KU LEUVEN

FACULTY OF ECONOMICS AND BUSINESS

Feasibility of blockchain application as medium for collaborative systems or databases

How to replace a previously thought indispensable middleman with technology

Pelle Jacobs
r0364018

Thesis submitted to obtain the degree of

Masters in Information Systems Engineering
Majoring in Data Science

Promotor: Prof. Dr. Jochen De Weerdt
Assistant: Vytautas Karalevicius

Academic year: 2016-2017

Contents

Preface

Leuven, July 18, 2017.

Chapter 1

Introduction

THIS IS OLD AN NEEDS A REWRITE

Hitting a market capitalization of $13.8 Billion in December 2013, it was clear that a once fringe cryptocurrency called 'Bitcoin' had hit mainstream. By building on decades of research in decentralized currencies and cryptography, Bitcoin is the first successful implementation of a truly decentralized currency. Most of this success has been attributed to an innovation called 'Blockchain'.

Now, the question is: 'How can this blockchain technology be leveraged to power collaborative, distributed databases?' To provide an answer to this question, this thesis will be split in three chapters:

In the first chapter, the blockchain technology will be thoroughly discussed. The chapters starts out with a description of the concept 'blockchain', the specific characteristic of blockchains and differences between implementations of blockchains. The following part looks into why the blockchain was developed in the first place, looking at the advantages a blockchain provides compared to its alternatives. Finally, several implementations of blockchains and blockchain applications are reviewed.

The second chapter describes distributed and decentalised databases. To maintain consistency, it follows a similar structure as the first chapter: a definition of what is meant with the terms 'distributed' and 'decentralized' and how these concepts can be applied to databases. Next follows a comparison between distributed and decentralized databases and fully centralized databases, looking at the advantages and disadvantages of both implementations. Finally, some common implementations of distributed and decentralized database are reviewed.

The final chapter combines the previous two chapters by focusing on how to bring collaborative, distributed databases to a blockchain. A first section will discuss the general

advantages but also the trade offs of bringing a database to the blockchain. The second section suggests several possible implementations.

Chapter 2

Prior Research

2.1 Public key cryptography

Public key cryptography is the backbone of most distributed systems as it provides both a way to encrypt a message as to validate the source of a message, without the need to agree upon a shared key. The most used implementation of this idea is the RSA encryption method, named after inventors Rivest, Shamir and Adleman [3].

2.1.1 Basics

In previous encryption systems before the public key cryptography, two parties who wanted to safely exchange messages needed to agree on a common cypher first. One of the more famous of such encryption systems is the Ceasar cypher [2], in which the cypher consisted of offsetting every character with a previously agreed upon number. However, if a third party manages to intercept transmission of this cypher itself, all encrypted messages can easily be decoded. Therefore, a trusted third party network is needed to securely exchange the cypher.

Public key cryptography uses a pair of hashes instead. These hashes are often referred to as the keys. These hashes are algorithmically generated so that they are cryptographically connected: data encrypted with one key can only be decrypted with the other key and visa versa [4].

One of these keys can be publicly broadcasted. As a result, everyone knows that this key is connected to the owner's identity. Therefore, it is referred to as the public key.

The other key will be held privately. It is very important that nobody except the owner

know the content of this key, as it will be used to prove the owner's identity. It is therefore referred to as the private key.

Because no cypher has to be exchanged, there is no need for the trusted middleman to exchange the cypher, eliminating a key weakness of cypher encryption.

2.1.2 Uses: encryption and source validation

A first use case of public key cryptography is encryption. If a person called Alice wants to send a message to Bob, she encrypts her message with Bob's public key. As a result, only Bob can decrypt this message as only Bob knows the corresponding private key. The message can now be send over any insecure network, such as the internet, email or even carrier pigeon, without any danger of leaking its contents.

Besides encryption, public key cryptography can also be used for source validation. If Alice wants to broadcast a message to the world, she can encrypt the message with her private key. As the encrypted message can only be decrypted with Alice's public key, everyone knows that only Alice could have encrypted it. This case is often referred to as a digital signature, as Alice signed the message with her private key.

For full security, a secure message is often first encrypted with the private key of the sender and then the public key of the recipient. Now only the recipient can decrypt the message and can then validate the origin of the message. This full process is modeled in figure 2.1.

