FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Visualization of Bitcoin Transactions for Forensic and Security Analysis

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Visual Computing

eingereicht von

Alexandra Mai, BSc
Matrikelnummer 01125691

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Priv.Doz. Dr. Edgar Weippl

Wien, 30. September 2017

Alexandra Mai	Edgar Weippl

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.ac.at

FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Visualization of Bitcoin Transactions for Forensic and Security Analysis

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Visual Computing

by

Alexandra Mai, BSc
Registration Number 01125691

to the Faculty of Informatics

at the TU Wien

Advisor: Priv.Doz. Dr. Edgar Weippl

Vienna, 30th September, 2017

Alexandra Mai

Edgar Weippl

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.ac.at

Erklärung zur Verfassung der Arbeit

Alexandra Mai, BSc
Lehmanngasse 25/4/13

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 30. September 2017

____
Alexandra Mai

Acknowledgements

I want to express my thankfulness to Dr. Katharina Krombholz. She continuously provided input and comments to make this thesis possible. My gratitude also goes to my advisor Privatdoz. Dipl.-Ing. Mag. Dr. Edgar Weippl.

Furthermore, I want to thank my colleague Matthias for the collaboration on the prototype. I also want to thank my boyfriend Thomas for moral support.

My sincere thanks go to SBA Research, my employer. They gave me the opportunity to work on this topic during my internship, allowing me to continue with it as an employee and supported my work.

Last but not least I want to thank my parents Verena and Günther and my sister Astrit for supporting me throughout my studies.

Kurzfassung

„Zeit für Plan₿“ – Aufgrund der steigenden Unzufriedenheit mit zentralgesteuerten Währungen und der enormen Wertsteigerung ihres Kurses gewinnt die Kryptowährung Bitcoin immer mehr an Bedeutung. Bitcoin basiert auf einem Peer-to-Peer System welches die durchgeführten Transaktionen in einer dezentralen Datenbank, der sogenannten enquoteBlockchain, für jeden online zugänglich speichert. Das Ziel dieser Arbeit ist es, einen neuartigen Weg zu finden, die Transaktionen des Bitcoinsystems visuell und interaktiv darzustellen. Mit Hilfe der Visualisierung wird es Analytikern und Forensikern ermöglicht, einen Einblick in aktuelle und vergangene Transaktionen zu bekommen. Hierbei liegt ein besonderes Augenmerk auf der Zeitkomponente, um zeitliche Zusammenhänge schneller erfassen zu können.

Der in dieser Arbeit vorgestellte Prototyp (Plan₿) kombiniert Visualisierungs und Clusteringtechniken sowie statistische Wahrscheinlichkeiten, um dem Benutzer ein breites Spektrum an Informationen bieten zu können. Plan₿ ist eine webbasierte Applikation, um sicherheitsbezogene Fragen durch visuelles Feedback genauer beantworten zu können (z.B.: Pseudoanonymität, Diebstahl etc). Um eine reibungslose Interaktion mit den Daten zu ermöglichen wird eine GPGPU Methode eingesetzt welche die Informationen kontinuierlich lädt, diese wurde von Matthias Gusenbauer implementiert. Des Weiteren wird in dieser Arbeit auch eine Evaluierung der Applikation vorgenommen und auf die Stärken und Schwächen dieser eingegangen. Abschliesend werden die weiteren Pläne für die Nutzung und Weiterentwicklung des Prototypen erläutert.

Abstract

"Time for Plan₿" – the increasing dissatisfaction with centralized currencies and the enormous increase in value caused the cryptocurrency Bitcoin to gain in importance over the last few years. Bitcoin is a peer-to-peer system with a publicly available decentralized database, the so-called "Blockchain", where all transactions are stored. The main goal of this thesis is to explore a new way to visualize this transaction information in an interactive form. Our visualization gives a better understanding of the (pseudo-)anonymity and security of Bitcoin and enables analysts to get insight into past and current transactions. A special emphasis is placed on the time component to better understand chronological correlations. This helps forensic and security analysts to understand cash flows.

The prototype presented in this thesis (Plan₿) combines visualization techniques, clustering algorithms and statistical probabilities to provide exact information. Plan₿ is a web-based application to extract security-related visual feedback of the Blockchain. To ensure a smooth interaction with the data, a GPGPU method implemented by Matthias Gusenbauer is used to load the information constantly in the background. Further, this thesis evaluates the prototype, highlights strengths and weaknesses and describes future work.

Contents

CHAPTER 1

Introduction

The Blockchain of Bitcoin currently contains over 150 GB (February 2018 [15]) of transaction data and is publicly accessible to everyone. It is getting larger every minute, as payments with bitcoins have become more and more popular. The popularity of Bitcoin can be also seen in the tremendously increase of market value during 2017, which is directly related to the demand. The value increased by over 800% to a value of 8,241 USD for 1 BTC on the 8th of Februar 2018. According to Khairuddin et al. [55] the increasing usage of bitcoins can be ascribed to three main causes:

First, Bitcoin is assumed to have a huge societal and future financial impact as it is seen as the money revolution of this century. The second motivation is the empowerment of users and the circumvention of traditional banks, which is an important factor for people using bitcoins. This empowerment is achieved by the network structure of Bitcoin which is decentralized, open source and (pseudo-)anonymous. The last factor mentioned by Khairuddin et al. is the perceived real value, which participants claim to be like that of gold.

1.1 Problem Statement

Although many users think that Bitcoin is per se anonymous [57], all transactions are publicly stored in the Blockchain, and with different methods it is possible to connect (at least some) Bitcoin addresses with real persons/companies as proven by various researchers in recent years (for example Hirshman et al. [52], Reid et al. [66] and Androulaki et al. [34]). However, it is still hard for investigators to track anonymized cash flows due to mixing services, multiple wallets and other anonymization mechanisms used for Bitcoin transactions.

One important goal for the forensic and security analysis is to reveal correlations between transactions and filter (suspicious) activities through a time-based overview. As there is a huge variety of information hidden in the Blockchain, it is an important goal to provide security experts with insights into transactions over time with respect to specific addresses and to furthermore provide details of these transactions (amount of bitcoins transferred, hash values, etc.) and their further connections to other wallets/addresses.

A useful analysis tool for security and forensic experts, therefore, has to fulfill two major aspects: On the one hand, the interactivity is important, and on the other hand, the availability of the complete data of the Blockchain, which enables the user to see the relations between the transactions without any short-cuts. Currently there are different tools available which deal with either of the two aspects (see for example chapter 3).

1.2 Aim of this Work

Our main motivation is to propose a novel way to analyse transaction data of the Bitcoin ecosystem and implement an application prototype (Plan₿), which combines the interactivity and the availability of the data and gives the user the possibility to get a visual insight into the complex world of bitcoins. Our tool will enable forensic and security analysts to track masked cash flows and search for specific patterns and addresses to see their positions in the overall network. Furthermore it is possible to search for specific connections between two different Bitcoin adresses to see if there are any connections.

Bitcoin is pseudonymous, as the transactions are secured with public key cryptography. As mentioned above, Bitcoin addresses are hard to link to a specific user by name as many Bitcoin users employ more than one Bitcoin address or other masking techniques. Caused by the (pseudo-)anonymous nature of Bitcoin, it is prone to be used for criminal or fraudulent purposes [13]. We want to focus on the visualization of different Bitcoin addresses (to filter over time) as well as their correspondence to specific transactions. Especially reoccurring payments and extremely high and very low amounts are of special interest, if they both arrive at one specific address in the end.

The display of relations between different addresses and transactions gives the analysts insight in transaction habits as for example recurring payments. Furthermore, the user is provided with additional information as the direction of the cash flow and the involved parties in case mixing services or other anonymization tools were used. Thanks to the visualized information, it should be easier to keep track of certain suspicious users, transaction behaviours and the probability that anonymity tools were used (e.g. mixing services, multiple wallets).

Our prototype is a web-based application providing multiple views for the user to get an optimal insight into the world of Bitcoin transactions. On the one hand, the user is enabled to search for specific addresses and their relations within the network, and on the other hand, he/she is provided with a time-based search function. To ensure the usability, an expert for usable security and an expert for information visualization were involved to provide feedback regarding the prototype. The source code of the prototype is open source to give analysts the chance to use it or to expand it further.

1.3 Structure of this Work

In chapter 2 an overview of Bitcoin and its network, the Blockchain, is provided. In this chapter the basic systematics of this cryptocurrency and the most important transaction types are discussed.

Chapter 3 describes related cryptocurrencies and alternatives to Bitcoin. Furthermore, visualization tools which have the same or similar goals are discussed and compared.

The abstract as well as used technology is described in chapter 4, highlighting the main methods which are used in developing the prototype Plan₿.

In chapter 5 the implementation of the web-based prototype is presented. It explains the usage of existing viusalization techniques and how they are integrated in the framework.

The final chapter 6 highlights the advantages and flaws of the prototype and compares it to existing visualization frameworks. Furthermore, the chapter provides an outlook to future work and some concluding thoughts.

CHAPTER 2

Bitcoin and Blockchain

New payment systems gained a lot of attention in recent years and many different innovative systems for exchanging money have been released [45]. Those payment systems are mostly built upon digital platforms enabling users to access their money flexible in time and from all over the world in a very fast way. There are many factors affecting the success of a digital payment system, as Van der Heijden [73] stated.

Two important factors influencing a wide consumer acceptance for cryptocurrencies are security and trustworthiness. As many systems relate on one single trust unit a trend towards decentralized systems can be observed, explaining the major interest in Bitcoin and similar virtual currencies. Cryptocurrencies in general are a monetary unit using encryption to control, verify and transfer financial means. They have no intrinsic value as the (market) value depends on the users demand. Furthermore they have no physical form as their existence is only in the digital network. The last major characteristic is the decentralization as no central bank system controls the transactions, but the network attendees do.

Bitcoin was the worldwide first known and used decentralized cryptocurrency on the market [32]. The background as well as well as the most important technologies of Bitcoin and its subjacent network, the Blockchain, are described in the following sections. They provide the reader with basic knowledge for the further chapters, especially the detailed explanation of the prototype.

2.1 History

The name and the cryptocurrency payment system of “Bitcoin” was first mentioned in the paper “Bitcoin: A peer-to-peer electronic cash system”, published under the pseudonym Satoshi Nakamoto in 2008 [63]. It is a peer-to-peer system, which makes Bitcoin the first decentralized digital currency. The basis of this cryptocurrency is the Blockchain, a continuously growing sequence of transactions. This network works with a decentralized cryptographic proof of concept instead of trust and one single institution managing the transactions, which is used in normal (centralized) currency/bank systems. After the paper of Satoshi Nakamoto was published, Bitcoin was released as an open source software on the platform sourceforge. The first block, also called genesis block, was created before its official release without the network and was stored in the Blockchain on the 3rd of January 2009. The block is hard-coded in the source code with a block reward of 50 BTC [19]. The first official mentioning in a mailing list was on the 11th of January 2009. The first block with an actual transaction (not only the transaction of newly generated currency) was created on the 12th of January 2009 with the block number 170. The recipient of this transaction was Hal Finney, receiving a sum of 10 BTC [44] . After its launch it only slowly gained attention and, therefore, also the exchange value to USD was very low (see Figure 2.1).

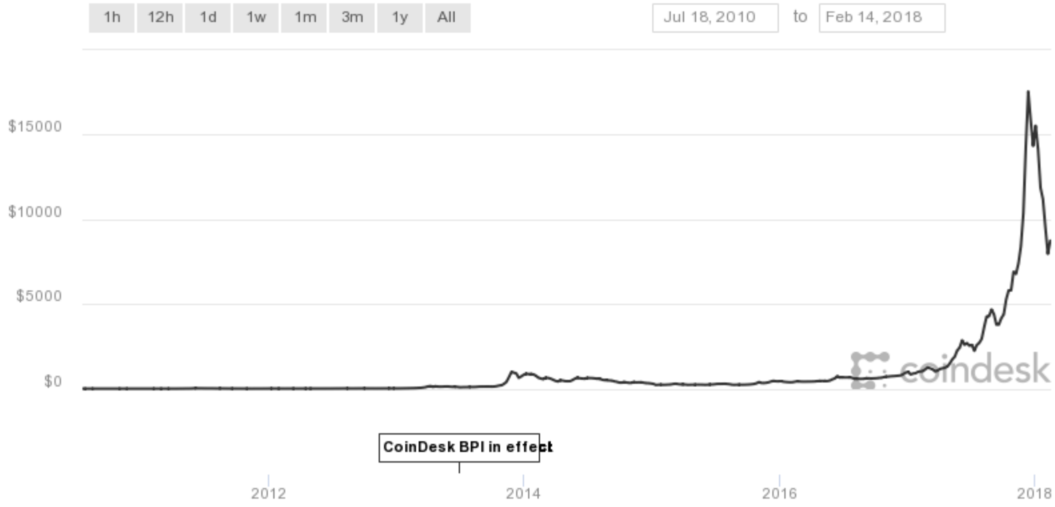


Figure 2.1: Bitcoin price development from mid 2010 to 1st September 2017 [11]

In the beginning, pioneers invested and tried Bitcoin because of the cryptographic and decentralized aspect. Afterwards, the interest slowly rose and increased tremendously first in 2013, with a market value above 1000$. This huge raise in value got Bitcoin an increasing attention of potential users. However, it also sparked criticism because of its value fluctuation [50]. It got discredited as a speculative asset instead of being accepted as a serious new currency. This was also the reason why regulatory responses increased massively. The responses were diverse throughout the world as some countries legally allowed Bitcoin and determined it as an official currency and others prohibited it due to a lack of legal framework. This and the splitting of Bitcoin into two networks led to a price fall of more than 50% and shortly even trade stagnation.

