CioSy: A Collaborative Blockchain-Based Insurance System

Faiza Loukil, Khouloud Boukadi, Rasheed Hussain, Mourad Abed

electronics

MDPI

Article

CioSy: A Collaborative Blockchain-Based Insurance System

Faiza Loukil [1,*], Khouloud Boukadi [2], Rasheed Hussain [3] and Mourad Abed [1,4]

1 Univ. Polytechnique Hauts-de-France, LAMIH, CNRS, UMR 8201, F-59313 Valenciennes, France; mourad.abed@uphf.fr
2 Miracl Laboratory, Sfax University, Sfax 3018, Tunisia; khouloud.boukadi@fsegs.usf.tn
3 Networks and Blockchain Lab., Innopolis University, 420500 Innopolis, Russia; r.hussain@innopolis.ru
4 INSA Hauts-de-France, F-59313 Valenciennes, France
* Correspondence: faiza.loukil@uphf.fr

Abstract: The insurance industry is heavily dependent on several processes executed among multiple entities, such as insurer, insured, and third-party services. The increasingly competitive environment is pushing insurance companies to use advanced technologies to address multiple challenges, namely lack of trust, lack of transparency, and economic instability. To this end, blockchain is used as an emerging technology that enables transparent and secure data storage and transmission. In this paper, we propose CioSy, a collaborative blockchain-based insurance system for monitoring and processing the insurance transactions. To the best of our knowledge, the existing approaches do not consider collaborative insurance to achieve an automated, transparent, and tamper-proof solution. CioSy aims at automating the insurance policy processing, claim handling, and payment using smart contracts. For validation purposes, an experimental prototype is developed on Ethereum blockchain. Our experimental results show that the proposed approach is both feasible and economical in terms of time and cost.

Keywords: blockchain; insurance; smart contract; Ethereum; security

Citation: Loukil, F.; Boukadi, k.; Hussain, R.; Abed, M. CioSy: A Collaborative Blockchain-Based Insurance System. *Electronics* **2021**, *10*, 1343. https://doi.org/10.3390/electronics10111343

Academic Editor: Mariusz Nowostawski

Received: 29 April 2021
Accepted: 31 May 2021
Published: 3 June 2021

1. Introduction

The insurance industry has seen an unprecedented growth during the last decade due to, at least in part, advancements in communication and computation technologies. The new futuristic technologies have positively impacted our lives in many sectors, such as health, transport, business, and so on. Like other beneficiaries of the today's cutting edge computation and communication technologies, the insurance industry is no exception. To keep up with the emerging trends, the insurance industry is also harnessing the benefits of the existing futuristic technologies. It is worth mentioning that the insurance industry covers many dimensions among which life, Property and Casualty (P&C), and health are primarily important. Without loss of generality, the processes involved in the insurance industry depend on the transacting entities for initiation, maintenance, and closure of the insurance policies [1]. Each insurance policy, which is a contract between the insurer and the insured, referred to as the policyholder, determines the claims that the insurer is legally required to pay, as well as the premium that the insurer promises to pay periodically (e.g., monthly).

The generic existing insurance systems require manual interactions across different transaction processes resulting in slow processing and lengthy payment settlement time. Moreover, the insurance industry spends tens of millions dollars each year on processing claims and loses millions of dollars to fraudulent claims [2]. To address these limitations, several researchers have investigated the application of the blockchain technology in the insurance industry [3,4]. For instance, insurance policies can be transformed into smart contracts that will eventually help in automating claim processing, verification, and payment. It will provide several-fold advantages, for instance, saving time, reducing costs, and preventing potential fraud. Indeed, a smart contract can be written to register customers

interested in purchasing policies offered by the insurer, enable them to follow their claims, and automatically receive refunds. While existing blockchain-based approaches use smart contracts to improve asset transfers, limit fraud, and reduce administrative costs, these solutions do not focus on collaborative insurance, which allows people with similar profiles to come together in small communities to insure themselves in a more supportive, fair, and transparent way [5]. Over the past few years, several collaborative insurance initiatives have been launched around the world; for instance, Guevara, a UK peer-to-peer car insurance platform that enabled its users to pool their car insurance premiums online and protect each other [6]. Collaborative insurance provides many advantages and benefits to both the insurer and the insured, namely reducing the insurance premium, improving the customer experience, and allowing the traditional insurer to invest in the innovation field in order to stand-up/resist in the increasingly competitive insurance environment. Although the existing collaborative insurance systems aim at providing a peer-to-peer platform to enable direct communication for customers, none of them relies on the blockchain technology in order to propose an untrustworthy, transparent, and tamper-proof network. To the best of our knowledge, the combination of collaborative insurance and blockchain technology has never been explored.

In order to address the aforementioned limitations of the existing solutions, we propose a new collaborative architecture, called CioSy that aims at executing and monitoring insurance policies using blockchain technology. The proposed solution saves time, reduces administrative and operational costs, and enables manual interactions across different insurance system entities. More precisely, a smart contract is provided to insurers (e.g., individuals, banks, insurance companies, etc.) to team up, pool their contributions/premiums, and protect each other. This smart contract includes a set of the created insurance policies which are the agreements among the insurers and the insured. The insurance policy is mapped into a smart contract that communicates with external oracles in order to automatically invoke a new claim and transfer to the insured (i.e., the insurance policy smart contract's owner) the claimed amount. The reasons behind using the blockchain technology are: to guarantee that the collaborative insurance model is followed by each participant, to improve the insurance transaction transparency, to guarantee the non-repudiation principle conformity among several untrustworthy collaborative entities, to automate and speed up business processes in the insurance industry from the insured registration till the claim handling, and to reduce administrative and operational costs by eliminating manual interactions across insurance system entities.

The main contributions of this paper are summarized as follows:

1. We highlight the benefits of using the blockchain technology and smart contracts to enable peer-to-peer collaborative insurance, where customers rely on each other to meet their insurance needs while eliminating a centralized authority and adopt them in a new scheme, called CioSy. To the best of our knowledge, none of the existing solutions for digitized insurance consider covering all the insurance business process steps (from the insurer teaming up until the claim handling).
2. We introduce the machine-readable and self-enforcing insurance policies and claims based on voting mechanisms and external oracles as well as detail the working principles of the proposed blockchain-based insurance system.
3. We evaluate CioSy performance and scalability when the number of insurance claims and insurers increases. We analyze the performance of the proposed system in terms of execution time, scalability overhead, and the consumed gas.