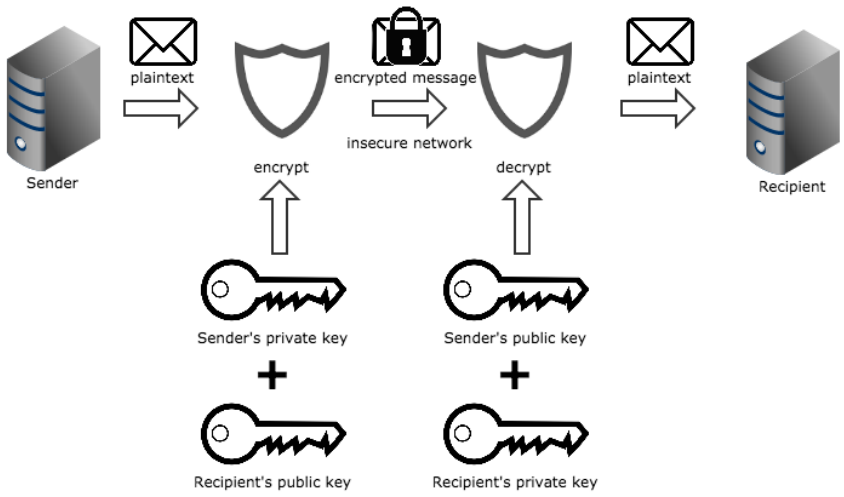

Figure 2.1

2.2 Blockchain

2.2.1 Definition of a blockchain

In the last few years, the blockchain as a concept has been given a wide range in meanings and definitions. Therefore, we will start out this paper by thoroughly defining this concept, to avoid any misunderstanding or confusion further on.

Antonopoulos (2014) [1] probably defines it best: "The blockchain data structure is an ordered, back-linked list of blocks and transactions." (p. 159). There are several aspects in this definition that need further explanation.

Back-linked lists as data structure

First of all, we are talking about a data structure: "A specialised format for organising and storing data" [5]. There are several types of data structures such as arrays, lists, graphs, trees, etc.

Next, a blockchain is a specialised version of an "ordered, back-linked list". A list is data structure that combines a number of ordered values (Abelson 1996 [6]). Incidentally, the "ordered" in the definition provided by Antonopoulos is superfluous, as by definition every list is ordered. An unordered list would no longer be a list, but would be a set instead. A blockchain will use back-linking to preserve the order of the values. This means that all values have a reference to the previous value in the list. For a blockchain, this means that every block (see subsubsection **??**) in the blockchain contains the "hash" of the previous block. We will go further into hashes in a next section.

Transactions

Antonopoulos (2014) [1] defines transactions as: "data structures that encode the transfer of value between participants in the bitcoin system" (p. 109). Although it gives a good idea of what transactions are, even in the context of the bitcoin blockchain is this definition not completely correct: despite most transactions being value transactions between participants, a transaction can also store data on a blockchain. Even the bitcoin blockchain allows non-monetary transactions to be included.

Therefore, we can conclude there are two types of transactions: monetary transactions and non-monetary transactions. In a monetary transaction, there is a transfer of value (eg. bitcoin) between participants. It must be noted that this does not have to be an instant transfer. Several blockchain protocols allow for a custom conditions to be

scripted into the transaction.

On the other side, there are non-monetary transactions that store data onto the blockchain. An example of a common used non-monetary transaction is the encoding of a document on the blockchain. The specific advantages of such practices will be discussed further in section **??** "Advantages of the blockchain".

Transactions are the raison d'être of the blockchain. All aspects that we will discuss in this chapter exist to make sure that transactions can be validated, propagated over the network and added to a distributed ledger.

Blocks

Antonopoulos (2014) [1] defines a block as: "a container data structure that aggregates transactions for inclusion in the public ledger, the blockchain." (p. 160). A block is literally a wrapper for multiple transactions to make it possible for them to be included into the blockchain. These blocks then form the ordered values of the back-linked list addressed in the beginning of this section, each linking to the previous block in the list.

To conclude, the blockchain is a list of blocks, with each block containing several transactions.

2.2.2 Properties of a blockchain

2.2.3 Examples of blockchains

Bitcoin

Ethereum

2.3 Distributed databases

2.3.1 Distributed, decentralized and centralized databases

2.3.2 Examples of distributed databases and networks

Git

BitTorrent

Non-persistent P2P networks

Chapter 3

Aim of research

Chapter 4

Proposed solutions

4.1 Storing onto the bitcoin blockchain

4.2 Blockchain distributed storage

4.3 Timestamping a database

4.4 Applied example: replacing notary will services

Chapter 5

Conclusion

Bibliography

[1] Antonopoulos, A. M. (2014). Mastering Bitcoin. Sebastopol, CA.

[2] Cipher, C., & Cipher, M. (2004). Introduction to cryptography. EEC, 484, 584.

[3] Rivest, R.L. and Shamir, A. and Adleman, L.M. US Patent 4,405,829 issued in 1983

[4] Calderbank, M. (2007). The RSA Cryptosystem: History, Algorithm, Primes.

[5] data structure (February 2006). Retrieved 23rd March 2017, from http://searchsqlserver.techtarget.com/definition/data-structure.

[6] Abelson, H. et al. (1996). Structure and Interpretation of Computer Programs. MIT Press.

Appendix A

A.1 Appendix

FACULTY OF BUSINESS AND ECONOMICS
Naamsestraat 69 bus 3500
3000 LEUVEN, BELGIË
tel. + 32 16 32 66 12
fax + 32 16 32 67 91
info@econ.kuleuven.be
www.econ.kuleuven.be