

University POLITEHNICA of Bucharest
Faculty of Automatic Control and Computers,
Computer Science and Engineering Department



BACHELOR THESIS

Exploring Decentralized Naming Services using Namecoin

Scientific Advisers:

Drd. Ing. Lucian Mogoşanu
As. Dr. Ing. Sergiu Mihai Costea
Prof. Dr. Ing. Răzvan Victor Rughiniş

Author:

Adriana Dinca

Bucharest, 2017

I would first like to thank my thesis advisors for the support and cooperation in writing this scientific paper. The direction of this paper was drawn with the helpful suggestions of my tutors. Particularly I want to express my gratitude for the active involvement and relevant feedback to drd. ing. Lucian Mogoşanu.

I would also like to thank as. dr. ing. Sergiu Mihai Costea for the interesting suggestions of research and for continuous innovation ideas. Finally, I like to acknowledge prof. dr. ing. Răzvan Victor Rughiniş for guiding me to the know-how people and for offering his view of this paper possible directions.

Abstract

Naming and identity services is a widely used concept implemented in multiple technologies. These systems should have three basic attributes: secure, decentralized and human-meaningful. Traditional technologies perform well but none of them provide all three properties. The Namecoin claims to fulfill all three attributes by building a distributed network but it is quite young, the documentation is sparse, there are unexplored features. The protocol is developed/used by a small community thus it lacks maturity and is unexplored enough.

We propose an exploring solution for filling the existing gaps. There is no tool available for discovering Namecoin's naming services so we build a prototype for exploring the Namecoin's domain name specifications. As a result of our exploration, we validate protocol technical implementation details. Our evaluation outlines that a small group of stackholders has the majority of Namecoin domain names so the Namecoin network is less trustworthy.

Contents

Acknowledgements	i
Abstract	ii
1 Introduction	1
1.1 State of the art	1
1.2 Motivations and Objectives	2
1.3 Contributions	2
1.4 Summary	3
2 Background	4
2.1 Naming and Identity Systems	4
2.2 Bitcoin Protocol	5
2.2.1 Bitcoin History	5
2.2.2 Bitcoin Competition	6
2.2.3 Proof-of-work	6
2.2.4 Blockchain	6
2.3 Discussion	7
3 Study of Namecoin Protocol	8
3.1 Namecoin Overview	8
3.2 Namecoin High Level Design	9
3.3 Namecoin Transactions	10
3.4 Namecoin's Naming Services Properties	11
3.5 Namecoin's Domain Name Services	12
4 Design	14
5 Implementation	17
5.1 Implementation Decisions	19
6 Evaluation	21
6.1 Discussion	22
7 Summary and Future Work	24
A Namecoin Records Distribution according to <code>expires_in</code> field	25
A.1 Namecoin Records relationship between <code>expires_in</code> and value fields	25

List of Figures

2.1	Zooko's triangle	5
2.2	Blockchain Structure using Merkle Tree	7
3.1	Transactions in Namecoin Network	11
4.1	Namecoin Explorer Design	14
4.2	Namecoin Explorer Block Diagram	16
5.1	Main panel - Search by name	18
5.2	Main panel - Dumping list of domains	19
5.3	Dump file content	20
5.4	Second panel - Advanced filtering options	20

List of Tables

3.1	Common Namecoin Naming and Identity Transactions	11
6.1	Common attributes from Namecoin record's value field	21
6.2	Namecoin records distribution	22

Chapter 1

Introduction

This thesis presents Namecoin technology specific information as an attempt to document and systematize the protocol's implementation details. In order to validate the information described in this paper we build an exploring tool that provides insights into the particularities of the Namecoin system. The structure of this paper outlines the challenges that the traditional naming systems are facing and how the blockchain resolves some of these issues focusing on the Namecoin solution.

In this chapter we present a brief description of the current state of the naming systems, the limitations of these technologies and what are the basic attributes accordingly to Wilcox-O'Hearn [21]. Furthermore, we present what motivated us to analyze this protocol and what are the main objectives that we want to achieve as well as our contributions to the Namecoin community.

1.1 State of the art

Naming services define relationships between real entities and names. The need for creating a name service is due to the difficulty of working with identity information such as addresses, IDs, etc. The large number of entities that are connected by the same network or that are using the same system made the identification mechanism quite complex so associating human-meaningful names with different identity information improved considerably the user experience.

One popular naming system is DNS . As described in RFC 1034 [12] the DNS has three components: domain name space, name servers and resolvers. The domain name space is represented by a tree where each node defines a domain that is under the management of an authoritative name server. Each server delegates responsibility for its sub-domains to other servers and so forth. Resolvers are responsible for extracting the identity data associated with name as a response to a client name request. ICANN is a nonprofit organization that manages namespaces of the internet. The main issue of using a central authority is having a single point of failure.

There are also other naming systems such as Tor services and OpenID. Tor services are used for anonymous communication. Tor aims to remove trace between user's identity and his online activity but there is no guarantee for anonymity. OpenID protocol allows us to use an account to sign in to multiple websites without having a username and a password for each website.

These naming systems have proven efficient but they failed to resolve the tree basic attributes of a naming system as stated by O'Hearn [21]. This problem is known as the Zooko's triangle and consists on building systems that solve only two of the three desired properties: decentralized, human-meaningful and secure. For example, DNS and OpenID is human-meaningful

and secure but it is not fully decentralized. They rely on third parties to perform properly. As a solution to this problem, Satoshi Nakamoto [13] developed the Bitcoin project that is based on a blockchain structure that offers fully decentralization. Satoshi describes the solution to the Byzantine Generals' Problem and how it can be implemented for achieving consensus on the Bitcoin distributed network. Starting from the Bitcoin project, other blockchains projects were developed such as Ethereum ¹, Namecoin ² or Ripple ³. The Namecoin project is the first fork of Bitcoin ⁴ that claims to solve all three properties of Zooko's triangle.

1.2 Motivations and Objectives

The thesis' motivations are based on thorough revision of the Namecoin system that lead us to the conclusion that the protocol documentation is sparse, it lacks maturity and it is unexplored enough. Developed by a small community, the Namecoin technology is still in a beginning phase and there are features not documented or incomplete specified. As a result of this research, we reach the conclusion that filling some of the gaps mentioned above is valuable and based on this we build some short and long term objectives:

- **Exploration phase:** discovering Namecoin technical implementation details by installing a Namecoin node on a local machine and testing different functionalities such as the registering a domain name, querying blockchain for naming and identity information for names already registered, running a local DNS server that resolves `.bit` domains, etc.
- **Validation phase:** building an argument for validating or invalidating that the Namecoin technology resolves the Zooko's triangle limitation. Namecoin technology claims to offer the first solution to Zooko's triangle but there are some security issues and unexplored features.
- **Evaluation phase:** analyzing current state of the Namecoin system by developing a tool to explore Namecoin domain name specifications. The Namecoin Explorer tool must extract data from the blockchain and displayed it in a user-friendly interface. The tool will also do some accounting based on the complete list of Namecoin records in order to provide insights into the system.

1.3 Contributions

The study of the Namecoin protocol was a tough challenge from the point of view of sparse documentation, reading Namecoin source code and following forum discussions. Firstly, chapter 3 systematizes Namecoin naming and identity information after researching and testing specifications from different sources. The Namecoin naming services features are validated by creating the Namecoin Explorer tool. The purpose of this application is to explore naming information stored in the blockchain in order to evaluate some Namecoin system parameters(usability) and available trading functionalities for currency exchange and naming transfer. This paper claims to contribute to the Namecoin technology systematization by testing and validating the information from forums, IRC channels, wiki pages and other websites. Another contribution is the Namecoin Explorer tool that offer insights into the blockchain using a user-friendly interface.

¹<https://github.com/ethereum/wiki/wiki/White-Paper>

²<https://github.com/namecoin/namecoin-core.git>

³<https://github.com/ripple>

⁴<https://github.com/bitcoin/bitcoin>

Based on the results extracted with this tool we offer an evaluation of the current state of the Namecoin system.

1.4 Summary

The thesis contains seven chapters that are structured as follows:

Chapter 1 describes the state of the art of this thesis by presenting the Namecoin protocol in a global context. Also, the introductory chapter contains the motivations and objectives of developing this paper alongside our contributions to the Namecoin community.

Chapter 2 provides insights into the basic concepts that build the development foundation of Namecoin naming services. Popular naming systems examples, brief description of blockchain structure and a discussion about the reasons of exploring the Namecoin protocol are the main aspects presented in the background chapter.

Chapter 3 aims to systematize the Namecoin protocol and to validate the Namecoin naming/identity services specifications accumulated during the research period. This chapter consists of a study of the Namecoin protocol that focus on naming services. The study represents our contribution to the Namecoin community.

Chapter 4 and chapter 5 presents the design and implementation details, respectively, of the Namecoin exploring tool developed in order to validate the statements done in chapter 3. Programming language decision, system architecture, brief discussion about the arguments of choosing a design beside other are discussed in these chapters.