The rest of this paper is organized as follows. Section 2 discusses the existing blockchain-based insurance approaches and Section 3 introduces some key concepts followed by our proposed collaborative blockchain-based insurance scheme in Section 4. Section 5 describes the main framework functionality, and Section 6 discusses the implementation details and performance evaluation of the proposed scheme. Finally, Section 7 concludes the paper with possible future directions.

2. Related Work

Blockchain technology has proved to be a disruptive technology for many sectors including finance, governance, trade, ownership, and so on. Indeed, blockchain is used for cost reduction, transparency, and tokenization of objects. For instance, Fallucchi et al. [7] aimed to "certify the data" without the need for a centralized organization using blockchain to ensure traceability by storing data hashes in blockchain transactions and IPFS to guarantee the data availability. Several governments are launching blockchain technology pilot projects [8,9] to adapt to new technologies. According to the authors, their framework could be easily applicable to the Valls city council open data portal project, which publishes the datasets in the municipal web portal using blockchain and IPFS technologies [8] or to the blockchain project funded by the Department of Homeland Security (DHS) for secure digital identity management [9].

Similarly, blockchain technology has also penetrated the insurance industry and it can be a game-changer technology to address the limitations of the existing insurance solutions including claim processing, automated payment, asset transfers, and limit fraud [2]. In this context, we can distinguish between three categories, namely automated claim handling insurances [10,11], pay-per-use insurances [3,4], and peer-to-peer insurances [12,13].

In [10], oracles are used to gather information from the real-world and invoke automatically the smart contract pertaining to the claim. For instance, in travel insurance, an oracle can periodically check flight status, then a smart contract can read these external data, and trigger a payment to refund the insured travelers in case of a flight delay. Insure chain [11] is another interesting proposal that is based on a smart contract which includes the rules associated with setting the premium and the settlement verification. The verification of reimbursement conditions is based on the services of an oracle whose task is to certify the corresponding weather data and ensuring its authenticity. Using oracles can speed up claim handling and reduce manual administrative mistakes; however, relying on external oracles can be adopted only for a limited number of use-cases. The majority of claims are processed by insurance companies that still need to be evaluated by an expert before being settled.

Smart contract-based payments have enabled new revenue sources, such as pay-per-use insurances [3,4]. Lamberti et al. [3] proposed an on-demand car insurance system using smart contracts and Internet of Things (IoT) for decreasing policy modification costs and limiting insurance fraud. Similarly, Vo et al. [4] proposed a blockchain-based pay-as-you-go car insurance application. This application allows drivers who rarely use cars to only pay the insurance premium for particular trips they would like to travel. Blockchain-based pay-per-use insurances can save the insured money compared to classic insurance offers and bring the insurance company a competitive advantage by attracting young customers and technology-enthusiast customers. However, these pay-per-use mechanisms require near-real time data of the insured in order to limit the insurance fraud. Thus, these approaches need to focus on one non-functional requirement of insurance applications which is privacy protection for the insured.

Peer-to-peer insurance models enable the transfer of an asset without the need for an intermediary. For instance, Friendsurance [12] is a digital, scalable, and modular digital bank-assurance platform for banks, insurers, and other partners who want to retain and monetize their customers through meaningful services. Moreover, Dynamis [13] is a blockchain-based peer-to-peer insurance system aiming at providing unemployment insurance for a community of self-managed people in terms of underwriting and claims acceptance and processing. However, according to [2], existing peer-to-peer insurances are not "real" peer-to-peer models because they have a traditional insurance model or risk carrier behind them to support the heavy part of the insurance business.

Permissionless blockchains are decentralized systems designed to allow anyone to join the system, including Bitcoin and Ethereum, whereas permissioned blockchains are blockchain systems in which the participation in some or all roles is restricted to a set of users, such as Hyperledger Fabric. In the insurance industry, several distributed ledger technologies, such as Ethereum [10–13], Hyperledger Fabric [14,15], and IOTA [16] have

been used. For instance, BlockCIS [14] is a cyber insurance system that aimed to provide an automated, real-time, and immutable feedback loop among the involved parties for assessing cyber risks. It has been built using the permissioned Hyperledger blockchain framework to isolate enterprise transactions from public access. While permissioned blockchains solve low performance and limited data confidentiality capabilities of permissionless blockchains, they come at the cost of sacrificing complete decentralization.

To this end, we note that the existing blockchain-based solutions for insurance used blockchain technology to automate the payment while eliminating the intermediary entities. However, they still use a traditional insurance model. While our solution also uses the automation feature of the smart contract, it is to be thought of as a continuous processing peer-to-peer insurance system. Such collaborative insurance could be dedicated to electric cars, pet care, etc., but is not proposed to replace the traditional insurance model. Therefore, we aim at proposing a new collaborative insurance model to allow people who have similar profiles to come together in small communities to insure themselves in a more supportive, fair, and transparent way in an untrustworthy and tamper-proof network. We also note that the current solutions do not address the necessity of claim expert views in terms of claim verification in both cases of automating claim handling and peer-to-peer insurances. Therefore, in this paper, we address this limitation by relying on oracles only to automate a claim creation. Then, in order to take into consideration the created claim, authorizations from both the insurer and the insured are required.

3. Blockchain Technology and Smart Contracts for Insurance

In this section, we present the key concepts of blockchain and smart contracts, and then we introduce the impact of using blockchain in the insurance industry.

3.1. Blockchain and Smart Contracts

Blockchain is a distributed and chronological database of transactions where the transactions are stored in blocks. It is almost impossible to tamper with the blocks in blockchain and thus it can be trusted. Trust is built in blockchain without the need for a central authority. Such a distributed ledger can contain digital or physical assets that can be shared in a network throughout many institutions and geographical locations where all members of the network must have their identical copy of the ledger [17]. Blockchain technology was originally designed to play a role primarily in the financial field, but in recent years it has also been exploited in other areas, such as healthcare information exchange [18], fairness-based packing of industrial IoT data [19], supply chain management for food traceability [20], and blockchain-based solutions for insurance [10–13].

Smart contracts are computer programs deployed on the blockchain. They are triggered and perform pre-defined actions when specific conditions are met. The smart contract functions will always respond when invoked and they cannot be censored or altered once deployed [17]. Smart contracts gave network automation and the ability to convert paper contracts into digital contracts. Compared to traditional contracts, smart contracts enabled users to codify their agreements and trust relations by providing automated transactions without the supervision of a central authority [21]. It is also worth mentioning that smart contracts cannot communicate directly with the external systems, thus this communication is carried out by oracles. An oracle is a third-party information source that provides information to the smart contract in blockchain through Application Programming Interfaces (APIs). For instance, Provable Ethereum API is the leading oracle service for smart contracts and blockchain applications [22].