Figure 2.2 shows the legal status of Bitcoin in 2014. The countries marked in green have legalized Bitcoin, either together with legal guidelines and tax as properties or as private money. Yellow displays countries that have certain restrictions for the usage of bitcoins and its transactions. Countries coloured in red prohibited the usage of bitcoins (in 2014 Thailand and Iceland), for the rest of the countries, which are grey, there was no data available.

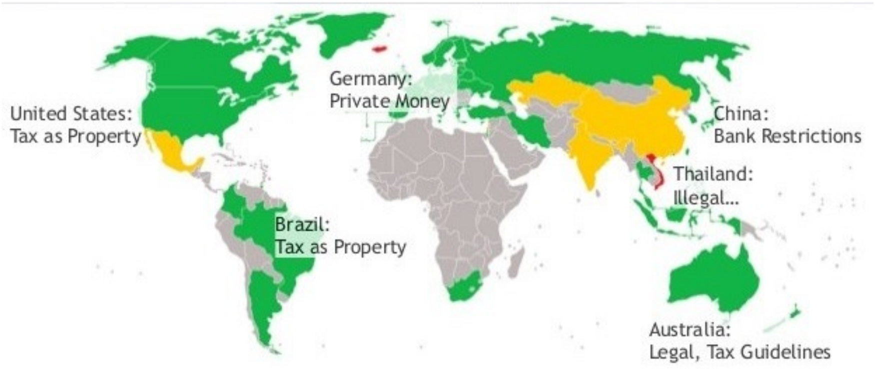


Figure 2.2: Bitcoin legal status worldwide in 2014 [9]

The number of businesses accepting Bitcoin increased between 2013 and 2015 to an estimated amount of 160,000 merchants. Since late 2015 the value of Bitcoin therefore managed to increase again, growing slowly back to its value of 1000$ at the end of 2016. In the beginning of 2017 the value of Bitcoin sky rocket within months. Figure 2.3 shows the tremendous gain in value in the eleven months of 2017 and its fluctuations in December. Its highest gain was over 1900%, however, afterwards its value dropped to a price of 8500 USD in the beginning of February 2018, which lead to much publicity and a great research interest.

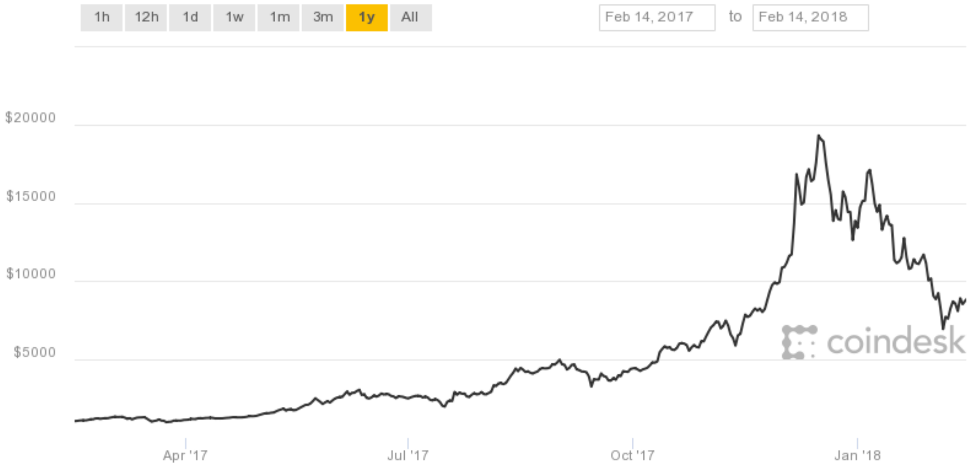


Figure 2.3: Bitcoin price development from 1st January to 14th Feburary 2018 [11]

2.2 Blockchain

The basis of Bitcoin is *Blockchain*, a decentralized database system. It stores the information publicly and permanently in an unalterable way to prevent double spending and (unwanted) modifications of already performed transactions. A blockchain consists of several individual blocks which contain one or more transactions. Each block is validated by a consensus rule system. For more information about consensus rules see section 2.4

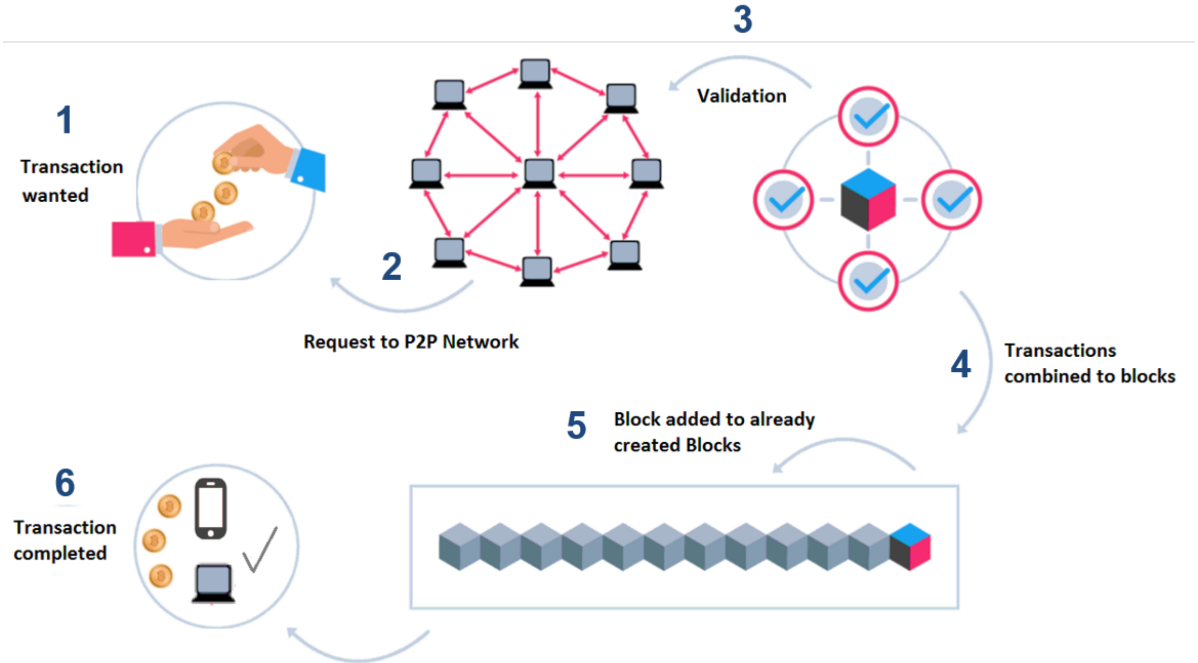


Figure 2.4: Simplified overview of Blockchain and transaction process [29]

Figure (2.4) shows a simplified version of the Blockchain technology and its creation process. The first step in the process is the decision to perform a transaction. This decision is afterwards requested from the peer-to-peer (P2P) network. The P2P network consists of all nodes which are currently actively participating in the network. They validate the requested transactions (as found in section 2.4) and combine all the transactions within a certain period of time (~10 min) to one block. The grouping process creates a block with the proof-of-work concept (as found in section 2.3). As soon as a block is created, it is added to the previous blocks. As soon as a block is added to the Blockchain, the transaction is completed and cannot be changed any more.

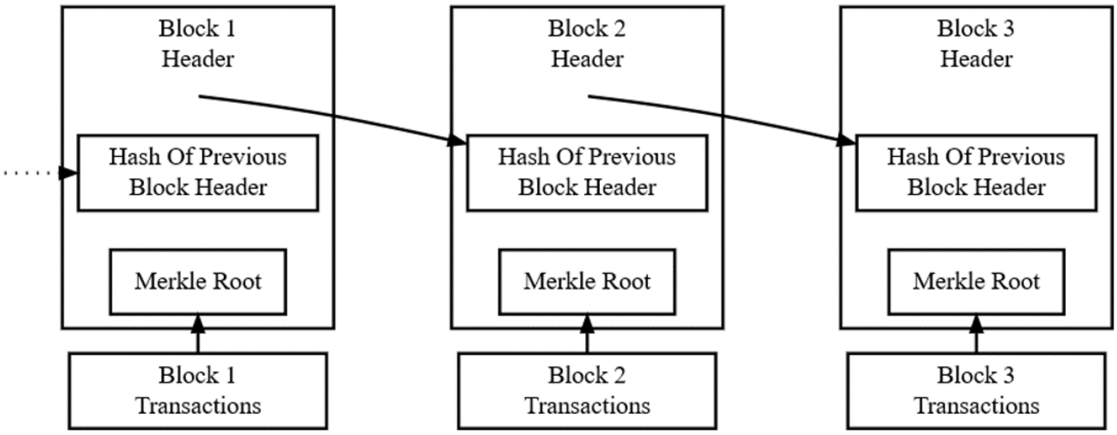


Figure 2.5: Structure of one block [7]

A block, as mentioned above, consists of at least one transaction. This transaction consists of the newly generated bitcoins (for details of the reward for the mining process see section 2.6) which get transferred to the miner's Bitcoin address. The main parts of one block are the header and the transaction(s) (see Figure 2.5). The header consists of a newly generated hash (merkle root) and the hash of the previous block header, thus an explicit order of blocks is given. A merkle root is the hashed value of all transaction included in one block. It is calculated by repeating the pairing and hashing process as long as only one value is the output. Figure 2.6 shows how the grouping and hashing is done for an odd amount of transactions included. Hereby the "leftover" transaction is duplicated and paired with itself until an even amount of pairs is left (even amount of transaction is analogue without the duplication step).

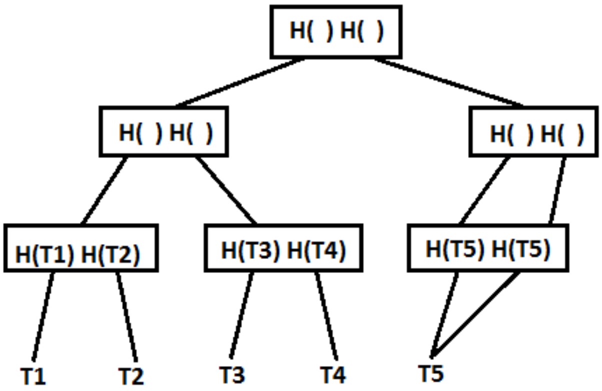


Figure 2.6: Pairwise grouping of transactions to create the merkle root hash value

2.3 Proof of Work

The main characteristics of Blockchain are the distributed trustless census and the prevention of double spending. Therefore, for each block must be checked if a specific amount of effort was put into its creation, ensuring that double spending and a modification of past blocks is (nearly) impossible. A modification would include modifying all following blocks and convincing more than half of the network peers that this new blocks are trustworthy which would require enormous computation power.
Cryptographic hashes have a seemingly random nature and even small changes lead to a completely different hash value. A peer can prove the amount of work put into the finding of a good hash, by making sure that it is below a certain difficulty threshold value.

The proof-of-work scheme (see Figure 2.7) is based on adding a nonce to the hash value of the previous and current block hashes. Afterwards another hash is generated, which is either smaller or bigger as the difficulty threshold, determining whether the proof of work was solved or not.

To ensure the difficulty threshold value is neither too strong nor too simple, every 2,016 blocks the threshold is recalculated. In general, a generation of one single block should take about 10 minutes, which adds up to around two weeks for 2,016 blocks. If the generation took less than two weeks, the threshold is increased, and if it took longer, it is decreased. It happens rarely that two or more nodes prove the work for a block at the same time, which would lead to two branches of the Blockchain. As soon as another block is solved, afterwards this new nodes switch to the longer (stronger) branch. The other block(s) will be discarded and the nodes agree on the order of the blocks, which again results in one valid branch [63].

2.4 Consensus Rules

The nodes in the Bitcoin network validate a block and all its included transactions to ensure the immutability and security of the Blockchain. In general, this consensus rules can be determined as a set of rules of important properties a block has to fulfill to be valid. Basic consensus rules apply to the amount of the mining reward, the data format, the double-spending of transaction inputs and the correctness of signatures.