Chapter 6 outlines Namecoin system parameters as well as usability information. The number of Namecoin records registered in a certain period of time, the list of the valid names registered and advanced filtering options are key elements for discovering who is using the Namecoin system and how is it used.

Chapter 7 builds the thesis conclusion by resuming our contributions and offers other research directions that are unexplored. The future work section is uncompleted thus we have to study in depth the information available in order to assure that is relevant to choose a certain path.

Chapter 2

Background

In this chapter we describe the naming systems particularities by offering popular examples, the Bitcoin basic concepts which apply to Namecoin protocol and we propose a discussion that outlines the motivation for dedicating chapter 3 to Namecoin study. In section 2.1 we describe traditional naming systems, existing protocols limitations ending with a brief presentation of the basic attributes of a naming system. The section 2.2 offers a high level view of the Bitcoin project focusing on relevant attributes for naming services. Finally, in section 2.3 we open a discussion about what motivated us to continue the exploration of Namecoin protocol.

2.1 Naming and Identity Systems

A naming system is a mechanism which offers an association between a name and an identification data object. The naming systems were developed as a solution to the exponential numeric growth of users of a network or system. This growth made the identification and authentication algorithms quite complex. The naming protocols faced security and privacy challenges thus the subject of user identification drawn the attention of computer scientists. O'Hearn [21] studied naming systems and reached the conclusion that traditional protocols have only two of the three desired attributes of a naming service. This conclusion is known by the community as the Zooko's triangle.

The three desirable attributes of a naming service are: human meaningful, secure and decentralized. The idea of placing these three attributes in a corner of a triangle can be described as follows. In a triangle (see Figure 2.1) only two corners can be connected by a single line so the three attributes will never be connected by the same line when placed in corners just like a system that couldn't focus on all attributes.

For example, OpenID which is a naming protocol that allows us to carry our identity across other websites without the need to register again focus only on security and human meaningful. The decentralized aspect is not solved so we have to trust at least one service provider. This protocol is used by software companies like Google, Twitter, Yahoo!, etc.

To solve the Zooko's triangle problem, the Namecoin protocol was developed and until now there is no other protocol that was able to focus on the three attributes. The Namecoin is a third-party identity provider, the blockchain, that can be used as an intermediate step between someone (maybe a service) that wants to know your identity and you. Until 2011, the design of a system that follows the Zooko's triangle was impossible. In order to have a naming system that is secure and easy to work with, a user should be able to choose name/identity strings

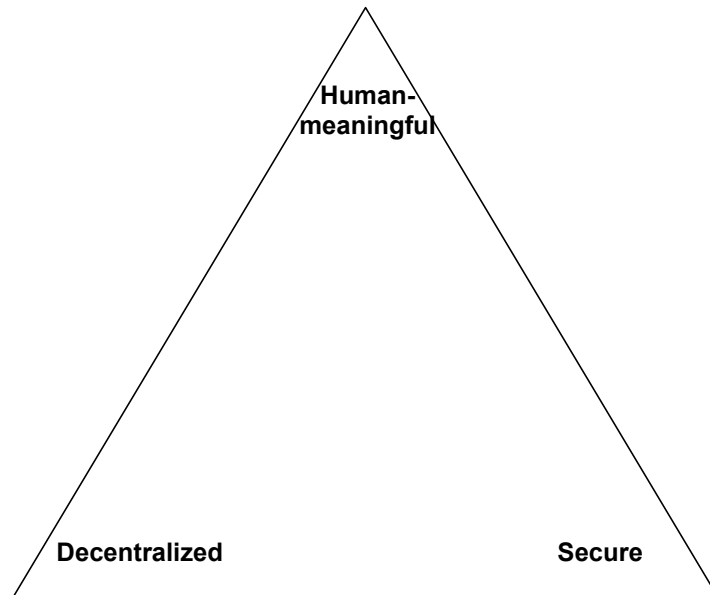


Figure 2.1: Zooko's triangle

that mapped with a certain value are uniquely identifying a network node and that no one can convince a node that other value for that pair is correct.

Although, the Namecoin system is not free of security attacks and errors this is the first system that solved the Zooko's triangle by using cryptocurrency technology. This protocol is limited to the human-legible phrases but has the advantage of making a name choice. More than that, the namecoin is the first alt-coin developed by forking the Bitcoin project. Namecoin implements both a merged mining and a decentralized DNS.

Building a system that is decentralized based on the Bitcoin project was done by creating another blockchain using a new genesis block. This system has also several specific transaction types in order to provide domain name system features. The only TLD used is `.bit`.

2.2 Bitcoin Protocol

The Bitcoin protocol is a blockchain based project that was developed as a solution to trusting the third party issue. Before introducing Bitcoin, we offer a brief presentation of the currency exchange technologies available and the Nakamoto's motivation for building Bitcoin [13]. This section outlines Bitcoin fundamental concepts such as Hashcash [3], Merkle Tree [11], Proof-of-work [10] and Byzantine Generals' Problem[8].

2.2.1 Bitcoin History

Electronic cash wouldn't be possible without the existence of a mechanism that offers a common language for financial message exchange. As a result, the Society for Worldwide Interbank Financial Telecommunication was funded. SWIFT network has standardized messages for financial information transfer. Some examples of ISO standards used for transfer of electronic

cash are ISO 15022 MT and ISO 20022 MX. The main issue of electronic cash protocols is the centralization. Satoshi Nakamoto is the creator of the first distributed system used for financial transfer by introducing the Bitcoin protocol. Antonopoulos [2] states that the Bitcoin is the evolution of electronic cash, payment systems, how money are transferred over the world and even how the business looks nowadays. Prior Bitcoin, electronic cash and digital currencies were available only in centralized systems.

2.2.2 Bitcoin Competition

Based on prior work, Bitcoin introduces proof-of-work[10] for cash minting, uses the blockchain[15] for data storing and a consensus protocol based on the Byzantine Generals' Problem[8] to eliminate double-spending.

The Bitcoin project is a popular blockchain based application but there are other blockchain projects that deserves to be brought to attention such as Ethereum, Namecoin or Ripple. The Ripple project started before Bitcoin in 2004 and it passed through many changes over time. Today, Ripple is a solution that enables banks and clients to exchange value. Similar with Bitcoin, Ripple is a distributed system with its own blockchain and a native currency called ripples. Ethereum is another blockchain based application that has its own Turing-complete programming language which makes it more powerful than Bitcoin by allowing users to create their applications on any programming language. Namecoin was the first fork of the Bitcoin project. It offers a domain name registry similar to the Internet's DNS. It is an alternative domain name service for the root domain `.bit`. The project has its own currency called namecoins (symbol NMC) which is used to pay transactions fee for registration, update and transfer of domains name.

One interesting aspect about Bitcoin technology is how it solves both the problem of double-spending and mining in a decentralized system. The protocol is developed on "sound cryptographic" principles that guarantees proof-of-work, proof-of-ownership and classic currencies attributes such as fungibility and scarcity.

2.2.3 Proof-of-work

The idea of using a proof-of-work system to legitimate the user of an application was used before in other software systems. For example, Hashcash [3] is a proof-of-work system introduced by Black that was invented to limit email spam and denial-of-service attacks. The idea is quite easy to follow: a legitimate sender needs to spend a reasonable amount of computing time in order to send an email. If a legitimate user sends a reasonable number of emails, a spammer wants to send thousands of emails making the spamming effort very expensive. The receiver's role is to validate the hash, which is quite easy. Hashcash is conceptually similar to the proof-of-work system used by Bitcoin miners.

Solving the problem of double spending using a proof-of-work system provides security guarantees for naming/identity registration and transfer.

2.2.4 Blockchain

The blockchain is a public ledger of all transaction sent through the network that is known by every one that holds a Bitcoin node. More precisely, each node has a copy of the blockchain in order to validate or invalidate a transaction. The majority of nodes decide if a transaction is valid and if so that transaction is added in the latest created block by one of the miners. The choice of using blockchain for data storing is related with the fact that the blockchain's deep history is immutable so it ensures the Bitcoin security.

The blockchain has no central authority for managing bitcoin flow along the bitcoin nodes network. This technology is recognized as the “fifth evolution” of computing, the missing trust interface of the Internet as stated by Laurence [9].

The blockchain structure has the following components as described in Figure 2.2:

- **The block** is an array of transactions recorded into a ledger. Depending on the blockchain the transaction has assigned a different type of value. Each block has its own hash and a merkle tree with hashed transactions. Storing transactions in a merkle tree is a design decision that saves disk space. Nakamoto [13] explains the reason why using a merkle tree makes disk space recover possible. The explanation for this is that "Once the latest transaction in a coin is buried under enough blocks, the spent transactions before it can be discarded to save disk space. To facilitate this without breaking the block's hash, transactions are hashed in a Merkle Tree, with only the root included in the block's hash. Old blocks can then be compacted by stubbing off branches of the tree. The interior hashes do not need to be stored" [13].