3.2. Blockchain at the Service of Insurance

Blockchain provides many advantages and benefits to financial engineering and particularly the insurance sector:

Automation: Smart contracts provide a high degree of automation by encoding the business rules of an insurance policy in software code deployed on the blockchain. The business processes in the insurance industry are automated and fast (from the customer registration all the way to claim handling).

Time saving: Without involving the banks, asset transfers can be made faster because cryptocurrencies are directly moved from a wallet's address to another without intermediate steps. Thus, the blockchain-based transactions are quicker than traditional bank transfers (especially in the case of overseas asset transfer).

Reduced cost: By removing intermediaries, the cost of money transfers can be reduced (e.g., bank commissions are not needed). Moreover, by eliminating manual interactions across insurance system entities, the administrative and operational costs can be reduced.

Improved transparency: Transparency is guaranteed because the blockchain can be accessed worldwide. In addition, the blockchain can become the repository of a huge amount of information that cannot be repudiated and can be used for data analytic in the insurance sector. Thus, such transparency enables regulators and auditors to detect suspicious transaction patterns and market behaviors.

In this section, we present a system model and the description of the involved entities in blockchain-based insurance system.

3.3. System Model Main Goals

Although multiple researchers have studied the impact of blockchain technology on the insurance industry, there are still many outstanding challenges to be addressed to enable collaborative insurance. In essence, the collaborative insurance model is inspired by the concept of a collaborative economy. The latter *"is an economic model where ownership and access are shared among corporations, startups, and people. This results in market efficiencies that bear new products, services, and business growth"* [23]. Therefore, collaborative insurance, as defined/considered in this paper, is a peer-to-peer network where customers rely on each other instead of traditional insurance companies to meet their insurance needs with the help of a web-based middleman. Several profit-driven models of the collaborative economy, such as Airbnb, Uber, etc., exist. For the insurance industry, eliminating the necessity to trust a middleman (e.g., a web-based middleman) is required in order to incite customers to join the peer-to-peer insurance network and share their money. Moreover, a distributed data storage is needed to eliminate the single point of a trust/failure problem where all the collaborative insurance transactions are stored by a centralized authority. Furthermore, a machine-readable and self-enforcing insurance policy need to replace the traditional insurance policy in order to automate and speed-up the insurance business process. Finally, a collaborative insurance solution needs to take advantage of the digital signature in order to guarantee the three security properties, i.e., confidentiality, integrity, and the sender's identity (i.e., authentication data) during insurance data exchange to limit any potential fraud. To the best of our knowledge, none of the existing solutions for digitized insurance consider all the aforementioned requirements throughout the insurance business process (from the insured registration until the claim handling). For this purpose, we propose CioSy, a blockchain-based collaborative insurance solution that harnesses the benefits of both blockchain technology and smart contracts. The main idea is to develop a continuous monitoring and processing collaborative insurance system by (i) managing the collected money of the insurers using a smart contract to eliminate the need to trust the involved insurance parties, (ii) implementing the insurance policies and the claims as smart contracts, and (iii) deploying the contracts in a distributed platform using blockchain for both automating the execution of the agreement between the insurer and the insured as well as (iv) recording all the insurance transactions in a transparent and tamper-proof manner.

3.4. Functional Entities

As depicted in Figure 1, there are mainly four entities in our system: insured, insurer, third-party web APIs, and auditor. The role of these entities is described as follows.

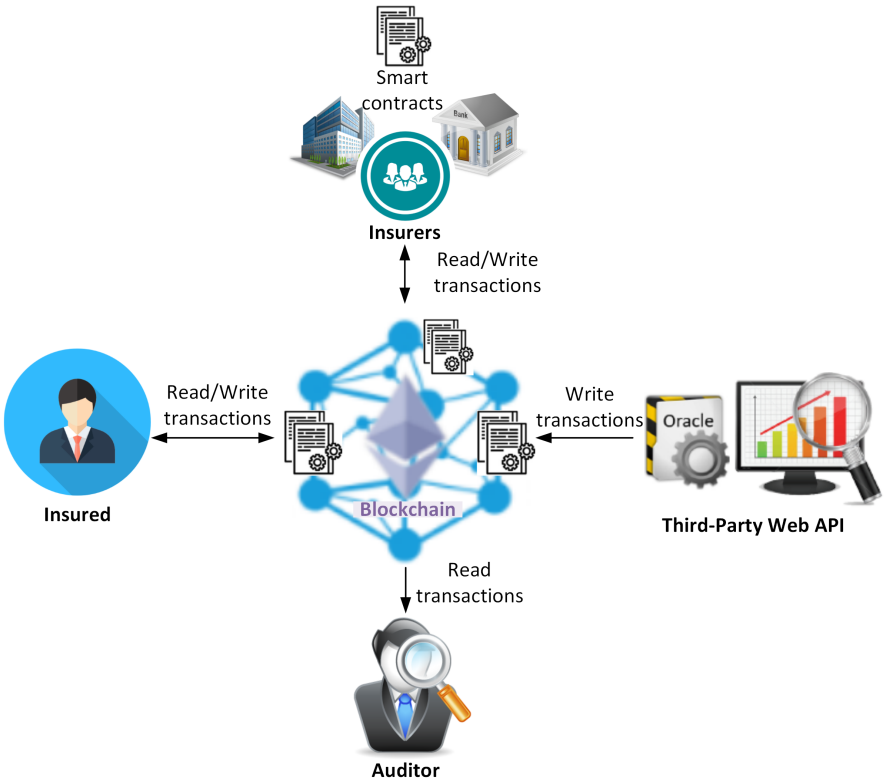

Figure 1. CioSy system model overview.

4. System Model for Blockchain-Based Insurance System

1. **Insured**: This entity is interested in purchasing policies offered by an insurer in order to be covered (depending on the type of insurance). In case of a claim request, the insured can receive refunds from the insurer (subject to verification).
2. **Insurer**: This entity is represented by a smart contract shared among several customers (e.g., people, banks, insurance companies, etc.). These customers collaborate together to pool their contributions/premiums and protect each other by refunding any customer facing a situation that warrants a refund. The refunded entity is then required to contribute to the fund that is used to pay future claims. This smart contract provides insurance for the insured by proposing multiple insurance policy types. Each insurance policy determines the claims that the insurer is legally required to pay. To reduce manual interactions, claim requests are automatically invoked after notifications/warnings are sent by third-party web APIs.
3. **Third-Party Web APIs**: These entities provide specialized services that are useful to invoke claim requests; for instance, an airline's API that notifies the policy smart contract in case of a flight delay or IoT-equipped vehicles that report near-real time accidents. In case of a dispute between the insurer and the insured, there may be a need for an auditor.
4. **Auditor**: This entity investigates and audits the insurance transactions stored on the blockchain to settle some legal dispute between the insurer and insured. The blockchain-enabled distributed platform facilitates the auditor's task, thanks to the salient features of blockchain, i.e., transparency, tamper-proof, and the non-repudiation.

