Hindawi
Scientific Programming
Volume 2021, Article ID 6787406, 11 pages
https://doi.org/10.1155/2021/6787406

Research Article

Secured Insurance Framework Using Blockchain and Smart Contract

Abid Hassan,[1] **Md. Iftekhar Ali**,[1] **Rifat Ahammed**,[1] **Mohammad Monirujjaman Khan**,[1] **Nawal Alsufyani**,[2] **and Abdulmajeed Alsufyani**[2]

[1]*Department of Electrical and Computer Engineering, North South University, Bashundhara, Dhaka 1229, Bangladesh*
[2]*Department of Computer Science, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia*

Correspondence should be addressed to Mohammad Monirujjaman Khan; monirujjaman.khan@northsouth.edu

Received 9 October 2021; Accepted 6 November 2021; Published 24 November 2021

Academic Editor: Punit Gupta

Traditional insurance policy settlement is a manual process that is never hassle-free. There are many issues, such as hidden conditions from the insurer or fraud claims by the insured, making the settlement process rough. This process also consumes a significant amount of time that makes the process very inefficient. This whole scenario can be disrupted by the implementation of blockchain and smart contracts in insurance. Blockchain and innovative contract technology can provide immutable data storage, security, transparency, authenticity, and security while any transaction process is triggered. With the implementation of blockchain, the whole insurance process, from authentication to claim settlement, can be done with more transparency and security. A blockchain is a virtual chain of data blocks that is a decentralized technology. Any transaction or change in the blocks is done after the decentralized validator entity, not a single person. The smart contract is a unique facility stored on the blockchain that gets executed when the predetermined conditions are met. This paper presents a framework where smart contracts are used for insurance contracts and stored on blockchain. In the case of a claim, if all the predetermined conditions are met, the transaction happens; otherwise, it is discarded. The conditions are immutable. That means there is scope for alteration from either side. This blockchain and intelligent contract-based framework are hosted on a private Ethereum network. The Solidity programming language is used to create smart contracts. The framework uses the Proof of Authority (PoA) consensus algorithm to validate the transactions. In the case of any faulty transaction request, the consensus algorithm acts according to and cancels the claim. With blockchain and smart contract implementation, this framework can solve all the trust and security issues that rely on a standard insurance policy.

1. Introduction

Insurance is an agreement in which an individual or institution receives financial protection or compensation from an insurance provider in the event of a loss, represented by a policy. Insurance is a widely practiced method of security all over the world. According to a statistical report, the global insurance market is valued at over 5050.3 billion US dollars for 2021 [1]. There are various types of insurance policies for health, business, and vehicles. These policies are prevalent in developed countries around the world. In much of Europe, Latin America, Canada, Australia, and Japan, national health insurance schemes are in existence through national policies [2]. Though insurance policies are prevalent, settling claims is not always a fault and a hassle-free procedure. There are often situations when insurance companies refuse to pay the insured by misrepresenting conditions and terms. Again, false claims are another set of problems that are troubling the insurance companies. Conventional contractual methods are not fault-proof. These contracts lack transparency and have loopholes. These loopholes lead to exploitation in many cases by both insurers and insureds. These conditions can be disrupted using smart contracts on the blockchain, as it reduces the need for trust and financial risk in existing agreements and provides legal clarity.

This research applies the method of creating a conceptual framework using blockchain and smart contracts to applications in insurance. Its primary goal is to use blockchain and smart contracts to ensure secure, insurance fraud-free transactions. This framework covers client registration, issuing a policy, and refund settlement activities using blockchain technology, making the whole insurance system more robust.

A blockchain is an accumulation of blocks that hold data. Each block contains the previous block's cryptographic key [3], timestamp, and transaction information. This technology has been prevalent for a series of applications. Since Satoshi introduced the Bitcoin platform based on the blockchain system in 2009 [4], blockchains have been attracting attention with various applications in various fields. Till now, the most accepted usage of blockchain technology has happened in Bitcoin's distributed transactions [5]. However, researchers have found other practical applications of blockchain in the government's public services [6]. IOT [7] and the most important financial and banking sector [8] are two major fields where the proper usage of blockchain technology can bring more productivity. Blockchain technology has several unique properties that make it ideal for financial transaction applications. The main characteristics of the blockchain are decentralized, consensus, provenance, immutability, and finality. Decentralized means no single most potent entity controls the whole blockchain, and it is a crucial feature of the blockchain. The whole system runs on the standard agreement of its participants. This standard agreement is called consensus. Again, consensus is one of the most important characteristics of the blockchain. When all the participants of a blockchain network agree on a transaction, the transaction gets executed.

A typical agreement is a must for a transaction on a blockchain. These consensus characteristics make the system very trustworthy among the participants [9]. The provenance feature ensures the traceability of data blocks. In the blockchain, each block's whereabouts are traceable. Suppose an item is sold on a blockchain system. In that case, every aspect of its development must be recorded in its blocks, from the moment it was built to the previous owner's information. When a valid transaction is completed and recorded in the specific block, no one in the network can change or alter it. This immutable nature of the blockchain makes it very secure compared to other transactional methods. All these immutable blocks are linked in one single ledger. In a blockchain system, there is only one ledger with one common truth [10]. There is a whole system that has one policy that operates everything. For any query about the transaction, the blockchain ledger is the only information hub. In general, there are three types of blockchain: public (or unauthorized), private (or permitted), and consortium (or allowed). Due to the general uniqueness of the network's geographic region, each one has its distinct features [11].