- **The chain** is simply a hash that connects two blocks in chronological order. The hash for a new block is generated based on the data that was in the previous block. The hash algorithm used by Bitcoin is Secure Hash Algorithm 256 that creates an unique (collision probability is negligible), fixed-size (256 bits) hash.

- **The network** is represented by full nodes. We can think of it as a computer that runs an algorithm for securing the network. Mining is expensive and it requires a lot of computer power so the blockchain algorithm rewards miners for their service. The payment is usually a currency or a token.

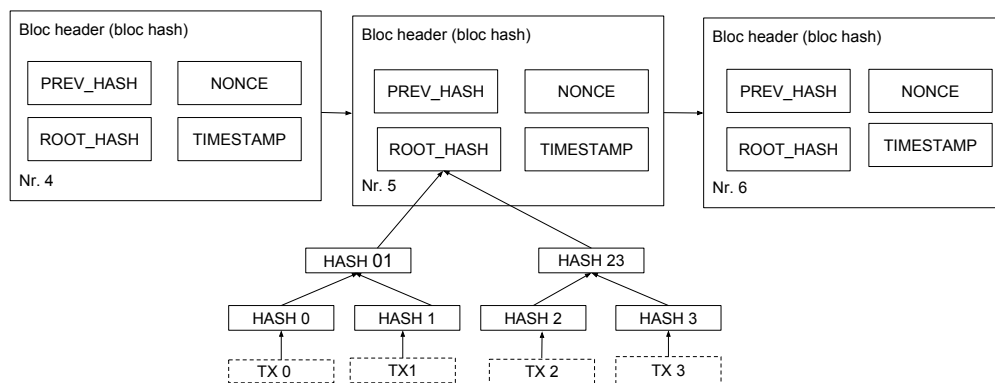


Figure 2.2: Blockchain Structure using Merkle Tree

2.3 Discussion

As a resume of this chapter we described the traditional naming systems properties and limitations. Furthermore, we introduced the Namecoin protocol as a solution to the trust issue that exists in traditional protocols. For a clear understanding of the Namecoin protocol and what are the security guarantees it provides we offer a description of the basic principals of Bitcoin protocol on which Namecoin is developed. Based on the information presented above we reach the conclusion that there is not clear yet how the Namecoin protocol provides naming and identity services thus we dedicated the next chapter to studying only Namecoin specifications.

Chapter 3

Study of Namecoin Protocol

The Namecoin protocol is an open-source project based on the Bitcoin project that offers naming services. The protocol provides the possibility to register a name/identity and to associate data with it. There are many naming services available but none of them offer the three basic properties: secure, decentralized and human-meaningful. The Namecoin protocol claims to provide all three properties. It is human-meaningful because the naming services involve registering a domain name `.bit` and attaching metadata for managing names and human readable keys that makes the protocol more user-friendly.

3.1 Namecoin Overview

The Namecoin protocol was forked from Bitcoin project and improved in order to offer naming services. All features available in Bitcoin are valid in Namecoin.

Namecoin protocol solves the trust issue of traditional naming systems by storing data in a blockchain structure in a decentralized network. Details on how the Namecoin protocol removes the third party element are discussed on chapter 2. Eliminating the third party is a feature of the Bitcoin protocol that was inherited by Namecoin project alongside all Bitcoin's properties. According to Raval [15], the blockchains are "secure by design and are an example of a distributed computing system with high Byzantine fault tolerance. Decentralized consensus has therefore been achieved with a blockchain". The decentralization property of Namecoin protocol is provided by the existence of the Namecoin network. Each Namecoin node has voting right so the validation is done by the majority of network entities by solving the Byzantine Generals' Problem [16]. The Namecoin is secure because it is very difficult to associate Namecoin nodes with real identities. The Zooko's triangle outlines that traditional naming systems can respect only two of the three desired properties mentioned above. Stated in Wilcox-O'Hearn [21] having all three properties was doubtfully until the release of Namecoin protocol.

The software exists in two versions: `namecoin-qt`(with GUI) and `namecoind`(without GUI). Both versions offer the same functionalities. After installation, start the namecoin node and wait until the blockchain is downloaded on disk. The size of the blockchain increases with every new block added to the Namecoin chain. After the blockchain was downloaded the Namecoin node is connected to the Namecoin network and can access different features of the protocol.

3.2 Namecoin High Level Design

As Bitcoin, the Namecoin is using a blockchain structure for storing transactions. The types of transactions that are store in the Namecoin chain are different from the Bitcoin. The protocol has its own blockchain apart from Bitcoin blockchain. The Namecoin protocol provides support for naming commands beside the common financial transactions available in Bitcoin protocol. The naming commands are Namecoin specific transactions. They are discussed in more details in section 3.3. The financial messages that exists in Namecoin blockchain are similar with Bitcoin transactions. The only difference is that the traded currency is namecoin (NMC).

Namecoin protocol uses an identity system based on addresses. Namecoin addresses are Base58 encoded hashes of receivers' public keys. All payments and records are made to addresses. Each Namecoin node is identified by a Namecoin address [19].

Namecoin specific features offer support for naming and identity registration using Namecoin records. Namecoin records are key-value pairs. In order to register key-value pair the user has to spend namecoins. Once the user owns their key-value pair the value field is available for 35999 blocks. With every new block added in the blockchain the `expires_in` field decreases by one block. When reaching zero blocks the value will expire and you need to spend Namecoins to update it. For the key field there is no expiration condition. Namecoin protocol specifies the possibility of organizing keys that have a common scope into groups. The keys can be grouped by different criteria. Each criterion defines a namespace. A key can belong to only one namespace or to none. The list of namespaces is unlimited. In order to register a key and associated it with a namespace, the key should have the following format `<namespace name>/<key name>`. Popular namespaces are `a` (application specific data), `d` (domain name specifications), `ds` (secure domain name), `id` (identity), `is` (secure identity), etc. Each user can create new namespaces or organize his keys based on already existing namespaces. Some examples of domain name specific information that are stored in the blockchain:

```

1 {
2   "name": "d/nf",
3   "value": "{\"map\":{\"\":\"94.23.252.190\"}, \"fingerprint\":
4     [\"69:16:99:8B:A7:62:6F:BE:2A:F6:AF:62:E4:DA:4D:8F:32:B8:52:28\"]}",
5   "expires_in": 34458
6 }
7
8 {
9   "name": "d/nh",
10  "value": "{\"for_sale\" : 1, \"website\" : \"http://blockchained.com
    \",
11  \"name\" : \"phelix\"}",
12  "expires_in": 34921
13 }
14
15 {
16  "name": "None",
17  "value": "http://PeerName.com",
18  "expires_in": 29917
19 }

```

3.3 Namecoin Transactions

The protocol offers support for two types of transactions: currency exchange transactions and naming transactions. Currency exchange transactions are similar with other cryptocurrencies transactions based on bitcoins, ethereum, etc. Naming transactions are Namecoin's specific features. They are represented by a group of transactions for identifier registration: `name_new`, `name_firstupdate` and `name_update`.

The registration mechanism uses two commands: `name_new` and `name_firstupdate`. A user who wants to register a new domain must initiate the command `name_new <name>`. After issuing this command the user will receive two hexadecimal numbers of different sizes. The larger hexadecimal number identifies the pre-order name and the shorter is a random value that identifies a user as the first one that pre-ordered that key. The `<rand>` is used in step two. This command has a small network fee that is destroyed by the protocol. The next command is `name_firstupdate <name> <rand> [<tx>] <value> [<toaddress>]` where `<name>` is the key that will be registered, `<rand>` is the shorter hexadecimal number, `<tx>` is optional and represents the previous transaction id, `<value>` is the data that a user associate with that key and `<toaddress>` is optional and is used for sending the registration to another node. This field `<value>` has no fixed structure and everyone can add different type of data such as digital certificates, files, contact information and so on. The `<value>` field must meet the JSON standard [17]. This transaction type must be done 12 blocks after the `name_new` transaction was generated and it is called only once for a certain domain.

The command `name_update` is used for updating the value field, for changing the ownership or for renewing the name. The command `name_update <name> <value> [<toaddress>]` has the following meaning: `<name>` is the name registered, the `<value>` is the data associated with that name and the `<toaddress>` is optional and represents the new owner address. If someone wants to buy a domain the only available solution is to search if the owner of the name left external contact information in the value field.

Although, this protocol provides naming and identity services, it can be used as a normal cryptocurrency exchange protocol. The protocol uses namecoins as exchange currency. Both registration transactions (except for `name_new`) and currency exchange transaction has a fee that gets to the miner that added the transaction in a block.

The network fee is a protection against an attacker that wants to register all alphanumeric names for financial reasons or for sabotaging the Namecoin network. The higher the number of names the more difficult is for miners to create new cryptocurrencies. This fee prevents attackers to buy all alphanumeric names by imposing a fee for each domain registered. The fee's value was discussed and for now it is fixed. There are different opinions on how to set the price of these transactions and the issue is subject to debates on IRC channels and forums. The command `name_new` has two fees: the network fee that is worth 0.01 NMC and the transaction fee that is worth 0.005 NMC. The protocol guarantees that the network fee is destroyed. Only the transaction fee goes to miners. The commands `name_firstupdate` and `name_update` have a 0.005 NMC fee that goes to miners.