In the following section, we discuss the proposed blockchain-based framework for insurance in detail.

5. Blockchain-Based Framework for Insurance

To exchange insurance-related transactions in an untrustworthy network, we leverage smart contracts to define a blockchain-based insurance framework. The latter aims at automating and speeding up business processes in the insurance industry, improving insurance transaction transparency, non-repudiation, and reducing administrative and operational costs by eliminating manual interactions across insurance system entities.

5.1. Proposed Smart Contracts

In order to automate the execution of the agreement between the insurer and the insured, three smart contracts are proposed: `InsurancePool`, `InsurancePolicy`, and `Claim`. Figure 2 shows the proposed smart contracts, which automatically enforce the agreement between the insurer and the insured. The smart contract's functions are executed when a set of predefined conditions are satisfied.

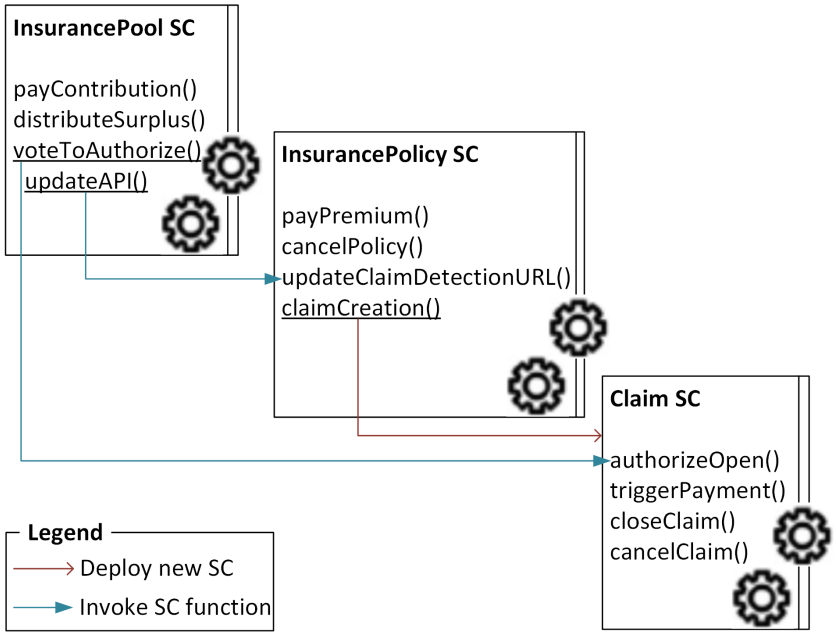

Figure 2. Smart contracts for insurance processes.

- `InsurancePool` **smart contract**: it is hosted on the blockchain and used by multiple clients interested in proposing multiple insurance offers. The `InsurancePool` smart contract is designed to enable several insurers to collaborate on a common project that offers a collaborative insurance to refund the insured for possible damage(s) during designated incidents. This smart contract defines a set of functions, i.e., a (i) `payContribution` function that enables each insurer to participate by paying an amount of money to the insurance pool, (ii) `updateAPI` function that enables the insurers to update the link to the third-party web API, this function invokes automatically one of the `Claim` smart contract's functions, (iii) `voteToAuthorize` function that enables the insurers to decide whether to authorize or not opening a claim, this function also invokes automatically one of the `Claim` smart contract's functions, and (iv) `distributeSurplus` function that is invoked at the end of the year in order to distribute the surplus of the collected money to all insurers who have not had any claims during the last year (each according to its contribution).
- `InsurancePolicy` **smart contract**: it is created by the customer interested in purchasing policies offered by the insurer and hosted on the blockchain. The `InsurancePolicy` smart contract is designed to enable the insured, known as the *policyholder*, to hold

a machine-readable and self-enforcing insurance policy. This smart contract defines a set of functions, namely a (i) `payPremium` function that enables the policyholder to pay periodically a premium which is a fixed amount of money to the insurer, (ii) `cancelPolicy` function that enables the policyholder to cancel a purchased insurance policy, thus the insurance policy status is updated to "Canceled" and the insurance policy is canceled, (iii) `updateClaimDetectionURL` function that is invoked by the `updateAPI` function defined in the appropriate `InsurancePool` smart contract instance in order to update the link to the third-party web API, and (iv) `claimCreation` function that is connected to an external third-party web API, then in case of a received claim notification, this function creates/deploys automatically a new instance of the `Claim` smart contract. The `InsurancePolicy` smart contract inherits the `usingProvable` smart contract [22] which helps to connect our proposed smart contract with the external data provided by the third-party web APIs.

- `Claim` **smart contract**: it is created by an `InsurancePolicy` smart contract and hosted on the blockchain. The `Claim` smart contract is designed to automate the claim processing, verification, and payment. This smart contract defines a set of functions, namely a (i) `authorizeOpen` function that enables the insurer to update the status of the claim from "Created" to "Open" or to "Rejected", this function is invoked by the `voteToAuthorize` function defined in the `InsurancePool` smart contract, (ii) `triggerPayment` function that is automatically invoked in order to refund the claimed amount and ask for closing the claim, (iii) `closeClaim` function that is automatically invoked once the insured is refunded by updating the claim's status to "Closed", and (iv) `cancelClaim` function that enables the insured to cancel a claim, then the claim status is updated to "Canceled" and the claim is canceled.

Due to the lack of space, we provide the full definition of the smart contracts at our Github-repository (https://github.com/Floukil/BlockchainBasedInsurance, accessed on 1 June 2021). Now, we explain the main functions offered by and working principles of the proposed framework for blockchain-based insurance.

5.2. Main Functions and Working Principles of the Proposed Blockchain-Based Insurance

Based on the proposed smart contracts, our blockchain-based insurance framework includes the following functions: (i) gathering the insurers as a collaborative community, (ii) purchasing an insurance policy offered by an insurer, (iii) creating a claim by the insurance policy, and (iv) automating claim processing and refunding payment.

5.2.1. Gathering the Insurers as a Collaborative Community

In order to gather several insurers as a collaborative community, an interested insurer can initiate a collaborative insurance network through the following steps:

- Step 1: Create (i.e., write and compile) an `InsurancePool` smart contract.
- Step 2: Send a transaction to deploy the created smart contract onto the blockchain. Once hosted, the `InsurancePool` smart contract instance got a unique blockchain address.