Smart contracts are simple scripts that are enforced when forethought conditions are fulfilled and are recorded on a blockchain [12]. They are frequently used to automate the implementation of an agreement so that all parties are guaranteed a timely conclusion without the need for any middlemen or wasted time. They can also automate a workflow by initiating the following step when certain circumstances are fulfilled [13]. A generic smart contract's life cycle begins with the parties entering the conditions of a contract on a distributed ledger. Then they connect to internal or external databases and systems. The contract waits for predefined conditions to be evaluated by external variables. Finally, the contract self-executes when criteria are met via triggers. The ease with which smart contracts may be deployed on public blockchains, also known as public smart contracts, has sparked a slew of business uses [14]. Using smart contracts for insurance can be very useful for claim settlements.

Several research studies have shown the possible disruption of the insurance industry by using blockchain and smart contracts. In this paper [15], the researchers concentrated on developing a blockchain-based infrastructure for processing insurance-related transactions. They created a prototype using Hyperledger Fabric, an open-source permissioned blockchain architecture platform. Researchers in [16] used blockchain in a user-based insurance model for vehicle insurance. They presented an application of blockchain to the Pay as You Drive (PAYD) and Pay How You Drive (PHYD) insurance schemes. Pay As You Drive (PAYD) is a common type of usage-based insurance. The insurance premium is determined by the number of kilometers driven in the vehicle throughout the covered period.

Customers who drive less get a lower insurance rate. The insurance duration can be tailored to the customer's specific needs. Pay How You Drive (PHYD) is a type of usage-based insurance that is quickly gaining popularity in the industry because of its numerous advantages. PHYD calculates insurance rates depending on how the vehicle is driven, rather than the vehicle's type and model, the driver's age, employment, or other factors. Because the driving pattern is a significant predictor of how likely the user is to make a claim, this evaluation method is more suitable. A reckless driver, for example, is more likely to be involved in an accident and, as a result, to submit a claim. As this information is stored using a blockchain, it is immutable. In [17], researchers predict that, when fully implemented, blockchain might pose a threat to the existing insurance business model since it suggests substantial cost reductions. However, by pooling resources and collaborating with these new players, this danger may be mitigated. However, there are also different views. In [18], the authors state that blockchain smart contracts are not fully secured. Smart contracts are regularly targeted by hackers, with devastating results in certain situations. This element may pose a particular danger to peer-to-peer insurance policies, which rely heavily on smart contracts for governance. In [12], the authors mention that the "ordinary user's" engagement with the blockchain is still complicated. Understanding the fundamentals of wallets, transactions, mining, and other related concepts necessitates some technical knowledge. At the same time, Bitcoin has been linked to a pyramid scheme or a scam on several occasions. As a result, there is still much misunderstanding about blockchain. The authors state that blockchain

technology is still not mature enough for insurance applications yet. However, the authors also mentioned that the technology would gain more acceptance among insurance consumers with time. As of 2021, Bitcoin is more accepted all over the world than at any time in history. In [19], the authors discussed the influence of blockchain on the payments sector and the technology's disruptive nature. Based on these findings, they looked into how blockchain influences important areas of radical innovation and developed contributions from innovation management. Blockchain technology can provide immutable security for payments in banks, Equated Monthly Installment (EMIs), installments, or regular billing. In [20], the authors discuss that blockchain has the ability to innovate and drastically disrupt the insurance industry as we know it by providing cryptographically secure forms of distributed records. In [21], the authors presented a secured blockchain-based data exchange platform that can fight against fraudulent activities regarding insurance. Though blockchain technology is comparatively new in fintech, blockchain as a technology has the potential to disrupt the insurance sector by promoting honesty and openness, as well as influencing consumer risk perceptions, which might impact how insurers promote mutualization. The authors of [22] present a survey of blockchain disruption in various domains, most notably blockchain and how blockchain can improve the insurance domain by eliminating fraud, automating claims, analyzing data with the Internet of things, and preserving reinsurance.

The above research works present blockchain and smart contracts for use or, in some cases, the possibility of future use in the insurance industry with increased security, immutability, and accountability. With blockchain implementation, the traditional insurance industry can be disrupted and accountable to both insurers and the insured. This paper presents a secure framework for insurance using blockchain technology and smart contracts. The whole framework is implemented on a private Ethereum network. Smart contracts for the system are developed using the Solidity language. The framework covers all the expectations for insurance, client registration, client query, policy initialization, issuing, claiming, and refund.

Problem Statement: General people get registered for insurance policies to use the insurance money in case of any danger. On the other hand, insurance companies are an excellent money-making business that creates jobs and pays taxes like any other business. So, whether it is a claim settlement for the insured or a false claim causing trouble for the insurance companies, none is expected. So, it is crucial to have a fast and fraud-proof settlement process. The insurance industry is going through various settlement and trust issues in the traditional method of insurance. Implantation of blockchain in insurance can be a robust solution to the security and trust issues related to these problems.

Motivation: Blockchain technology uses a decentralized, secure authentication process to store data in blocks. For any transaction or change in the blocks, the transaction request has to pass through an established consensus algorithm that makes any unauthorized change virtually impossible. Also, it encrypts data during the transaction and puts time stamp blocks that make the whole transaction process secure and immutable. The whole process is made more efficient by the introduction of smart contracts that are transparent and very secure. Suppose the test cases match transaction requests' triggers and eventually get executed by passing through a consensus algorithm. Implementing this whole process for insurance makes the whole insurance structure more trustworthy, efficient, and secure for both stakeholders, the insured and the insurer.

The introduction of the paper is presented in section one, and section two describes the method and materials. The results and analysis of this paper are provided in section three. Section four discusses the conclusion.

