

Use Cases and Design of Web Applications in Decentralized Finance

Johannes Hüsters



BACHELORARBEIT

eingereicht am
Fachhochschul-Bachelorstudiengang

Software Engineering

in Hagenberg

im Februar 2021

Advisor:

DI Martin Harrer

© Copyright 2021 Johannes Hüsters

This work is published under the conditions of the Creative Commons License *Attribution-NonCommercial-NoDerivatives 4.0 International* (CC BY-NC-ND 4.0)—see <https://creativecommons.org/licenses/by-nc-nd/4.0/>.

Declaration

I hereby declare and confirm that this thesis is entirely the result of my own original work. Where other sources of information have been used, they have been indicated as such and properly acknowledged. I further declare that this or similar work has not been submitted for credit elsewhere. This printed copy is identical to the submitted electronic version.

Hagenberg, February 1, 2021

Johannes Hüsers

Abstract

—0.5 to 1 page—

Kurzfassung

—0.5 to 1 page—

Contents

Declaration	iv
Abstract	v
Kurzfassung	vi
1 Introduction	1
1.1 Motivation	1
1.2 Goals	1
1.3 Structure of the Thesis	1
2 Fundamentals	2
2.1 Cryptography	2
2.1.1 Public Key Infrastructures	2
2.1.2 Elliptic Curve Cryptography	2
2.1.3 Hashing Functions	2
2.2 The Ethereum Blockchain	2
2.2.1 Blockchain Fundamentals	2
2.2.2 Clients	2
2.2.3 Wallets	2
2.2.4 Transactions	2
2.3 Smart Contracts	2
2.3.1 Solidity	3
2.4 Decentralized Finance	3
2.4.1 State of the Art	3
3 Use Cases of Decentralized Finance	4
3.1 Classification of Financial Services	4
3.2 Store of Value	4
3.2.1 Proof of Work	4
3.2.2 Proof of Stake	5
3.2.3 Switching the Consensus Algorithm	5
3.3 Payments	6
3.3.1 Bitcoin	6
3.3.2 Stable Coins	6
3.3.3 Dynamic Supply Protocols	6

3.3.4	Stability Protocols	6
3.4	Lending & Borrowing	6
3.5	Exchanging	6
3.5.1	Characteristics of Decentralized Exchanges	7
3.5.2	Liquidity Pools	7
3.5.3	Risks	7
3.6	Investing	7
3.6.1	Tokens	7
3.6.2	Derivatives	7
4	Design and Architecture	8
4.1	Architecture of Decentralized Web Applications	8
4.1.1	Technologies	8
4.2	Lending and Borrowing Application	8
4.3	Token Exchange	8
4.4	Asset Management Platform	8
5	Closing Remarks	9
5.1	Criticism	9
5.2	Risks	9
5.3	Prospective Impact	9
A	Technical Details	10
B	Supplementary Materials	11
B.1	PDF Files	11
B.2	Code-Files	11
B.3	Reference Files	11
References		12
Literature		12
Online sources		12

Chapter 1

Introduction

1.1 Motivation

—1.25 pages—

1.2 Goals

—0.75 pages—

1.3 Structure of the Thesis

—0.5 pages—

Chapter 2

Fundamentals

2.1 Cryptography

— 1.5 pages —

2.1.1 Public Key Infrastructures

2.1.2 Elliptic Curve Cryptography

2.1.3 Hashing Functions

Blockchain Grundlagen Schritt 11: Hashfunktionen in der Realität, S. 99ff Allgemeine Anwendungen von Hashfunktionen: * Data comparison * Detect data change * Computational complex problems

Blockchain Anwendungen: S 109 * Save transaction data * Digital fingerprint of transaction data * Implement generation costs

2.2 The Ethereum Blockchain

— 2.5 pages —

2.2.1 Blockchain Fundamentals

2.2.2 Clients

2.2.3 Wallets

2.2.4 Transactions

2.3 Smart Contracts

— 1 page —

2.3.1 Solidity

2.4 Decentralized Finance

— 2 pages —

2.4.1 State of the Art

current applications, relevance on the market, technologies, ...

