International Journal of Innovation and Applied Studies
ISSN 2028-9324 Vol. 41 No. 3 Jan. 2024, pp. 717-728
© 2024 Innovative Space of Scientific Research Journals
http://www.ijias.issr-journals.org/

Blockchain for Peer-to-Peer Insurance: Design and implementation of a P2P Insurance application using smart contracts

Saad Slaoui Hasnaoui and Hafid Barka

Research Team in Economics and Management of telecommunications and information technologies,
National Institute of Posts and Telecommunications, Rabat, Morocco

ABSTRACT: This article investigates the feasibility of designing a comprehensive insurance solution without the involvement of insurers or intermediaries. The study begins by the introduction of the blockchain and smart contract concept and their transformative potential on many sectors, of which the insurance market. A novel Peer-to-Peer home insurance model using the parametric concept is presented, answering a real-world issue. To validate the concept, the model is realized on an Ethereum blockchain testbed. Results of the demonstration reveal the feasibility of the solution and the potential of this type of application to streamline insurance processes, enhance transparency, and reduce reliance on intermediaries. This article contributes to the ongoing evolution of the insurance by examining the paradigm shift from centralized to decentralized models. And by addressing the ability of blockchain and smart contracts to realize this shift.

KEYWORDS: blockchain, smart contract, P2P insurance, parametric insurance.

1 INTRODUCTION

Bypassing central agencies and intermediaries was always the biggest ambition of Blockchain initiators and pioneers. In fact, bitcoin itself has been created mainly to use a form of a currency without the intervention of a central bank. This leads to a new economy based on decentralized finance.

For insurance sector in particular, many applications of the decentralized economy are under development, but the biggest disruption would be a fully Peer to Peer insurance product used in large scales.

But can we really design insurance products without insurers or intermediaries? Are there any useful cases of this type of product? What are the technological choices to implement those products?

In this article and to answer the questions above, I will first explain Blockchain technology and it's potential, then introduce the concept of smart contracts. Then I will discuss the applications under development of these two technologies in the insurance sector. Among which applications of P2P insurance, and parametric insurance.

Afterwards, I will propose a P2P insurance model for a home insurance without a central organization (insurer). The demonstration of the model will be run on the Ethereum testbed. Finally, I will discuss the results of this demonstration, and the prospects and limitations of this type of insurance.

2 THEORETICAL FOUNDATIONS: STATE OF KNOWLEDGE

2.1 BLOCKCHAIN: A REVOLUTIONARY TECHNOLOGY

Blockchain is a relatively recent technology, developed in the last decade, yet it is so promising that its transformative potential has been compared to that of the Internet in the 90s-2000s [1].

Blockchain can simply be defined as a distributed ledger that enables several distinct entities connected to a network (e.g., the Internet), to exchange value in a secure way. The best-known example of this exchange is crypto-currency (bitcoin is the most widespread of all).

Major characteristics that clearly distinguish this technology from other existing distributed connection protocols are [2]:

- Trust is ensured by the protocol and robust encryption technologies, and not by a third-party entity (i.e. bank, notary, state entity...).
- Transfer of Value: Until now, Internet technology has enabled the free exchange of information: files, music, audio... Using Internet Applications (email, ftp...), only the copy of the digital content is distributed, not the original. On the other hand, the blockchain enables the transfer of items holding value such as money, votes, property titles...
- Immutability and Transparency: Once a transaction is recorded on the blockchain, it becomes impossible to alter it or tamper it. Each new block in the chain contains a cryptographic hash of the previous block, creating a permanent and transparent record of all transactions. These characteristic increases trust and enable audibility, making the blockchain ideal for applications like supply chain management, voting systems, and financial transactions.
- Security: Blockchain utilizes advanced cryptographic algorithms to secure transactions and data. Transactions are validated and added to the blockchain through consensus mechanisms such as Proof of Work (PoW) or Proof of Stake (PoS), which require significant computational power or stakeholder participation. This makes it extremely difficult for malicious actors to manipulate or corrupt the blockchain.
- Potential for Disruptive Applications: The revolutionary nature of blockchain lies in its potential to disrupt various industries. By providing a secure and transparent platform for trusted transactions, blockchain can transform industries such as finance, healthcare, supply chain, real estate, voting, intellectual property, and more. It enables new business models, fosters innovation, and empowers individuals by granting them greater control over their data and assets.

While blockchain technology is still evolving, its revolutionary potential stems from its ability to redefine trust, security, and decentralization in a wide range of applications. However, it’s important to note that the technology is not without limitations and challenges [2], including scalability, energy consumption, regulatory considerations, and integration with existing systems.

2.2 Smart contracts

Among the biggest applications of blockchain technology, the smart contracts have a special place especially in some industries and sectors.

The term "Smart Contracts" was first used in 1994 by Nick Szabo, a computer scientist specializing in cryptography. He described this concept as follows: "A set of promises, specified in digital form, including protocols within which the parties perform on these promises [3]. Aaron Wright and Primavera de Filippi provide in 2015 the following definition "digital, computable contracts where the performance and enforcement of contractual conditions occur automatically, without the need for human intervention" [4].

The vending machine illustrates for Nick Szabo the first practical example of an intelligent contract. Indeed, the machine executes a very simple contract: when the user inserts a coin corresponding to the price of a soda, for example, the machine fulfills its part of the contract and dispenses the requested product. Conversely, if the consumer introduces less money than requested, the machine does not dispense the product and returns the coin.

Apart from articles written by Nick Szabo and a few other scientists in the 1990s, the interest in this concept has been very low for a long period, mainly due to the absence of a technology to support its implementation. The advent of bitcoin in 2008 [5], and mainly the underlying blockchain technology, revived the interest in this concept. Few years after the bitcoin, a specific blockchain platform customized for smart contracts have been developed by Vitalik Butterin: The Ethereum [6].

In parallel, several ecosystems and consortia have been formed to develop "Smart Contracts", often around a particular industry: for example, the consortium (Etherisc, n. d.) for insurance. These consortia bring together manufacturers, developers, and legal experts, who work together to develop and promote practical development cases.