There are also Namecoin transactions that meet other purposes such as identifier lookup (`name_filter`, `name_history`, `name_list`, `name_scan` and `name_show`), maintenance (`name_clean`), general commands for controlling (`getinfo`, `help`, `stop`) and network commands (`addnode`, `getaddednodeinfo`, `getconnectioncount`, `getnettotals`). The Namecoin protocol has also support for Bitcoin specific transactions related with blockchain, mining or wallet.

Beside the registration operations a Namecoin name owner can transfer the name/identity to another Namecoin address. The decision of transferring a name to another entity belongs to the name/identity owner. The common transferring process is an external trade between

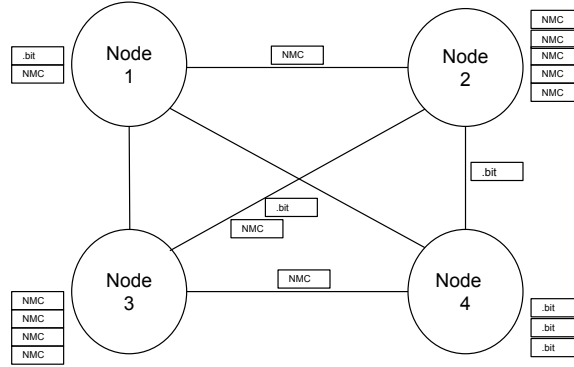


Figure 3.1: Transactions in Namecoin Network

a seller and a buyer. The protocol doesn't offer support for name/identity selling process. There is no possible way to contact a domain owner using Namecoin protocol. If the owner of a name/domain wants to sell that name it is common to include in the value field of the Namecoin record contact information such as email, name, website, etc. The Namecoin protocol provides the possibility of transferring a name by issuing the command `name_update` with parameter `<toaddress>`. The transfer of the name to another address is record in the blockchain and there is no way to alter the data stored in the Namecoin blockchain. Applying proof-of-work mechanism [10] when adding transactions into a block diminishes the chances to double spend a transaction. Selling a name is done by creating a `name_update` transaction and broadcasting it to all Namecoin nodes. The transaction has the address of the new name's owner. If the seller tries to double sell that domain, it needs to clone the blockchain, redo the block with that transaction and all blocks after it thus catching up is difficult. The scheme is called tampering and the change to succeed decreases exponentially as the chain gets longer.

Table 3.1: Common Namecoin Naming and Identity Transactions

Command	Syntax
<code>name_clean</code>	clean unsatisfiable transactions from the wallet
<code>name_firstupdate</code>	perform a first update after a <code>name_new</code> reservation
<code>name_history</code>	list all name values of a name
<code>name_list</code>	list my own names
<code>name_new</code>	pre-order a name
<code>name_pending</code>	list all pending name operations
<code>name_scan</code>	scan all names
<code>name_show</code>	show values of a name
<code>name_update</code>	update and possibly transfer a name

3.4 Namecoin's Naming Services Properties

In this section we present Namecoin naming properties and argue that the Namecoin protocol solves the Zooko's triangle [21]. The human-meaningful property consists in associating meaningful data with a name that is human readable, easy to remember and use. The Namecoin records are key-value pairs that allow registering a name using alphanumeric characters that is user-friendly and meaningfully for a user.

Naming systems aim to be decentralized by removing the third party entity. Namecoin is a decentralized system that solves the trust issue by using a network of blockchain nodes. The system achieves consensus under the limits of Byzantine fault tolerance [16]. The validation power is evenly distributed between Namecoin nodes. The majority decides if a transaction is valid or not. The decision mechanism is discussed in depth in chapter 2.

Having a naming system that is secure means preserving secret the identity of a .bit name owner. As stated in Bitcoin whitepaper [13] the privacy means "keeping the public keys anonymous". The information about the identity of a node is not stored in transactions so this removes traces to a certain entity. The network has access to details such as the amount of the transaction and the destination address. This property of the Bitcoin protocol works the same for Namecoin technology so this protocol protects the identity of a name owner as well as the identities involved in currencies exchange. Taking these arguments into consideration we reach the conclusion that the Namecoin's assertions involving Zooko's triangle [21] are valid.

3.5 Namecoin's Domain Name Services

Traditional naming systems failed to provide all three properties of Zooko's triangle [21]. Technologies used for naming are Tor services, DNSSec, I2P, OpenID, etc.

As stated in RFC 7686 [7] .onion domains are special TLD that can be accessed via Tor network. They are different from the internet DNS domains by relying upon Tor servers to resolve .onion domains. OpenID [5] is a authentication solution that allows webmasters to use a third party entity that handles login services. From the point of view of a user, this technology offers the possibility to access different websites using the same credentials. DNSSec [4] is an DNS extension that resolves DNS security issues such as authentication of denial of existence, data integrity, etc. I2P is technology for message communication between applications in a pseudonymous way via a secure network. Thus Tor services are decentralized, secure but not human-meaningful and OpenID, I2P, DNSSec are secured and human-meaningful but not decentralized.

The Namecoin's domain name services are decentralized, secure and human-meaningful. These properties apply to all names and identities registered in Namecoin network as stated in 3.4. The .bit domains can have internet addresses associated and can be used for hosting different websites. The .bit domains cannot be resolved by public DNS but there are other means of resolving their IP addresses. Namecoin records are key-values pairs that can provide domain name services. A key for a domain name belongs to namespace d and has the following syntax: d/<name>. The value for that key should match the DNS namespace specification so it must contain addressing information such as TLD or IPs. An example of a simple domain name registration:

```
name_new d/test
```

```
name_firstupdate d/test "ip":"5.5.5.5" <rand>
```

The protocol offers multiple solutions for domain name related issues. For example, the internet is trying to protect free-speech rights thus Namecoin is proposing a solution web censorship. The protocol is used also for accessing websites via .bit domains.

There are many solutions for enabling .bit domains browsing: replacing the DNS server with a local server, changing the DNS server with a OpenNIC Tier 2 DNS Resolvers or installing a browser plugin that connects to an external DNS. The simplest solution is to replace the local DNS server with an OpenNIC server. OpenNIC resolves multiple domains such as TLD, .bit, .emc, .coin, etc. According to [1], the OpenNIC project is a centralized server that stores .bit domains - IP address pairs from the Namecoin blockchain generating a DNS area

for different namespaces. Other solutions are proxy DNS (.surf) and browser extensions like PeerName, FreeSpeechMe and MeowBit. Each one has limitations and requires special environment conditions in order to function properly. There is also possible to have a local DNS server that queries the Namecoin blockchain that runs on the same machine and retrieve the IP address of the desired .bit domain. The NMControl project is an open-source project that add the .bit zone to the existing DNS. The NMControl server resolves only .bit domains. For enabling this feature the NMControl server must access a local Namecoin node blockchain. All other TLD are forwarded to the DNS server from the local network.

Chapter 4

Design

This chapter presents the design decisions and the main components of a Namecoin exploring tool. The Namecoin Explorer tool was developed in order to validate the naming statements described in section 3.5. The purpose is to help Namecoin node owners to discover, validate or explore information stored in the blockchain that are naming specific. The basic design of this tool is described in Figure 4.1. The operating principle is quite simple. The Namecoin Explorer behaves as client by sending requests to the Namecoin node which plays the server role. Based on the request sent the Namecoin node offers responses in JSON standard format.

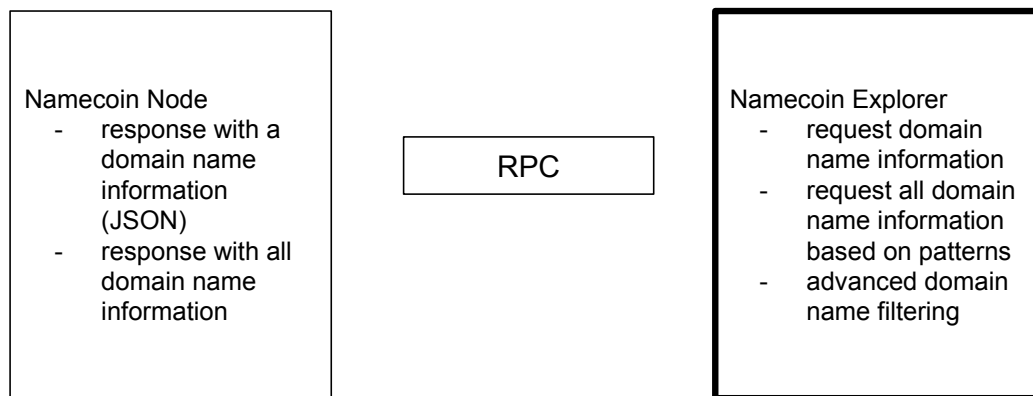


Figure 4.1: Namecoin Explorer Design

The design of this tool aims to achieve many purposes such as validating and testing naming information stored in blockchain, finding answers to Namecoin usability related scenarios, etc. The design was built taking into consideration all purposes of this tool. Before working on the tool design we thought about common use cases. For example, the value field of a name/identity may be of interest for a Namecoin user that wants to buy an already registered name. Another possible scenario is when a Namecoin user wants to see all the names registered in the blockchain that follows a certain pattern. There is common for a user to own more than one name grouped by a pattern that is meaningful for that user. Taking these details into consideration providing dumps of naming specific information that follows a certain pattern is useful and it represents a common use case. Also, the Namecoin Explorer aims to offer a clear and high level view on

the Namecoin network usability thus we propose computing name numbers that follows one or more criteria. This functionality is offered via advanced filtering options. The filtering criteria are email, IP addresses, websites and BM addresses. As a result of using these filters an user can compute the number of names/identities registered that added a certain attribute in the value field and the unique names/identities that stores the same value in that attribute.

