TECHNISCHE UNIVERSITÄT DRESDEN

Faculty of Computer Science Institute of Software and Multimedia Technology

Chair of Software Technology

Bachelor Thesis

Contract to Contract Calls for Financial Applications in Algorand

Vasil Petrov

Born on: April 8, 1999 in Burgas, Bulgaria
Matriculation number: 4818111
Matriculation year: 2018

to achieve the academic degree

Bachelor of Science (B.Sc.)

Supervising professor
Prof. Dr. Uwe Aßmann

Submitted on: August 17, 2022

Contents

Abstract

here goes the abstract, I am going to write it when I am done with the rest, see you then ;)

1 Introduction

1.1 Motivation

Blockchain technology has the potential to revolutionize the financial world. Its industrial adoption is growing at a rapid rate mainly due to the rising popularity of cryptocurrency. Big names in the banking sector such as Bank of America, Agricultural Bank of China, HSBC, BNP Paribas and many more have already invested a lot of capital into researching and experimenting with blockchain solutions for their financial services with the hope of reducing processing costs and time.[4]

In simple terms, blockchain represents a decentralized immutable ledger where transactions and assets can be securely stored and processed. Through cryptography and other mechanisms used by this technology, the need for a trusted central authority as a middleman in the transaction between two individuals is redundant. This could open the doors to the global financial market for a lot of people that don't have access to modern banking services.[6] Fintech startups are quickly trying to capture this new market by developing financial applications on various blockchain platforms like Algorand for example. Algorand is a cutting-edge blockchain technology created by Silvio Micali, a professor at the Massachusetts Institute of Technology and a Turing Award winner, and his team of leading scientists. In this thesis, we will look at the qualities and possibilities of developing financial applications specifically on Algorand.

Smart contracts play a key role in the development of blockchain-based applications. They can be interpreted as programs that are stored on the distributed ledger and execute some logic automatically when certain conditions are met. As suggested in this paper[7] by T. Mattis and R. Hirschfeld, an analogy could be made between deployed smart contracts and objects in object-oriented programming. Each contract has its own *states* and *methods* and plays its specific part in the application.

Object-oriented programming (OOP) is a paradigm that is used for designing modular, reusable software. With this approach, a complex problem could be solved by dividing it into a set of subproblems among different classes, making the development process easier and more intuitive. In the blockchain world, this could be achieved using contract composability which refers to combining multiple smart contracts, where each solves a certain aspect of the global problem, achieving again a modular structure. The way to achieve composability is by contract-to-contract calling. This allows one smart contract to interact and execute transactions to another contract on its own. Thus, with just one transaction sent by the client of the application, a chain of complex logic and communication between different contracts can be triggered, performing the requested task. This can serve as an example that despite the new ways of solving existing problems through blockchain, the well-established

principles and practices for software development and design remain the same.

An important feature of blockchain is that once something is stored on the ledger, it becomes immutable. However, this can sometimes be bad for software development, because often certain parts of the program need to be optimized or rewritten. Luckily there are ways to get around this drawback by using upgradable smart contracts. The idea is that all transactions and values that have already happened on-chain will be visible and immutable on the ledger until it exists. However, we can still swap the logic of a smart contract with a new upgraded one. In this thesis, we will examine how this can be achieved on Algorand and other famous blockchain protocols.

Combining the OOP programming approach with blockchain could be a very powerful concept. However, there still exists a gap in research and development into blockchain for financial applications from an academic perspective, and this gap could hurt the adoption of this relatively new technology by the banking sector. For this reason, the main focal point of this thesis will address the topic of role-based programming in the context of blockchain development using Algorand. As mentioned in [11], the idea behind role-oriented programming is that the same object (entity) in different contexts could play different roles. Each role serves a specific purpose and can overwrite the behavior of the object allowing it to adapt to a certain context dynamically. For example, a customer of a bank can be at the same time a borrower and depositor. There is no point to model borrower and depositor as separate objects, but instead, as roles that can be added or dropped dynamically by customer objects. This way we can achieve a more accurate and natural representation of the real world in our program.

The primary task of this thesis will be to answer the following research questions:

1. Is it possible for the Algorand blockchain to interact and be a part of a role-based application?
2. How features such as contract-to-contract calling and upgradable smart contracts can help the Algorand blockchain to implement a role-oriented approach?
3. What are the drawbacks and advantages of developing role-based applications using the Algorand blockchain?