2.3 P2P Insurance: or collaborative insurance

In Peer-to-Peer insurance (or social/collaborative insurance), a group of individuals or homogeneous entities agree to share a common risk. Everyone in the group is both insured and insurer. The use of the Blockchain ensures trust between the parties, and smart contracts ensure the management and execution of the terms and agreements between the parties. The majority of the startups that penetrated this market use most often a hybrid model that applies the principles of P2P but still uses traditional insurers, with the following model:

- Each member pays a premium, a percentage of the pot (often over 50%) is reserved for settling incurred claims that are approved by the majority of members (automatic approval using the consensus mechanisms of blockchain).
- The remaining percentage of premiums in the pot is allocated to subscribe a policy with a classic insurer to cover the biggest risks (like reinsurance concept).

The compensation of the startup is thus made up of brokerage commissions generated by the placement of these complementary insurances, in other models the compensation may also include a small percentage of the initial premiums paid by the members.

At the end of each period, these startups generally distribute the remaining profits in the pot to the contributing members (this is why it’s sometimes called a social insurance).

Several startups have started this type of insurance since Friend assurance in Germany, which is considered by many as a pioneer in this field. We can mention for example lemonade in the United States or Teambrella (the first P2P insurance using bitcoin).

2.4 Parametric insurance

In the context of a parametric insurance, the coverage is based on the use of one or more indices that are correlated with the damages suffered by the clients. For example, A farmer may experience loss of income from a certain level of drought, he is then compensated directly when the level of rain falls below a certain threshold, independently from the real damages that he experienced.

During the execution of the contract, advanced technologies, such as connected objects (IoT), satellite imagery, radar or sonar, are used to monitor and measure the contract parameters/indices. Once the threshold predefined in the contract is triggered, the compensation intervenes systematically. The amount of the compensation is fixed and known in advance by both parties.

From this definition, it appears clearly that the determining element of the success of this coverage model is the parameter used for each case. These indices must validate several criteria to be retained, mainly the ease of measurement and the reliability of the data collected. And it is precisely for these reasons that it is best to use external data providers or trusted third parties called "ORACLES".

The advantages of this solution are obvious, starting with:

- The simplicity of use and thanks to new technologies such as connected objects for measurement and smart contracts for automation of the process.
- Expert firms will no longer have to carry out missions within the framework of this insurance model, because compensation does not depend on the evaluation of losses. This will result in cost savings and faster claims handling.
- Limiting fraud: with this model, the possibilities of classic fraud are virtually eliminated since it would be useless to falsify or inflate a loss...

On the other side, the main drawback is the accuracy of the correlation between the used threshold and the actual losses of the insureds. In some cases, an insured who realizes a huge operating loss cannot claim compensation because the index was not triggered. The opposite is also possible when an insured - having suffered no loss - is compensated following the triggering of an index. Moreover, the parametric insurance can be costly to set up, especially to pay reliable data providers or to invest in proprietary solutions to acquire data that is unavailable from providers.

Until now, this insurance model has been closely linked to climatic events, for obvious reasons. Indeed, the sources of meteorological data are available, reliable, and abundant enough to be used in this type of insurance. On the other hand, the sector of activity that depends the most on climatic variations is agriculture. It is therefore normal that most solutions currently developed around this concept cover first the damages caused by the weather on the primary sector of the economy. Among global insurers, we can mention the following platforms: POP STORM from Swiss RE, One Cat from Munich RE or AXA Climate. Startups such as FloodFlash, Arbol market or Blink parametric are also exploiting this niche. However, the opportunities of the parametric insurance go beyond the protection against the unleashing of nature. Sectors such as renewable energies, infrastructures, the health sector, tourism... are promising to create new insurance and innovative solutions.

2.5 P2P PARAMETRIC INSURANCE AND SMART CONTRACTS

In previous paragraphs, we have reviewed the definition of P2P and parametric insurance products as well as the smart contract as a technological concept. in this paragraph we will review the literature regarding the combination of the two worlds to use the smart contracts in the P2P parametric insurance development. In fact, this type of product has been identified early as a use case for smart contracts. Mainly because the latter is a self-executing program that enforces the terms of a contract.

Several studies have demonstrated the potential benefits of using smart contracts in P2P parametric insurance, [7] shows that Smart contracts can reduce the need for intermediaries, resulting in lower costs and increased efficiency. And [8] demonstrated in their thesis research that the use of this type of contracts provides greater transparency and accountability, as all parties involved can see and track the execution of the contract.

Other published studies explored the use cases of smart contracts in P2P parametric insurance. For example, a study [9] looked at the particular use case for affordable products to Indian farmers found that the use of smart contracts greatly improves the efficiency and transparency of the platform. Another study [10] investigated the use case for natural disasters in Europe and found that the platform was able to provide payouts quickly and accurately using smart contracts.

On the other hand, several challenges and limitations need to be addressed before we can see this evolution as a mainstream in the industry. One of these limitations is compliance with regulations. The insurance sector is known to be highly regulated for good reasons, and Morocco is not an exception to this fact. Smart contract may not align yet with existing laws and regulations [11]. Risk management is also high for the use of smart contracts, the DAO hack that resulted in the loss of millions of dollars is a good example of those risks [12]. Finally customer adoption needs to be taken seriously, as customer education and trust in the technology are not there yet [13]

According to the review above, we can conclude that blockchain and smart contracts can potentially disrupt the industry and facilitate the rise of the P2P parametric new products in the market. Thanks to the several benefits listed earlier. However, further studies need to be carried to test the use cases of this technology in real-world applications and address also the challenges associated with it.

3 EXPERIMENTAL STUDY AND RESULTS

3.1 THE PROPOSED APPLICATION: A P2P COLLABORATIVE HOME INSURANCE

The experiment used in this study tries to answer a real-world issue. let's take a touristic village in the north of Morocco of a hundred houses. The owners of these houses are from other cities in Morocco and use their secondary residences -either for rent or own use - only 2 months a year. During the 10 months when the house is uninhabited, maintenance is usually very poor and the owners face several problems, one of them is the damage due to water leaks in their houses. The consequences of this problem are often an excessive energy bill, damage to the furniture and installations, and sometimes loss of expected revenues during the peak season.