2. Method and Materials

The methods and materials utilized to achieve the goal are discussed in this section. The goal is to create an insurance ecosystem using blockchain technology. The main idea is to deploy the whole execution and storage of the contract. Its conditions and logic for execution will be structured as smart contracts and written using the Solidity programming language. The deployment of these contracts will be on a blockchain-enabled distributed platform, in our case, Ethereum. The first subsection here presents a basic system model for the framework. The following parts present the characters involved, mechanism of the framework for insurance, network platform, consensus algorithm, blockchain blocks, smart contracts, and framework components and algorithm. Section 3 presents the results and analysis of the framework. Finally, in Section 4, the whole framework is concluded.

The major contribution of this proposed system is the implementation of blockchain technology in the field of all kinds of insurance processes. It also includes the use of a specific consensus algorithm (in this case: Proof of Authority) in the system and the detailed algorithm and the explanation of the whole process.

2.1. Outline of Full System. Figure 1 shows the whole system model diagram of the proposed framework. It indicates that the client will be able to register and issue a policy, claim, and refund with the help of their corresponding agent. With the help of the corresponding agent, the information gets placed into the Ethereum private network. The agent is fully responsible for submitting all client requests within the network. When the transaction happens, the validators are responsible for validating the transactions.

It is to be noted that it is supposed to be a framework for an insurance company. Thus, the diagramed model is only of the members and clients of the company. The blockchain network in this case is the private Ethereum network. The validators are previously selected and will validate transactions per the rules of the Proof of Authority algorithm.

2.1.1. Authentication. Though the system is mainly a proposed algorithmic framework, it has a barebones frontend which includes the authentication process as well for the

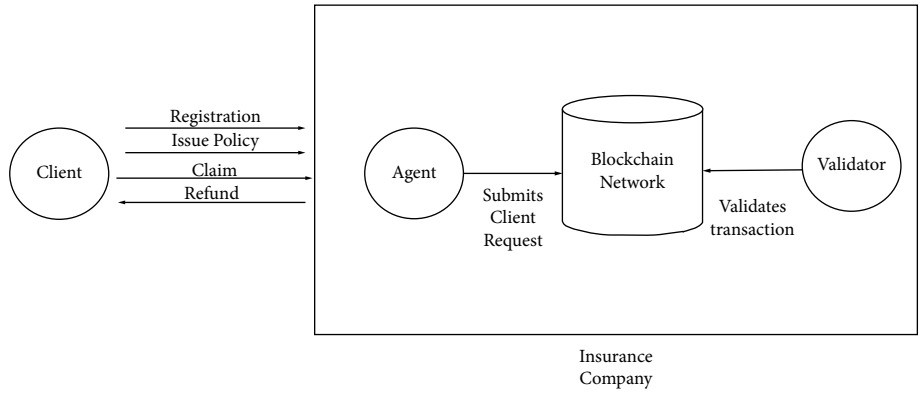


Figure 1: Full system diagram.

stakeholders involved in the system. Figures 2(a)-2(b) show the authentication User Interface (UI) for the system. The authentication is maintained by Google's Firebase system. The same login is used for both the client and the authority. A hard coded filter is enabled to filter the client and the authority after being logged in. The authority members are registered manually as it has to be private and discreet inside the agency. So, the sign-up serves a purpose for the clients only.

2.1.2. Agent Panel. Whether a desktop, console, or Web application, the agent panel is essential as it is the vital stakeholder of the system. They control important features like policy issuing, policy initializing, and different client queries. The agent and the validator have the same portal, because in some cases, the agency can select one for both roles. Figure 3 shows the agent panel interface where he can control all the features, including looking into clients' lists, the agency policy lists, the transaction history, etc.

2.2. Participation of Characters in This Model. The main character that is directly involved in the model is the client. He can register for the contracts, request insurance policies, submit claim requests, apply for a refund, and more. There is a middle man, an agent, who will process all clients' documents and demands and put them on the blockchain network. In the framework of some validators present in this system, they are responsible for verifying contract policies and transactions and storing the contracts in the ledger.

The stakeholders are as follows:

(i) Clients.

(ii) Insurance company authority

(iii) Agents

(iv) Validators

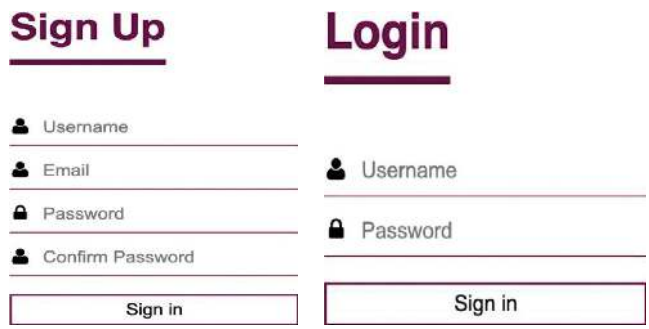
2.3. Mechanism of the Framework for Insurance. The client has to register with a unique id along with other necessary attributes as values. These IDs are to be stored in a DB. Previously, all policies and regulations would be written in the form of smart contracts. They are designed to be triggered when all the requirements or logics are met for the transactions. When a transaction is made, the record logs and execution results will be stored in a ledger in the blockchain network. In between the transactions, there is a set of endorsers and validators who verify the transaction and validate and store the transaction block in the blockchain ledger.