In order to participate in the collaborative insurance, other interested customers can contribute to the insurance pool through the following steps:

- Step 3: Send a transaction to call the `payContribution` function defined in the created `InsurancePool` smart contract to pay periodically an amount of money.

Then, each insurer can purchase an insurance policy in order to be insured.

5.2.2. Purchasing an Insurance Policy Offered by an Insurance Pool

In order to facilitate the management of the insurance, an interested customer can purchase an insurance policy offered by an insurance pool through the following steps:

- Step 1: Create (i.e., write and compile) an `InsurancePolicy` smart contract.
- Step 2: Send a transaction to deploy the created smart contract onto the blockchain with a precise insurer's blockchain address (i.e., the blockchain address of the `Insurance`

`Pool` smart contract) and the fixed premium payment amount. The sender of this transaction becomes the owner of the `InsurancePolicy` smart contract instance, known as the policyholder.

- Step 3: Send a transaction by the defined insurer to call the `updateClaimDetectionURL` function defined in the `InsurancePolicy` smart contract to update the link to the third-party web API, which is responsible for claim notifications.

5.2.3. Creating a Claim by the Insurance Policy

In case of a designated incident (e.g., car accident, flight delay, etc.), an `InsurancePolicy` smart contract instance can create a new claim through the following steps:

- Step 1: Receive a notification from an associated third-party web API that a potential claim has happened.
- Step 2: Call internally the `claimCreation` function defined in the `InsurancePolicy` smart contract by the `callback` function of the `usingProvable` smart contract [22] in order to deploy a new `Claim` smart contract instance onto the blockchain. The same `InsurancePolicy` smart contract instance can create multiple `Claim` smart contract instances.

Once created, both the insurer and the insured receive the blockchain address of the new `Claim` smart contract instance.

5.2.4. Automating Claim Processing and Refund Payment

Both the insurer and the insured can interact with the `Claim` smart contract instance through the following steps:

- Step 1: The insured sends a transaction to call the `cancelClaim` function defined in the `Claim` smart contract to cancel the claim and update the claim's status from "Created" to "Canceled". One of the most common reasons why the insured might want to cancel a claim is not wanting to pay the deductible, which is the amount of money paid by the insured before the insurer refunds the claimed amount.

The objective of the previous step is two-fold: either the insured abandons the refund by canceling the claim or authorizes the claim handling. In case of an authorization, the insurer starts the claim handling through the following steps:

- Step 1: Send a transaction by the insurer to call the `authorizeOpen` function defined in the `Claim` smart contract to authorize opening the claim or reject it after a claim verification by claim experts and the insured confirmation.
- Step 2: Once the claim is open, call internally the `triggerPayment` function in order to transfer the deductible amount from the insured account to the insurer account and transfer the claimed amount from the insurer account to the insured account.
- Step 3: Call internally the `closeClaim` function to update the claim's status to "Closed".

6. Performance Evaluation

This section provides experimental results to evaluate performance and demonstrate the feasibility of the proposed framework. We first introduce the experimental setup, define a use-case for insurance policy handling, and evaluate the performance of the proposed blockchain-based solution. Finally, we discuss the relationship of premiums, refund payments, and risk management in collaborative insurance systems.

6.1. Experimental Setup

Ethereum is currently the most commonly used blockchain platform for the development of smart contracts [17]. Hence, we implemented our proposed smart contract using the Solidity language [24] and deployed it to the Ethereum test network. The experiments were performed based on a Remix tool that supports testing, debugging, and deploying smart contracts on Ethereum blockchain. In order to deploy a lightweight blockchain, we used Ganache as a personal blockchain for Ethereum development [25]. Therefore,

we created a test system using a Truffle development framework [26], which is the most popular development framework for Ethereum. The reason behind the use of the Ethereum blockchain is that Ethereum enables developing and executing advanced and customized smart contracts using the programming language Solidity, which is supported by other blockchain platforms, such as Hyperledger Fabric that supports multi-language smart contracts. Thus, the proposed design can be supported by several blockchain platforms. While alternative blockchains are emerging, Ethereum is considered as the most solid and widespread blockchain that allows decentralized applications to be built on top of it. All the experiments were conducted on a computer with an Intel Core i5 CPU (running at 2.30 GHz with 8 GB RAM).

6.2. Use-Case: Insurance Policy Handling

We implemented a test system that consists of several nodes, namely 50 insurers, 1000 insured, 1 third-party Web API, and 1 auditor. We assumed that each node (except the third-party Web API) is represented by an Ethereum address associated with a pair of public/private keys. In our experiments, we used the contract events to automate the actions taken by the different nodes. Then, we implemented event callbacks in the test system using the web3.js library [27].

Suppose a home is equipped with sensors monitored by smart applications that can notify a smart contract in case of serious damage or a designated event. Thus, such an application can initiate a claim or contact a repairer for a quicker assistance when needed. Let the insurer be a group of individuals, the insured be the home's owner, and the third-party Web API be an API that is provided by the home applications. First, an individual creates an instance of the `InsurancePool` smart contract on the blockchain to create a collaborative group that insures group members against water damage. Each group member can purchase an insurance policy to be insured. Thus, the home's owner creates an instance of the `InsurancePolicy` smart contract on the blockchain to generate an insurance policy while indicating the blockchain address of the insurer. The created smart contract serves as the home's insurance policy that is connected to the home application API. In case of damage, the home application API notifies the `InsurancePolicy` smart contract that waits for both the insurer and the insured authorizations to deploy automatically an instance of the `Claim` smart contract. Suppose that the home's owner does not want to pay a lot of money if a group member claims damages so he/she sets a fixed limit amount per single premium in his/her insurance policy. To ensure fairness, in case of a claim, the home's owner will not receive more than his/her fixed limit premium per group member even if the other group members accept to pay a higher premium in their insurance policies. At the end of the year, if the insurers have no claims, they recover part of their contribution thanks to the `distributeSurplus` function.

For validation purposes, we show in the rest of this section the insurance policy handling feasibility by implementing smart contracts, and interacted with them by sending a set of transactions. During our experiments, we recorded the computing time, in milliseconds, of each aforementioned step. Each step consisted of one or several transactions that invoked the appropriate smart contract's functions in order to read or write on the deployed smart contract.

6.3. Performance Evaluation Metrics

To evaluate the performance of our proposed scheme, we consider computation overhead in terms of time cost, scalability overhead, the consumed gas, and computational cost for claim authorization.