To overcome this issue, of course they can subscribe to home insurance with classic insurers, but the problem is that the latter consider secondary residence as higher risk for the reasons explained earlier and thus ask for high premiums or simply decline the coverage. In the case of acceptance of this risk by the classic insurer, the settlement of claims is subject to several conditions that are hard to satisfy by the insured. The result is that many of them are not covered by any insurance and face possible damages with their own means.

To avoid those constraints, I propose a collaborative Peer to Peer (P2P) insurance without insurers, managed in a completely digitalized way using blockchain and smart contracts. The simplified model for this application is shown below:

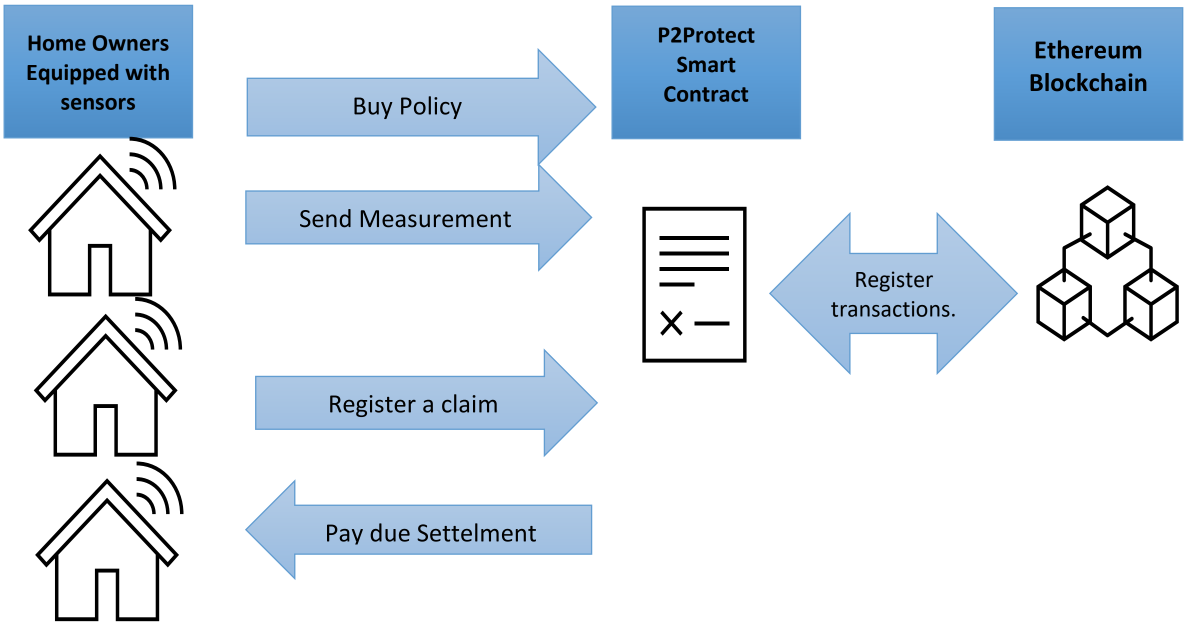


Fig. 1. High Level Model for P2Protect application

Each insured/Homeowner is equipped with two smart sensors, the first one installed within the Water company counter in each house to determine if the water consumption is excessive compared to normal, the second is installed in the floor to detect the presence of a certain amount of water. The measurements from those two sensors are used as criteria to determine the existence of leakage of water at the inhabitant's house. And therefore, the eligibility to the settlement.

The compensation is in the form of a flat rate according to the severity of the leak. These levels are reviewed each period for adjustments to the actual damage.

In order to implement this type of contracts the following steps must be conducted:

- Analysis and Identification of contractual elements: Stakeholders identify the opportunities for cooperation between them and the expected outcomes of that cooperation. For the case of home insurance, it is necessary to identify the value to be insured, the process of claims assessment and validation, the process of money transfer, and the distribution of premium excess at the end of contract.
- Determining the conditions of execution: smart contracts are initiated by the realization of certain conditions: for example, the underwriting of a new membership/policy, claim declaration...
- Programming the contract logic: A computer program is written in such a way that the arrangement is executed automatically when the conditional parameters are met.

After encryption, the contract is registered in the blockchain so that it can be executed. Once this step is done, all participants of the concerned blockchain network (network computers) update their registers to reflect this execution. No changes are possible after that.

3.2 Technical choice for the development of the solution

As for any Web 3.0 development in general, before coding our solution, it is particularly important to make several technological choices that will be crucial in the future for the evolution or modification of the solution. The most important selections that we should perform are: the blockchain platform, own Node or Node provider, The wallet for crypto money, the coding language for both the back end and for the front-end application, and finally the environment of the development.

3.2.1 The Blockchain Platform: Ethereum vs Hyperledger

Both Ethereum and Hyperledger are blockchain platforms that are widely used to develop smart contracts and Dapps (decentralized applications). For our application we have chosen Ethereum mainly because it's a public blockchain platform that is designed to be decentralized and open to all. While Hyperledger, is a private or permissioned blockchain platform, where only approved participants can access the network and participate in transactions [14].

The Ethereum also offers a possibility to deploy real solutions and applications in Blockchain testbed that has the same functionalities except that the used token has no real value. The P2Protect is deployed on the GOERLI Ethereum https://goerli.etherscan.io/.

3.2.2 Own Node vs Node Provider

Once the blockchain platform is chosen, we need to decide whether to deploy our own node or use an external provider. For the first option, we need to install the EVM (Ethereum Virtual machine) in our computer and run a normal Ethereum node on the network. This gives us full control over the node and the data it processes, making it more secure and trustworthy. The second option is to host the node in a provider (similar to hosting a website in an external server in Web 1.0 or Web 2.0 development).

According the official Ethereum documentation (Nodes as a service, ethereum.org), using the second option while it's not giving full control over the data, it gives the advantage to focus on building your product rather than managing the nodes and worrying about the maintenance of the infrastructure.

For this reason, we have chosen to use one service provider called Alchemy (Alchemy, s.d.) for its reliability and functionalities. Of course, for an industry grade application or to scale the application, this choice can be reviewed according to new needs [16]

3.2.3 Web3 Wallet

Although it's possible, to interact with the blockchain without a wallet, but having one is the more common and convenient, as it provides users with greater control, security, and privacy for their digital assets [17]. We have chosen for this test, the most popular one: MetaMask.