In order to have access to Namecoin naming details an user must run a Namecoin node and have the blockchain downloaded on a local machine. As mentioned in section 3.1 there are two Namecoin versions. In order to be able to use our tool the Namecoin version installed must offer namecoin-cli features. The namecoin-cli binary provides access to Namecoin RPC [6]. The Namecoin client offers support to query the blockchain by exposing an API based on RPC. This API enables external applications to connect to Namecoin blockchain and send requests that are known by the RPC server and that can be handle by sending responses in a JSON [17] format.

Based on the information presented above there were multiple possibilities of extracting the data from blockchain. Our tool aims to explore naming information stored in the blockchain based on request types. We created an application that connects to a Namecoin node and gets the relevant data based on the type of the request sent. The Namecoin project offers an API to ask for some specific information about Namecoin records using RPC [18] mechanism. The Namecoin node has the server role providing data from blockchain to a client. The client is a Java application that displays both information for a `.bit` domain or for all `.bit` registered domains. As described in Figure 4.2 the tool has the following components:

- **Data Acquisition:** this component is responsible for sending and receiving messages from/to the Namecoin node. The data acquisition handles connection errors and triggers events for notifying the user about the state of the Namecoin client.
- **Parser:** this component main purpose is parsing the data received from the acquisition module and storing into objects that can be processed by the user interface module.
- **User interface:** this component displays the data in a user-friendly interface and offers different features such as requesting data for a particular name or for a group of names that respects a given pattern. Also the user interface offers filtering options for exploring the naming related data from blockchain.
- **Filtering:** this component aims to compute numeric results based on a certain filter that offers insights into the Namecoin system usability after evaluation.

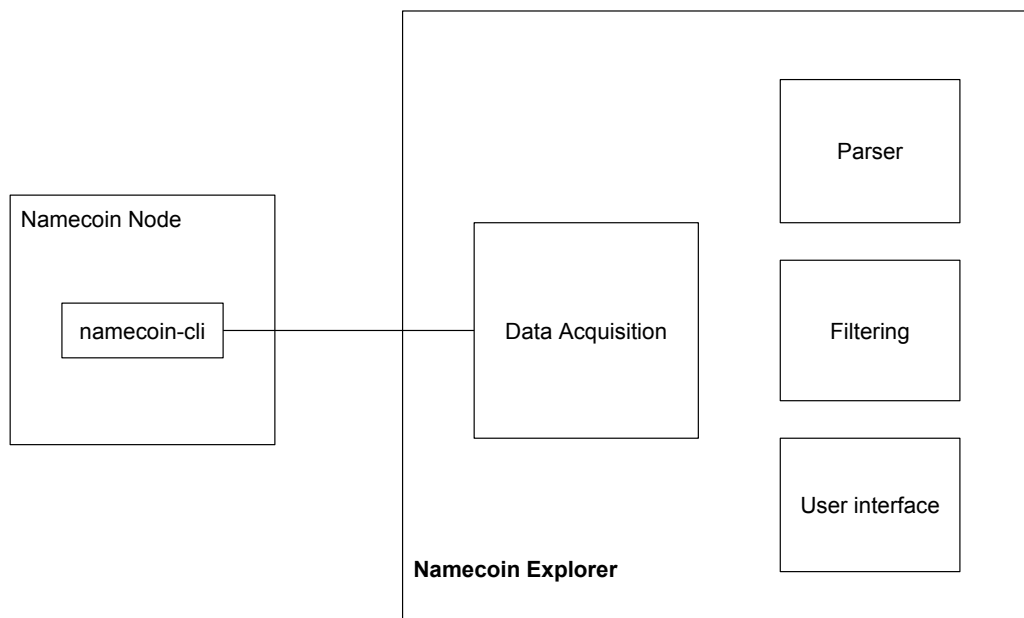


Figure 4.2: Namecoin Explorer Block Diagram

Chapter 5

Implementation

The Namecoin Explorer is a Java application that establishes a connection to the Namecoin node and sends different commands that the protocol recognize. The Namecoin Explorer has a client side that was implemented for receiving the results set from the Namecoin node.

In this chapter the Namecoin Explorer implementation details are described alongside the reasons for choosing a technology instead of another. The Namecoin Explorer was developed in Java programming language for simplicity reasons. The scope of this prototype was to be able to connect to a running Namecoin node, to bring domain name specific information and to compute numeric data that reveals the Namecoin network stakeholders and in which way is the network used. The use of Java technology was strongly connected with the fact this tool was created with the thought of allowing users with no technical background to run it without many requirements.

The Namecoin Explorer was split into three different parts in order to provide modularity and simplicity. The three parts are the following:

- **Namecoin Explorer's Connection to Namecoin Node** was implemented using an existing API provided by Namecoin protocol that allows other applications to connect to nodes by using RPC. As stated by [6] and [18] the RPC is an inter-process communication mechanism that allows the execution of a procedure in another address space as a local call without having the implementation details of the remote connection described explicitly in the code sections. The Namecoin Explorer tool initiates a RPC request to the Namecoin node to execute a certain procedure and waits that the server side sends the response to the Namecoin Explorer. The mechanism is similar to a client-server connection. The types of requests that the Namecoin Explorer sends to the Namecoin node are domain name specific information requests for a specified name or for a range of names that matches a certain pattern. Also the explorer displays information about the Namecoin node that runs on the local machine.

- **Namecoin Explorer's UI** was implemented using SWT [14] library. The UI component has two panels that provides different functions. The main panel contains information about the Namecoin node that runs on the machine and two available ways in which a user can request details about a Namecoin domain name. The first function provided in the main panel is searching for a certain domain `.bit` that has to be registered in the Namecoin network as described in Figure 5.1. The name of the domain should be complete and correct in order to receive all the details that the domain has associated with it. The second function is the possibility of dumping multiple domains specific information in a file on disk. This functionality is displayed in Figure 5.2. The dump contains a list of domain names and their specific information based

on the following two parameters: start name and max returned value. The dump list starts with the first domain whose name is the start name or comes first in lexicographic order from the list of valid domain names. Figure 5.3 contains Namecoin records that are stored in a dump file.

The second panel provides advanced filtering criteria as described in Figure 5.4 . This function was developed in order to collect numeric data about the registered domains and the content of the value field. There are four filtering criteria: ip, email, website and BM addresses. The second panel has functionalities that allows selecting one or multiple filtering criteria and displays the number of the domains that have the attributes selected in the value field and the number of unique values for the selected attributes from the main set of filtered domains.