6.3.1. Computation Time Cost

We compute the processing time needed to validate a premium payment transaction that invoked a chain of functions that include a `payPremium` function, an API link update transaction that invoked the `updateClaimDetectionURL` function, a claim creation trans-

action that invoked the `claimCreation` function, and a policy canceling transaction that invoked the `cancelPolicy` function. Thus, we measure the processing time of invoking the aforementioned functions defined in the `InsurancePolicy` smart contract. Figure 3 depicts the computational cost of the invoked functions for one insured entity. Only 222 milliseconds are needed to deploy a new `Claim` smart contract instance by an insured instead of the several minutes or hours it usually takes when using a traditional insurance policy contract. The processing time of the other invoked functions varies from 100 to 160 ms. Thus, our solution is able to meet our fixed objectives, namely fast insurance business processes, reduce administrative and operational costs by eliminating manual interactions, and record the insurance transactions in a tamper-proof manner.

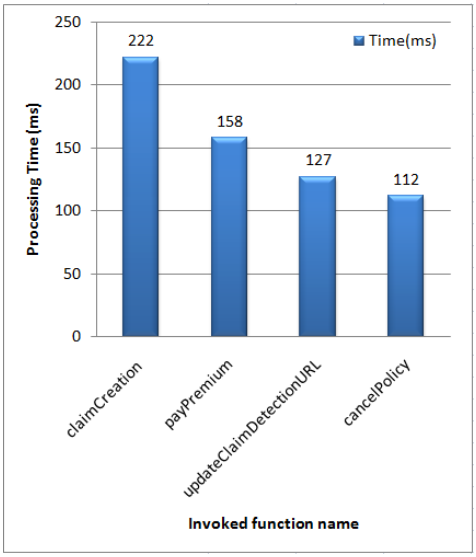

Figure 3. Computational cost for one insured's insurance policy.

6.3.2. Scalability Overhead

We generated up to 1000 different insured accounts, and compiled and deployed an `InsurancePolicy` smart contract for each account. Figure 4 depicts the time taken while invoking the `InsurancePolicy` smart contract functions. We observe that the processing time increases with the number of insured accounts, ranging from 0 to 240 s. Furthermore, Figure 4 shows that the total time required for smart contract interaction is strictly linear to the number of the insured accounts.

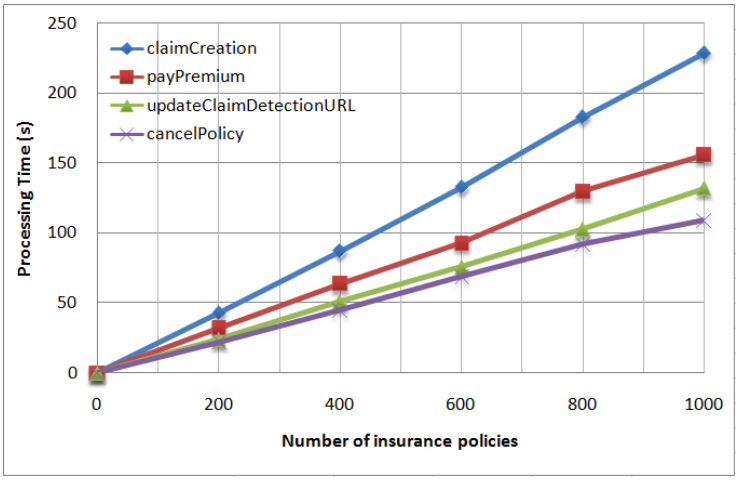

Figure 4. Computational cost for 1000 insureds' insurance policies.

6.3.3. Cost Overhead

We also evaluated the consumed gas by a transaction while invoking one of the `InsurancePolicy` smart contract's functions, namely `claimCreation`, `payPremium`, `updateClaimDetectionURL`, and `cancelPolicy` after deploying it in the blockchain. Figure 5 depicts the cost incurred in terms of gas by different invoked functions of the `InsurancePolicy` smart contract. We observe that the consumed gas is changed by changing the invoked function. This can be explained by the fact that the functions that need more computational resources cost more gas than functions that require fewer computational resources. As expected, the deployment of a new `InsurancePolicy` smart contract instance and the invocation of the `claimCreation` function, which deploys automatically a new `Claim` smart contract instance, require more gas than the invocation of the rest of functions.

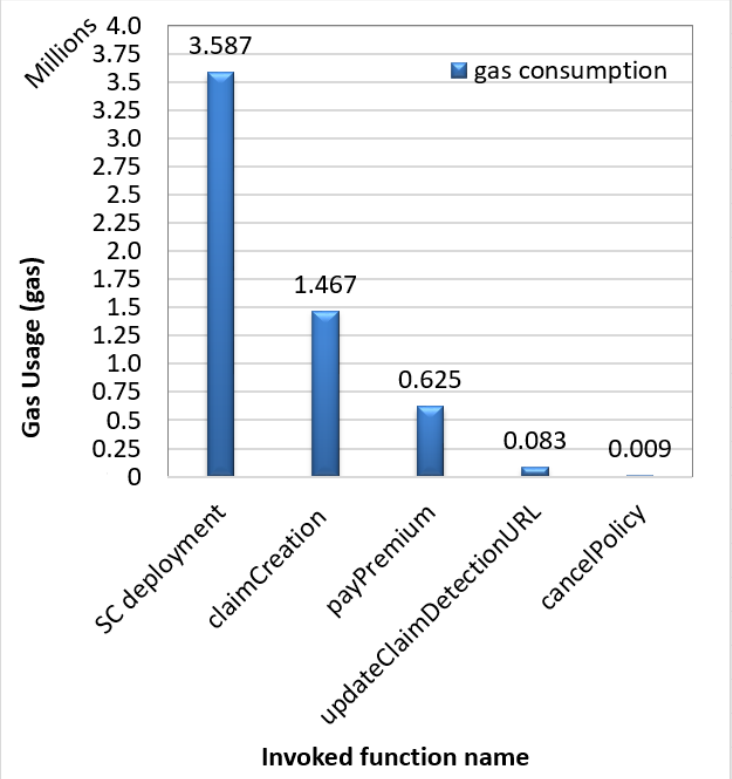

Figure 5. Gas usage with different invoked functions.

Table 1 illustrates the cost overhead of deploying a new `InsurancePolicy` smart contract instance and invoking each predefined function.

Table 1. Cost overhead.

