Smart Contract for Insurance: Opportunities and Challenges for Insurers

Aditya Acharya[a], Khushal Sharma[a], Soham Singhal[a]

[a]*Student, Bachelor in Technology (Artificial Intelligence)*
NMIMS Mukesh Patel School of Technology Management and Engineering Mumbai, India

---***---

Abstract - The world is rapidly moving towards digitalization and automation, and the insurance industry is no exception. While other industries have adopted blockchain technology and smart contracts, the insurance industry has been slow to adopt these new technologies. However, with the growing trend of InsurTech, insurers are increasingly turning to smart contracts as a way to streamline their operations and offer more efficient and cost-effective insurance products and services.

keywords *- Smart Contract, Blockchain, Insurance, Security.*

Introduction

Smart contracts were first proposed as a framework for digital transactions to uphold a contract's terms in the 1990s[4].Smart contracts[1] are contracts that automatically carry outthe conditions of the parties' agreement after being written directly into lines of code. They allow for the automation of digital processes, which increases their speed, efficiency, and security[3]. Blockchain[2] technology is used in smart contracts to guarantee decentralization,transparency, and immutability, which means that no single entity has authority over the system. They can execute business rules in a peer-to-peer network without the need for centralized authority since they are decentralized, auto-enforcing, and verifiable. Contracts are agreements that are enforceable by a centralized legal entity between two or more parties and have a legal effect. Contracts are agreements that are enforceable by a centralized legal entity between two or more parties and have a legal effect. Smart contracts, however, do away with the necessity for neutral third parties and middlemen between the parties. In a decentralized blockchain, they are carried out using code that is automatically checked by network modes.

Smart contracts allow for transactions between entrusted parties without the involvement of a third party or direct communication, which lowers the cost of intermediary commissions[9].

The insurance sector[5] needs smart contracts since the procedures used now are laborious and time-consuming. The manual intervention required in the standard insurance claim process might result in mistakes and delays[6]. Insurance companies may automate the entire claims process with smart contracts, from confirming the claim to making payments to the claimer.

Insurance Services : The Conventional Practice

In the traditional insurance process, the claimant reports the loss to the insurer through a broker, and the insurer verifies the claim by checking the information provided by the claimant and other sources. The loss adjuster assesses the loss, and if the claim is approved,the insurer makes a payment to the claimant. The entire process is depicted in fig1.

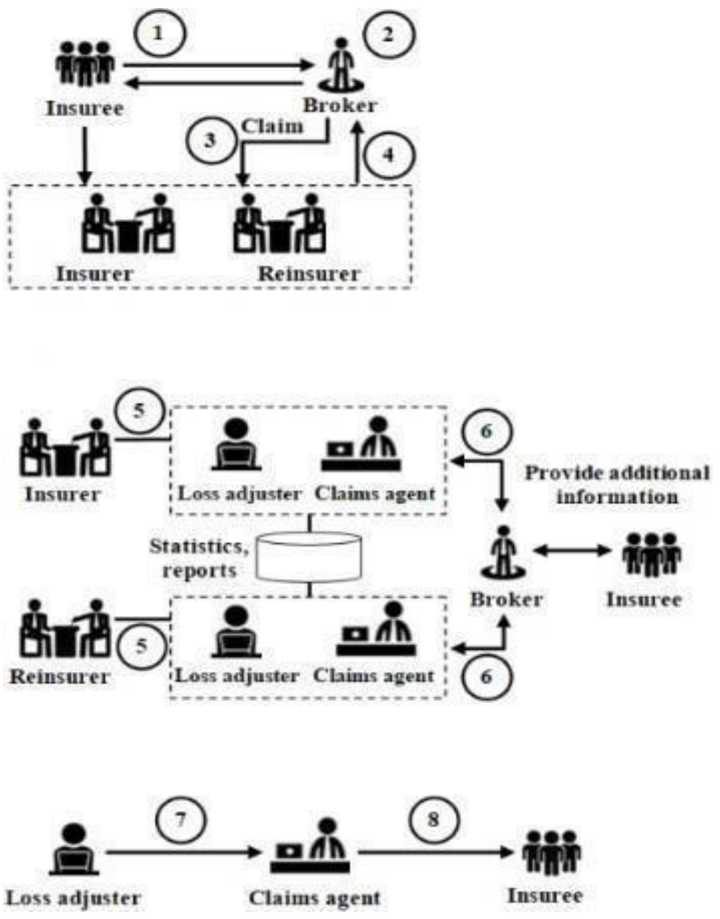


Fig 1: Traditional Insurance

Insurance Services A Blockchain Approach

Under traditional insurance, the loss is reported to the insurer by the claimant through a broker. The insurer evaluates the claim based on the information submitted by the claimadjuster and other information. If the claim is accepted, the insurer pays the claim to theclaimant.