3.2.4 Back-End programming language

the backend application code for a smart contract on Ethereum is written in Solidity, [18] which is similar to JavaScript, and allows developers to specify the rules and conditions under which the contract executes. The code defines the variables, functions, and events that make up the contract's behavior, as well as the conditions under which the contract can be initiated, modified, or terminated. Any other scripts needed by the applications but not related to Ethereum blockchain, can be written simply with JavaScript.

3.2.5 Ethereum development environment

Before choosing a smart contract development environment, it is crucial to understand the specific needs of the developers. For beginners looking to write smart contracts in Solidity, Remix -an Ethereum-compatible IDE (integrated development environment) - is a suitable option for compiling, and debugging the code [19] However, as smart contract complexity grows, it is necessary to use an efficient testing and deployment tool.

Remix IDE has limitations such as the need to manually copy the ABI (Application Binary Interface) code of smart contracts, which is an interface between two program modules [18]. In contrast, the development environment such as Hardhat and Truffle automate this process and offer a more flexible and configurable development environment. For this application, we chose Hardhat due to its range of plugins and features that can improve development efficiency [20].

Hardhat allows for easy setup, compilation, testing, and deployment of smart contracts while also providing built-in support for other popular Ethereum testing networks. Furthermore, it integrates seamlessly with other development tools such as React, which we used for our front-end applications.

3.2.6 Front end Development

The development of this section is identical to Web2 development, and thus, the interface used to interact with smart contract would be similar in any point to any other interface used to interact with centralized systems (i.e., a classic insurer website for example). The simple interfaces developed in these applications used the popular front-end libraries: REACT.Js.

3.3 Description of the application

3.3.1 Front-End

It's the client interface, where the members will subscribe to the insurance products as well as getting their claims the compensation. Three simple pages have been developed on the client side:

A home page: this page contains three parts:

- A link to the subscription page, for new members or renewals.
- A link to claim a loss and receive the indemnity.
- A link to the explanation of the P2Protect service.

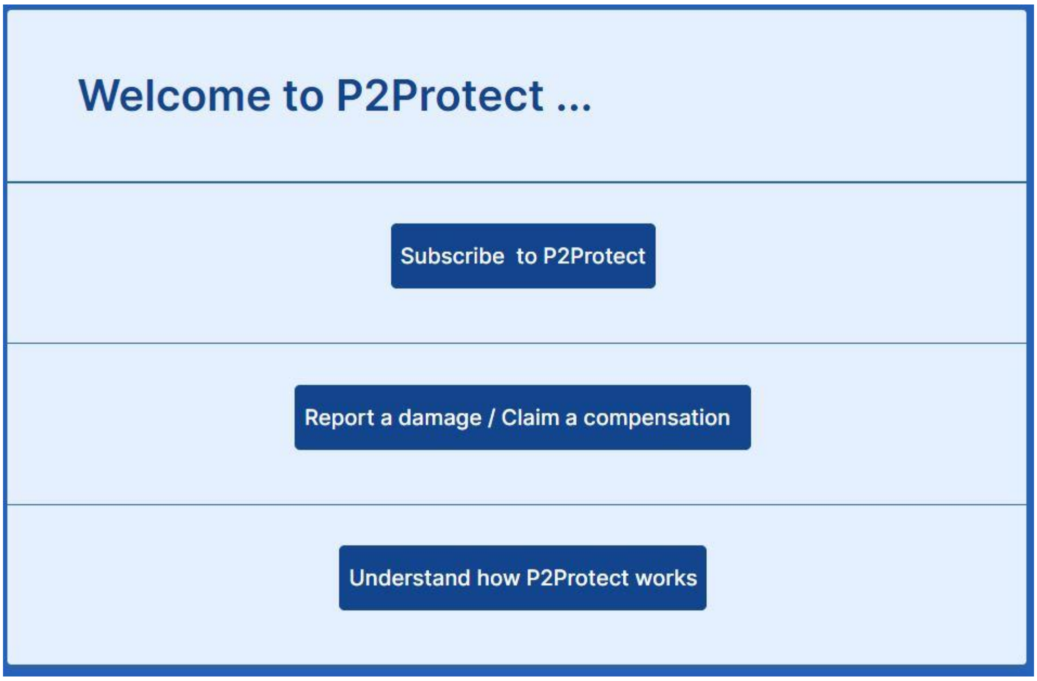

Welcome to P2Protect ...
Subscribe to P2Protect
Report a damage / Claim a compensation
Understand how P2Protect works

Fig. 2. Home page of the P2Protect application

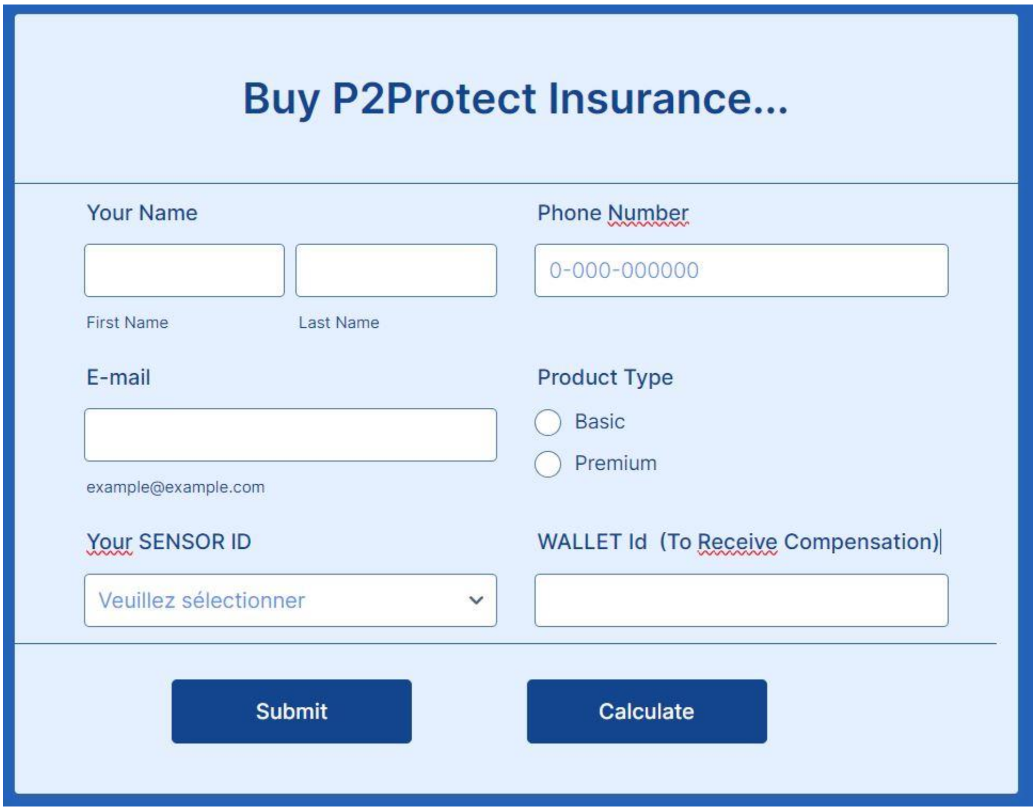
A subscription page: the member enters in this page the necessary elements for the contract, namely:

- Name and first name of the member.
- The telephone number and the email address for communications with the insured.
- The choice of the level of coverage: basic or premium. For test purposes, only two choices have been proposed, but we can normally have several options, each level gives the right to a compensation package in case of claim. Obviously, the premium to be paid depends on the chosen level.
- Sensor number: very important information because it identifies the sensor that will provide the daily measurements to the smart contract. The application is fed with all the sensor numbers of the residents, the new member will have to choose from a list.
- And finally, the portfolio ID of the member: this important information will be used during the subscription for the transfer of money from the customer account to the smart contract account (Pool Account), it will also be used to pay the indemnity in case of a verified claim.

Once the client fills in the form, two action buttons are proposed:

- Calculate: The result is the amount of the premium to be paid according to the choices of the customer.
- Subscribe: once the customer clicks on this button, he buys the insurance, and receives an email containing the insurance certificate with all the details of the contract and the validity period.

The smart contract deployed on the blockchain is then updated with the new insurance and money is transferred from the member's account to the wallet of the P2Protect Contract.



Buy P2Protect Insurance...

Your Name

First Name

Last Name

Phone Number

0-000-000000

E-mail

example@example.com

Product Type

○ Basic

○ Premium

Your SENSOR ID

Veuillez sélectionner

WALLET Id (To Receive Compensation)

Submit

Calculate

Fig. 3. P2Protect Subscription Page

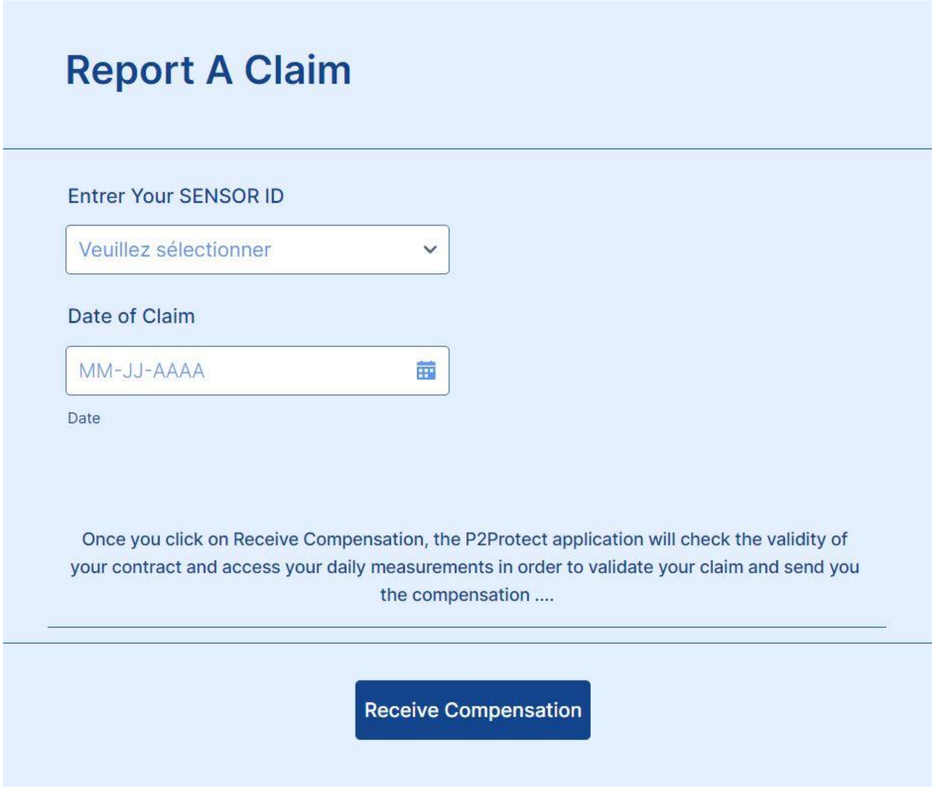
A claim Page: in this form the customer must provide only two pieces of information:

- The number of the sensor
- The date of the claim

Once the customer submits this form, the application will use the provided information to verify the validity of the claim according to the contractual elements. To make this verification, the application check, first, the existence of a valid contract that covers the date of the claim, and then use the measurements – received on daily basis – by all the sensors in the residence and compare the measurement of the claim date with the agreed threshold. The chosen trigger for the compensation is the following:

$$\text{Consumption of the date of the disaster} > 5 * \text{Average consumption of the 5 days before the date of the disaster}$$

If this condition is met, the compensation is paid immediately by the program to the address of the client portfolio. The insured is required to report any loss or damage to the insurance company to receive compensation. This means that compensation will not be automatically provided if the loss threshold is reached without the insured making a claim.



Report A Claim

Entrer Your SENSOR ID

Veuillez sélectionner

Date of Claim

MM-JJ-AAAA

Date

Once you click on Receive Compensation, the P2Protect application will check the validity of your contract and access your daily measurements in order to validate your claim and send you the compensation ....

Fig. 4. P2Protect Claim Page

3.3.2 Back-End

The backend consists of the actual contract deployed on the ETHEREUM blockchain and written in Solidity language.