Each node should theoretically have the same consensus rules, although during an update of the consensus rules there is the possibility that some have already the new ones and others have the old ones. This case can lead to two possible results:

- ***Hard Forks***: Nodes which are not upgraded yet, reject a block and refuse to add newly created blocks to the the same chain as upgraded blocks, are reffered to as “hard forks”. It leads to a split of the block chain into two separate chains (see Figure 2.8) as it is a neither forward nor backward compatible change to the rules. To prevent that it leads to a permanent split into two payment networks,

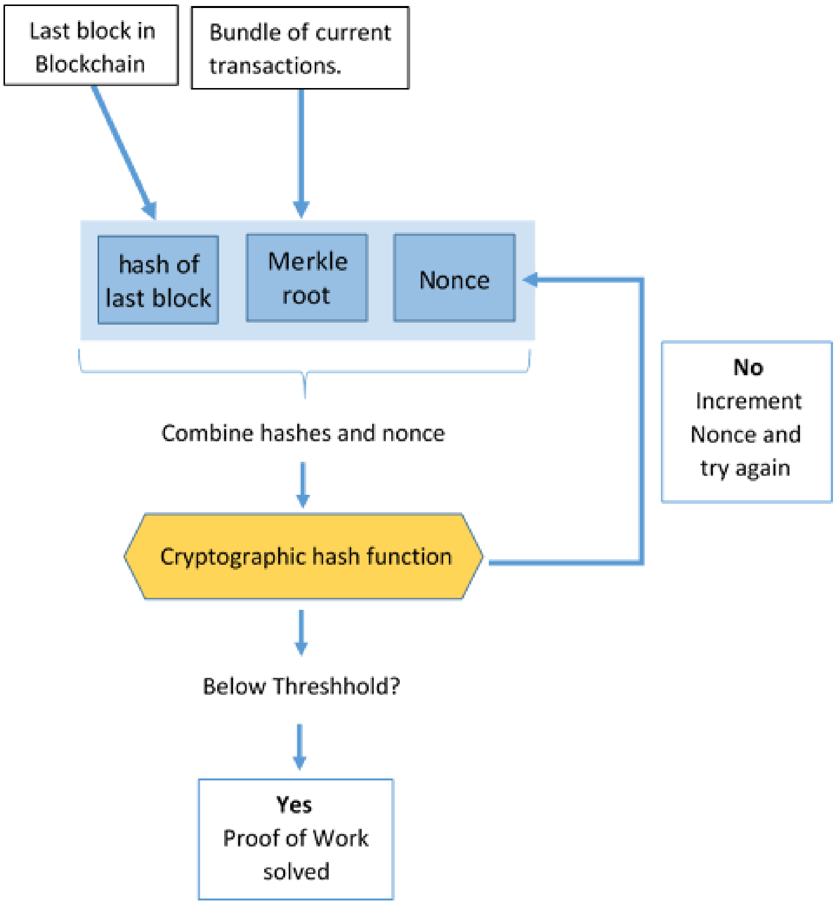


Figure 2.7: Proof-of-work scheme

(nearly) all active nodes have to be updated to the new rules [14]. An example of a (unintentional) hard fork was BIP50 [6] (downgrading capabilities of the already upgraded nodes and removing this again later on).

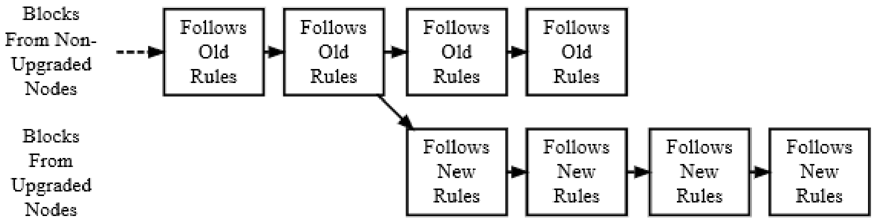


Figure 2.8: Hard Fork [7]

- ***Soft Forks***: The rejection of not upgraded blocks (do not fullfill new rules) by the upgraded ones are refereed to as soft forks. As the nodes following the old rules still accept the blocks created by the updated nodes (forward compatible), this chain gets stronger as more and more nodes get upgraded (see Figure 2.9). The stronger chain will be accepted as the main one to follow all nodes [7]. Examples of soft forks within the last five years are: BIP16 [3] (new standard transaction type: pay-to-script), BIP30 [4] (prevent double spending – transaction identifiers are only allowed once), BIP34 [5](unique value added for coinbase transactions; block update to version 2).

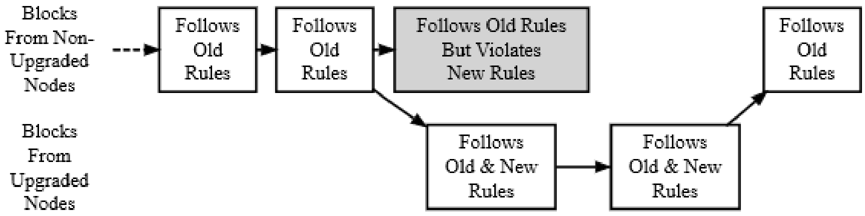


Figure 2.9: Soft Fork [7]

2.5 Transactions

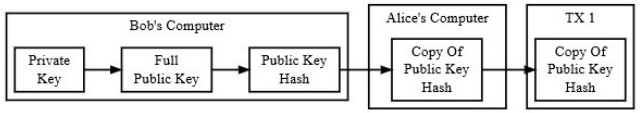
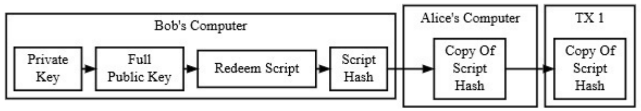
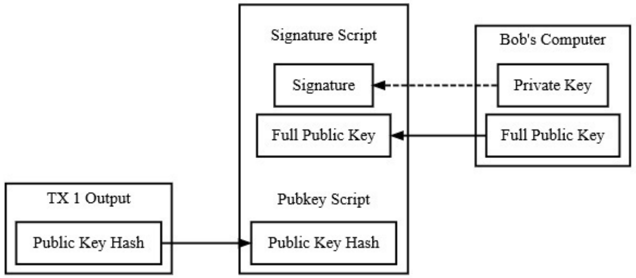
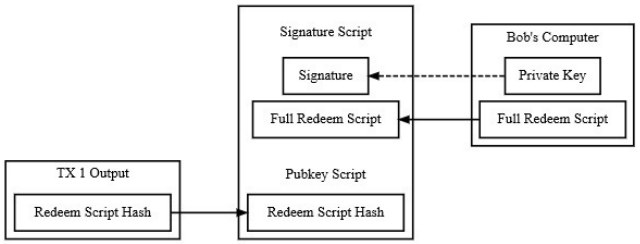
A transaction is the possibility for a user to get and/or spend money. The basic parts of a transaction are: input, output, the version number and the locktime. Every transaction can have several inputs and outputs, but an output can be used only once as an input (prevention of double spending). Each output is marked as an Unspent Transaction Output (UTXO) waiting for its usage as an input for a new transaction. As soon as the UTXO is used as an input, it has to be spent entirely (either by transferring money to others and/or to the user itself). A special transaction is the mining reward which is transferred as a reward for a new block (the early blocks consists only of this “currency generating” transaction).

Currently there are six different standard transaction types [10]:

- Pay-to-Public-Key-Hash
- Pay-to-Script-Hash
- Pay-to-Address
- Multisignature Transaction
- OP_RETURN Meta Coin
- Non-Standard Transactions

An overview of the transaction system is shown in Figure 2.10. The transaction on the left side with the number 0 has an input of 100k. To transfer money to two different addresses, two outputs are created and used for transaction 1 and 2 as an input. The transactions 3 and 6 have UTXO as their output. The amount of output transferred between the individual transactions sums up to the input minus a certain transaction fee. In this example the transaction fee amounts to 10k.

Table 2.1: Overview of Transaction Types

	Pay To Public Key Hash (P2PKH)	Pay To Script Hash (P2SH)
Creating Hash to Receive Payment	 Bob's Computer Private Key Full Public Key Public Key Hash Alice's Computer Copy Of Public Key Hash TX 1 Copy Of Public Key Hash Creating A P2PKH Public Key Hash To Receive Payment	 Bob's Computer Private Key Full Public Key Redeem Script Script Hash Alice's Computer Copy Of Script Hash TX 1 Copy Of Script Hash Creating A P2SH Redeem Script Hash To Receive Payment
Spending the Output	 Signature Script Signature Full Public Key Pubkey Script Public Key Hash Bob's Computer Private Key Full Public Key TX 1 Output Public Key Hash Spending A P2PKH Output	 Signature Script Signature Full Redeem Script Pubkey Script Redeem Script Hash Bob's Computer Private Key Full Redeem Script TX 1 Output Redeem Script Hash Spending A P2SH Output

Script	`Pubkey script: OP_DUP OP_HASH160 <PubKeyHash> OP_EQUALVERIFY OP_CHECKSIG`
	`Signature script: <sig> <pubkey>`
		`Pubkey script: OP_HASH160 <Hash160(redeemScript)> OP_EQUAL`
		`Signature script: <sig> [sig] [sig...] <redeemScript>`

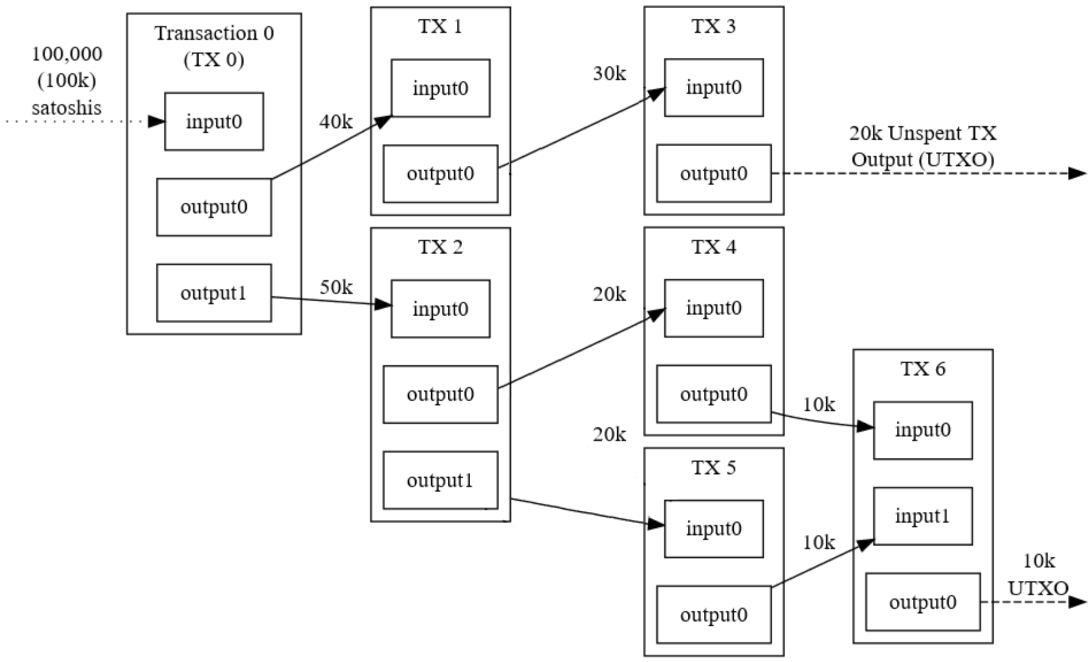


Figure 2.10: Transaction to transaction payments [7]

The two most commonly used transaction types are Pay-to-Public-Key-Hash (P2PKH) and Pay-to-Script-Hash (P2SH). In table 2.1 an overview of those two transaction types can be seen. The creation of the hash to receive payments, the spending of the output and the scripts are contrasted between P2PKH and P2SH.

2.6 Mining

The process of generating and adding new blocks to the Blockchain is called mining. Hereby a hash value has to be found which fulfills the proof of work, the quantification that a certain amount of work was put into the finding of the hash value (as found in section 2.3) and qualifies the miner to receive a reward for her/his efforts. Mining is therefore also the process to place new bitcoins into circulation. The reward is composed of the transaction fees and a specified block reward (newly "generate" Bitcoin).
Currently there are two possibilities of mining:

- ***Solo mining:*** The user tries to generate the block on his/her own to receive the complete reward. As many users participate in the network, it is very unlikely to find the hash before all the other miners.
  Advantage: There is a large reward when a new block is found.

Disadvantage: The probability of finding a new block is quite low. It takes a long time to find one .

- ***Pool mining:*** Several users join a pool to share their computing power resources. As more computing power can be found in a mining pool, it is more likely to find a new block. The reward is shared among the miners according to their contribution of hashing power (see Figure 2.11).
  Advantage: The probability of finding a new block is, in comparison to solo mining, much higher. It takes a shorter period of time.
  Disadvantage: The reward is smaller as all contributors get a part.

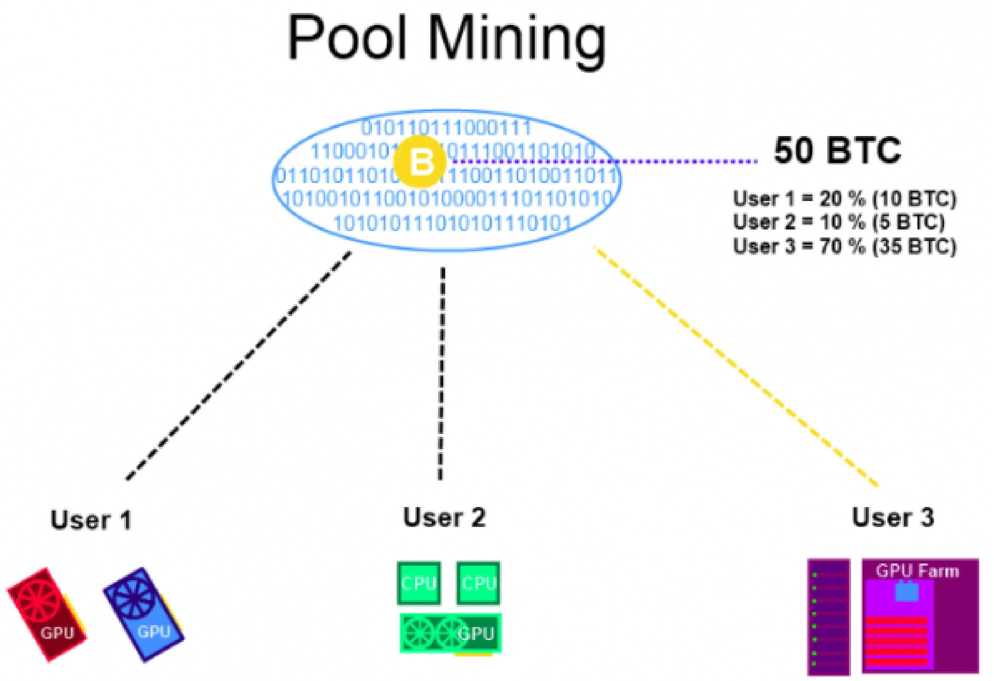


Figure 2.11: Reward sharing according to contribution in pool mining process [8]

2.7 Wallets

A Bitcoin wallet stores all the digital information necessary (the private key(s) of the user, user preferences, version number, etc.) in order to perform a transaction. According to their dependency of location and currency-/budget-safety [12], a wallet can be categorized into three main groups:

- ***Desktop wallets:*** The credentials are stored on the computer, enabling the user to spend bitcoins offline. Therefore it provides the user with full control over the money and its safety.