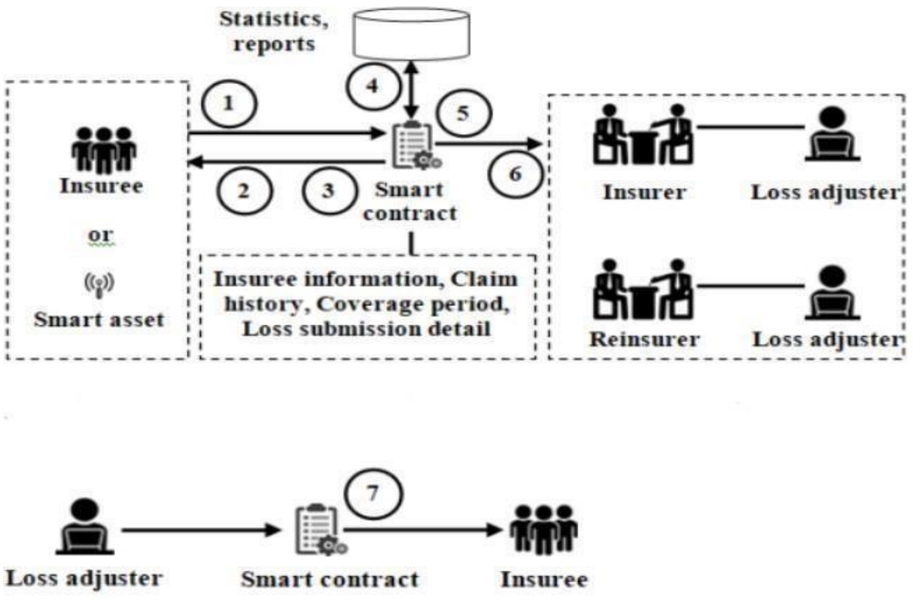
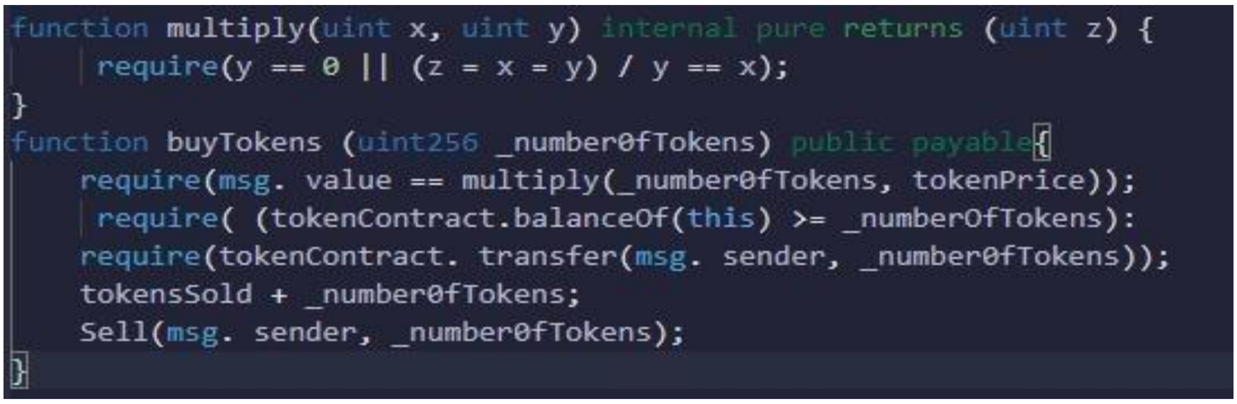


Fig 2: Smart Contract based insurance.