1.2 Goal and objectives

To answer the research questions asked in Motivation, a simple banking app will be created. The Algorand blockchain platform is going to be used for the development of the program together with python as a general-purpose language. In its implementation, I will demonstrate the relationship between OOP objects and smart contracts. Roles will also be presented by a client and an investor class that are going to communicate with a smart contract representing the bank. Depending on the role, different functionalities will be accessible to the object. Services such as opening a bank account, depositing, withdrawing and transferring funds will also be used to demonstrate contract-to-contract calling features. Contract composability and upgradeability will be applied to show how smart contracts could communicate and cooperate to build a more complex application.

1.3 Structure of the thesis

This thesis is divided into seven chapters. First and foremost, the topic of the research is laid out together with the goals and objectives in the Introduction section. The reader should be familiar with some basic concepts to better grasp and follow the research process, this is

done in the Background chapter. Concepts like role-based programming, blockchain, smart contracts and the Algorand protocol will be explained in more detail.

Then comes Problem analysis, dedicated to the description of the bank application to be created. The structure of the app will be presented in detail with the help of Class and Sequence UML diagrams, explaining each important functionality. It will also be answered how this app should solve the research questions. The tech stack used for its implementation will also be mentioned.

In section four comes the practical part. A detailed description supported by code snippets will illustrate the development process and how the bank application part by part is being created.

The test results regarding the functionality of the developed app are shown in the Evaluation section of this thesis. An argument will be presented on how these results provide an answer to the research questions.

In the Discussion chapter, the pros and cons of developing applications on the Algorand blockchain will be considered. The arguments will also be supported by performance tests. Reference and comparison to another popular blockchain platform, namely Ethereum, will also be made here.

Finally, the Conclusion section presents perspectives for the extension of the presented research and app as well as a summary of the researched questions and process.

2 Background

In this chapter, the necessary basic knowledge of different concepts is provided which is key for understanding the rest of the thesis. We begin with a more in-depth definition of blockchain technology and what advantages it offers. After that, smart contracts are going to be discussed with an explanation of features such as contract-to-contract calling and upgradeable contracts. Then we will continue with a description of the Algorand blockchain and its perks. Finally, we will end with a more detailed explanation of the role-based programming approach.

2.1 Blockchain technology

Many consider blockchain as a revolutionary technology that is the next big thing. It is even compared in the potential to the early years of the internet. However, many do not know that its basic idea and fundamentals predate the emergence of cryptocurrencies.

2.1.1 History and Structure

David Chaum in his dissertation from 1982 called "Computer Systems Established, Maintained, and Trusted by Mutually Suspicious Groups"[1] is one of the first to suggest a protocol that resembles blockchain as we know it today. He proposed the implementation of a decentralized system that can be maintained and trusted by achieving mutual consensus between its participants. Almost a decade later S. Haber and W. Scott Stornetta proposed in their work[3] a time-stamping mechanism for digital documents. The process involves a one-way hash function where the hashes of the different time-stamped documents are linked together, forming a tamper-proof cryptographically secured chain. Based on these and some other early works, Satoshi Nakamoto published a whitepaper titled "Bitcoin: A Peer-to-Peer Electronic Cash System"[8], where blockchain was introduced as a public distributed ledger for bitcoin payment transactions. From this point on, the rising popularity of this technology and cryptocurrencies began.

As explained by A. Panwar and V. Bhatnagar in [9], blockchain is just a type of distributed ledger technology (DLT). This is important to note because many people use both terms interchangeably which is technically wrong. DLT is a decentralized database distributed as copies across multiple computer nodes connected via a peer-to-peer (P2P) network. If a modification occurs, then all copies of the ledger get autonomously updated and synchronized. With the things said so far, it sounds exactly like the definition of blockchain. However, the key difference is that DLT doesn't necessarily need to represent its data in a block

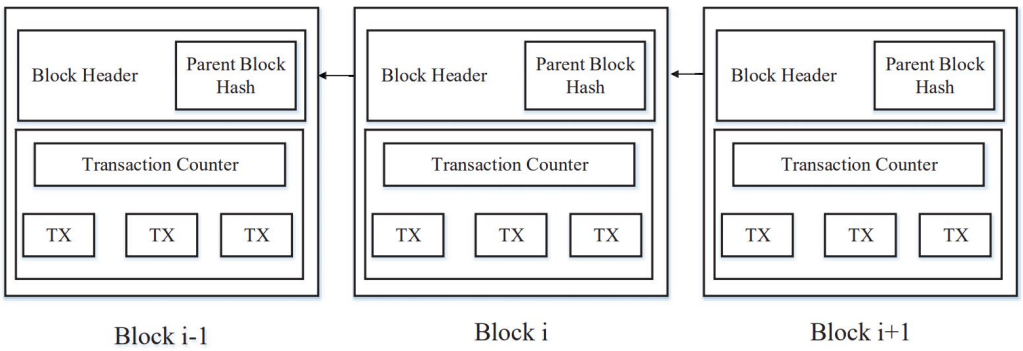

Figure 2.1: An example of a basic blockchain structure (Figure 1 in [12])