2.4. Network Platform. The whole network distribution is going to be deployed on a private Ethereum platform. This is an access-controlled blockchain. Participants are invited into this network by the insurance company authority. Based on access controls, this dedicated network will limit individuals who can participate in the network. Here, the network will allow the distribution of the ledger to a specific group of participants without making the transaction information public to all.

2.5. Consensus Algorithm Used in This System. This framework uses the Proof of Authority (PoA) consensus algorithm to validate and generate transaction blocks before adding them to the blockchain network. The validators will be preselected by the insurance company authority. Only those people who have proven their reliability as authorities get the right to generate new blocks or transaction logs. Once the validators are selected, they are allowed to make transaction logs and other monitoring stuff.

Figure 4 shows the flow diagram for the Proof of Authority algorithm in the specific use case of this research topic. The members of the authority are essentially the insurance authority. The algorithm has to go through the configuration option of the period. Then the transaction nodes or blocks, after being validated, are added to the Ethereum private network of the agency by the help of the issuing power of the insurance agent. Else, the block gets discarded.

On a broader perspective, the algorithm will be in need of configuration settings to work with the relevant system network. The configuration will have the chain-data, gas-limit, other relevant information, and so on. Now the authority members, in this case, the validators, when they receive a new block, will need to solve complex mathematical



(a) (b)

Figure 2: Sign-up and login and interface.

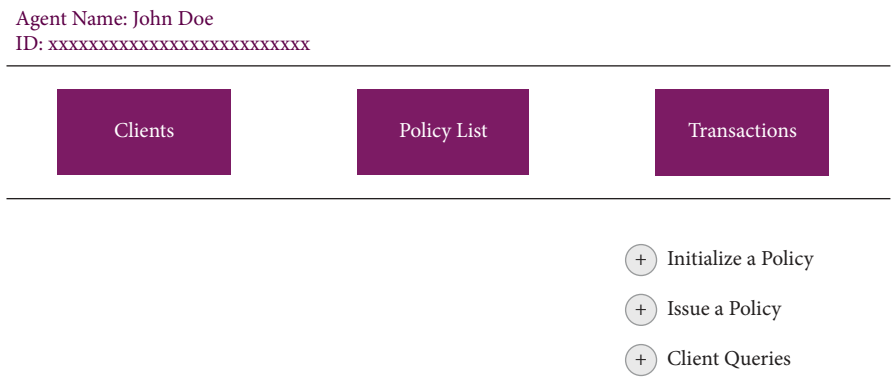


Figure 3: Agent panel.

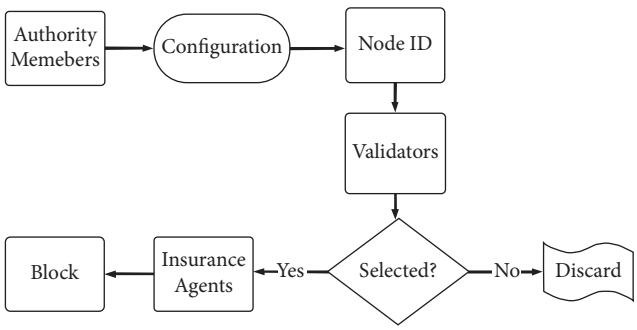


Figure 4: Proof of authority (PoA) flow diagram.

instructions to validate the block. When a validator has successfully mined the correct block, it will be selected to be added to the main blockchain network. The validator or the corresponding agents will be responsible for adding the block to the mainframe. Otherwise, the block will be stashed.

Figure 5 shows a genesis.json file, which is necessary for a PoA consensus-based network. The configuration ensures that all the known protocol changes are available. It also, importantly, configures the Clique Engine of PoA consensus.

2.6. Blockchain Blocks for Insurance. A blockchain is a chain consisting of blocks or data packages where a block consists of multiple transactions. The blockchain increases its length

```
const config = {
  "config": {
    "chainId": 7,
    "homesteadBlock": 0,
    "eip150Block": 0,
    "eip155Block": 0,
    "eip158Block": 0,
    "byzantiumBlock": 0,
    "constantinopleBlock": 0,
    "petersburgBlock": 0,
    "clique": {
      "period": 5,
      "epoch": 30000
    }
  },
  "difficulty": "1",
  "gasLimit": "8000000",
  "extradata": "0x7ff8b875a174b3bc565e6424a0050ebc1b2d1d82",
  "alloc": {
    "7df9a875a174b3bc565e6424a0050ebc1b2d1d82": { "balance": "300000" },
    "f41c74c9ae680c1aa78f42e5647a62f353b7bdde": { "balance": "400000" }
  }
}
```

Figure 5: Period configuration for a PoA network.

with every addition of the blocks. Figure 6 presents the structure for each block. Each block in the blockchain is validated by a specific validator before it is added or executed. Each block contains timestamps and hashes that allow it to be distinguished from the rest of the blockchains. Other than the hash and time stamp, there is information stored in blocks. This information varies according to the application needs. Below we have shown a generic block for any insurance application through the blockchain. Essential data

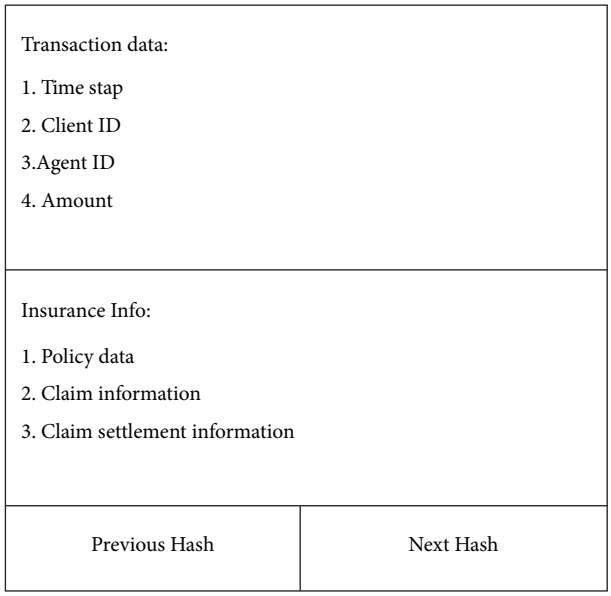


Figure 6: Structure of each block.

for insurance are , for example, Client ID, Agent ID, Amount, and essential insurance information.