Methodology And Implementation

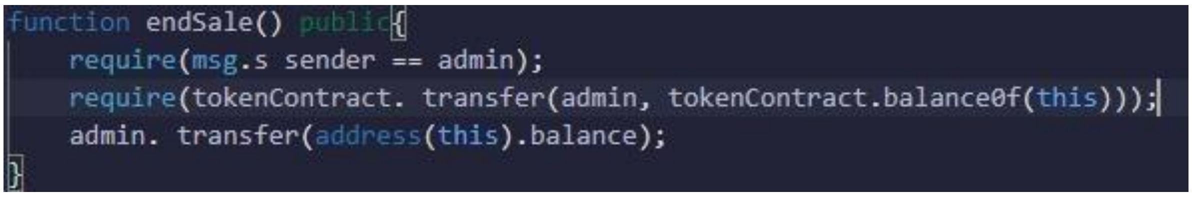
For insurers to use smart contracts in their business, they need to implement blockchain technology and smart contract platforms. One of these platforms is Ethereum. Ethereum isa free, open source, blockchain platform that allows developers to create smartcontracts. Insurers can use Metamask (a browser extension) to interact with and execute smart contracts on the Ethereum network. Metamask is also an Ethereum wallet.



```
function multiply(uint x, uint y) internal pure returns (uint z) {
     require(y == 0 || (z = x = y) / y == x);
}
function buyTokens (uint256 _number0fTokens) public payable{
    require(msg. value == multiply(_number0fTokens, tokenPrice));
     require( (tokenContract.balanceOf(this) >= _numberOfTokens):
    require(tokenContract. transfer(msg. sender, _number0fTokens));
    tokensSold + _number0fTokens;
    Sell(msg. sender, _number0fTokens);
}
```

Fig 3: Purchase of Token

Truffle is a popular system used for implementing smart contracts on the Ethereum network. Itallows for the compilation and uploading of smart contracts, including those implementing the ERC20 standard. To ensure proper functionality of the smart contract, additional functions fortoken sale, sale termination, and payment to the contract owner are required



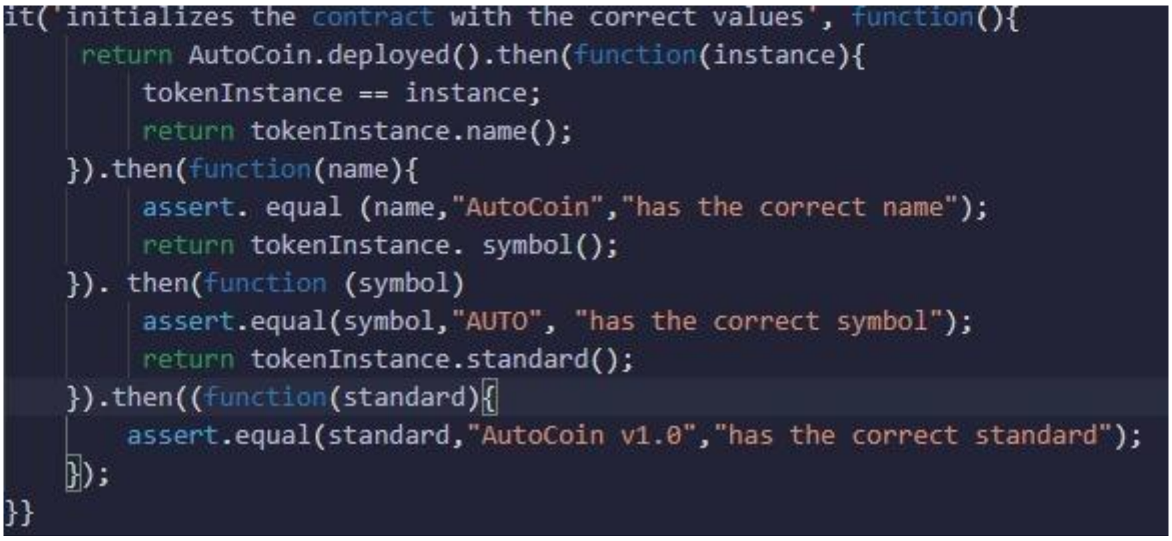
```
function endSale() public{
    require(msg.s sender == admin);
    require(tokenContract. transfer(admin, tokenContract.balance0f(this)));
    admin. transfer(address(this).balance);
}
```

Fig 4: Termination of sale

One of the most important functions is “endSale,” which only the account administrator can call. This function sends any outstanding tokens to the administrator’s account address, along with all collected Ethers. This stops any further buying of tokens after the sale ends, making sure that the remaining tokens aren’t sold and that the collected funds go to the right person. By using Truffle and performing the right functions, you can deploy smart contracts on the Ethereum network quickly and easily, ensuring a safe and efficient process for token sale andmanagement.