structure, but in any type of structure. Furthermore, DLT doesn't necessarily store data in a specific sequence. Many blockchains also use some kind of native currency/token on the network, used to pay fees for interacting with the network and also as an exchange value between different participants. It is also given as a reward for the people who keep the blockchain alive and add/validate new blocks to the chain. However, not all forms of DLT require a token or a currency to keep the system going. These are just some of the key differences between the two technologies.

As described by Z. Zheng et al.[12], blockchain can be viewed as a sequence of blocks that are linked together in a chronological order using cryptography. As can be seen on Figure 2.1.1, each block is composed of a *block header* and a *block body*. The block header holds the hash value that points to the previous block (also called a parent block) and a timestamp that shows when the block was validated and added to the chain. In each block body, a certain number of transactions is being stored depending on the size of the block and each transaction. Any attempt to modify some information in an existing block will affect its hash address and all others before and after it because everything is connected. The participants will compare the change with their current copy of the ledger and immediately notice the change and reject it. This is why blockchain is considered immutable and tamper-proof.

As further mentioned in [12], asymmetric cryptography (public-key cryptography) is one of the key components of blockchain technology. Each participant in the network owns a pair composed of a private key and a public key. The private key is used to verify transactions and prove ownership of a blockchain address and the public key is used for the unique account address, allowing users to receive cryptocurrencies or other digital assets.

2.1.2 Types of blockchain

Blockchain networks can be divided into 3 categories: public, private and consortium.[5]

Public blockchains are permissionless, meaning that they are open networks, where anyone can participate and read/write to the ledger without needing the approval of any trusted authority. The transactions are completely transparent and accessible for anyone to see, but in most cases they are anonymous and you can't directly link them to the person behind them. Here we see anonymity and transparency, qualities for which blockchain is liked and which are absent in the standard financial system.

Private blockchains on the other hand require permission to join them. They are more suitable for organizations and companies which want to benefit from its implementation but don't want everyone to have control over the network. Normally, it is again a centralized structure, where only some participants have the right to verify and add new data to the

chain.

In consortium blockchains, there is more than one company/organization in charge to decide who has what rights on the blockchain. It is best suited for organizational collaboration. It can be viewed as a hybrid between public and private blockchain because the power is not only in one central structure, but at the same time, not everyone can participate and interact with the network.

From this moment forth, the word blockchain will refer to the public version of the network, because this thesis will revolve around Algorand, which is a permissionless public blockchain.

2.1.3 Consensus mechanisms

Based on the provided knowledge so far, blockchain is a decentralized system available for everyone to take part in it. Everyone has an updated and synchronized copy of the ledger. The control of the network is not in the hands of some central entity but all participants in the network. However, what guarantees that the copy of each node will be up to date and correct? Who validates and adds new transactions/data to the chain? How is it verified that these new transactions are valid at all? Here comes the important role of a consensus mechanism. The most used ones are Proof of Work (PoW) and Proof of Stake (PoS), which are going to be briefly explained in the following paragraphs.

PoW was the first consensus mechanism ever for blockchain. It was introduced in the whitepaper for Bitcoin[8], where Nakamoto argued that this will be the mechanism to keep the distributed ledger secure and consistent while being decentralized and without a central authority to regulate. As explained in the research article "On the Security and Performance of Proof of Work Blockchains"[2] by A. Gervais et al., the PoW principle of selecting a node to add a new block to the chain is by propagating to the network a cryptographic puzzle that needs to be solved. Every node also called a miner, is in direct competition with the others and spends computational resources and time to find the solution. Once an answer is found, it gets propagated to the network together with the new block and the other nodes verify if they are valid or not. If yes, the new block gets appended to the chain and all participants update their ledgers. If not, the block will be rejected and a new proposal will be expected. This process is repeated and thus the chain is kept consistent, valid and secure.

PoS has a different approach to reaching a consensus. As described by F.Saleh in [10], PoS chooses the proposer (forger) of a new block and the validators based on their stake or in other words on the proportion of native coins they are holding. The protocol chooses a random coin each round from the supply and the node to whom it belongs will be entitled to propose a new block on the chain. This is one of the reasons why PoS is considered a more elegant mechanism than PoW and its use in new blockchain projects is increasing.[10] They are better in terms of scalability and require much fewer resources such as electricity or investment in equipment to participate.