2.7. Smart Contracts in Insurance. The blockchain containing insurance details will be contractualized on the Ethereum smart contract platform, where each peer or validator of the system will implement access control for its resources. Smart contracts are digital contracts between the client and service provider. As the contracts are well specified and transparent, there is a slight possibility to manipulate the contract conditions as the processing is done under the supervision of validators who have well-defined contract clarity.

In the insurance smart contract presented in Figure 7, when a client claims a refund, it sends account details to the validator. The validators check the contract details and send confirmation of their decision. Then it goes through the execution process and creates essential changes in the blockchain.

2.8. Framework Components and Algorithms. In this framework, maintaining and processing part of the insurance environment is done using blockchain technology. Blockchain technology ensures the security factor of false claims and the accountability of insurance.

Figure 8 shows the basic use case functionalities for the framework. These functionalities of an insurance policy are maintained through distributed smart digital contracts that are safe, reliable, and almost temper proof. Smart contracts will register the client and policy details as an object in a database. By taking a look, it can be seen that the client has the ability to register and to apply for the policy initialization, claiming, and refund. Agents and validators are internal personnel of the insurance company. The agents will be the direct interactors with the clients. They will help the client or

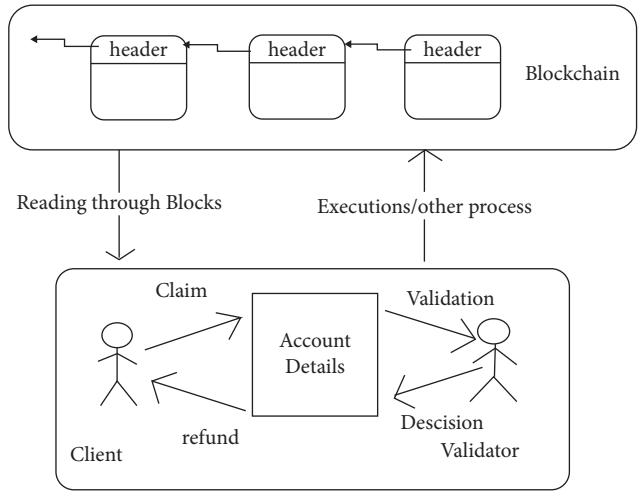


Figure 7: Structure of smart contract.

have access to the registration and perform client queries, policy initializing, and issue. The validators have access to policy claiming, refund, and transaction block validation.

2.8.1. Client Registration. With the help of smart contract, clients get registered with the insurance system. To initialize a client, a client object structure ($\text{Struct}_{\text{OC}}$) is created in the database. The client object contains attributes like a unique id, name, age, contact, history, etc. To register that client object, a composite key (C_{key}) is created by an agent and the client object (O_{C}) is created using the C_{key}. (Algorithm 1).

2.8.2. Client Query. After a client is registered, his information with all the attributes is stored in the blockchain network. Now, to retrieve any specific client details, the insurance agent has to create a composite key (Algorithm 2).

2.8.3. Policy Initialization. The smart contract will contain the policy issuance, claims, refunds, etc. To initialize a policy ($\text{Struct}_{\text{OP}}$) and policy-client ($\text{Struct}_{\text{OPC}}$), their structures are created in the database. The policy structure is going to have its id, name, premium, reimburse, and term information. The policy-client structure will have its id, policy id, amount claimed, claim acceptance indicator, claim submission date, etc. (Algorithm 3).

2.8.4. Policy Issuing. To issue a policy, a client chooses a policy from the available policy (Policy_id) in the database. After choosing the policy, the client submits a premium to the agent. If the transaction passes all the verification and checks, a corresponding policy-client object ($\text{Object}_{\text{Policy-Client}}$) is made and stored in the database (Algorithm 4).

2.8.5. Policy Claiming. To process a claim, the client submits his credentials to his corresponding agent. All the necessary conditions go through verification, and the refund is initiated accordingly (Algorithm 5).

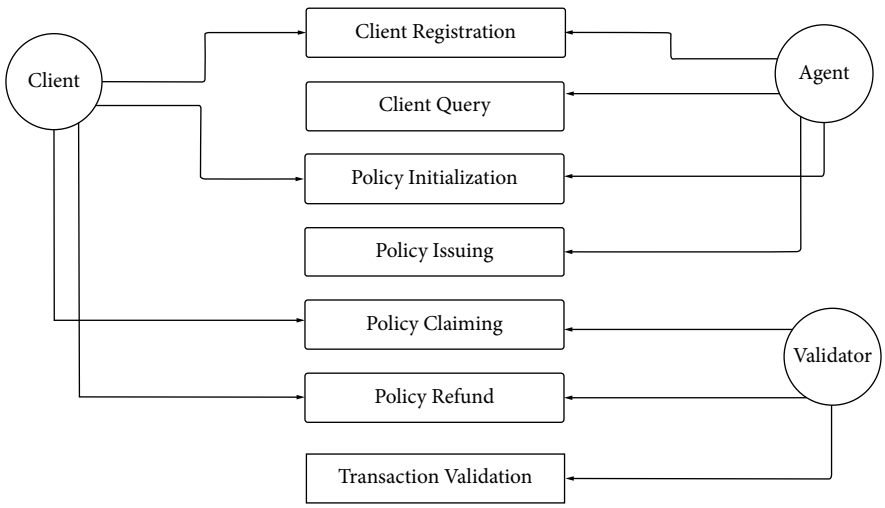


Figure 8: Use case diagram for the framework.