Chapter 3

Use Cases of Decentralized Finance

— 9 pages —

This chapter aims to introduce the five most relevant use cases of Decentralized Finance. Each use case is built on top of the previous one and rises in complexity. For example storing value in digital systems is quite easy nowadays but moving real physical assets such as gold or real estate into a blockchain is still pretty demanding. Note, that each type of financial service is not specific to Decentralized Finance, which means that they are applicable to every financial environment.

3.1 Classification of Financial Services

—0.5 pages—

3.2 Store of Value

No matter of the financial product or service, they all need one essential thing. In fact, our entire economic system depends on it: a currency, or in simpler terms, something to store value. Storing value is a fundamental trait of money alongside with being exchangeable and having a unit of account. The last two characteristics are comparatively easy to accomplish in the online world. Storing value in a digitalized way without having a central trusted authority, however, was a similar challenge as the people had to face in the early days when they started to switch from bartering to a real currency. A common consensus needs to be created where everyone can easily verify that a specific piece of money is authentic and wasn't created illegally. Even further, each person that uses this money needs a proof that it is storing real value. This wasn't possible until Satoshi Nakamoto, an anonymous person or group, published a specification [4] in 2008, on how digital money could be implemented. The first cryptocurrency was founded: Bitcoin.

3.2.1 Proof of Work

Bitcoin uses a mechanism called proof-of-work in order to give each block its value. This utilizes a cost-function which is designed to be easily verifiable but quite expensive to compute [3]. In order to create a new block, a value needs to be found that matches with

the correct number of zero bits of the block hash. Once the correct solution has been found, the new block represents real value, because operating this computationally intensive task on a CPU costs a lot of electricity. The integrity of this block is guaranteed as well, because in order to change the history of the blockchain, each changed block needs to be re-computed. The longest chain of blocks on the network is accepted as the “truth”, so if dishonest people want to cheat and change the history of the blockchain, they would need to create the longest chain of blocks which can be only achieved by having more than 50% of the computing power of the whole network. While these so-called 51% attacks are a threat on blockchains with a smaller number of network members, it is very unlikely to happen on a well established network like the Bitcoin blockchain [5]. The concept of proof-of-work in order to achieve consensus in a decentralized manner became very popular and can be used in other areas of application too, such as elections, lotteries, asset registries, digital notarization and more [1].

3.2.2 Proof of Stake

A different approach to storing digital value is proof-of-stake. Instead of making the participants of the network solve computational expensive problems, the consensus is created by proving that someone owns specific funds. A blockchain that uses a proof-of-stake algorithm to achieve consensus has a set of validators. These validators are running a master node which gives them the opportunity to vote. In order to do so, the validators need to prove that they own a specific amount of funds. This is done by sending a special transaction, which locks away their currencies for a specific time. If the validators are honest and their vote corresponds to the result of the majority, they will get their funds back and an additional reward proportional to their deposited stake. If their vote gets rejected, the dishonest validator risks losing his money [2].

3.2.3 Switching the Consensus Algorithm

At the time of writing this in October 2020, the Ethereum blockchain uses a similar proof-of-work algorithm like Bitcoin, called *Ethash*. However, Ethereum plans to switch to a proof-of-stake algorithm called *Casper* in the near future [8]. Algorithms that are based on proof-of-stake have the big advantage of less energy consumption because they are not focused on computational intensive tasks. The Bitcoin network, for example, has an annual electrical energy consumption of approximately 72.18 terawatt hours which is comparable to the consumption of Austria [7]. But proof-of-stake comes also with a caveat. Implementing incentives on a voting system which are based on the amount each validator is willing to stake means that the rich validators get richer and the poor validators stay poor.

The process of changing a consensus algorithm of a blockchain is not easy to accomplish. Because the network is decentralized, there is no single entity which can force all members to change the algorithm from proof-of-work to proof-of-stake. Just letting the people choose what they prefer to use is also not a good idea, because it is very unlikely that all agree on one type of algorithm. That would probably trigger a hard fork of the blockchain resulting in two separate networks, similar to what happened to Bitcoin in August 2017.