Invoked Function	Average Gas Usage (Gas)	Average Gas Cost (USD)
`SC Deployment`	3 586 838	158.61
`claimCreation`	1 467 150	64.88
`payPremium`	62 4547	27.62
`updateClaimDetectionURL`	82 882	3.66
`cancelPolicy`	9 409	0.42

Currently, 1 gas costs about 20 Gwei (i.e., 20×10^{-9} Ether) and the exchange rate is about 2211 USD for 1 Ether at the time of writing. Thus, we compute the gas cost by multiplying the used gas by the gas price for transactions that invoke the smart contract functions.

6.3.4. Computational Cost for Claim Authorization

We compute the confirmation time to authorize opening or rejecting a claim while varying the number of insurer accounts from 1 to 50. Thus, we measure the processing time of invoking the `authorizeOpen` function defined in the `Claim` smart contract. As aforementioned, the `authorizeOpen` function is invoked by the `voteToAuthorize` function defined in the `InsurancePool` smart contract once all the insurers give their votes. Figure 6 depicts the correlation between the number of collaborative insurance community members (i.e., insurers) and the corresponding time taken for opening or rejecting one claim. We observe that the community size is proportional to the authorization time. Therefore, the larger the number of insurers, the higher the computational cost for claim authorization. This is caused by the requirements for a larger number of insurer votes. Even with a significant number of collaborative insurance community members, the computational cost for claim handling is lower that of a traditional insurance system that requires manual interaction and human confirmation.

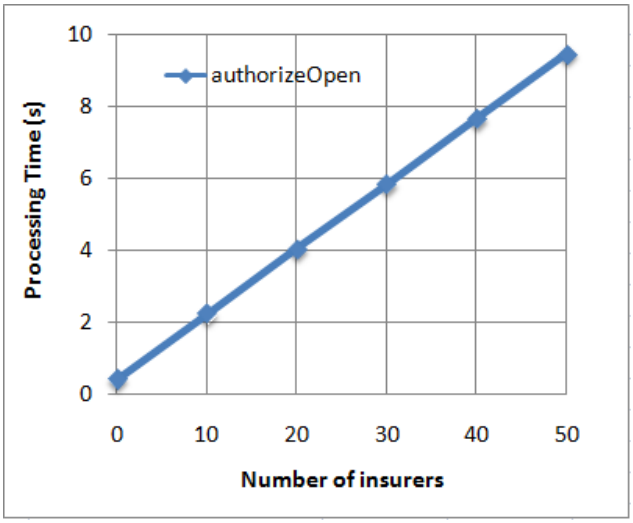

Figure 6. Computational cost for claim authorization by 50 insurers.

6.4. Discussion

The proposed framework is a peer-to-peer insurance where the insured are comfortable just sharing risk with each other. Thus, CioSy is more suitable for insurance products with a very low expected value of risks. In fact, solidarity is a major rule in such collaborative insurance. By using the premiums, everybody pays for the damages of the others and may not wish to generate profit. However, catastrophes may happen, such as a hurricane that might disrupt flights for a few days or a volcano eruption that closes down numerous airports for several days. Thus, premiums could be insufficient to fulfill all the claims. Therefore, a collaborative insurance requires a multi-layer risk management strategy to offload and manage the risks involved. For instance, villagers setting up a collaborative insurance need to manage a catastrophe risk in case of a disaster that hits their village and affects everyone. Therefore, two possible solutions could be considered, including reinsurance by traditional insurance companies and collateralization with cryptographic tokens. The first one is relying on insuring the actual insurance pool by another insurance company, which in turn is often insured by yet another, usually state-owned, insurer. The second solution consists of tokenizing one or more of the insurance pools and making them available to any investor seeking opportunities to earn passive income on their crypto assets [28].

In order to keep the decentralization characteristic of our proposed framework and prevent relying on one traditional reinsurance company, CioSy could be extended to include a risk pool, where a portion of the premium originally paid by the insureds is paid to investors who are willing to accept a catastrophe risk. In this context, risk pool

tokens, which are proposed based on the Etherisc Protocol [29], could be used. Therefore, investors could buy risk pool tokens by paying some ethers to an appropriate smart contract. In the absence of a catastrophe, the token holder periodically receives a portion of all premiums paid by insureds as compensation for insuring the risk of a catastrophe, and on the expiration date of the token, the smart contract returns the original investment to the token holder's wallet. If there is a catastrophe, the tokens' holders may lose some or all of the original investments. All the previous improvements in the proposed framework need more investigations in future work.

7. Conclusions

In this paper, we proposed CioSy, a collaborative blockchain-based insurance system for monitoring and processing insurance transactions. We discussed the proposed framework's main functionality and implemented it on Ethereum blockchain with smart contracts. We also conducted experiments to evaluate the performance of the proposed scheme. The obtained results showed that our approach is feasible and enables time and cost savings. The most relevant issue on Ethereum blockchain is the gas price [30]. Developments are in progress and several improvement proposals would solve the problem. One of the most well-known proposed solutions is the Ethereum Improvement Proposal (EIP-1559) [31], while Ethereum 2.0 will use proof-of-stake consensus instead of proof-of-work, which is less expensive and more energy-efficient.

As future work, we plan to provide a formal security proof for the proposed model. Furthermore, we also plan to examine the possibility of investing the collected money by an insurance pool using the blockchain technology in order to incite banks and insurance companies to join the proposed collaborative insurance system.