The contract is named "P2PROTECT," and contains a set of functions and events needed to execute the requests received from the front end. The main function is the one used to buy a policy by providing the needed information like the user’s address on the Ethereum. The other important one allows the users to make claims and receive reimbursement in case of water leakage or unexpected consumption of a Water.

The contract emits events for policy activation, Claim Paid. The latter can be used to trigger other actions outside the scope of this program or notify external entities.

Below the description of the function included the contract:

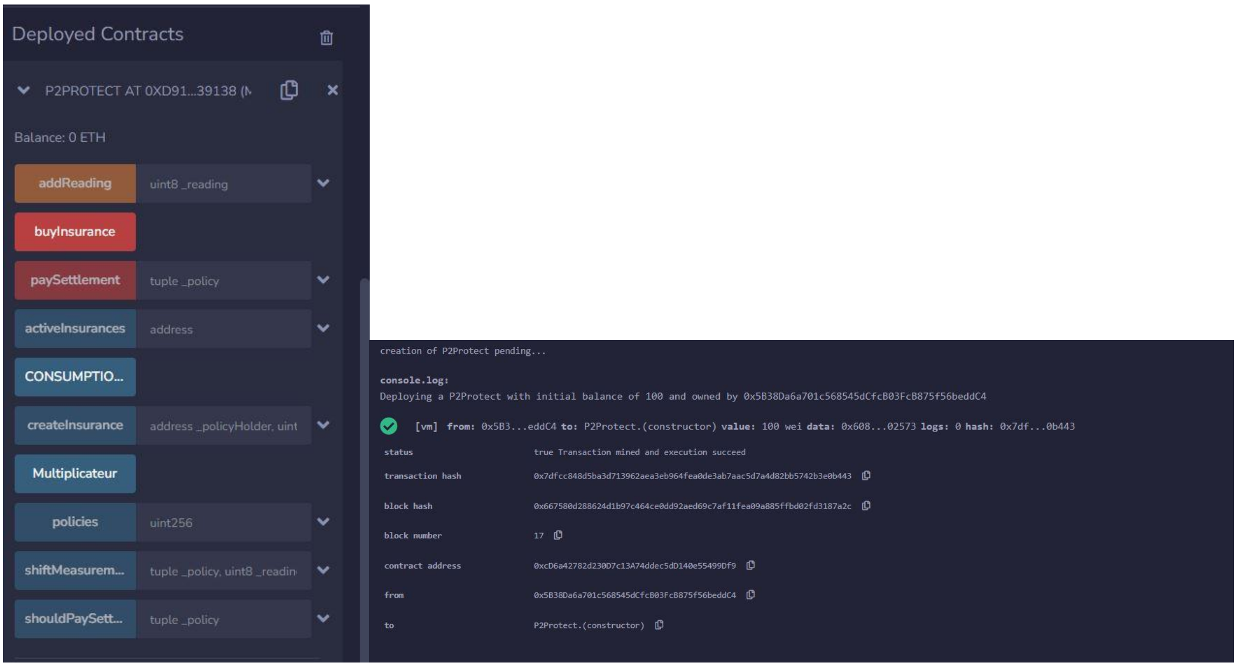
- *createInsurance ():* This function creates a new insurance policy for the policy holder with the provided policy premium and 5 days of consumption readings. The policy holder address and policy premium are stored in a Policy struct.
- *buyInsurance ():* This function allows a policy holder to buy a new insurance policy by calling createInsurance and updating the activeInsurances mapping to mark the policy holder as having an active policy. The policy holder must have sufficient funds to pay for the policy.
- *addReading ():* This function allows adding a new consumption reading. It calls shiftMeasurement on all policies to shift the readings and add the new reading to the end of the list.
- *shiftMeasurement ():* This function shifts the 5 days of consumption readings for a policy and adds the new reading to the end of the list.
- *shouldPaySClaim ():* This function checks if the policy holder is eligible for a settlement payment based on the latest consumption reading. A policy holder is eligible for a settlement if the latest consumption reading is greater than or equal to the CONSUMPTION_THRESHOLD and LEAKAGE= True.
- *payClaim ():* This function pays the settlement to the policy holder if the policy holder is eligible for a settlement, based on the result of shouldPayClaim. The settlement amount is calculated as the policy premium multiplied by the greed Multiplier (10 in this case).
- *ClaimPaid ():* This is an event that is triggered when a settlement payment is made to a policy holder.

In addition, there are other Node.js scripts necessary to operate this type of application.

3.4 DISCUSSION OF THE RESULTS

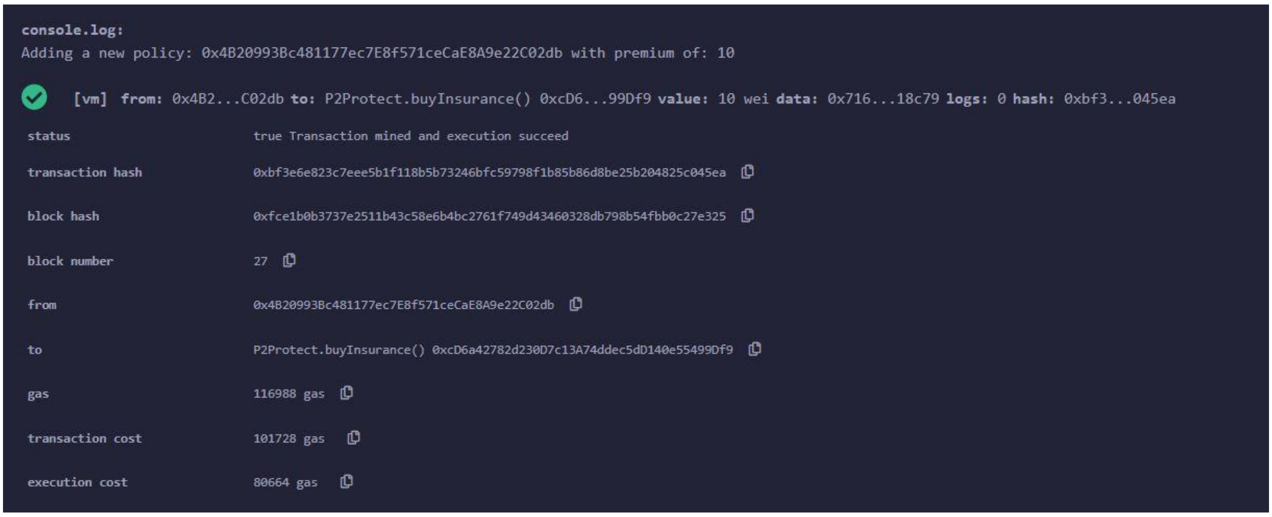
In this section we demonstrate the functionalities of P2Protect using the Remix platform. First, we compiled the code using the solidity compiler, and corrected the errors and warning that it contains. Next, we deployed the smart contract to the Ethereum blockchain (the goerli Ethereum testbed). Once the contract is deployed, we were able to interact with it using Remix user interface. We tested then all possible scenarios and functionalities, such as creating a new insurance policy, adding a new measurement, submitting a claim... We were able to visualize the contract state variable as well as the transactions performed during the testing process, using the debugger offered by Remix. Allowing us to verify that the P2Protect smart contract is working as intended in term of policy subscription and claim settlement.