(1) $Struct_{OC} \leftarrow$ *(id, name, age, contact, history)*;
(2) *Database* $\leftarrow Struct_{OC}$;
(3) $C_{key} \leftarrow f$ *(Agent_id, Client_id)*;
(4) *Store* $(C_{key}, Object_{Client})$ *in theDatabase;*

Algorithm 1: Client registration.

(1) $C_{key} \leftarrow f$ *(Agent_id, Client_id)*;
(2) *Search* (C_{key}) *in the Database*;
(3) *If(exists) retrieve desired* $Object_{Client}$ *Else return Error;*

Algorithm 2: Client query.

(1) $Struct_{OP} \leftarrow$ *(Policy_id, Policy_name, Policy_Premium, Policy_Reimburse, Policy_Term)*;
(2) *StructOPC* $\leftarrow$ *(Policy_id, Client_id, Amount, Acceptance, Date)*;
(3) *Store the structures in the Database;*

Algorithm 3: Policy initialization.

(1) *Check if the* $Object_{PolicyClient}$ *already exists*;
(2) *Check Client Smart Contract if Client with an id is registered to an Agent;*
(3) *Check if Client Premium matches Policy Premium*;
(4) *CkeyPolicyClient* $\leftarrow f$ *(Agent_id, Client_id, Policy_id)*;
(5) $Object_{PolicyClient} \leftarrow$ *new* $Struct_{OPC}$*(Client_id, Policy_id, 0, yes, date);*
(6) *Store* $(Ckey_{PolicyClient}, Object_{PolicyClient})$ *in the Database;*

Algorithm 4: Policy issuing.

(1) $Ckey_{PolicyClient} \leftarrow f$ (Agent_id, Client_id, Policy_id)
(2) Check if the $Object_{PolicyClient}$ exists using the $Ckey_{PolicyClient}$
(3) If $Object_{PolicyClient}$ exists, check acceptance in $Object_{PolicyClient}$;
(4) if (acc==Yes) then
        if amt + Client_Reimburse <= Policy_Reimburse then
        Refund(Agent_id, Client_id, Policy_id, Client_Reimburse)
        else
        Refund(Agent_id, Client_id, Policy_id, Policy_Reimburse–amt)
    end;

Algorithm 5: Policy claiming.

2.8.6. Policy Refund. The refund process is initiated from the earlier claim process. Here, the client's total claimed amount in the policy-client object is updated in the database (Algorithm 6).

3. Result and Analysis

This chapter analyzes the result of the framework from the point of view of outputs, security, and scalability. The potential threats and their corresponding prevention along with the used consensus algorithm are analyzed here. Also, it points out the average response time per size of blocks.

3.1. Transactions and Outputs. The transactions in the blockchain stay like a chain. To get it like a printed list output, here one transaction block is shown in Figure 9. The list itself is an array of transaction block objects. The first property is its index number. Secondly, there is the previous hash property, one of the main important concepts of the blockchain technology. The proof here is the Proof of Authority (PoA). There is a timestamp property which is the time the transaction happened. Next on the list is the transaction array. It is the combination of the above information and the client-agent information, along with the transaction cost amount. Finally, the list includes the corresponding policy data for that particular transaction block.

The transaction array in that block is very vital. Figure 10 breaks down the transaction array and shows the properties inside. The first property is the amount, which indicates the number of times the agent sends and receives a separate unique number for each transaction. Next is the client_id. To keep it simple and agent-friendly, the id is simply a string which is the id_name of the client. The last property is the sender or agent hash_id. This whole transaction array is also linked to that certain timestamp of the block. During transactions, the consensus protocol helps to agree on the list contents, which guarantees the integrity of the blocks and transaction. If validated, then the blocks get added to the main blockchain. These transactions depend on the hash and their values. If fraud or any suspicious block gets detected, it will easily get detected and will not be added.

The whole test-development phase ran inside the Rinkeby testnet. It is an Ethereum test network that allows the development testing phase to be done, before being deployed on the main network. As the research's target network is the Ethereum private network, this testnet has been pretty useful in the test phase. The corresponding transactions took place on the testnet, and the data could be retrieved for that network. Figure 11 shows the retrieved transaction data from the testnet. The retrieved data includes the transaction's both parties, the transaction hash, gas values, and so on. The data is safe and secure.

3.2. Security of the Proposed Framework. Table 1 indicates the potential threats to the framework and how it is going to prevent them. Though the blockchain network is already secured with its immutable design, the table further indicates how different malicious activities can harm the framework. Unwanted modification and deletion of client data will be handled by our chosen Proof of Authority (PoA) consensus algorithm. The endorsement and other policies will be scripted in the smart contract on the Ethereum network, so any kind of wrong endorsement will be noticeable and can be figured out as well.