2.1.4 Summary of blockchain's characteristics

As a summary of the previous three subsections, blockchain is an immutable, decentralized, distributed, transparent and secure network that is maintained and managed by all its participants that run its software. These participants are called nodes and each has its copy of the ledger that holds all of the transactions and information that are so far written on the chain. This supports the claim of distributed and transparent nature of the system. Furthermore, the authority and control belong to all the nodes, not to a single central structure. With the help of a consensus mechanism, the participants can collaborate without problems in adding new blocks to the chain and maintaining its integrity and consistency. This

achieves decentralization and security of the network and a transaction is accepted only if the majority of nodes agreed that it is valid. Thanks to the cryptography used in the protocol, already existing records on the chain can't be edited or deleted. This guarantees the immutability of the transactions made on the blockchain.

2.2 Smart contracts

2.3 Algorand

2.4 Role-based programming

3 Problem analysis

4 Implementation

5 Evaluation

6 Discussion

7 Conclusion

Bibliography

[1] Chaum, David: Computer Systems Established, Maintained and Trusted by Mutually Suspicious Groups, Ph.D. dissertation, Dept. Comput. Sci., Univ. California, Berkeley, 1982

[2] Gervais, Arthur ; Karame, Ghassan O. ; Wüst, Karl ; Glykantzis, Vasileios ; Ritzdorf, Hubert ; Capkun, Srdjan: On the Security and Performance of Proof of Work Blockchains. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. New York, NY, USA : Association for Computing Machinery, 2016 (CCS '16). – ISBN 9781450341394, S. 3–16

[3] Haber, Stuart ; Stornetta, Wakefield: How to time-stamp a digital document. In: *Journal of Cryptology* 3 (1991), S. 99–111

[4] Hassani, Hossein ; Huang, Xu ; Silva, Emmanuel: Banking with blockchain-ed big data. In: *Journal of Management Analytics* 5 (2018), Nr. 4, S. 256–275

[5] Khan, Abdul G. ; Zahid, Amjad H. ; Hussain, Muzammil ; Farooq, M ; Riaz, Usama ; Alam, Talha M.: A journey of WEB and Blockchain towards the Industry 4.0: An Overview. In: *2019 International Conference on Innovative Computing (ICIC)*, 2019, S. 1–7

[6] Kshetri, Nir: Potential roles of blockchain in fighting poverty and reducing financial exclusion in the global south. In: *Journal of Global Information Technology Management* 20 (2017), Nr. 4, S. 201–204

[7] Mattis, Toni ; Hirschfeld, Robert: Activity Contexts: Improving Modularity in Blockchain-Based Smart Contracts Using Context-Oriented Programming, Association for Computing Machinery, 2018 (COP '18). – ISBN 9781450357227, S. 31–38

[8] Nakamoto, Satoshi: Bitcoin: A Peer-to-Peer Electronic Cash System. (2008)

[9] Panwar, Arvind ; Bhatnagar, Vishal: Distributed Ledger Technology (DLT): The Beginning of a Technological Revolution for Blockchain. In: *2nd International Conference on Data, Engineering and Applications (IDEA)*, 2020, S. 1–5

[10] Saleh, Fahad: Blockchain without Waste: Proof-of-Stake. In: *The Review of Financial Studies* 34 (2020), Nr. 3, S. 1156–1190. – ISSN 0893–9454

[11] Steimann, Friedrich ; Stolz, Fabian U.: Refactoring to Role Objects. In: *Proceedings of the 33rd International Conference on Software Engineering*, Association for Computing Machinery, 2011 (ICSE '11). – ISBN 9781450304450, S. 441–450

[12] Zheng, Zibin ; Xie, Shaoan ; Dai, Hongning ; Chen, Xiangping ; Wang, Huaimin: An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. In: *2017 IEEE International Congress on Big Data (BigData Congress)*, 2017, S. 557–564

List of Figures

List of Tables

Statement of authorship

I hereby certify that I have authored this document entitled *Contract to Contract Calls for Financial Applications in Algorand* independently and without undue assistance from third parties. No other than the resources and references indicated in this document have been used. I have marked both literal and accordingly adopted quotations as such. There were no additional persons involved in the intellectual preparation of the present document. I am aware that violations of this declaration may lead to subsequent withdrawal of the academic degree.

Dresden, August 17, 2022

Vasil Petrov