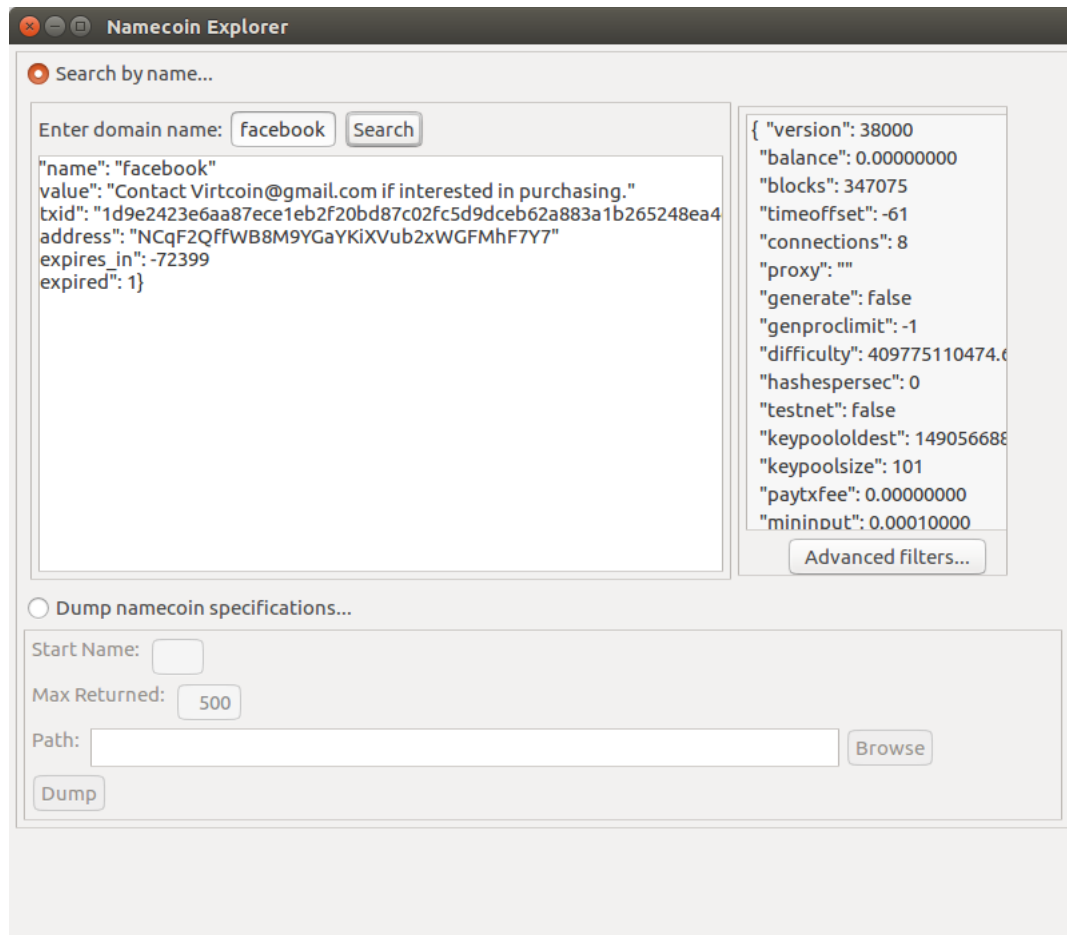


Figure 5.1: Main panel - Search by name

• **Implementing Advanced Filtering Technics** aims to extract Namecoin network information in order to discover the stakeholders involved and how are they using this protocol. This information is used in the evaluation process that is going to be discussed later on chapter 6. The panel displays the number of domains that have the selected attribute and the number of domains that have unique values for that attribute.

The data is transferred between Namecoin Explorer and Namecoin node using JSON [17]. The library used when working with JSON objects is org.json ¹ library. The decision of using this

¹<https://mvnrepository.com/artifact/org.json/json>

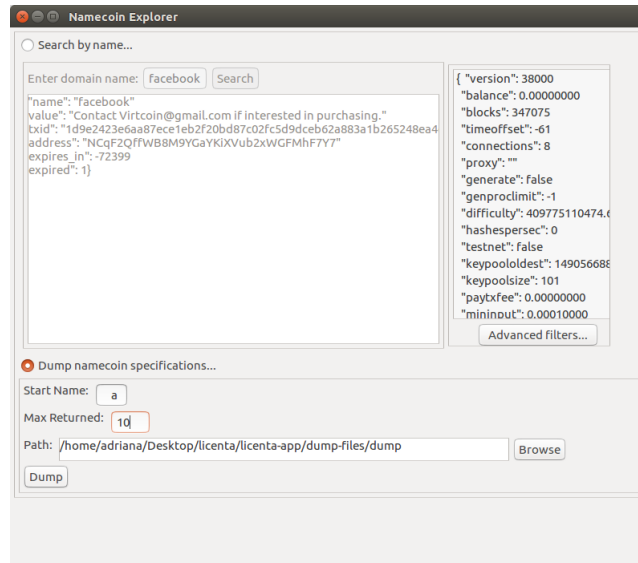


Figure 5.2: Main panel - Dumping list of domains

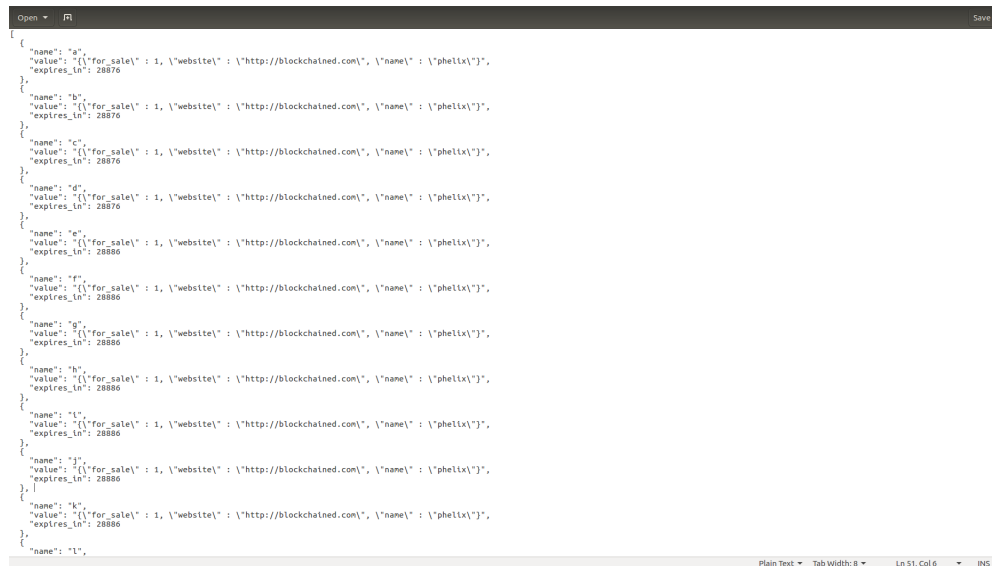
library is based on the following package's features: light-weight, language independent, data interchange format.

5.1 Implementation Decisions

The decision of implementing the Namecoin Explorer using RPC is based on a detailed research of the available exploring tools developed for Namecoin blockchain. There is no tool dedicated to extracting name specific information. All available explorers are web clients that display the transactions that are going to be added in the blockchain. The problem we faced at the beginning was determining the best extraction solution. After researching we discovered a project that creates a SQL database with all transactions stored in a blockchain. The project is called bitcoin-abe¹ and it is developed in order to extract data from Bitcoin blockchain. The bitcoin-abe parses the blockchain and extracts the data in a SQL database. In order to offer updated information about the data stored in the blockchain we had to recreate the database from scratch. The process of regenerating the database was difficult and time consuming so we stopped using it.

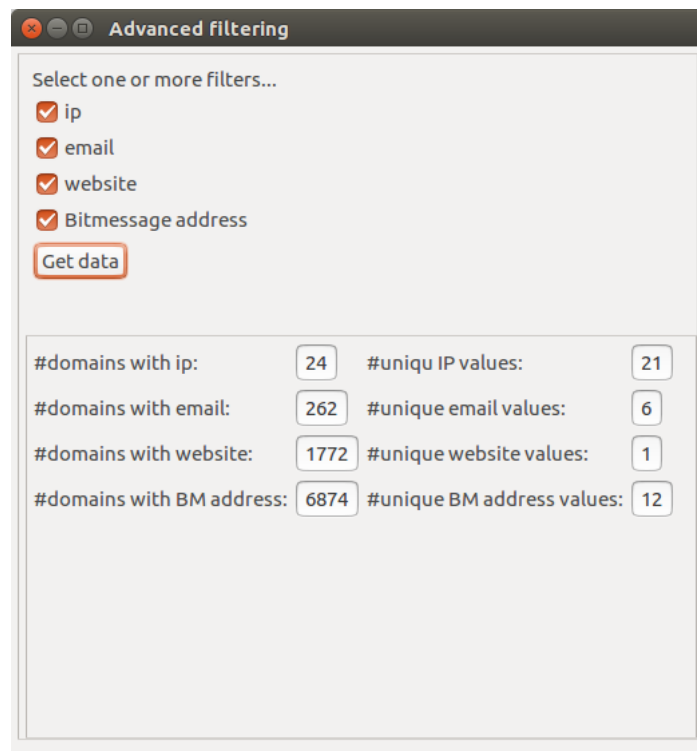
The tool has to provide updated information so we decided to lookup into a solution to connect to the Namecoin node and extract the data we need without intermediary. The Namecoin documentation is sparse so we started reading naming specification from source code. As a result we discovered that the protocol offers the possibility to extract naming specific data from blockchain using RPC technology. The Namecoin RPC server delivers responses to certain requests. Using the existing functionalities provided by Namecoin client we developed an external tool that makes requests based on the types of calls supported by the Namecoin node.