- ***Internet wallets:*** The credentials are stored with the online provider, making it easier for the user to access the money location independent at any time. As the privacy is provided by a third party, there is no (complete) control of the budget safety.

- ***Mobile wallets:*** The addresses are scanned as QR-codes or NFC while the funds are stored on the users' hardware (mobile, tablet, etc.). The mobile version of *Blockchain* is an internet wallet, combing the advantages of easy access and control.

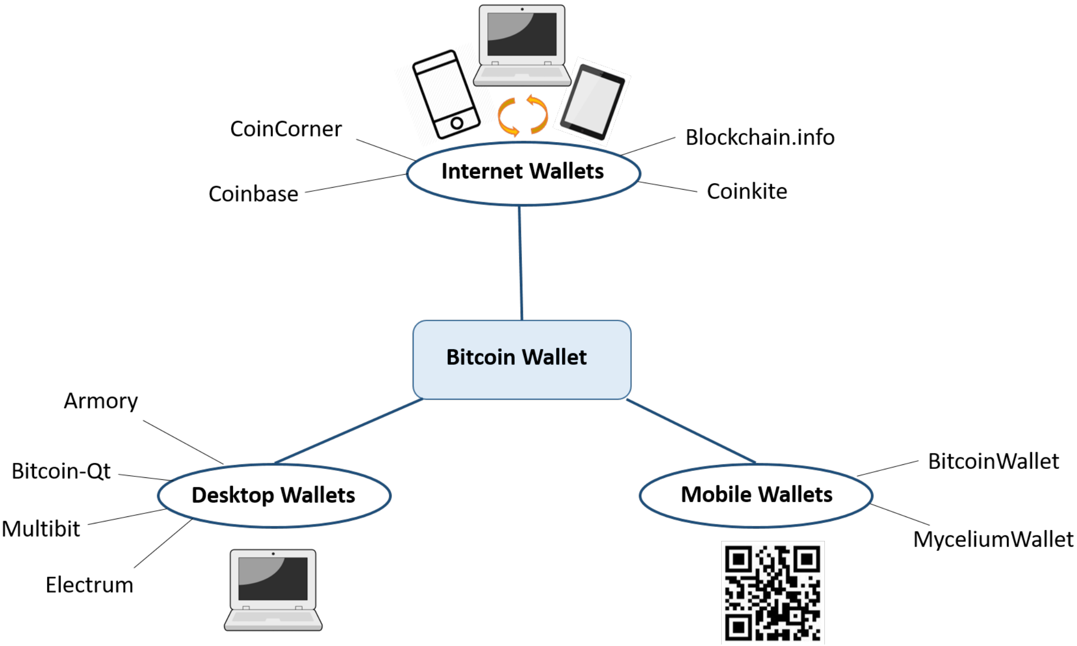


Figure 2.12: Types of wallets with example wallets

In Figure 2.12 an overview of the three wallet types can be seen with corresponding wallet examples.

2.8 Mixing Services

All activities of a Bitcoin user are stored and available to the public in the Blockchain. Although each user has his/her own address which is not directly related to the users themselves, there are possibilities to connect a user and her/his Bitcoin address (see examples in section 3.2).

To anonymize the individual cash flow, so-called mixing services (or tumblers) are used. Hereby, several Bitcoin funds are used to mix the money, intending to mask the original source of a transaction (similar to using accounts in countries with strict bank secrecy laws). To optimally obfuscate the connection between the origin and the target address, it is advised to use trustworthy mixing services with many users and split the payments into smaller transactions. There are two types of mixing currently available:

- ***Centralized mixing services:*** A third-party service where users send their money, and receive for a small fee different bitcoins than they had sent in.
- ***Peer-based mixers:*** Services which provide a space where users meet and exchange (mix) without a third-party their money.

The importance of trustworthy mixing services is emphasized by diverse incidents that happend in the last years. Thereby fake mixing services (scams) stole the transferred bitcoins from the users (e.g. Onion Wallet, Easy Coin, and Bitcoinwallet.in [1]).

New laws concerning the illegality of mixing bitcoins increase the pressure on those services. In summer 2017, one of the biggest mixing services shut down [25] and there are still speculations whether they were forced by political and legislative pressure or changed their mind about the right of privacy for criminals.

CHAPTER 3

Related Work

In recent years, extensive research was conducted on cryptocurrencies as they gained a lot of attention due to the uprise of Bitcoin. Major fields of interest concerning cryptocurrencies, for both the industry and the research community, are anonymity and privacy, as well as the used concepts (e.g. proof of work, consensus rules, etc). Another aspect which got attention, especially over the last years, is the possible usage of the Blockchain system beyond cryptocurrencies (e.g. decentralized governance [35], anti-counterfeits in the postal supply system [72], medical data records [36], etc.).

3.1 Cryptocurrencies

In this section, important work is highlighted in the context of cryptocurrencies in general and with a special focus on systems alternative to Bitcoin. In 2015, Bonneau et al. [40] highlighted the general perspectives and challenges of different cryptocurrencies. The technological background of cryptocurrencies and the dezentralization by the consensus system are described by Serapaglia et al. [68] and Narayanan et al. [64].

In 2014, an alternative cryptocurrency was proposed by Wood called Ethereum[74]. It operates through a message-passing framework with multiple resources in comparison to the transactional singleton machine with shared-state which Bitcoin is using.
A cryptocurrency based on Bitcoin, called Darkcoin, was proposed 2014 by Duffield et al. [48]. In comparison to Bitcoin, however, the main focus of this currency lies on the privacy-centric cryptography instead of the dezentralization part. The anonymisation of a block transaction is incorporated directly into the client. Furthermore, the SHA 256 algorithm is replaced by a chain of hashing algorithms (X11 [30]) for an additional security effect as an improvement to the proof-of-work concept performed by Bitcoin. This technology replaces and advances the anonymity of the user in a more effective way than Bitcoin.

Another alternative was proposed in 2017 by Jaag et al. [53], called Postcoin. It uses the Blockchain technologies for postal financial services, with exchange rates that are not as volatile as those of bitcoins. To date, there are more than 900 different alternative bitcoins (altcoins) [17] with a total market cap as high as Bitcoin on its own.

3.2 Security

There are many strategies to manage bitcoins and ensure privacy. Caused by the variety of possibilities to ensure the anonymity when using Bitcoin, many users have significant misconceptions of their management tools, as the first large-scale user survey conducted by Krombholz et al. [57] shows. The results of the study point out that the usability of the security mechanisms have potential for improvements to ensure better privacy and usable security. Another cause of the misconceptions is the gap between the theoretical understanding of this new cryptocurrency and the practice. Judmayer et al. [54] presented a book in 2017 which aims to bridge this gap, providing detailed descriptions of the inner workings of the bitcoins protocol.

A guide for the most common Bitcoin transaction patterns was introduced by Ferrin in 2015 [49]. The guide describes on the basis of (real) Bitcoin transactions with the help of generic parameters specific patterns (e.g. peel transactions – simple spend of money, distribution transactions – payout, etc.) for a better understanding of the mechanisms behind them.

Androulaki et al. [34] proposed an evaluation which determines the privacy and anonymity of bitcoins. They evaluated the privacy implications provided by Bitcoin, finding that those are not enough to protect the privacy of the users. As bitcoins are not per-se anonymous it is important to emphasize this circumstance to achieve awareness. Andoulaki et al. [34] and Brugere et al. [52] for example highlighted the pseudo-anonymity by introducing different ideas as unsupervised learning and heuristics about multi-input transactions and “shadow” addresses to associate Bitcoin addresses to the controlling entity/user.

Reid et al. [66] analyzed the Bitcoin system based on its anonymity. They demonstrate the possibility to associate the public keys to a specific user by combining the public available structure of the Bitcoin system with external informations. Another survey was done by Maurer et al.[60], comparing privacy solutions for the Bitcoin system. According to their study, CoinJoin [16] can be considered a good approach for anonymizing the transactions.

In 2013, Meiklejohn et al. [62] proposed another method to connect users and their Bitcoin addresses with the main focus on identifying big institutions and their transactions. This method uses (manual) labelling of the input or change addresses combined with clustering, which exploits a current idiom of use in the network.

One year later, another approach which determines a connection between a bitcoin address and a real person or company was proposed by Baumann et al. [37]. It uses recent data provided by the Blockchain, which is explored with a combination of statistics and network analysis tools to get an insight in the past transactions. The overall analysis is done by graph mining showing the different connections. In comparison to our prototype this approach focuses only on recent data stored in the Blockchain and does not provide any interactive exploration possibility of the graphs.

3.3 Visualization

The Blockchain technology became famous together with the Bitcoin ecosystem. It provides the unique opportunity to record the history of digital transactions and events based on a consensus system which is publicly available to everyone. The public available transaction data stored in the Blockchain gives the research community the possibility to analyze the current and past cash flows.

The great price boom [23] of bitcoins in the last four years led to an increasing interest among laypeople and academia. Consequently, the fields of visualization, privacy and usable security became of great importance for the analysis and exploration of this virtual cryptocurrency under different aspects and circumstances. The great importance of visual output for the discovery and analysis of digital data was presented by Nordbo [65] who conducted a literature study on the challenges and possibilities of different visualization tools and methods.

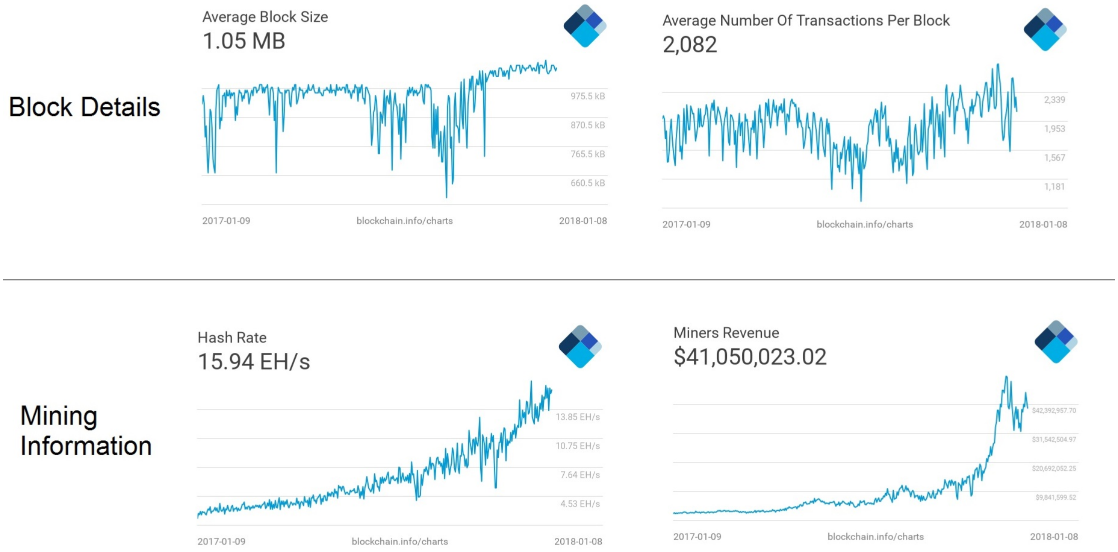


Figure 3.1: Example graphs from Blockchain.info containing information about blocks and mining data[15]

The website Blockchain.info [15] gives an insight into the available information provided by the Blockchain. Individual blocks can be inspected including the corresponding transactions in textual form. Furthermore, currency statistics, block details (see example of block size average in Figure 3.1), mining information, network activity and wallet activities can be inspected in different line or circle charts. Blockchain.info provides the user with all information available in the Blockchain, which is also the goal of our approach; however, the information is not connected and scattered all over the website. Furthermore, the transactions and its connections are provided only via textual information, which must be clicked through by the user by hand. Plan₿ enables the user to search for specific Bitcoin addresses and display their surrounding (neighbourhood) or search for a connection to another address.

The different visualization applications can be categorized into three groups according to their usage and goals: Economic visualizations, real-time (transaction) visualizations and security and analytic-related visualizations. In the following influential visualization approaches in those categories are described.

3.3.1 Economic

Saublet et al. [67] proposed a visualization tool for economists to evaluate the Bitcoin payment system. It is one of the first web-based applications (for a screenshot of the application see Figure 3.2) with the major goal to determine the utilization of bitcoins, answering the question if it is used as a currency or a commodity. The graphs provide no interactivity and solely focus on general statistics (not on individual transaction levels).