The snapshots below show some of the key steps involved in the deployment process.



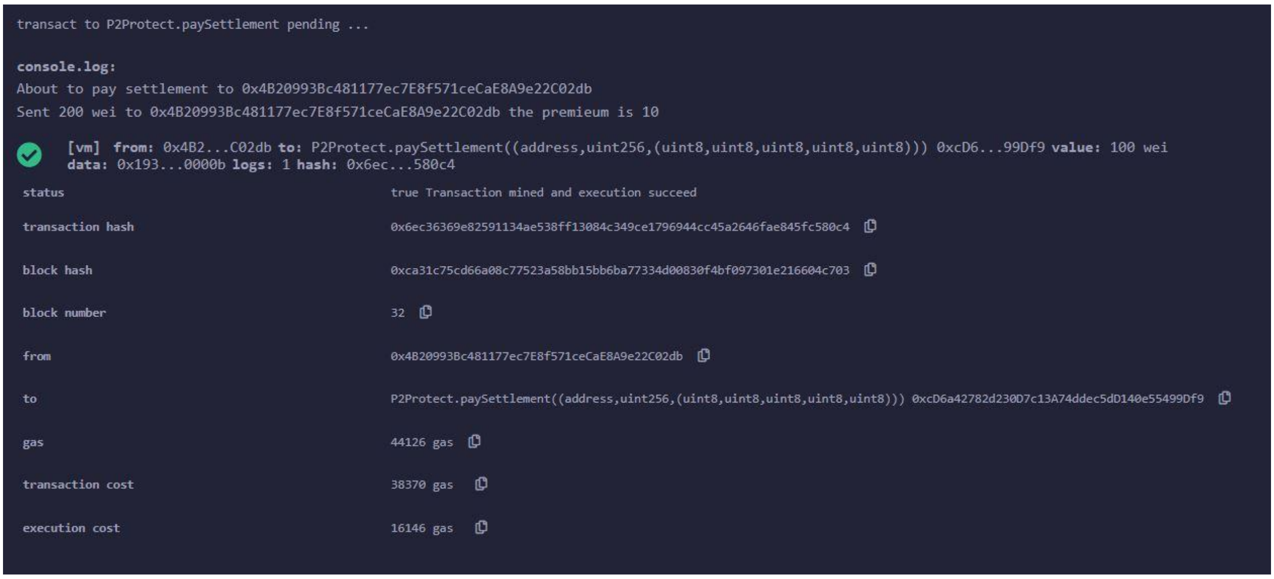
```
creation of P2Protect pending...

console.log:
Deploying a P2Protect with initial balance of 100 and owned by 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

[vm]  from: 0x5B3...eddC4 to: P2Protect.(constructor) value: 100 wei data: 0x608...02573 logs: 0 hash: 0x7df...0b443
 status                   true Transaction mined and execution succeed
 transaction hash         0x7dfcc848d5ba3d713962aea3eb964fea0de3ab7aac5d7a4d82bb5742b3e0b443

 block hash               0x667580d288624d1b97c464ce0dd92aed69c7af11fea09a885ffbd02fd3187a2c

 block number             17

 contract address         0xcD6a42782d230D7c13A74ddec5dD140e55499Df9

 from                     0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

 to                       P2Protect.(constructor)
```

Fig. 5. Snapshot of the P2Protect contract deployed on Ethereum



```
console.log:
Adding a new policy: 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db with premium of: 10

[vm]  from: 0x4B2...C02db to: P2Protect.buyInsurance() 0xcD6...99Df9 value: 10 wei data: 0x716...18c79 logs: 0 hash: 0xbf3...045ea
 status                   true Transaction mined and execution succeed
 transaction hash         0xbf3e6e823c7eee5b1f118b5b73246bfc59798f1b85b86d8be25b204825c045ea

 block hash               0xfce1b0b3737e2511b43c58e6b4bc2761f749d43460328db798b54fbb0c27e325

 block number             27

 from                     0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db

 to                       P2Protect.buyInsurance() 0xcD6a42782d230D7c13A74ddec5dD140e55499Df9

 gas                      116988 gas

 transaction cost         101728 gas

 execution cost           80664 gas
```

Fig. 6. Snapshot of a new insurance policy purchased



```
transact to P2Protect.paySettlement pending ...

console.log:
About to pay settlement to 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db
Sent 200 wei to 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db the premieum is 10

   [vm]  from: 0x4B2...C02db to: P2Protect.paySettlement((address,uint256,(uint8,uint8,uint8,uint8,uint8))) 0xcD6...99Df9 value: 100 wei
   data: 0x193...0000b logs: 1 hash: 0x6ec...580c4
 status                        true Transaction mined and execution succeed

 transaction hash              0x6ec36369e82591134ae538ff13084c349ce1796944cc45a2646fae845fc580c4

 block hash                    0xca31c75cd66a08c77523a58bb15bb6ba77334d00830f4bf097301e216604c703

 block number                  32

 from                          0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db

 to                            P2Protect.paySettlement((address,uint256,(uint8,uint8,uint8,uint8,uint8))) 0xcD6a42782d230D7c13A74ddec5dD140e55499Df9

 gas                           44126 gas

 transaction cost              38370 gas

 execution cost                16146 gas
```

Fig. 7. Snapshot of a successful claim payment according to contract parameters

Finally, we tested the interoperability of P2Protect smart contract with the external wallet (our test MetaMask wallet in this case). We confirmed that it can be used in conjunction with these external tools for secure and transparent transactions.