¹<https://github.com/bitcoin-abe/bitcoin-abe.git>



```
{
  "name": "a",
  "value": "{ \"for_sale\" : 1, \"website\" : \"http://blockchained.com\", \"name\" : \"phelix\" }",
  "expires_in": 26876
},
{
  "name": "b",
  "value": "{ \"for_sale\" : 1, \"website\" : \"http://blockchained.com\", \"name\" : \"phelix\" }",
  "expires_in": 26876
},
{
  "name": "c",
  "value": "{ \"for_sale\" : 1, \"website\" : \"http://blockchained.com\", \"name\" : \"phelix\" }",
  "expires_in": 26876
},
{
  "name": "d",
  "value": "{ \"for_sale\" : 1, \"website\" : \"http://blockchained.com\", \"name\" : \"phelix\" }",
  "expires_in": 26876
},
{
  "name": "e",
  "value": "{ \"for_sale\" : 1, \"website\" : \"http://blockchained.com\", \"name\" : \"phelix\" }",
  "expires_in": 26886
},
{
  "name": "f",
  "value": "{ \"for_sale\" : 1, \"website\" : \"http://blockchained.com\", \"name\" : \"phelix\" }",
  "expires_in": 26886
},
{
  "name": "g",
  "value": "{ \"for_sale\" : 1, \"website\" : \"http://blockchained.com\", \"name\" : \"phelix\" }",
  "expires_in": 26886
},
{
  "name": "h",
  "value": "{ \"for_sale\" : 1, \"website\" : \"http://blockchained.com\", \"name\" : \"phelix\" }",
  "expires_in": 26886
},
{
  "name": "i",
  "value": "{ \"for_sale\" : 1, \"website\" : \"http://blockchained.com\", \"name\" : \"phelix\" }",
  "expires_in": 26886
},
{
  "name": "j",
  "value": "{ \"for_sale\" : 1, \"website\" : \"http://blockchained.com\", \"name\" : \"phelix\" }",
  "expires_in": 26886
},
{
  "name": "k",
  "value": "{ \"for_sale\" : 1, \"website\" : \"http://blockchained.com\", \"name\" : \"phelix\" }",
  "expires_in": 26886
},
{
  "name": "l",
  "value": "{ \"for_sale\" : 1, \"website\" : \"http://blockchained.com\", \"name\" : \"phelix\" }",
  "expires_in": 26886
}
```

Figure 5.3: Dump file content



Advanced filtering

Select one or more filters...

- ☒ ip
- ☒ email
- ☒ website
- ☒ Bitmessage address

#domains with ip:	24	#unique IP values:	21
#domains with email:	262	#unique email values:	6
#domains with website:	1772	#unique website values:	1
#domains with BM address:	6874	#unique BM address values:	12

Figure 5.4: Second panel - Advanced filtering options

Chapter 6

Evaluation

In this chapter we offer an evaluation of the Namecoin protocol current state. It focuses on determining who are the Namecoin client owners and how are they using this protocol. Network reliability is determined by many parameters such as the number of users, the network infrastructure, the proof-of-work [10] that need to be done, etc. The naming service is only a component of the Namecoin system. We focus on exploring this component as there are many gaps related with the naming features offered by Namecoin protocol. A Namecoin record is a key-value pair (see section 3.2 for more details) that stores naming and identity specific information. The value field is configurable therefore a name owner can store anything as long as it respects the JSON standard format [17]. This field has a maximum size of 520 bytes. This limitation prevents overloading the Namecoin network and increasing significantly the Namecoin blockchain size.

Using the Namecoin Explorer tool we were able to extract all valid Namecoin records that are registered in the blockchain and analyze information related with the content of the value field. After analyzing different dump files that contains the value field of registered Namecoin names we reach the conclusion that we need to compute the number of attributes that appear in the value field. As a result we discovered that there were many domains that contain the same attributes: email, BM address, website, ip/map, for_sale, owner's pseudonym, etc. The list with the most used attributes of the value fields is available in Table 6.1. This data can suffer modifications with a new registration.

Table 6.1: Common attributes from Namecoin record's value field

Attribute stored in record value field	Number of occurrences	Number of records
Bitmessage address	6874	10214
website	1772	10214
email	262	10214
ip	24	10214

After reading samples of the dump files we observed that there different records have similar content in the value field. Taking this observation into consideration we decide to offer filtering options for the most popular attributes stored in the value field. The result of name filtering based on a certain criterion is displayed in Table 6.2. This table outlines the number of names that has a certain attribute and the number of the names that stores a unique content of that attribute.

Table 6.2: Namecoin records distribution

Attribute	Number of occurrences	Number of unique occurrences
Bitmessage address	6874	12
website	1772	1
email	262	6
ip	24	21

The Namecoin system allows both the possibility to trade NMC cryptocurrency and to register/use naming services for different purposes. Analyzing this data we discovered that:

- **BM addresses** are present in percentage of 67.2 of Namecoin records from which 0.17 percentage are unique.
- **Websites** are present in percentage of 17.3 of Namecoin records from which 0.05 percentage are unique.
- **Emails** are present in percentage of 2.5 of Namecoin records from which 2.29 percentage are unique.
- **IPs** are present in percentage of 0.23 of Namecoin records from which 87.5 percentage are unique.

This values outlines that the number of Namecoin records owners is small in relation with Namecoin records registered. Based on this observation we decided to explore into more details the usability of the Namecoin network and finding the reasons for registering a large number of Namecoin names/identities.

The next step was to determine when a name was registered based on `expires_in` field of Namecoin record. The record's ownership lasts 36000 blocks so the maximum value of `expires_in` field is 36000 blocks. We decided to group the record into 60 intervals of length 600 blocks. After splitting the names into intervals based on the number of blocks left until ownership expiration we computed the number of records for each interval and also the number of records with a unique value for a certain filter and discovered that the names that contain the same values for attributes from the value field of the Namecoin record have also the same number of blocks left until expiration. This information was determined by analyzing the blockchain naming records numeric computations from Appendix A.1.

6.1 Discussion

Another research direction is measuring the clustering of the Namecoin network. The percentage of Namecoin records that stores IP addresses information is quite small thus clustering analysis is not so relevant. Another interesting observation is that a significant percentage of Namecoin records have an Bitmessage address [20] attached to the value field. The Bitmessage protocol offers more privacy guarantees than displaying contact information such as websites, email accounts, etc. Adding the BM address in the value field of a Namecoin record doesn't compromise the security offered by the Namecoin network. From the financial perspective, investing in this cryptocurrency is a wise decision when taking into account that the protocol is based on the Bitcoin project and that all the security guarantees of Bitcoin extends to

Namecoin protocol too. There are also risks when investing in NMC currency especially Namecoin which lacks maturity, no updates or releases yet.

Chapter 7

Summary and Future Work

This chapter summarizes our contributions to the Namecoin community and proposes future research directions. This paper put together all the knowledge acquired and systematized it in chapter 3. At the end of this thesis we claim to achieve our objectives. We dedicated chapter 3 to the study of the Namecoin protocol filling the existing documentation gaps. In order to validate the naming information described in our study of the Namecoin protocol we build a tool that offers insights into the data stored in blockchain. The Namecoin Explorer tool performed well during our evaluation process by extracted naming data from blockchain and by filtering that information based on specific criteria. Using the tool's filtering feature we extracted naming services related information for evaluating the current state of the Namecoin system. In the end we offered a final forecast about the Namecoin system future based on trustworthy criteria.

There are still interesting Namecoin research directions that remained unexplored such as the geographic localization of top `.bit` domains (`google.bit`, `facebook.bit`, etc.). Determining the autonomous system (AS) of these domains, how top domains are distributed globally. Using the results obtained in the evaluation phase and computing the `.bit` domain global distribution we aim to offer new analyses and forecasts. What are the security guarantees offered by DNS limitations, what are the financial perspectives of a company that choose to enter the Namecoin cryptocurrency market are only a few questions that we propose to answer in the future.

Appendix A

Namecoin Records Distribution according to **expires_in** field

A.1 Namecoin Records relationship between **expires_in** and **value** fields

```
1 interval;number of names;number of unique names;attribute;
2 0;0;0;website;
3 0;0;0;Bitmessage address;
4 0;0;0;ip;
5 0;0;0;email;
6 1;0;0;website;
7 1;0;0;Bitmessage address;
8 1;0;0;ip;
9 1;4;1;email;
10 2;0;0;website;
11 2;0;0;Bitmessage address;
12 2;0;0;ip;
13 2;12;1;email;
14 3;0;0;website;
15 3;0;0;Bitmessage address;
16 3;0;0;ip;
17 3;0;0;email;
18 4;0;0;website;
19 4;42;1;Bitmessage address;
20 4;0;0;ip;
21 4;4;1;email;
22 5;0;0;website;
23 5;4;1;Bitmessage address;
24 5;0;0;ip;
25 5;0;0;email;
26 6;0;0;website;
27 6;0;0;Bitmessage address;
28 6;0;0;ip;
29 6;0;0;email;
30 7;0;0;website;
31 7;0;0;Bitmessage address;
```

```
32 7;0;0;ip;
33 7;0;0;email;
34 8;0;0;website;
35 8;0;0;Bitmessage address;
36 8;0;0;ip;
37 8;0;0;email;
38 9;0;0;website;
39 9;0;0;Bitmessage address;
40 9;0;0;ip;
41 9;0;0;email;
42 10;0;0;website;
43 10;0;0;Bitmessage address;
44 10;0;0;ip;
45 10;0;0;email;
46 11;0;0;website;
47 11;6;1;Bitmessage address;
48 11;0;0;ip;
49 11;0;0;email;
50 12;0;0;website;
51 12;2;1;Bitmessage address;
52 12;0;0;ip;
53 12;0;0;email;
54 13;0;0;website;
55 13;50;2;Bitmessage address;
56 13;0;0;ip;
57 13;0;0;email;
58 14;0;0;website;
59 14;0;0;Bitmessage address;
60 14;0;0;ip;
61 14;0;0;email;
62 15;0;0;website;
63 15;2;1;Bitmessage address;
64 15;0;0;ip;
65 15;0;0;email;
66 16;0;0;website;
67 16;0;0;Bitmessage address;
68 16;0;0;ip;
69 16;0;0;email;
70 17;0;0;website;
71 17;8;1;Bitmessage address;
72 17;0;0;ip;
73 17;0;0;email;
74 18;0;0;website;
75 18;0;0;Bitmessage address;
76 18;0;0;ip;
77 18;2;1;email;
78 19;0;0;website;
79 19;2;1;Bitmessage address;
80 19;0;0;ip;
81 19;6;1;email;
82 20;0;0;website;
83 20;0;0;Bitmessage address;
84 20;0;0;ip;
```


