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

The first chapter of this thesis deals with the basic motivation and defines the general problem to be researched. This is followed by an explanation of the objective and a description of the structure of the work.

1.1 Motivation

The industrial adoption of blockchain is growing at a rapid rate mainly due to the rising popularity of cryptocurrency. Companies of all sizes from different sectors such as healthcare, finance, education, etc. have already invested a lot of capital into researching and developing blockchain solutions for their services and products. This relatively new technology is greatly appreciated because of its decentralized architecture that enables the users to store immutable records of digital data without the risk of future tampering.

Smart contracts play a key role in the development of blockchain-based applications. They can be interpreted as programs that are stored on-chain and execute some logic when certain conditions are met. As suggested in this paper[1] 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 small analogy presented here can serve as an example that despite the new ways of solving existing problems through blockchain applications, the well-established principles for software development and design remain the same.

As described in [2], new design patterns are being created or existing ones are being adapted to solve the challenges connected with application development on blockchain. Standardized practices could only be beneficial for the young blockchain community and increase the speed of mass adoption of the technology. Combining the OOP programming approach with a blockchain working on the backend in an application could be a very powerful concept.

The main focal point of this thesis will address the topic of role-based programming in the context of blockchain development. As mentioned in [3], the same object (entity) in different contexts could play different roles. Each role serves a specific purpose and can overwrite the behaviour of the object allowing it to adapt to a certain context dynamically. The primary task of this thesis is to investigate whether role-based programming can be successfully used to design a blockchain application. How can features such as contract-to-contract calling and upgradable smart contracts help with the implementation of different context roles and why are they important for achieving separation of concerns. It will also be discussed if the role-based approach is at all beneficial for blockchain development.

1.2 Goal and objectives

To answer the research questions asked in Motivation, a simple banking application will be developed. 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 will also be applied to show how smart contracts could communicate and cooperate to build a more complex application. The Algorand blockchain platform is going to be used for the development of this bank app together with python as a general-purpose language.

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, Algorand platform, etc. 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

3 Problem analysis

4 Implementation

5 Evaluation

6 Discussion

7 Conclusion

Bibliography

[1] Mattis, T. ; Hirschfeld, R.: 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

[2] Rajasekar, Vijay ; Sondhi, Shiv ; Saad, Sherif ; Mohammed, Shady: Emerging Design Patterns for Blockchain Applications, 2020, S. 242–249

[3] 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

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