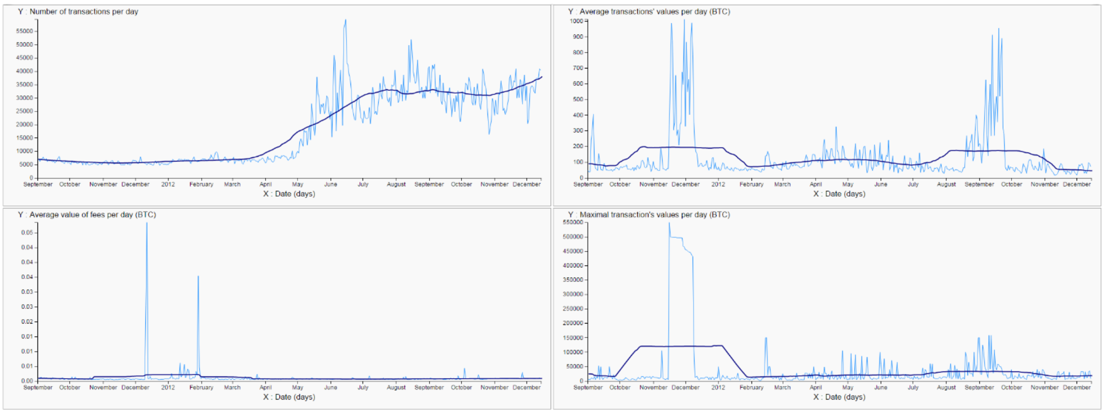


Figure 3.2: Quad-chart layout of the proposed interface by Saublet et al. [67]

The usage of Blockchain technology in both financial and non-financial applications was described by Crosby et al. [46] in 2016. In the same year, an explorative study was conducted by Lischke et al. [59] focusing on the transactional and economic side of the Bitcoin network of the first four years after the first transaction was performed in 2009. They were able to give an insight in business developments and gambling patterns within

the Bitcoin usage.
Alvarez-Pereira et al. [33] introduced a new way to explore the economic effects, for example economic bubbles. According to network and conversation analysis, they are able to relate those effects to Bitcoin. Due to their visual representation they were able to determine a correlation between discussion sentiment and the current price of the Bitcoin. Their visualizations in relation to our prototype, however, focused only on economic states and price relationships and not on transaction and time-based effects.

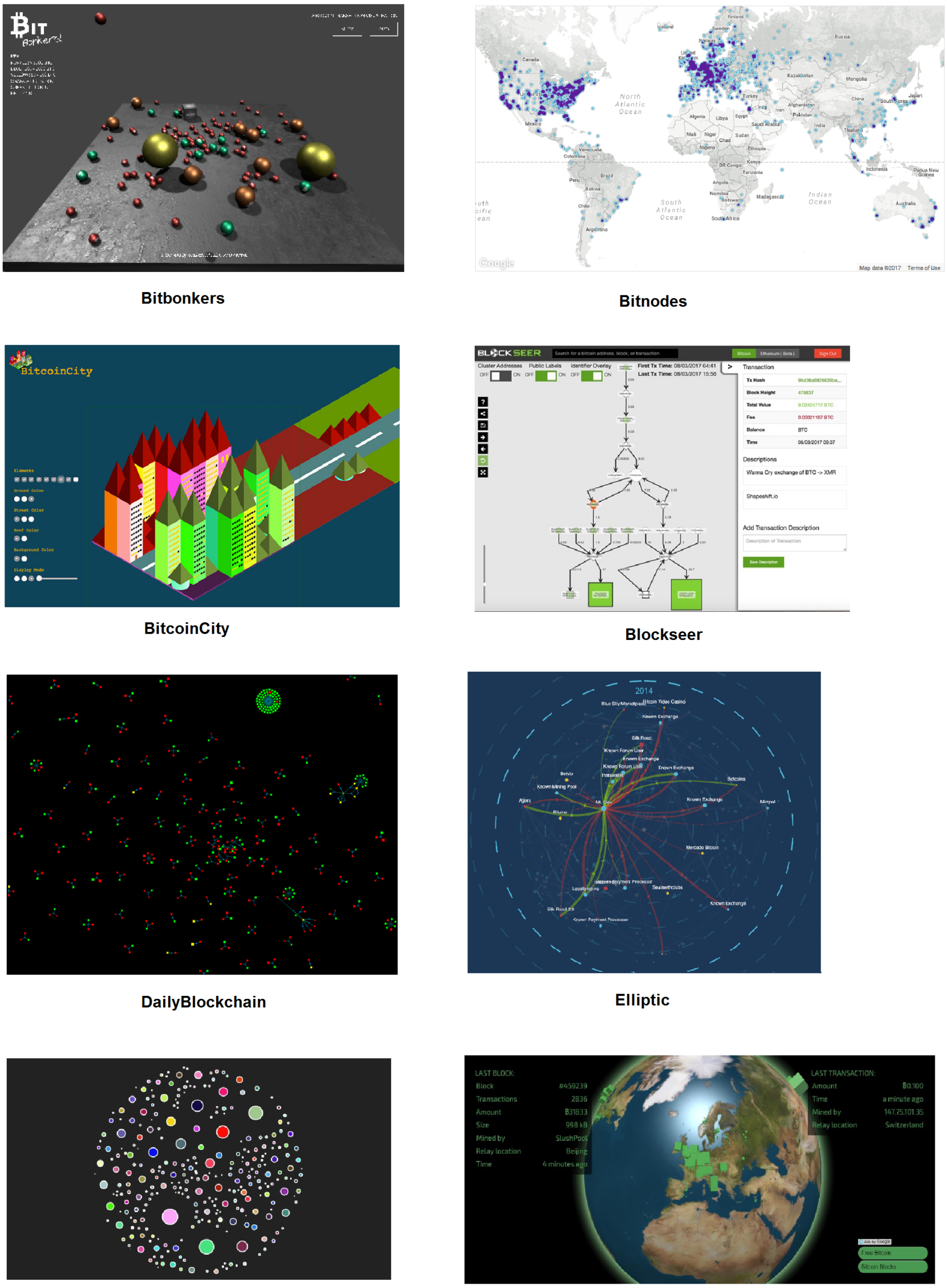
3.3.2 Real Time

Real-time visualizations of transaction flows are often published as web applications. Their two major goals are the displaying of current transaction values and sizes and their correlation among each other. The visualizations range from blob visualization and simple graphs to more complex world maps and city blocks. Those visualization mostly have a very catchy presentation of the current Blockchain transaction, however, the objective information which can be extracted of those is quite low. The main focus is the aesthetic representation of the real-time transaction. The focus of our project is to combine the real-time aspect with informative and interactive representations.

Addy Yeow [20] created a map that visualizes the distribution of the currently reachable nodes on a world map (see Figure 3.3). The data is fetched with the help of a crawler which uses the real-time data of the Blockchain.

Another real-time approach is a graph-based application published by dailyblockchain [27]. For every performed transaction a new node or separate sub-graph is created, which gives a general impression of the complex transaction network.

In the beginning of 2017, Sundara et al. [71] presented a general overview and a comparison of eight different Bitcoin visualization tools including two of the above-mentioned applications. They loooked at Bitbonkers, Bitnodes, BitcoinCity, Blockseer, DailyBlockchain, Elliptic, Interaqt and LiveGlobe for their study (see Figure 3.3). For our approach we had a look at Blockseer in more detail, as they have the most similarities to our approach compared to the other real-time applications (further details can be found in section 4.2).


Bitbonkers

Bitnodes

BitcoinCity

Blockseer

First Tx Time: 08/03/2017 04:41
Last Tx Time: 08/03/2017 15:56

Transaction

Tx Hash	95d36a6926639ba...
Block Height	478837
Total Value	9.02424717 BTC
Fee	0.00021107 BTC
Balance	BTC
Time	08/03/2017 09:37

Descriptions

Wanna Cry exchange of BTC -> XMR

Shapeshift.io

Add Transaction Description

Description of Transaction

DailyBlockchain

Elliptic

Interaqt

LAST BLOCK:

Block	#459239
Transactions	2836
Amount	B31833
Size	998 kB
Mined by	SlushPool
Relay location	Beijing
Time	4 minutes ago

LAST TRANSACTION:

Amount	B0.100
Time	a minute ago
Mined by	147.75.101.35
Relay location	Switzerland

LiveGlobe

Figure 3.3: Overview of different real-time visualization applications

3.3.3 Analytics and Privacy

Visualizations should on the one hand emphasize the pseudo-anonymity of Bitcoin, and on the other hand help forensic and security experts to backtrack cashflows and filter patterns (mining pools, mixing services, etc.).

Kurzuno et al. [58] introduced a system which is useful for investigating and filtering criminal actions done via the Bitcoin network, including past transaction data. Their system's main features are showing some statistics of the used addresses, a graphical transaction relation model and the visualization of transaction paths.

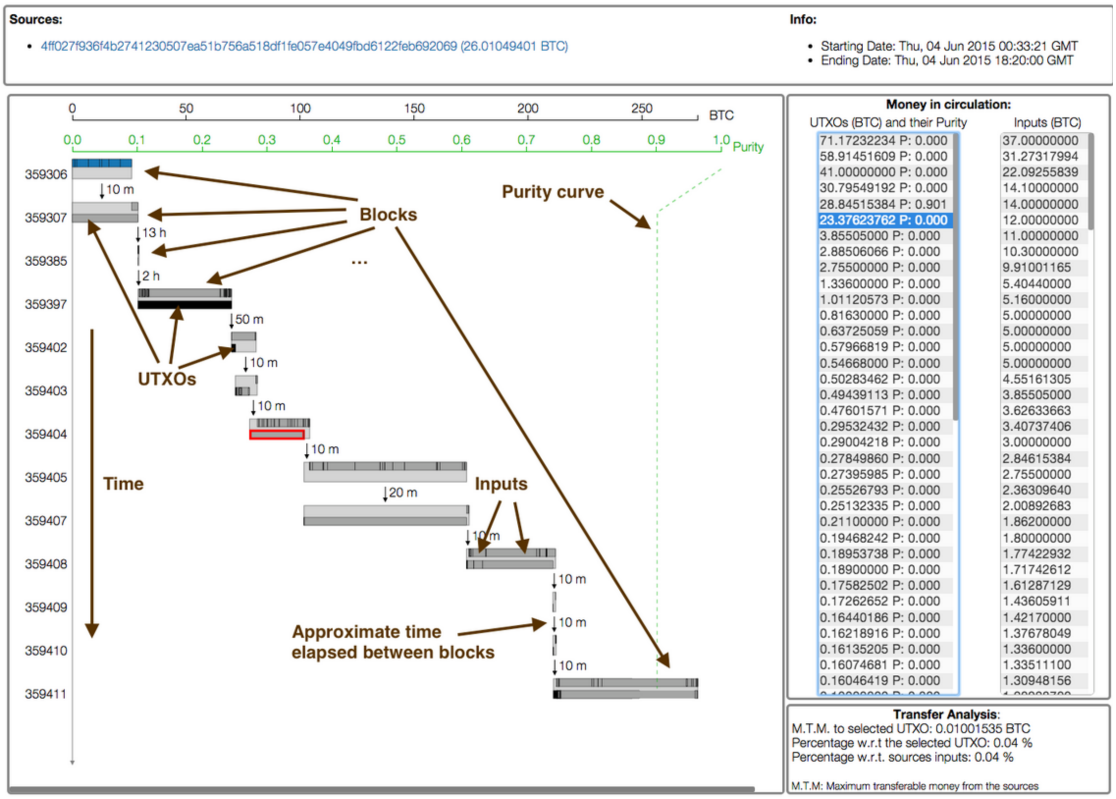
Spanguolo [69] finished his thesis about a semi-automatic system, called BitIodine, which labels active users and provides information about their actions for forensic analysis. Shortly afterwards, in 2014, Spagnuolo et al. [70] published their findings at the International Conference on Financial Cryptography and Data Security. They released their prototype as a library, which provides a baseline for further analysis tools.

A visual analysis system, called BitConeView, was introduced by Battista et al. in 2015 [47] for the evaluation of Bitcoin flows in the Blockchain. The focus of this application was the detection of flow patterns within the transaction graph for gaining a fast understanding of suspicious activities (see Figure 3.4). The general idea to detect suspicious patterns and determine connections between two addresses is similar to our approach, however, the implementation is very different. In comparison to our approach it has (nearly) no interactive component and the computation for larger information requests takes a long time, without a visual output. Furthermore, the BitConeView only uses bar charts to display the information.

In 2015 Greaves et al. [51] performed an analysis on the Bitoin network and its correlation and predictability to the Bitcoin price. Their findings are based on transaction graphs and enable them to predict the prices (with a 55% accuracy, as they state). 2015 Berini Sarrias [38] proposed different metrics which help to assess the security of the Bitcoin network with the help of simulator models. His main goal was to extract information of the network for a better understanding and give a visual output of the findings.

McGinn et al. [61] introduced a top-down visualization giving the users the possibility to collaboratively discover high frequency transaction patterns. Especially, they managed to show the evolution of denial-of-service attacks and laundering operations, which gave a great insight in the security aspects of the Bitcoin network. The most important difference to Plan₿ is the lack of transaction correlation over time.