Bitcoin’s network is by design very slow in verifying transactions. A block which

is mined approximately every 10 minutes has a size of about 1 MB. Due to the small block size, Bitcoin is only capable of processing 7 transactions per second. Some people wished for a Bitcoin network which is more suitable for day to day payments. That's why Bitcoin Cash was created, emerged out of a hard fork of the Bitcoin network. In order to prevent a hard fork on Ethereum due to the transition from proof-of-work to proof-of-stake, the Ethereum protocol [6] has a built in mechanism called *difficulty bomb* which makes it more difficult over time to be profitable with mining¹ by increasing the size of the problem to solve. This ensures a smooth transition of all members to switch to proof-of-stake.

For a simple user of the blockchain who is neither a miner nor a validator it doesn't matter which consensus algorithm is being used. In fact, it isn't even important when developing decentralized applications based on smart contracts like it's done in chapter 4. What's important is, that the blockchain is able to store real value on the network by reaching consensus by its members.

3.3 Payments

—1.5 pages—

3.3.1 Bitcoin

3.3.2 Stable Coins

3.3.3 Dynamic Supply Protocols

3.3.4 Stability Protocols

3.4 Lending & Borrowing

—1.5 pages—

3.5 Exchanging

—1.5 pages—

¹used synonymously to proof-of-work

3.5.1 Characteristics of Decentralized Exchanges

3.5.2 Liquidity Pools

3.5.3 Risks

3.6 Investing

3.6.1 Tokens

3.6.2 Derivatives

—1.5 pages—

Chapter 4

Design and Architecture

— 9 pages —

4.1 Architecture of Decentralized Web Applications

4.1.1 Technologies

React Truffle Ganache Web3 Solidity Mocha / Chai

4.2 Lending and Borrowing Application

4.3 Token Exchange

4.4 Asset Management Platform

Chapter 5

Closing Remarks

5.1 Criticism

—0.5 pages—

5.2 Risks

—0.5 pages—

5.3 Prospective Impact

—2 pages—

Appendix A

Technical Details

Appendix B

Supplementary Materials

List of supplementary data submitted to the degree-granting institution for archival storage (in ZIP format).

B.1 PDF Files

Path: /

thesis.pdf Master/Bachelor thesis (complete document)

B.2 Code-Files

Path: /media

.

B.3 Reference Files

Path: /online-sources

.

References

Literature

- [1] Andreas Antonopoulos. *Mastering Bitcoin: Programming the Open Blockchain*. 2nd ed. Sebastopol: O'Reilly Media, Inc., 2017 (cit. on p. 5).
- [2] Andreas Antonopoulos and Gavin Wood. *Mastering Ethereum: Building Smart Contracts and Dapps*. Sebastopol: O'Reilly Media, Inc., 2018 (cit. on p. 5).
- [3] Adam Back. *Hashcash - A Denial of Service Counter-Measure*. Aug. 2002. URL: <http://www.hashcash.org/papers/hashcash.pdf> (cit. on p. 4).
- [4] Satoshi Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. Oct. 2008. URL: <https://bitcoin.org/bitcoin.pdf> (cit. on p. 4).
- [5] Melanie Swan. *Blockchain: Blueprint for a New Economy*. Sebastopol: O'Reilly and Associates, 2015 (cit. on p. 5).
- [6] Gavin Wood. *Ethereum: A Secure Decentralised Generalised Transaction Ledger*. Version 3e2c089. Sept. 2020. URL: <https://ethereum.github.io/yellowpaper/paper.pdf> (cit. on p. 6).

Online sources

- [7] *Bitcoin Energy Consumption Index*. Oct. 2020. URL: <https://digiconomist.net/bitcoin-energy-consumption> (visited on 10/12/2020) (cit. on p. 5).
- [8] *Ethereum Casper*. Sept. 2018. URL: <https://twitter.com/i/events/1036281460704112645> (visited on 11/13/2020) (cit. on p. 5).

Check Final Print Size

— Check final print size! —



— Remove this page after printing! —