The result of these tests demonstrate that the proposed solution is feasible, we were able to automate the process for both subscriptions and claim processing, without the need of intermediaries. Automation leads to fast and cost-efficient process. The objective of this study which is the design and implementation of a real use case of a P2P home insurance using the parametric concept and the smart contract technology was met.

4 Conclusion

In This study, we have exposed the revolutionary blockchain technology, the concept and promises of the smart contracts, and their application in the insurance sector, as exemplified by the design and implementation of a Peer-to-Peer parametric home insurance application. By reviewing the literature on the intersection between the blockchain smart contract technology and the niche market of P2P parametric insurance, we have set the stage for our solution. The latter was derived from a real need in the touristic villages in the north of Morocco.

The main part of the research is dedicated to experimental study, which details all the steps involved in the implementation of the solution and its complexities. Within the implementation, we have given a special attention to the technical choices and tools used for the development. these choices have a big influence on the final application in the context of a Web3 Blockchain application. The backend part of the application requires deep knowledge of the blockchain technology and smart contracts, while the front-end provides the user interface to all types of applications (centralized or not). For this reason, only the backend was explained in detail in this article.

In conclusion, we have demonstrated in this study through the proposed P2P home insurance solution that a completely decentralized insurance product is feasible and can be a promising alternative in the future to traditional insurance models for real customers issues. Our findings suggest also that this system can provide homeowners with a coverage at reduced cost and more efficient processing of claims. The scope of this proposal included only the water leakage and resulting damage; however, it can be extended to other standard home damages that we find in the classic insurance contracts: electrical damage, fire, theft...

On the other hand, the study has limitations related mainly to real world testing with real customers to fully determine the effectiveness of the P2P insurance solution. User acceptance, and customer adoption of this type of products requires exploration and further research. Furthermore, common challenges for all application in the evolving landscape of blockchain and smart contracts needs to be addressed in other studies, such as performance under extensive loads, scalability issues and regulatory hurdles.

Looking ahead, this study contributes to the beginning of a long marriage and intersection between a cutting-edge technology and old established industry – the insurance -. The journey towards this positive transformation is still long and will be fueled by ongoing research and innovation.

REFERENCES

[1] Bonelli M (2020). Analogies Between Internet and Blockchain, Hype Cycles, and Securities Trading Patterns. doi: 10.2991/aebmr.k.201128.011.

[2] Gatteschi V, Lamberti F, Demartini C, Pranteda C, Santamaría V (2018). To Blockchain or Not to Blockchain: That Is the Question. IT Prof, 20: 62–74.

[3] Szabo N (1996). Smart contracts: building blocks for digital markets. EXTROPY J Transhumanist Thought16, 18: 28.

[4] Wright A, De Filippi P (2015). Decentralized Blockchain Technology and the Rise of Lex Cryptographia. doi: 10.2139/ssrn.2580664.

[5] Nakamoto S Bitcoin: A Peer-to-Peer Electronic Cash System. 9.

[6] Buterin V A NEXT GENERATION SMART CONTRACT & DECENTRALIZED APPLICATION PLATFORM. In: ethereum.org. https://ethereum.org. Accessed 9 Nov 2023.

[7] Gatteschi V, Lamberti F, Demartini C, Pranteda C, Santamaría V (2018). Blockchain and Smart Contracts for Insurance: Is the Technology Mature Enough? Future Internet, 10: 20.

[8] Salahshor A, Scherrer J (2020). Smart Contracts, Insurtechs and the Future of Insurance. 133.

[9] Jha N, Prashar D, Khalaf OI, Alotaibi Y, Alsufyani A, Alghamdi S (2021). Blockchain Based Crop Insurance: A Decentralized Insurance System for Modernization of Indian Farmers. Sustainability, 13: 8921.

[10] Schwarze R, Sushchenko O (2022). Climate Insurance for Agriculture in Europe: On the Merits of Smart Contracts and Distributed Ledger Technologies. J Risk Financ Manag, 15: 211.

[11] Ferreira A (2021). Regulating smart contracts: Legal revolution or simply evolution? Telecommun Policy, 45: 102081.

[12] Morrison R, Mazey NCHL, Wingreen SC (2020). The DAO Controversy: The Case for a New Species of Corporate Governance? Front Blockchain, 3: 25.

[13] Al Khalil F, Butler T, O'Brien L, Ceci M (2017). Trust in Smart Contracts is a Process, As Well. In: Brenner M, Rohloff K, Bonneau J, Miller A, Ryan PYA, Teague V, et al., editors Financ. Cryptogr. Data Secur. Cham: Springer International Publishing, 510–519.

[14] Xu X, Weber I, Staples M Architecture for Blockchain Applications. doi: 10.1007/978-3-030-03035-3.

[15] Nodes as a service. In: ethereum.org. https://ethereum.org. Accessed 24 Aug 2023.

[16] The 12 Best Blockchain Node Providers in Web3 (2023). https://www.alchemy.com/overviews/blockchain-node-providers. Accessed 24 Aug 2023.

[17] How to Choose the Right Crypto Wallet | Bitstamp. In: Learn Cent. https://www.bitstamp.net/learn/crypto-101/how-to-choose-the-right-crypto-wallet/. Accessed 24 Aug 2023.

[18] Introduction to Smart Contracts — Solidity 0.8.10 documentation. https://docs.soliditylang.org/en/v0.8.10/introduction-to-smart-contracts.html#using-remix-ide. Accessed 22 Feb 2023.

[19] Remix Docs Home. In: Remix. https://remix.run/docs/en/main. Accessed 9 Nov 2023.

[20] Hardhat Network | Ethereum development environment for professionals by Nomic Foundation. https://hardhat.org. Accessed 22 Feb 2023.