshi Reddy, Krishna S Bojja, Manjunatha K Gururaja, Yashaswi Doddaveerappa, and Loai Tawalbeh. Health care insurance fraud detection using

blockchain. In *2020 Seventh International Conference on Software Defined Systems (SDS)*, pages 145–152. IEEE, 2020.

[11] Najmeddine Dhieb, Hakim Ghazzai, Hichem Besbes, and Yehia Massoud. A secure ai-driven architecture for automated insurance systems: Fraud detection and risk measurement. *IEEE Access*, 8:58546–58558, 2020.

[12] Paolo Tasca. Insurance under the blockchain paradigm. In *Business Transformation through Blockchain*, pages 273–285. Springer, 2019.

[13] Ronny Hans, Hendrik Zuber, Amr Rizk, and Ralf Steinmetz. Blockchain and smart contracts: Disruptive technologies for the insurance market. 2017.

[14] Chuka Oham, Raja Jurdak, Salil S Kanhere, Ali Dorri, and Sanjay Jha. B-fica: 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)*, pages 1171–1180. IEEE, 2018.

[15] James Chen. Insurance Coverage. `https://www.investopedia.com/terms/i/insurance-coverage.asp`, 2020. [Online; accessed 18 March 2021].

[16] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. bitcoin. org. *URL: https://bitcoin. org/bitcoin. pdf (accessed: 24.02. 2020)*, 2008.

[17] Matthias Lischke and Benjamin Fabian. Analyzing the bitcoin network: The first four years. *Future Internet*, 8(1):7, 2016.

[18] Valentina Gatteschi, Fabrizio Lamberti, Claudio Demartini, Chiara Pranteda, and Víctor Santamaría. Blockchain and smart contracts for insurance: Is the technology mature enough? *Future Internet*, 10(2):20, 2018.

[19] Anuraag A Vazirani, Odhran O'Donoghue, David Brindley, and Edward Meinert. Blockchain vehicles for efficient medical record management. *NPJ digital medicine*, 3(1):1–5, 2020.

[20] Indranil Nath. Data exchange platform to fight insurance fraud on blockchain. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 821–825. IEEE Computer Society, 2016.

[21] Jake Frankenfield. Blockchain Wallet. `https://www.investopedia.com/terms/b/blockchain-wallet.asp`, 2020. [Online; accessed 18 March 2021].

[22] A Blockchain Platform for the Enterprise. `https://hyperledger-fabric.readthedocs.io/en/release-2.2/`, 2020. [Online; accessed 18 March 2021].

[23] Gabriela Ciocarlie, Karim Eldefrawy, and Tancrède Lepoint. Blockcis—a blockchain-based cyber insurance system. In *Proceedings of the 2018 IEEE International Conference on Cloud Engineering (IC2E), Orlando, FL, USA*, pages 17–20, 2018.

[24] Jake Frankenfield. Smart Contracts. `https://www.investopedia.com/terms/s/smart-contracts.asp`, 2021. [Online; accessed 18 March 2021].

[25] Nicola Atzei, Massimo Bartoletti, and Tiziana Cimoli. A survey of attacks on ethereum smart contracts (sok). In *International conference on principles of security and trust*, pages 164–186. Springer, 2017.

[26] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[27] Uma Srinivasan and Bavani Arunasalam. Leveraging big data analytics to reduce healthcare costs. *IT professional*, 15(6):21–28, 2013.

[28] Young-Taek Park, Jeong-Sik Yoon, Stuart M Speedie, Hojung Yoon, and Jiseon Lee. Health insurance claim review using information technologies. *Healthcare informatics research*, 18(3):215, 2012.

[29] Rayid Ghani and Mohit 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*, pages 325–333, 2011.

[30] Shinya Kimura, Toshihiko Sato, Shunya Ikeda, Mitsuhiko Noda, and Takeo Nakayama. Development of a database of health insurance claims: standardization of disease classifications and anonymous record linkage. *Journal of epidemiology*, 20(5):413–419, 2010.

[31] Fred Popowich. Using text mining and natural language processing for health care claims processing. *ACM SIGKDD Explorations Newsletter*, 7(1):59–66, 2005.

[32] Mayank Raikwar, Subhra Mazumdar, Sushmita Ruj, Sourav Sen Gupta, Anupam Chattopadhyay, and Kwok-Yan Lam. A blockchain framework for insurance processes. In *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–4. IEEE, 2018.

[33] Alpen Sheth and Hemang Subramanian. Blockchain and contract theory: modeling smart contracts using insurance markets. *Managerial Finance*, 2019.

[34] Lijing Zhou, Licheng Wang, and Yiru Sun. Mistore: a blockchain-based medical insurance storage system. *Journal of medical systems*, 42(8):1–17, 2018.

[35] Ayesha Shahnaz, Usman Qamar, and Ayesha Khalid. Using blockchain for electronic health records. *IEEE Access*, 7:147782–147795, 2019.

[36] M Thenmozhi, R Dhanalakshmi, S Geetha, and R Valli. Implementing blockchain technologies for health insurance claim processing in hospitals. *Materials Today: Proceedings*, 2021.

[37] Faisal Jamil, Omar Cheikhrouhou, Harun Jamil, Anis Koubaa, Abdelouahid Derhab, and Mohamed Amine Ferrag. Petroblock: a blockchain-based payment mechanism for fueling smart vehicles. *Applied Sciences*, 11(7):3055, 2021.

[38] Richard Patsch. *Survey on Hyperledger Fabric and comparison of Blockchain as a Service Providers*. PhD thesis, University of Applied Sciences, 2020.

[39] What is hyperledger fabric? *URL: https://www.ibm.com/topics/hyperledger (accessed: 17.08.2021)*, 2021.

[40] Deeptiman Pattnaik. Offchain storage in hyperledger fabric. `https://deeptiman.medium.com/offchain-storage-in-hyperledger-fabric-77e28bd99e0c/`, 2020. [Online; accessed 03 November 2021].

[41] Christopher Cordi. Hyperledger fabric security threats: What to look for. `https://www.hyperledger.org/blog/2021/11/18/hyperledger-fabric-security-threats-what-to-look-for/`, 2021. [Online; accessed 03 December 2021].

[42] Douglas Crawford. How does AES Encryption work. `https://proprivacy.com/guides/aes-encryption`, 2019. [Online; accessed 18 September 2021].

[43] Mohit 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].