3.3. Consensus Algorithm Analysis. The PoA algorithm suits the purpose of this framework pretty well. Proof of Authority (PoA) consensus is used in permissioned blockchain platforms. The consensus mechanism works on permissionless and permissioned platforms. In the permissionless ones, anyone can become a node. Meanwhile, in the permissioned one, all the nodes and validators are preselected, enabling the system to be more secure. PoA is a type of consensus that is super fault-tolerant and able to achieve high performance.

In this algorithm, only the nodes that have proven their reliability as authorities get the right to generate new blocks or transaction logs. Once the validators are selected, they are allowed to make transaction logs and other monitoring stuff. Unlike all other consensus algorithms we have seen here, the validators do not need to stack their coins or spend money on expensive storage or hardware; all they have to do is to use their reputation to get the right to validate and generate the blocks. Proof of Authority is based on the trust of the selected validators. This algorithm suits both private and public networks, where trust is distributed.

3.4. Scalability Analysis. Figure 12 shows the performance of the system with the response time for storage requests on the y axis and the peers (block size 20) on the x axis. The results

(1) $Ckey_{PolicyClient} \leftarrow f$ (*Agent_id, Client_id, Policy_id*)
(2) *Check if the* $Object_{PolicyClient}$ *exists using the* $Ckey_{PolicyClient}$
(3) *Update amt = amt + reimburse*$_{client}$ *in* $Object_{PolicyClient}$

Algorithm 6: Policy refund.

Figure 9: Genesis blocks and their properties.

Figure 10: Transaction array.

Figure 11: Retrieved transaction data from Rinkeby.

Table 1: Potential threats and prevention.

Potential threat	Prevention
Modify/delete client data	PoA consensus algorithm
Word endorsement	Endorsement policy
Wrong auditing result	PoA consensus algorithm

show that the response is greater as the number of peers increases. 50 peers can have less than 20 ms response time in average, which could easily be used in real time applications using blockchain.

The chain code of the smart contract is written in Solidity and all the tests are written in plain Javascript as the network is a private Ethereum network. The experiment was carried out on a system with a core i7 (2.60 GHz) with 8 Gigabytes (GB) of ram, running Windows 10.

3.5. Comparison with Existing Papers. Table 2 analyzes the comparison and difference between this paper and other existing papers. The table's comparison context is mainly the

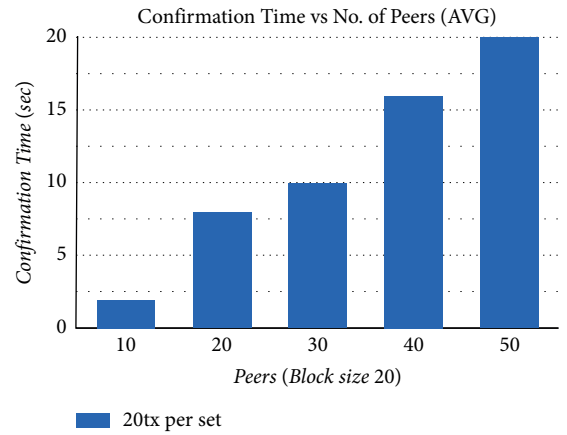


Figure 12: Average confirmation time vs. no. of peers (batch timeout: 2 sec).

Table 2: Comparison table with this paper and other papers.

Points of this paper	Points of other papers
1. This paper implements the Proof of Authority (PoA) consensus algorithm and uses the Ethereum private network to run the system.	1. There is no mention of a consensus algorithm and the system uses Hyperledger Fabric network to run the system [15].
2. This paper shows a generalized approach for all types of insurance.	2. A usage-based insurance model for cars is presented in the paper [16].
3. As this paper uses a private Ethereum network, it is solely for the authorized peers. Proof of Authority consensus needs the control of 51% of peers to attack the blockchain smart contracts, which is virtually impossible. So, it is secured.	3. This paper presents concerns about the security of smart contracts at a period of time when the Proof of Authority consensus algorithm was not established [18].
4. This paper presents a full framework for implanting blockchain technology into insurance.	4. Blockchain technologies' possibility of disrupting sectors like the insurance sector is only being discussed [19].

consensus algorithm, the network platform, methods, and so on. Apart from the differences between them, there are also several advantages and expanding scopes of this proposed system.

This paper is a novel work in the field of insurance, as it introduces some processes that were not used earlier in the corresponding field. This proposed system also has several impacts and advantages in the field of its use. It is basically a generalized approach to all types of insurance. Other existing research on this technology is pretty much focused on a

small or distinguished part. This system also introduces the use of the Proof of Authority consensus running in the core. As an insurance process mainly works within a closed circle or an authority, Proof of Authority is a very good choice in this case. This research also specifies the network system used for this whole system, a private Ethereum network. Overall, it is a complete framework for implementing blockchain technology into insurance.

In some cases, some papers did not mention or talk about their consensus algorithm. In this paper, the consensus algorithm is briefly discussed along with its config file and diagrams. Proof of Authority has been used here, which makes it unique in terms of being used in an environment like this. There are some papers in the table which did not talk about their internal feature workflow. Here in this paper, the corresponding feature algorithms are highly focused. The flowchart and use case of the system are discussed briefly in this paper. A paper talked about the smart contracts vulnerability. On the other hand, this paper solves the confusion with the use of the Proof of Authority algorithm. Also, it is considered as a whole new framework. The other papers were domain and field specific, whereas this paper focused on a generalized approach for all types of insurance.

4. Conclusion

The goal of this research is to present an insurance framework using blockchain and smart contracts. The insurance transaction process gets executed in a secure private Ethereum based decentralized system that increases security to a great extent. The usual contracts for insurance are made using smart contracts in this framework. This framework's decentralized Solidity smart contracts eliminate the complexities regarding claim settlements and insurance by their immutable nature. The use of the PoA algorithm in this framework saves a lot of storage and money. So, the framework provides an efficient and secure solution to insurance operations and functionalities.