Results

Smart contract testing is an essential aspect of ensuring their proper functioning. Varioustools are available for automated smart contract security vulnerability testing based on code-level analysis. In this regard, four related tools, namely Oyente, Mythril, Securify, and SmartCheck, have been summarized. However, the level of rigour in testing may vary depending on the underlying security testing technique of the tool. Until now, researchershave relied on the test tools implemented in Solidity[7] and written in Remix IDE[8].



```
it('initializes the contract with the correct values', function(){
    return AutoCoin.deployed().then(function(instance){
        tokenInstance == instance;
        return tokenInstance.name();
    }).then(function(name){
        assert. equal (name,"AutoCoin","has the correct name");
        return tokenInstance. symbol();
    }). then(function (symbol)
        assert.equal(symbol,"AUTO", "has the correct symbol");
        return tokenInstance.standard();
    }).then((function(standard){
        assert.equal(standard,"AutoCoin v1.0","has the correct standard");
    });
}}
```

Fig 5: Test transfer of tokens

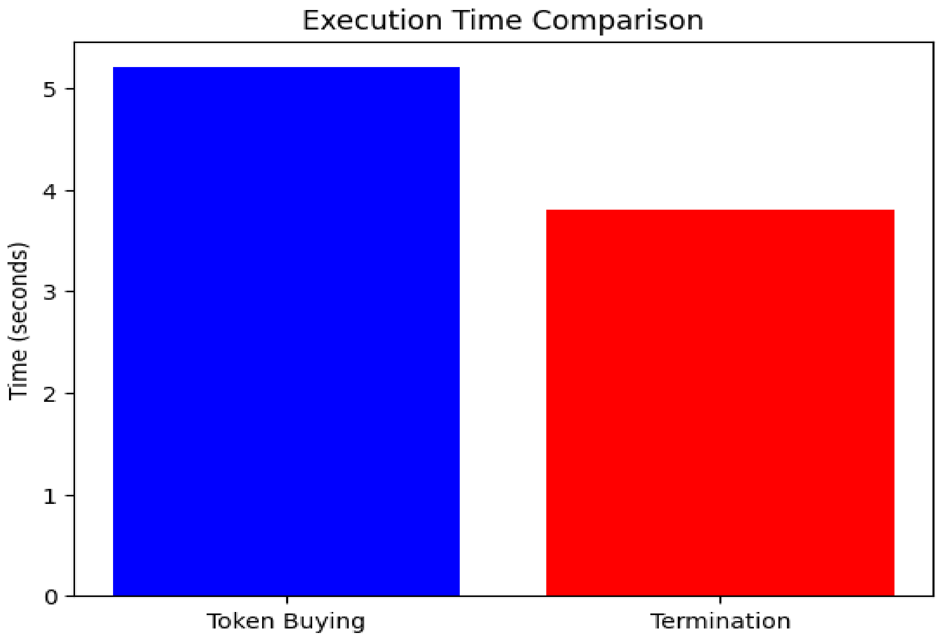
Two test codes are provided, namely token transfer test code and crypto-token disposal testcode. The former involves the direct transfer of 250,000 tokens from the administrator's address to the recipient's address. The transferred tokens are verified by capturing and checking the "Transfer" type event. The balance of the recipient's address is also checked to ensure the presence of transferred tokens.

The latter test code involves the delegated transfer of 100 tokens from the administrator's address to the address from which a delegated transfer will be allowed. Address 1 is allowed. to spend ten tokens, which are sent to address 3. After the completion of these actions, address 1 is expected to have 90 tokens, address 2 has 0 tokens, and address 3 has 10 tokens. The implementation results are shown with correct and incorrect parameters.

Graphical Results for the Performance of the Model:

Execution Time Comparison:

You can compare the execution times for both token buying and termination by creating a barchart or line chart. This chart will show the time taken for each operation, allowing one to seewhich operation is more time-consuming.