2017 the BitConduite tool was introduced by Kinkeldey et al. [56]. This application is based on a MongoDB document database combined with a MonetDB database to get fast access to the different data which is displayed afterwards. The visualization consists of several groups which are clustered by k-Means (see Figure 3.5) to get an overall feeling for the different clusters within the Blockchain. The difference to our approach is the lack of interactivity and the missing of different views of the transactions. Furthermore, the focus is different as the major goal of BitConduit is the deanonymization by connecting



Sources:

- 4ff027f936f4b2741230507ea51b756a518df1fe057e4049fbd6122feb692069 (26.01049401 BTC)

Info:

- Starting Date: Thu, 04 Jun 2015 00:33:21 GMT
- Ending Date: Thu, 04 Jun 2015 18:20:00 GMT

Money in circulation:

UTXOs (BTC) and their Purity	Inputs (BTC)
71.17232234 P: 0.000	37.00000000
58.91451609 P: 0.000	31.27317994
41.00000000 P: 0.000	22.09255839
30.79549192 P: 0.000	14.10000000
28.84515384 P: 0.901	14.00000000
23.37623762 P: 0.000	12.00000000
3.85505000 P: 0.000	11.00000000
2.88506066 P: 0.000	10.30000000
2.75500000 P: 0.000	9.91001165
1.33600000 P: 0.000	5.40440000
1.01120573 P: 0.000	5.16000000
0.81630000 P: 0.000	5.00000000
0.63725059 P: 0.000	5.00000000
0.57966819 P: 0.000	5.00000000
0.54668000 P: 0.000	5.00000000
0.50283462 P: 0.000	4.55161305
0.49439113 P: 0.000	3.85505000
0.47601571 P: 0.000	3.62633663
0.29532432 P: 0.000	3.40737406
0.29004218 P: 0.000	3.00000000
0.27849860 P: 0.000	2.84615384
0.27395985 P: 0.000	2.75500000
0.25526793 P: 0.000	2.36309640
0.25132335 P: 0.000	2.00892683
0.21100000 P: 0.000	1.86200000
0.19468242 P: 0.000	1.80000000
0.18953738 P: 0.000	1.77422932
0.18900000 P: 0.000	1.71742612
0.17582502 P: 0.000	1.61287129
0.17262652 P: 0.000	1.43605911
0.16440186 P: 0.000	1.42170000
0.16218916 P: 0.000	1.37678049
0.16135205 P: 0.000	1.33600000
0.16074681 P: 0.000	1.33511100
0.16046419 P: 0.000	1.30948156

Transfer Analysis:
M.T.M. to selected UTXO: 0.01001535 BTC
Percentage w.r.t the selected UTXO: 0.04 %
Percentage w.r.t. sources inputs: 0.04 %

M.T.M: Maximum transferable money from the sources

Figure 3.4: Interface of BitConeView [47]

addresses to a specific entity (person, firm). This is done by estimating and analysing the individual activity patterns and connect them to real persons (IP-Addresses).

A multi-view visual analytics approach was proposed by Bistarelli et al. [39] which filters undesired information to reduce the visualized data. The different views enable the user to get insights and/or overviews about miners, sources and the Bitcoin flows, including the used address and transactions. Furthermore, they grouped the transactions into disconnected “island” (see Figure 3.6), to give the user a better way to focus on a specific one. This approach is very similar to our prototype, however, the major difference is the different focus (for them the general overview with filter possibilities and for our approach the time-based transaction connection) and the different graphs we provide (for further details see section 4.2).

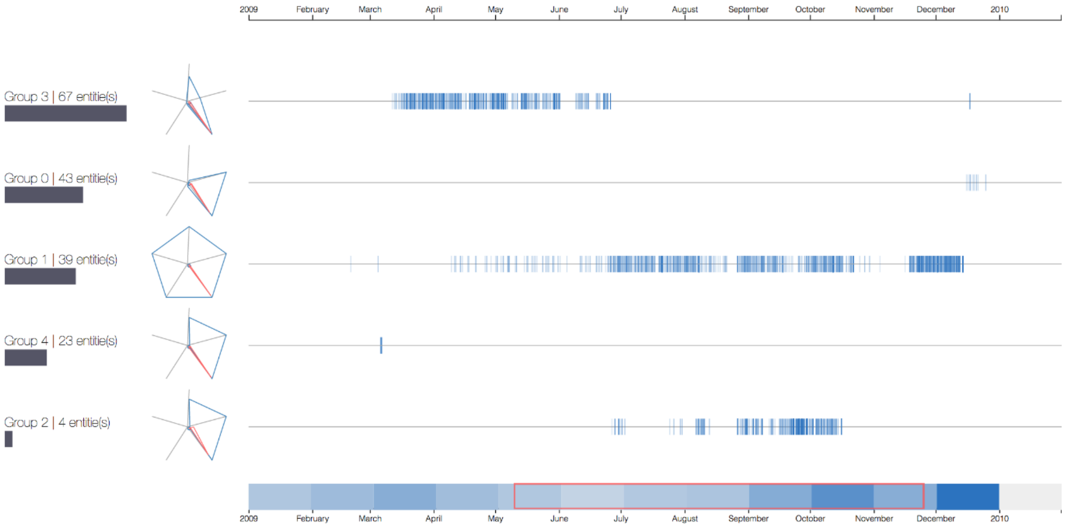
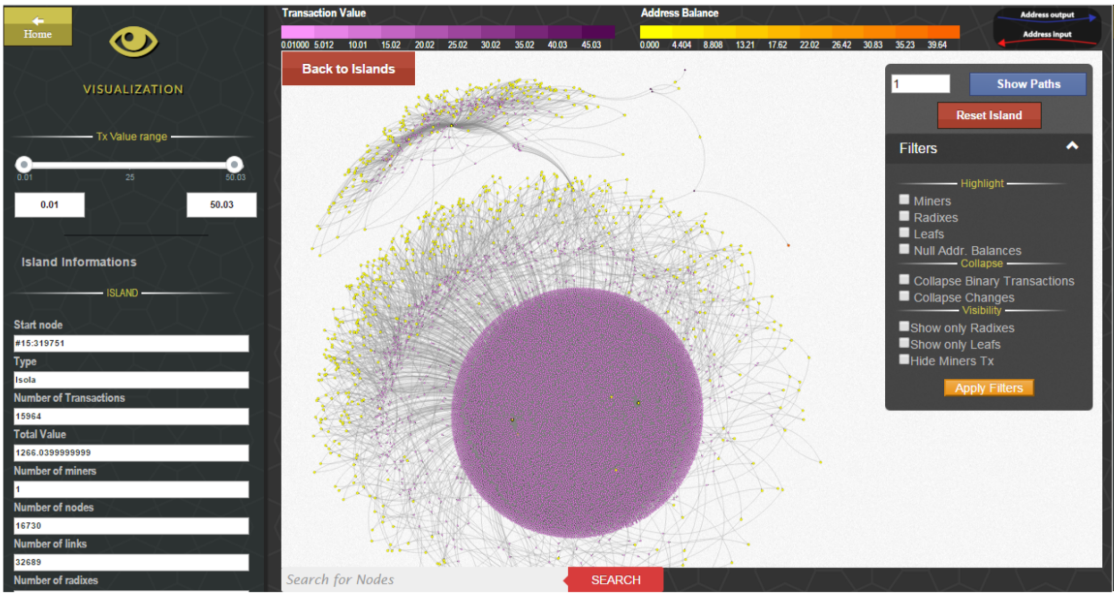


Figure 3.5: Transaction grouped by entity clusters - BitConduite[58]



Tx Value range: 0.01 – 50.03

Island Informations

ISLAND

Start node: #15:319751

Type: Isola

Number of Transactions: 15964

Total Value: 1266.0399999999

Number of miners: 1

Number of nodes: 16730

Number of links: 32689

Number of radixes

1

Highlight

- □ Miners
- □ Radixes
- □ Leafs
- □ Null Addr. Balances

Collapse

- □ Collapse Binary Transactions
- □ Collapse Changes

Visibility

- □ Show only Radixes
- □ Show only Leafs
- □ Hide Miners Tx

Figure 3.6: Overview of transactions with island cluster (see big pink dot circle) [39]

CHAPTER 4

Methodology

The methodology and design used for the prototype Plan₿ is based on qualitative and quantitative design methods of information visualization (as found in section 4.1.1) and visual analytics (as found in section 4.1.2).

Plan₿ is a web-based application and was developed based on the phases of the System Development Life Cycle (SDLC). The number of phases used in the SDLC vary between five to more than ten, depending on the project and its breakdown granularity. We considered the following five phases for this thesis:

1. **Planning:** We set up a project development plan. Hereby, the basic requirements were set and an overall goal was formulated. The goal to design and develop was: **an application which gives the user the possibility to interactively explore the Blockchain and determine corresponding transactions**.

2. **Analysis:** To get a better feeling of currently available applications we conducted an extensive literature research (as found in chapter 3). The related approaches were analyzed and compared among each other. During the literature research walkthroughs through three applications were performed (as found in section 4.2). Hereby, the main focus was to identify the flaws and improvement opportunities of those applications and to formulate (further) specifications for our prototype.

3. **Design:** After the specifications and requirements were formulated the design process started. Hereby the first sketches of our application were made and the name was retained (as found in section 4.3).

4. **Implementation:** The implementation phase started as soon as the basic design was finished. As the prototype consists of two major parts, the front- and the backend, the communication between those two was established. Details on the frontend can be found in chapter 5, including screenshots of the final prototype

version. During the implementation phase several feedback loops were performed to improve the layout and its usability.

5. **Operation:** The last phase of this work includes the source code publication as it is our goal to provide a basic framework for further visual research on the Bitcoin transaction system. Furthermore, the resulting prototype was evaluated, reviewed and future work was discussed.

4.1 Visualization

Various encyclopedies define visualization as "*The act or process of visualizing*" [18]. For this thesis the term visualization covers not solely the act of visualizing but also includes the communication of a message shown by images, diagrams or animations within a tool. This way of communication can be used effectively on the one hand for very abstract ideas and elements, and on the other hand for very concrete data to enable the user to form a mental vision, image or picture [Oxford Engl. Dict. 1989].

Visualization is an important area which is used in various fields of computer sciences (medicine, security, graphics, etc.) to provide users with visual information. It can be classified into the following three areas: Scientific Visualizations (Volume Visualization, Flow Visualization), Information Visualizations, and Visual Analytics. The three major goals of visualization are to explore (when nothing is known beforehand), to analyze (verify or falsify the hypotheses), and to present (to communicate the already known aspects). For this thesis the two main aspects of visualization used to design the prototype Plan₿ are information visualization and visual analytics, which will be discussed in the following two sections.

4.1.1 Information Visualization

The main goal of information visualization is to enable the user to explore a large amount of (abstract) data. The Blockchain currently has more than 155 GB (February 2018) of data stored. A simple comparison and look through this tremendous amount of data is undoable for a human being. Therefore, a visual representation is necessary to get a feeling of the structure and the transaction mechanisms which are publicly stored in the Blockchain.

4.1.2 Visual Analytics

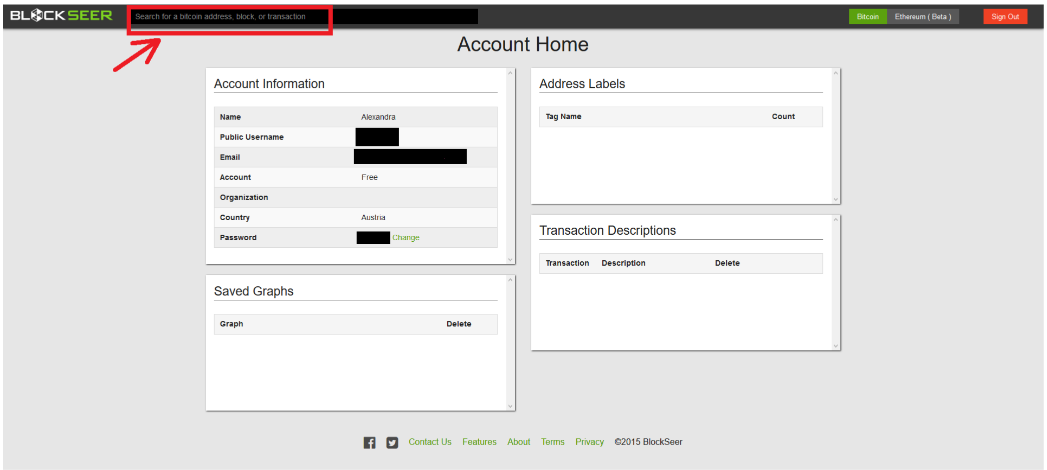
The field of visual analytics can be determined as an enhancement of scientific and information visualization. It supports the analytical reasoning with a visual system which is interactively explorable. Hereby, especially human strengths like creativity, flexibility and further background knowledge are facilitated with the processing powers of a computer based application and its storage capabilities. The interactive part of the prototype presented in this thesis is very important, as it gives the user an optimal possibility to explore the data and draw novel conclusions from them.

4.2 Application Walkthroughs

During our literature research about visualization applications for Bitoin transactions, we found several approaches (as described in chapter 3). However, most of them had no great similarity with our planned prototype. On the one hand, many of the applications only focused on one aspect (either real-time or only a limited period of time) which we defined as essential for our prototype Plan₿; on the other hand, many applications were not publicly available and only provided a very limited amount of screenshots/graphics of the framework. Based on the two constraints of availability and similarity, we decided to take two web-based applications into account for our walkthrough, which we describe in the following two sections.

4.2.1 Blockseer

The Blockseer application was published in 2015 (https://www.blockseer.com/ – walkthrough 11-08-2017) and offers the user a free account to test the application. Its mission is to bring more transparency to the participants of the Bitcoin network, aiming to reduce *"the level of disorder and chaos and increase the level of knowledge and analysis"* as they state in the "About" section on the website.