The framework presented in this paper is not a domain-specific one. It focuses on a standard approach for standard insurance policies. For any specific kind of insurance, this framework is also prevalent with customization in the smart contract. This framework provides a secure procedure to execute the whole process with security and transparency from registration to refund in insurance. In this framework, the scalability is tested by increasing the number of peers for a fixed block size of 20. It is shown that the confirmation time increases as there are more peers. Though the confirmation process gets slower with more peers, the security increases significantly with more validators.

The proposed system framework has plenty of room for improvement in the future. It is basically implemented as a central solution to all kinds of insurance processes. If it is thought to be a specific case like the IOT-based Car Insurance Process, or other insurance processes, or fields similar to supply chain, and so on, they can be implemented easily based on the proposed framework. Thus, it has plenty of opportunities to expand.

Data Availability

No data were utilized to support this research's findings.

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

Acknowledgments

The authors are thankful for the support from Taif University Researchers Supporting Project (TURSP-2020/115), Taif University, Taif, Saudi Arabia.

References

[1] "Estimated Size of the Global Insurance Market 2020, with Forecasts up until 2025," 2020, https://www.statista.com/statistics/1192960/forecast-global-insurance-market/.

[2] E. M. Immergut, "Health policy," in *International Encyclopedia Of the Social & Behavioral Sciences*, pp. 6586–6591, Elseiver, Amsterdam, Netherlands, 2001.

[3] A. Narayanan, J. Bonneau, E. W. Felten, A. Miller, and S. Goldfeder, *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*, Princeton University Press, Princeton, NJ, 2016.

[4] "Bitcoin.org," https://bitcoin.org/bitcoin.pdf.

[5] M. Lischke and B. Fabian, "Analyzing the bitcoin network: the first four years," *Future Internet*, vol. 8, no. 4, p. 7, 2016.

[6] Government Office for Science, "Distributed ledger technology: beyond block chain," in *Government of United Kingdom*, https://www.gov.uk/government/news/distributed-ledger-technology-beyond-block-chain, 2016.

[7] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[8] W. Tsai, R. Blower, Y. Zhu, and L. Yu, "A system view of financial blockchains," in *Proceedings of the 2016 IEEE Symposium On Service-Oriented System Engineering (SOSE)*, pp. 450–457, Oxford, UK, March 2016.

[9] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun, "A review on consensus algorithm of blockchain," in *Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2567–2572, Banff, Canada, October 2017.

[10] J. Ellul and G. Pace, "Blockchain and the common good reimagined," 2019, https://arxiv.org/abs/1910.14415.

[11] O. Alfandi, S. Otoum, and Y. Jararweh, "Blockchain solution for iot-based critical infrastructures: byzantine fault tolerance," in *Proceedings of the NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, Budapest, Hungary, April 2020.

[12] V. Gatteschi, F. Lamberti, C. Demartini, C. Pranteda, and V. Santamaría, "Blockchain and smart contracts for insurance: is the technology mature enough?" *Future Internet*, vol. 10, no. 2, 2018.

[13] H. Luo, M. Das, J. Wang, and J. C. P. Cheng, "Construction payment automation through smart contract-based blockchain framework," in *Proceedings of the 36th International Symposium on Automation and Robotics in Construction (ISARC 2019)*, pp. 1254–1260, Banff Alberta, Canada, May 2019.

[14] S. N. Khan, F. Loukil, C. G. Ghedira, E. Benkhelifa, and A. H. Bani, "Blockchain smart contracts: applications, challenges, and future trends," *Peer-to-PeerNetworking and Application*, vol. 14, pp. 1–25, 2021.

[15] M. Raikwar, S. Mazumdar, S. Ruj, S. G. Sen, A. Chattopadhyay, and K. Y. Lam, "A blockchain framework for insurance processes," in *Proceedings of the 2018 9th IFIP International Conference On New Technologies, Mobility And Security (NTMS)*, pp. 1–4, Paris- France, February 2018.

[16] P. K. Singh, R. Singh, G. Muchahary, M. Lahon, and S. Nandi, "A blockchain-based approach for usage-based insurance and incentive in its," in *Proceedings of the TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, pp. 1202–1207, Kochi, India, October 2019.

[17] K. Sayegh, *Blockchain Application in Insurance and Reinsurance*, 20 pages, SKEMA Business School, Lille, France, 2018.

[18] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on ethereum smart contracts (SoK)," in *Proceedings of the International Conference on Principles of Security and Trust*, pp. 164–186, Uppsala, Sweden, April 2017.

[19] F. Holotiuk, F. Pisani, and J. Moormann, "Radicalness of blockchain: an assessment based on its impact on the payments industry," *Technology Analysis & Strategic Management*, vol. 31, no. 8, pp. 915–928, 2019.

[20] P. Tasca, "Insurance under the blockchain paradigm," in *Business Transformation Through Blockchain*, pp. 273–285, Springer International Publishing, Cham, Switzerland, 2019.

[21] Nath, "Data exchange platform to fight insurance fraud on blockchain," in *Proceedings of the 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pp. 821–825, Barcelona, Spain, December 2016.

[22] W. Chen, Z. Xu, S. Shi, Y. Zhao, and J. Zhao, "A survey of blockchain applications in different domains," in *Proceedings of the 2018 International Conference On Blockchain Technology And Application - ICBTA 2018*, Xi'an, China, December 2018.