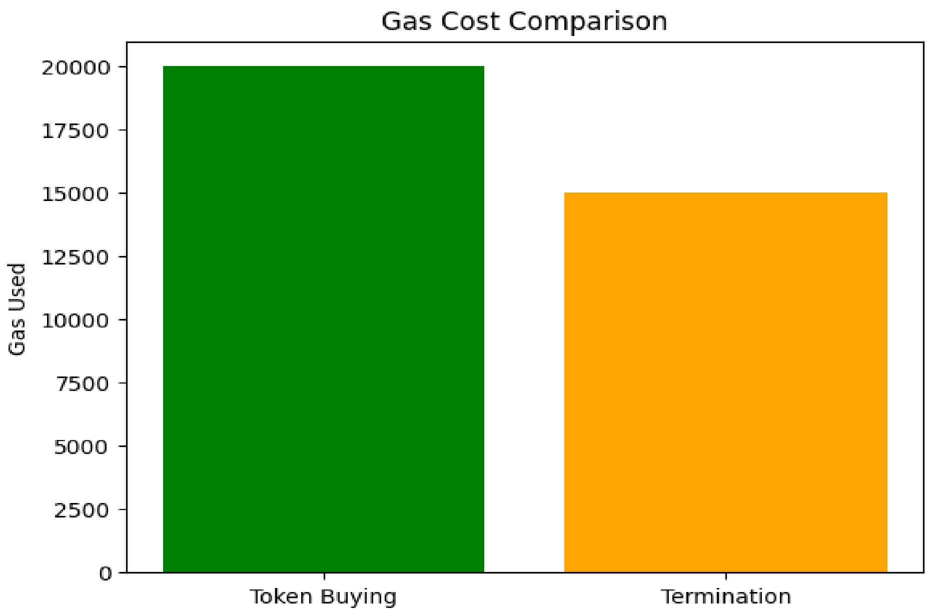


Gas Cost Comparison:

You can also compare the gas costs associated with token buying and termination by creating a similar chart, showing which operation consumes more gas.

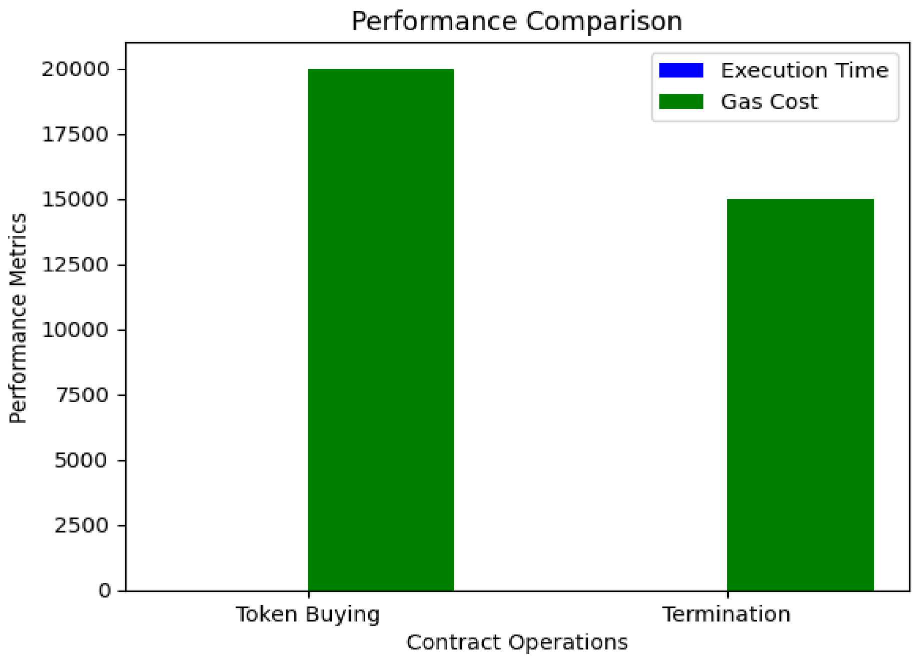
In this chart, "Token Buying" and "Termination" are represented on the x-axis, and the corresponding gas costs are shown on the y-axis. The different colors make it easy to distinguishbetween the two operations.

This chart will provide a clear visual comparison of the gas costs, helping you understand whichoperation consumes more gas.



Combined Comparison:

You can also create a combined chart to compare both execution times and gas costs simultaneously, providing a holistic view of the performance differences between the two operations used in the smart contract operations.



Future Application:

SALT Blockchain-Based Lending: How It Works, Benefits & The Bottom Line:

What is SALT blockchain-based lending? How does it work, what are the benefits and risks?What is SALT Blockchain-based Lending? SALT stands for SALT Automated Loan

Technology. It's an acronym for Secure Automated Technology. SALT lending is a platform that allows members to take out a loan with cryptocurrency as collateral. SALT was established in 2016 by a team of Bitcoin enthusiasts who wanted to provide crypto-backed lending to provide flexibility for investors holding digital assets.