[redacted]

Account Home

Account Information

Name	Alexandra
Public Username	[redacted]
Email	[redacted]
Account	Free
Organization	
Country	Austria
Password	[redacted] Change

Address Labels

Tag Name	Count

Transaction Descriptions

Transaction	Description	Delete

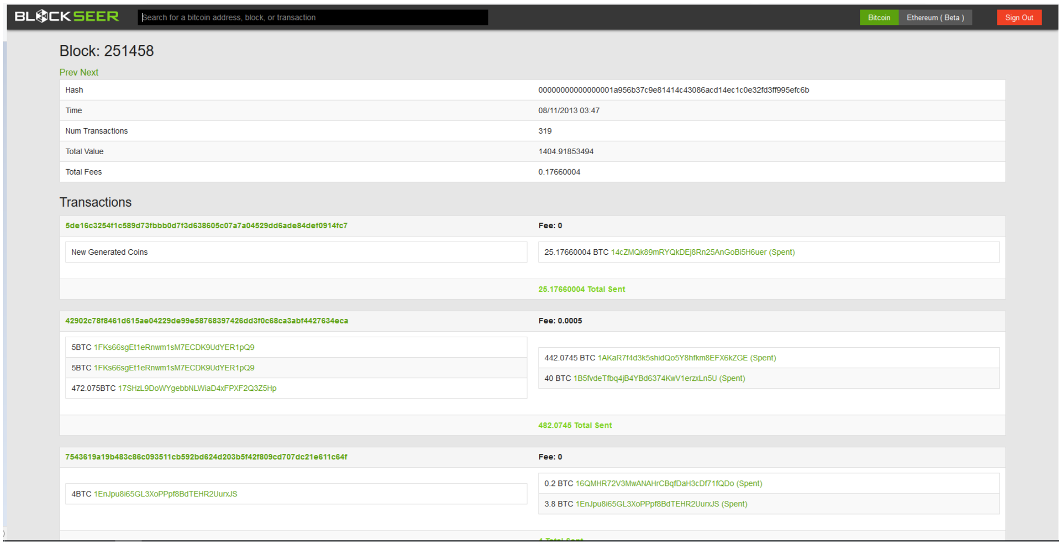
Saved Graphs

Graph	Delete

Figure 4.1: Home view of Blockseer application

Blockseer enables the user to visualize transactions of the Bitcoin system in an interactive way. First of all, the user has to search for either a Bitcoin address, a block number or a specific transaction. This information is added into the search field on the top left side of the website (see Figure 4.1). Afterwards the user is directed to another side displaying the searched information and its belonging attributes in a textual format, which is displayed in Figure 4.2. The displayed figure shows information about block 251458 with all its included transaction data. The text highlighted in green (light green –

Bitcoin addresses, darker bold green – hash values of transactions) signals the user the interactivity and can be inspected individually by clicking on them.



Block: 251458

Hash	00000000000000001a956b37c9e81414c43086acd14ec1c0e32fd3ff995efc6b
Time	08/11/2013 03:47
Num Transactions	319
Total Value	1404.91853494
Total Fees	0.17660004

Transactions

5de16c3254f1c589d73fbbb0d7f3d638605c07a7a04529dd6ade84def0914fc7 — Fee: 0

New Generated Coins → 25.17660004 BTC 14cZMQk89mRYQkDEj8Rn25AnGoBi5H6uer (Spent)

25.17660004 Total Sent

42902c78f8461d615ae04229de99e58768397426dd3f0c68ca3abf4427634eca — Fee: 0.0005

5BTC 1FKs66sgEt1eRnwm1sM7ECDK9UdYER1pQ9
5BTC 1FKs66sgEt1eRnwm1sM7ECDK9UdYER1pQ9
472.075BTC 17SHzL9DoWYgebbNLWiaD4xFPXF2Q3Z5Hp

442.0745 BTC 1AKaR7f4d3k5shidQo5Y8hfkm8EFX6kZGE (Spent)
40 BTC 1B5fvdeTfbq4jB4YBd6374KwV1erzxLn5U (Spent)

482.0745 Total Sent

7543619a19b483c86c093511cb592bd624d203b5f42f809cd707dc21e611c64f — Fee: 0

4BTC 1EnJpu8i65GL3XoPPpf8BdTEHR2UurxJS

0.2 BTC 16QMHR72V3MwANAHrCBqfDaH3cDf71fQDo (Spent)
3.8 BTC 1EnJpu8i65GL3XoPPpf8BdTEHR2UurxJS (Spent)

4 Total Sent

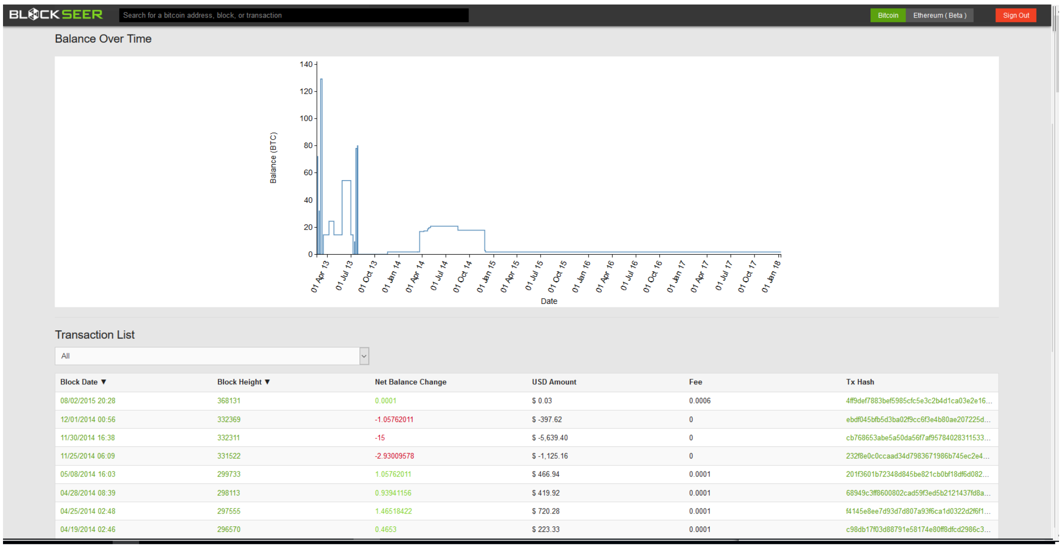
Figure 4.2: View after searching for Block 251458

A click on a Bitcoin address leads to a new side (an example can be seen in Figure 4.3), providing the user with further information about the individual address. Hereby the developer(s) decided to combine textual and visual information to display all the transactions linked to the address and its corresponding balance. This combination of text and graphic is similar to our prototype as a visual representation helps the human brain to analyse huge and complex data and a simple text is very useful for specific information (in this case a specific transaction, its fees and the other involved participants).

The other option by clicking on a specific transaction brings the user to an interactive graph board. In the beginning only the transaction which has been clicked on is displayed as a block as it can be seen in Figure 4.4. The circle with the dotted line represents the starting point of a newly mined block (non-expandable) and the square represents the transaction which was clicked on beforehand.

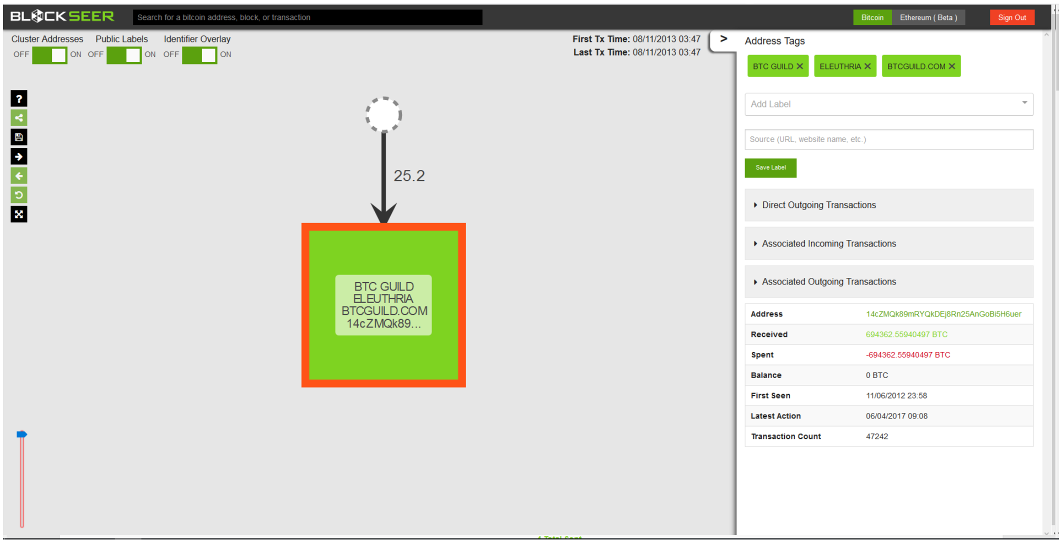
On the right hand side information is displayed concerning the transaction. This information includes the direct incoming and outgoing transactions as well as the associated incoming and outgoing transactions. Associated incomes and outcomes are not directly related to the transaction but indirectly, either through pool-mined incomes or other transactions. Furthermore, three different check boxes provide the following three options:

- *Clustering Addresses*
  OFF: transactions are displayed as rhomb with incoming and outcoming lines
  ON: transactions are displayed as straight lines



Block Date ▼	Block Height ▼	Net Balance Change	USD Amount	Fee	Tx Hash
08/02/2015 20:28	368131	0.0001	$ 0.03	0.0006	4ff9def7883bef5985cfc5e3c2b4d1ca03e2e16...
12/01/2014 00:56	332369	-1.05762011	$ -397.62	0	ebdf045bfb5d3ba02f9cc6f3e4b80ae207225d...
11/30/2014 16:38	332311	-15	$ -5,639.40	0	cb768653abe5a50da56f7af95784028311533...
11/25/2014 06:09	331522	-2.93009578	$ -1,125.16	0	232f8e0c0ccaad34d7983671986b745ec2e4...
05/08/2014 16:03	299733	1.05762011	$ 466.94	0.0001	201f3601b72348d845be821cb0bf18df6d082...
04/28/2014 08:39	298113	0.93941156	$ 419.92	0.0001	68949c3ff8600802cad59f3ed5b2121437fd8a...
04/25/2014 02:48	297555	1.46518422	$ 720.28	0.0001	f4145e8ee7d93d7d807a93f6ca1d0322d2f6f1...
04/19/2014 02:46	296570	0.4653	$ 223.33	0.0001	c98db17f03d88791e58174e80ff8dfcd2986c3...

Figure 4.3: Balance belonging to specific address, displayed in a line graphic and textual below, over time



First Tx Time: 08/11/2013 03:47
Last Tx Time: 08/11/2013 03:47

Address	14cZMQk89mRYQkDEj8Rn25AnGoBi5H6uer
Received	694362.55940497 BTC
Spent	-694362.55940497 BTC
Balance	0 BTC
First Seen	11/06/2012 23:58
Latest Action	06/04/2017 09:08
Transaction Count	47242

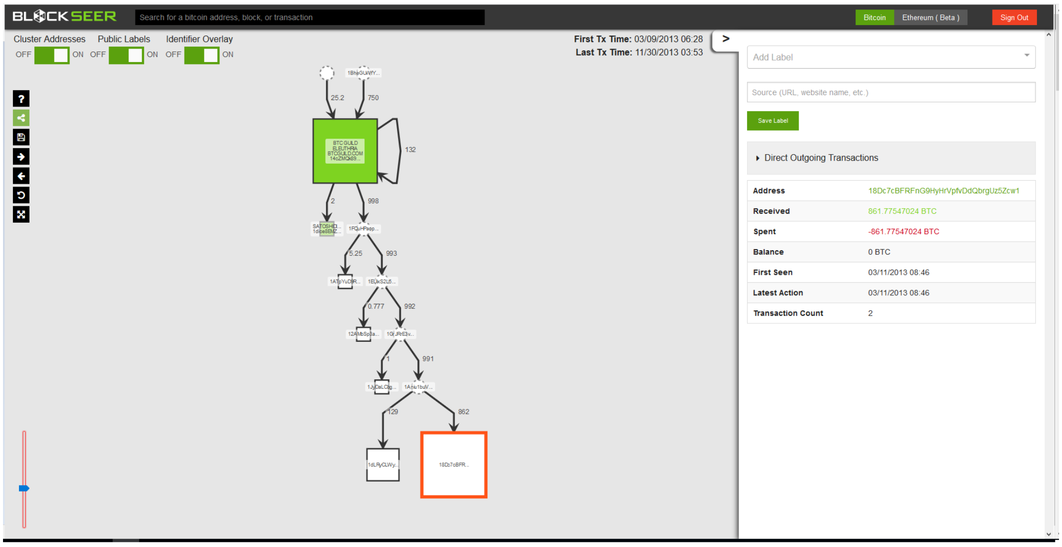
Figure 4.4: Path of direct outgoing transactions

- *Public Labels*
  OFF: no tags/labels are displayed on the right side
  ON: tags/labels are dislayed on the right upper side in green tag boxes

- *Identifier Overlays*
  OFF: no information is displayed on the squares, rhombs, etc. of the graph
  ON: information is displayed on the squares, rhombs, etc. of the graph

The labels/tags which were defined by the public can be adapted by creating individual ones and save them in an individual profile. To increase the usability, the user can furthermore save created graphs and share them with other people.

On this website the user may click him-/herself through the complete transaction timeline and build a graph with self-defined routes and properties (an example which was produced during the walkthrough can be seen in Figure 4.5). In comparison to Blockseer, PlanB generates graphs automatically depending on the input, instead of letting the user create it. This has the advantage of showing the user, in a prompt manner, correlations between specific addresses.



Address	18Dc7cBFRFnG9HyHrVpfvDdQbrgUz5Zcw1
Received	861.77547024 BTC
Spent	-861.77547024 BTC
Balance	0 BTC
First Seen	03/11/2013 08:46
Latest Action	03/11/2013 08:46
Transaction Count	2

Figure 4.5: Path of direct outgoing transactions

4.2.2 BlockChainViz

The BlockChainViz was published 2016 by students of the Italian Università degli Studi di Perugia. They offer a free test version on their homepage (http://www.dmi.unipg.it/blockchainvis/vis.php – walkthrough 10-08-2017) with a limited access to the Blockchain data and reduced functionality. It aims to provide users with a visual analysation tool to inspect the transaction of the Bitcoin network. Bistarelli et al. used visual analytics techniques to filter the available information according to user needs for a better insight.