85 20;0;0;email;
86 21;0;0;website;
87 21;2035;2;Bitmessage address;
88 21;0;0;ip;
89 21;0;0;email;
90 22;0;0;website;
91 22;26;1;Bitmessage address;
92 22;0;0;ip;
93 22;4;1;email;
94 23;0;0;website;
95 23;36;1;Bitmessage address;
96 23;0;0;ip;
97 23;0;0;email;
98 24;0;0;website;
99 24;0;0;Bitmessage address;
100 24;0;0;ip;
101 24;14;1;email;
102 25;0;0;website;
103 25;1260;1;Bitmessage address;
104 25;0;0;ip;
105 25;20;1;email;
106 26;0;0;website;
107 26;0;0;Bitmessage address;
108 26;0;0;ip;
109 26;0;0;email;
110 27;0;0;website;
111 27;0;0;Bitmessage address;
112 27;0;0;ip;
113 27;0;0;email;
114 28;0;0;website;
115 28;2;1;Bitmessage address;
116 28;0;0;ip;
117 28;0;0;email;
118 29;0;0;website;
119 29;2;1;Bitmessage address;
120 29;0;0;ip;
121 29;0;0;email;
122 30;0;0;website;
123 30;0;0;Bitmessage address;
124 30;0;0;ip;
125 30;0;0;email;
126 31;0;0;website;
127 31;0;0;Bitmessage address;
128 31;0;0;ip;
129 31;0;0;email;
130 32;0;0;website;
131 32;2;1;Bitmessage address;
132 32;0;0;ip;
133 32;0;0;email;
134 33;0;0;website;
135 33;3222;2;Bitmessage address;
136 33;0;0;ip;
137 33;60;1;email;

138 34;0;0;website;
139 34;30;2;Bitmessage address;
140 34;0;0;ip;
141 34;76;1;email;
142 35;0;0;website;
143 35;6;1;Bitmessage address;
144 35;0;0;ip;
145 35;0;0;email;
146 36;0;0;website;
147 36;0;0;Bitmessage address;
148 36;0;0;ip;
149 36;0;0;email;
150 37;0;0;website;
151 37;0;0;Bitmessage address;
152 37;0;0;ip;
153 37;0;0;email;
154 38;0;0;website;
155 38;4;2;Bitmessage address;
156 38;0;0;ip;
157 38;0;0;email;
158 39;0;0;website;
159 39;0;0;Bitmessage address;
160 39;0;0;ip;
161 39;2;1;email;
162 40;0;0;website;
163 40;1;1;Bitmessage address;
164 40;0;0;ip;
165 40;0;0;email;
166 41;0;0;website;
167 41;2;1;Bitmessage address;
168 41;0;0;ip;
169 41;0;0;email;
170 42;0;0;website;
171 42;0;0;Bitmessage address;
172 42;0;0;ip;
173 42;0;0;email;
174 43;0;0;website;
175 43;14;1;Bitmessage address;
176 43;0;0;ip;
177 43;6;1;email;
178 44;0;0;website;
179 44;0;0;Bitmessage address;
180 44;0;0;ip;
181 44;0;0;email;
182 45;0;0;website;
183 45;16;1;Bitmessage address;
184 45;0;0;ip;
185 45;0;0;email;
186 46;0;0;website;
187 46;2;1;Bitmessage address;
188 46;0;0;ip;
189 46;0;0;email;
190 47;0;0;website;

191 47;0;0;Bitmessage address;
192 47;0;0;ip;
193 47;2;1;email;
194 48;0;0;website;
195 48;28;1;Bitmessage address;
196 48;4;1;ip;
197 48;20;1;email;
198 49;0;0;website;
199 49;0;0;Bitmessage address;
200 49;0;0;ip;
201 49;0;0;email;
202 50;0;0;website;
203 50;0;0;Bitmessage address;
204 50;0;0;ip;
205 50;0;0;email;
206 51;0;0;website;
207 51;0;0;Bitmessage address;
208 51;0;0;ip;
209 51;0;0;email;
210 52;0;0;website;
211 52;4;1;Bitmessage address;
212 52;0;0;ip;
213 52;0;0;email;
214 53;0;0;website;
215 53;18;1;Bitmessage address;
216 53;0;0;ip;
217 53;0;0;email;
218 54;0;0;website;
219 54;0;0;Bitmessage address;
220 54;0;0;ip;
221 54;0;0;email;
222 55;0;0;website;
223 55;4;1;Bitmessage address;
224 55;0;0;ip;
225 55;22;1;email;
226 56;0;0;website;
227 56;12;2;Bitmessage address;
228 56;0;0;ip;
229 56;8;1;email;
230 57;922;1;website;
231 57;34;1;Bitmessage address;
232 57;0;0;ip;
233 57;0;0;email;
234 58;850;1;website;
235 58;0;0;Bitmessage address;
236 58;0;0;ip;
237 58;0;0;email;
238 59;0;0;website;
239 59;0;0;Bitmessage address;
240 59;0;0;ip;
241 59;0;0;email

Bibliography

- [1] .bit tld. <https://wiki.opennic.org/opennic:dot:bit>, 2017. Accessed: 2017-06-19.
- [2] Andreas M. Antonopoulos. *Mastering Bitcoin 2nd Edition*. 2017.
- [3] Adam Back. Hashcash-a denial of service counter-measure. 2002.
- [4] D. Eastlake. Domain name system security extensions. RFC 2535, March 1999.
- [5] Eric Eldon. Single sign-on service openid getting more usage. <https://venturebeat.com/2009/04/14/single-sign-on-service-openid-getting-more-usage/>, April 14, 2009. Accessed: 2017-06-23.
- [6] Network Working Group. Specifies version 1 of onc rpc. RFC 1057, June 1988.
- [7] A. Muffett J. Appelbaum. The ".onion" special-use domain name. RFC 7686, October 2015.
- [8] Robert Shostak Lamport, Leslie and Marshall Pease. *The Byzantine generals problem*. ACM Transactions on Programming Languages and Systems (TOPLAS) 4.3, 1982.
- [9] Tiana Laurence. *Blockchain For Dummies*. 2017.
- [10] Ben Laurie and Richard Clayton. *Proof-of-work proves not to work; version 0.2*. Workshop on Economics and Information, Security, 2004.
- [11] R. C Merkle. *A Digital Signature Based on a Conventional Encryption Function*. Advances in Cryptology - CRYPTO '87. Lecture Notes in Computer Science, 1988.
- [12] P. Mockapetris. Domain names - concepts and facilities. RFC 1034, November 1987.
- [13] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *The Cryptography Mailing list at metzdowd.com*, 2008.
- [14] Steve Northover and Mike Wilson. *SWT: The Standard Widget Toolkit, Volume 1*. Addison-Wesley Professional, Jun 28, 2004.
- [15] Siraj Raval. *Decentralized Applications: Harnessing Bitcoin's Blockchain Technology*. O'Reilly Media, 2016.
- [16] Nick Szabo. Secure property titles with owner authority. <http://nakamotoinstitute.org/secure-property-titles/>, Originally published in 1998. Accessed: 2017-06-22.
- [17] Ed. T. Bray. The javascript object notation (json) data interchange format. RFC 7159, March 2014.
- [18] R. Thurlow. Specifies version 2 of onc rpc. RFC 5531, May 2009.
- [19] Ralf Merkle US patent 4309569. *Method of providing digital signatures*. The Board Of Trustees Of The Leland Stanford Junior University, 1982.

-
- [20] Jonathan Warren. Bitmessage:a peer-to-peer message authentication and delivery system. <https://bitmessage.org/bitmessage.pdf>, November 27, 2012. Accessed: 2017-06-26.
- [21] Zooko Wilcox-O’Hearn. Distributed, secure, human-readable: Choose two. <https://web.archive.org/web/20011020191610/http://zooko.com/distnames.html>, 2001. Accessed: 2017-06-14.