SALT Lending is a blockchain-based lending platform that offers personal and business loans. Members purchase membership to the platform by purchasing SALT tokens, which are the platform's cryptocurrency. When a member joins SALT, they can borrow money from a wide network of lenders. The minimum loan amount for SALT is $1k, but members can borrow money for any purpose, including paying off credit card debt or purchasing a car. How SALT Works SALT is based on the ERC-20 standard, which is a standard that all Ethereum token contracts must adhere to allow for the exchange of tokens. A smart contract is a type of contract that not only sets out the terms of an agreement, but also enforces and executes thoseterms using cryptographic code. Borrowers are only allowed to use blockchain based cryptocurrencies as collateral, i.e., bitcoin (BTC), bitcoin cash (BCH), ether (ETH), litecoin (LCD), truUSD (USD), udex coin (USDC), paxos standard token (PAX), pax gold (PXG), salt token (SALT)

Once the loan is approved, the borrower sends the collateral to the "SALT collateral wallet."

The proceeds of the loan are then deposited into the borrower's bank account. The digital assetheld as collateral remains the borrower's property. Any price changes of the asset belong to theborrower. The borrower must make regular, regular payments to the lender throughout the life of the loan. Once the loan is repaid (known as loan completion), the borrower's collateral is available for withdrawal.

What are the benefits of SALT loans?

A SALT borrower is someone who holds their digital assets as collateral and believes that they will appreciate or remain the same value over time (known as a Holder in crypto jargon). SALT loans enable Holders to retain ownership of their digital assets and take advantage of any appreciation in the value of their digital currency, while also receiving cash for other uses.

The Bottom line: SALT blockchain lending provides investors with access to cash without the need to sell their crypto holdings. However, it is important to note that there are risks associated with this type of lending for the borrower. Cryptocurrency prices can fluctuate significantly, andif the digital asset used for collateral drops in price, the borrower may be required to pay down part of the loan or put up additional crypto assets as collateral. As Investopedia is a not-for- profit organization, Investopedia is not a financial advisor or financial advisor. Investopedia does not guarantee the accuracy, timeliness, or appropriateness of the information provided in this article, and everyone's situation is unique, so it is best to always consult a qualified professional before making any financial decision.

Conclusion

To sum up, the implementation of smart contracts in the insurance industry can result in numerous benefits, such as increased efficiency, reduced expenses, and improved insurance products and services. Nevertheless, there are several obstacles to overcome in order to successfully integrate smart contracts into the industry. Insurance companies should allocate resources towards building the necessary digital infrastructure and systems that support smart contracts. With these measures in place, insurers can take advantage of the potential of blockchain technology and smart contracts to transform the insurance sector.

References

1. Zheng, Z.; Xie, S.; Dai, H.-N.; Chen, W.; Chen, X.; Weng, J.; Imran, M. An overview on smart contracts: Challenges, advances and platforms. *Future Gener. Comput. Syst.* **2020**, *105*, 475–491.

2. Strebko, Julija & Romanovs, Andrejs. (2018). The Advantages and Disadvantages of the Blockchain Technology. 1-6. 10.1109/AIEEE.2018.8592253.

3. Ream, J.; Chu, Y.; Schatsky, D. Upgrading blockchains: Smart contract use cases in industry. *Retrieved Dec.* **2016**, *12*, 2017.

4. Szabo, N. The idea of smart contracts. *Nick Szabo's Pap. Concise Tutor.* **1997**, *6*, 199.

5. Gatteschi V., F.Lamberti, etc., "Blockchain and Smart Contracts for Insurance: Is the Technology Mature Enough?", MDPI, Basel, Switzerland, Future Internet 2018, p.16,2018.

6. Amponsah, Anokye & Adekoya, Adebayo & Weyori, Benjamin. (2021). Blockchain in Insurance: Exploratory Analysis of Prospects and Threats. International Journal of Advanced Computer Science and Applications. 12. 10.14569/IJACSA.2021.0120153.

7. Solidity: https://docs.soliditylang.org/en/latest/

8. Remix IDE: Remix - Ethereum IDE

9. Taherdoost, H. Smart Contracts in Blockchain Technology: A Critical Review. *Information* **2023**, *14*, 117. https://doi.org/10.3390/info14020117