Author Contributions: Conceptualization, F.L., K.B., R.H.; methodology, F.L., K.B., R.H.; software, F.L.; validation, F.L., K.B., R.H., and M.A.; formal analysis, F.L., K.B., R.H., and M.A.; investigation, F.L.; resources, F.L.; data curation, F.L.; writing—original draft preparation, F.L.; writing—review and editing, K.B., R.H., and M.A.; visualization, F.L.; supervision, K.B., R.H., and M.A.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Raikwar, M.; Mazumdar, S.; Ruj, S.; Gupta, S.S.; Chattopadhyay, A.; Lam, K.Y. A blockchain framework for insurance processes. In Proceedings of the 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Paris, France, 26–28 February 2018; IEEE: New York, NY, USA, 2018; pp. 1–4.
2. Gatteschi, V.; Lamberti, F.; Demartini, C.; Pranteda, C.; Santamaría, V. Blockchain and smart contracts for insurance: Is the technology mature enough? *Future Internet* **2018**, *10*, 20. [CrossRef]
3. Lamberti, F.; Gatteschi, V.; Demartini, C.; Pelissier, M.; Gomez, A.; Santamaria, V. Blockchains can work for car insurance: Using smart contracts and sensors to provide on-demand coverage. *IEEE Consum. Electron. Mag.* **2018**, *7*, 72–81. [CrossRef]
4. Vo, H.T.; Mehedy, L.; Mohania, M.; Abebe, E. Blockchain-based data management and analytics for micro-insurance applications. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 2539–2542.
5. Asquarepartners. Collaborative Insurance: Takeoff Impending? 2019. Available online: https://www.asquarepartners.com/collaborative-insurance-takeoff-impending/ (accessed on 1 April 2021).
6. InsurTech. Guevara, 'People to People' insurance. A Fairer and More Transparent Insurer. 2014. Available online: https://insur-tech.com/Startup/guevara/ (accessed on 1 April 2021).
7. Fallucchi, F.; Gerardi, M.; Petito, M.; De Luca, E.W. Blockchain Framework in Digital Government for the Certification of Authenticity, Timestamping and Data Property. In Proceedings of the 54th Hawaii International Conference on System Sciences, Maui, HI, USA, 5–8 January 2021; pp. 2307–2316.
8. Valls. Valls City Council Open Data Portal. 2019. Available online: https://dadesobertes.valls.cat/en/about (accessed on 15 May 2021).
9. Homeland Security. DHS S&T Announces New Collaborative Blockchain Innovation Solution. 2018. Available online: https://www.dhs.gov/science-and-technology/news/2018/12/04/news-release-st-seeks-collaborative-blockchain-innovations (accessed on 15 May 2021).

10. Bertini, T.; Butkute, K.; Canessa, F. Smart Flight Insurance—InsurETH. 2015. Available online: http://mkvd.s3.amazonaws.com/apps/InsurEth.pdf (accessed on 1 April 2021).
11. Replay. Insure Chain—Blockchain Insurance Management. 2017. Available online: https://www.reply.com/en/content/insurechain (accessed on 1 April 2021).
12. Kunde, T.; Herfurth, S.; Meyer-Plath, J. Friendsurance: The P2P Insurance Concept. 2017. Available online: https://www.friendsurance.com/ (accessed on 1 April 2021).
13. Joshua, D. Peer to Peer Insurance on an Ethereum Blockchain. 2018. Available online: http://www.dynamisapp.com/whitepaper.pdf (accessed on 1 April 2021).
14. Lepoint, T.; Ciocarlie, G.; Eldefrawy, K. BlockCIS—A blockchain-Based Cyber Insurance System. In Proceedings of the IEEE International Conference on Cloud Engineering (IC2E), Orlando, FL, USA, 17–20 April 2018; IEEE: New York, NY, USA, 2018; pp. 378–384.
15. Aleksieva, V.; Valchanov, H.; Huliyan, A. Implementation of Smart Contracts based on Hyperledger Fabric Blockchain for the Purpose of Insurance Services. In Proceedings of the International Conference on Biomedical Innovations and Applications (BIA), Varna, Bulgaria, 24–27 September 2020; IEEE: New York, NY, USA, 2020; pp. 113–116.
16. Suciu, G.; Nădrag, C.; Istrate, C.; Vulpe, A.; Ditu, M.-C.; Subea, O. Comparative analysis of distributed ledger technologies. In Proceedings of the 2018 Global Wireless Summit (GWS), Chiang Rai, Thailand, 25–28 November 2018; IEEE: New York, NY, USA, 2018; pp. 370–373.
17. Buterin, V. Ethereum White Paper: A Next-Generation Smart Contract and Decentralized Application Platform. 2014. Available online: https://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf (accessed on 1 June 2021).
18. Jiang, S.; Cao, J.; Wu, H.; Yang, Y.; Ma, M.; He, J. Blochie: A blockchain-based platform for healthcare information exchange. In Proceedings of the 2018 IEEE International Conference on Smart Computing (Smartcomp), Taormina, Italy, 18–20 June 2018; IEEE: New York, NY, USA, 2018; pp. 49–56.
19. Jiang, S.; Cao, J.; Wu, H.; Yang, Y. Fairness-based Packing of Industrial IoT Data in Permissioned Blockchains. *IEEE Trans. Ind. Inform.* **2020**, 1–11. [CrossRef]
20. Wu, H.; Cao, J.; Yang, Y.; Tung, C.L.; Jiang, S.; Tang, B.; Liu, Y.; Wang, X.; Deng, Y. Data management in supply chain using blockchain: Challenges and a case study. In Proceedings of the 28th International Conference on Computer Communication and Networks (ICCCN), Valencia, Spain, 29 July–1 August 2019; IEEE: New York, NY, USA, 2019; pp. 1–8.
21. Khan, S.N.; Loukil, F.; Ghedira-Guegan, C.; Benkhelifa, E.; Bani-Hani, A. Blockchain smart contracts: Applications, challenges, and future trends. In *Peer-to-Peer Networking and Applications*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 1–25.
22. Ethereum Team. Ethereum API—ProvableAPI 0.5 Smart Contract. 2019. Available online: https://github.com/provable-things/ethereum-api/blob/master/provableAPI_0.5.sol (accessed on 1 April 2021).
23. Owyang, J.; Tran, C.; Silva, C. *The Collaborative Economy*; Altimeter: San Mateo, CA, USA, 2013.
24. Solidity Team. Solidity Language. 2014. Available online: https://solidity.readthedocs.io/en/develop/ (accessed on 1 April 2021).
25. Ganache Team. Ganache: Personal Blockchain for Ethereum Development. 2016. Available online: https://www.trufflesuite.com/ganache (accessed on 1 April 2021).
26. Truffle Team. Truffle: Ethereum Development Framework. 2016. Available online: https://github.com/trufflesuite/truffle (accessed on 1 April 2021).
27. Web3. Web3.js—Ethereum JavaScript API. 2017. Available online: https://github.com/ethereum/web3.js/ (accessed on 15 May 2021).
28. Etherisc. Introduction to Risk Pool Tokens on the Etherisc Protocol. 2018. Available online: https://blog.etherisc.com/introduction-to-risk-pool-tokens-on-the-etherisc-protocol-1744de67a57e (accessed on 26 May 2021).
29. Etherisc. Etherisc—Decentralized Insurance Protocol to Collectively Build Insurance Products. 2017. Available online: https://etherisc.com/ (accessed on 26 May 2021).
30. Sassano A. Creating a Symbiotic Relationship. 2020. Available online: https://thedailygwei.substack.com/p/creating-a-symbiotic-relationship (accessed on 15 May 2021).
31. Buterin, V. Ethereum Improvement Proposal (EIP) 1559. 2021. Available online: https://github.com/ethereum/EIPs/blob/master/EIPS/eip-1559.md (accessed on 15 May 2021).