BlockChainViz offers three kinds of visualizations which can be selected on the main page (see Figure 4.6). The options are :

- *Single Transaction:* The user searches for a specific transaction hash value of interest. The graph displays the transaction with all its belonging input and output addresses.

- *Address Transaction:* The user searches for a specific Bitcoin address and the transactions in which it was involved either as sender or receiver.

- *Archipelago:* This graph shows so-called “islands” (within an island all nodes are connected, but with no connection to any other node or super-graph) and enables the user to filter those for further inspection. The data filters influence the following four factors of drawn graph elements: the block interval concerning its date or height, the amount of transactions, the Bitcoin value within an island and the number of miners of each island.

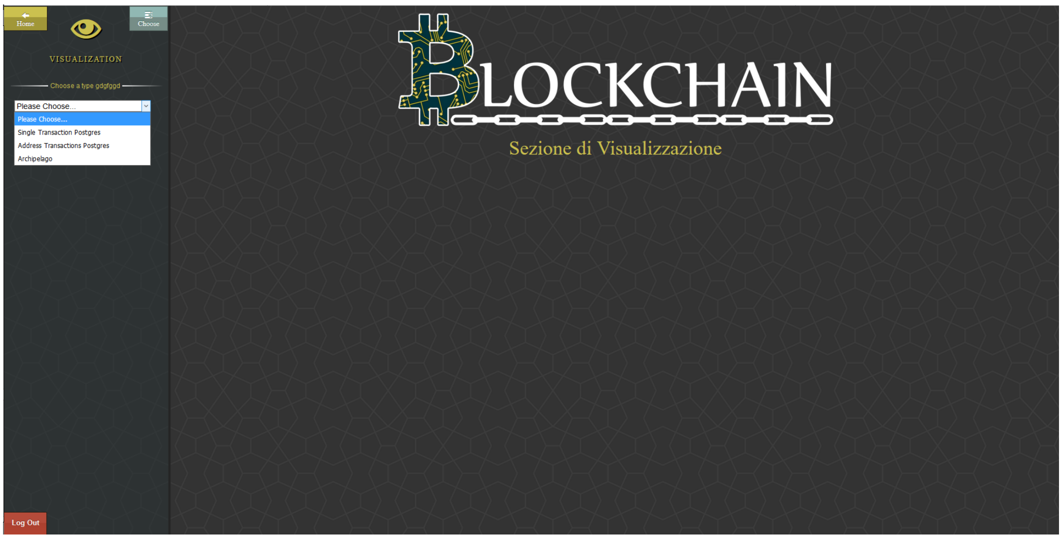


Figure 4.6: Home view of the BlockChainViz application – type selection

In the beginning the user has to decide which visualization type he/she wants to explore. To ensure all possibilities are taken into evaluation, they will be tested one after another. The single transaction option was picked therefore as the first choice. Hereby the user has to write the hash value of the transaction of interest into the provided field. After pressing the show button a graph with several addresses and one transaction in the middle is displayed (see Figure 4.7).

The application furthermore provides the opportunity to filter the displayed graph and/or highlight special characters within the system. A legend of the symbols is positioned on the right bottom of the page and the filters on the left side.

The address transaction option follows a similar work-flow, with the difference of searching for one address. It provides the user with a similar graph, however, this time the big node (address) is surrounded by the small nodes (transaction) it was part of (see Figure 4.8). By hovering over the the connecting lines between transactions and addresses, they dye it either red if the address functions as input, or blue if it functions as output address.

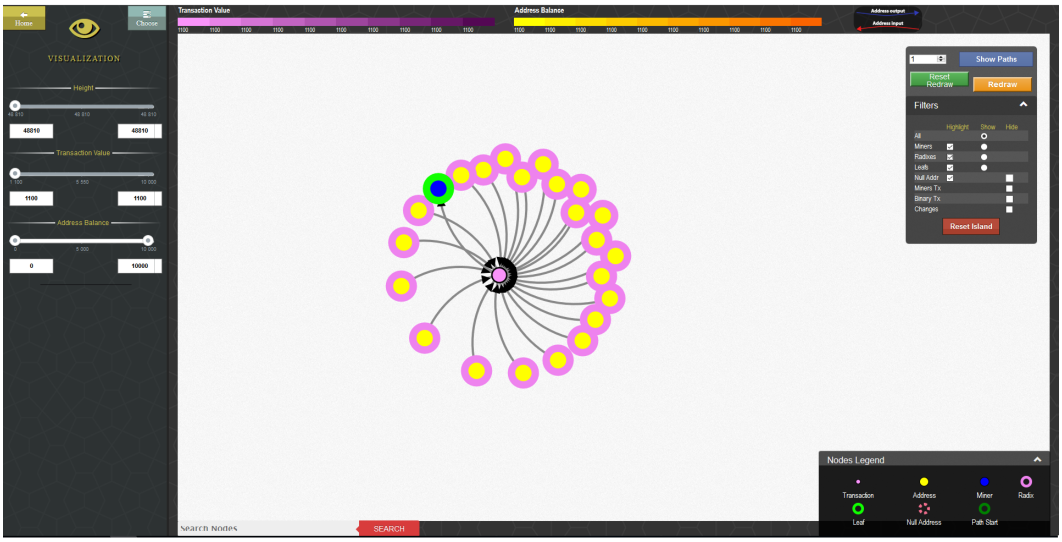


Figure 4.7: Single Transaction – the big nodes display the addresses and the small node in the middle the transaction

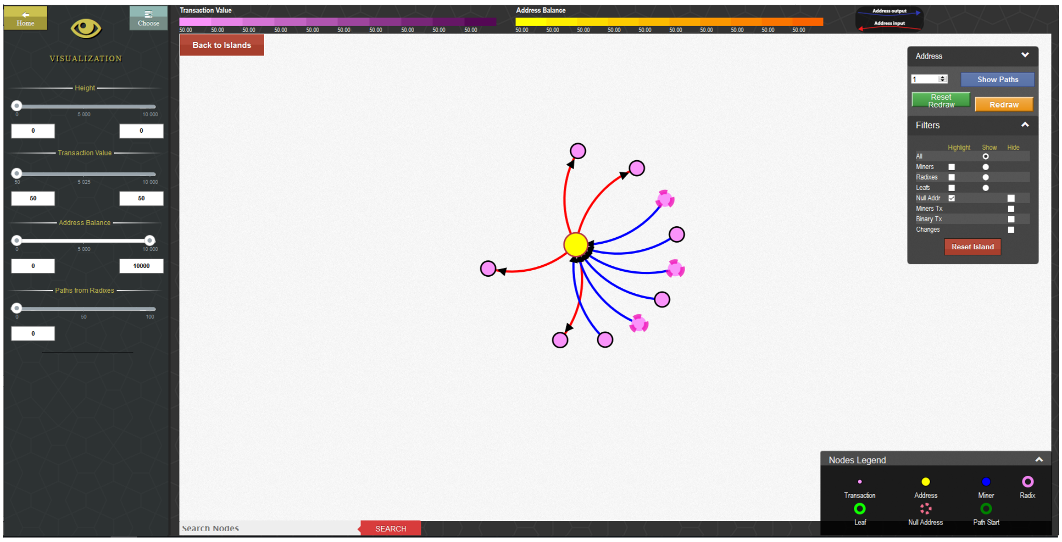
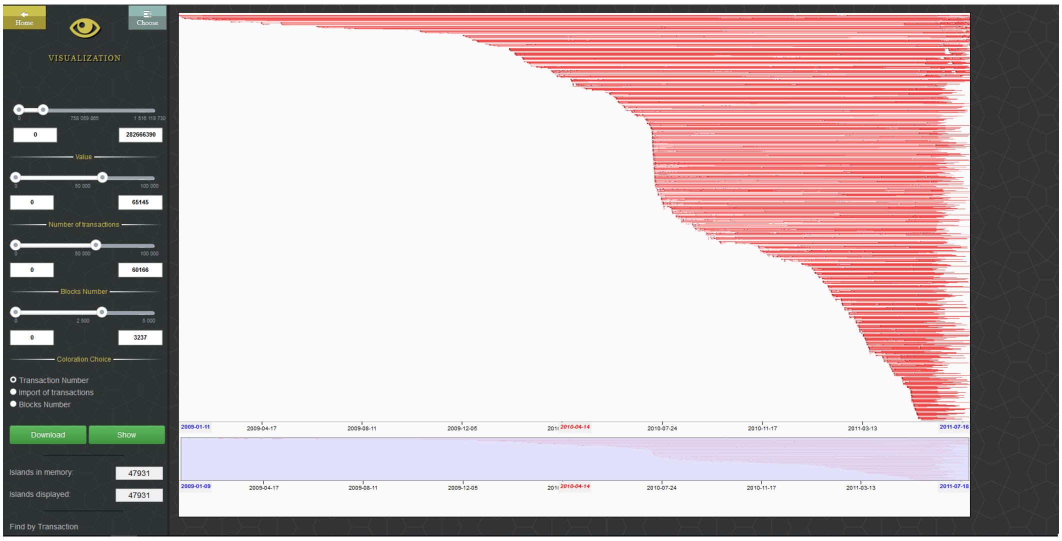


Figure 4.8: Address Transaction – the address was a sender when the transaction line is red and a receiver if it is blue

The archipelago was not fully provided in the test version; however, it displayed a bar-like chart of the requested data (see Figure 4.9). The graph enables the user to zoom inside and click on different transactions. This redirects to another view displaying the transaction in a line chart. As the visual output, which can be seen in Figure 4.10 and is described in [39], looks different this graph cannot be described in this section.

BlockchainViz is a good way to explore the Blockchain. The viusalization techniques are



0 – 758 059 865 – 1 516 119 730: 0 – 282666390

Value: 0 – 50 000 – 100 000: 0 – 65145

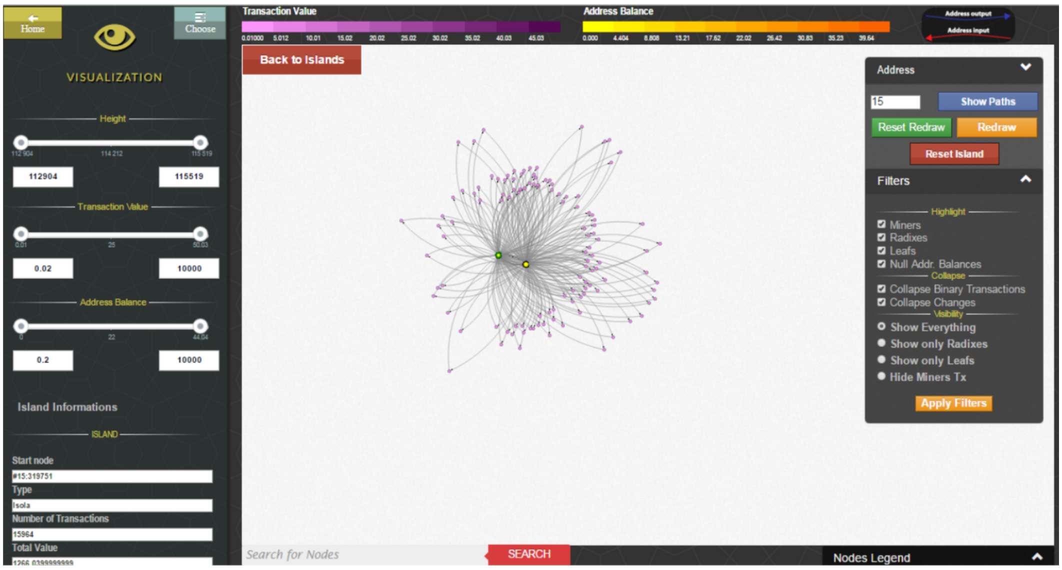
Number of transactions: 0 – 50 000 – 100 000: 0 – 60166

Blocks Number: 0 – 2 500 – 5 000: 0 – 3237

Coloration Choice
○ Transaction Number
● Import of transactions
● Blocks Number

Islands in memory: 47931
Islands displayed: 47931

Figure 4.9: Archipelago – result as it was displayed with the test version



Height: 112 904 – 114 212 – 115 519: 112904 – 115519

Transaction Value: 0.01 – 25 – 50.03: 0.02 – 10000

Address Balance: 0 – 22 – 44.04: 0.2 – 10000

Island Informations

ISLAND
Start node: #15:319751
Type: Isola
Number of Transactions: 15964
Total Value: 1266.0399999999

Address: 15

Filters
Highlight
☑ Miners
☑ Radixes
☑ Leafs
☑ Null Addr. Balances
Collapse
☑ Collapse Binary Transactions
☑ Collapse Changes
Visibility
○ Show Everything
● Show only Radixes
● Show only Leafs
● Hide Miners Tx

Figure 4.10: Archipelago – result as it was displayed with the test version

similar as the same web visualization tool is used. In comparison to PlanɃ, however, the correlation between two addresses is not easy to find and only possible after combining numerous filters and the knowledge of the user. Furthermore, BlockChainViz does not provide a multi-view of graphs on one side as PlanɃ does.

4.3 Design

After the first two phases of the SDLC the first sketch(es) of the prototype were developed. In the beginning the focus of attention was, as can be seen in Figure 4.11, to include as much as possible in the User Interface (UI). Furthermore, tabs were considered for individual graphs and numerous sliders to specify the requirements/filters.

After several revisions we decided to simplify the UI in order to increase the usability. Hereby the textual information was reduced to a minimum and the different graph visualizations were organized in a drop-down menu. Inspiration for the prototype came from Chang et al. [43] who developed a visual analytic application whose purpose is to discover the subjacent network of Bitcoin. It focuses on suspicious activities concerning (traditional) bank wire transactions. The visualization is performed by providing multiple connected views to the user, which was also important for our prototype.

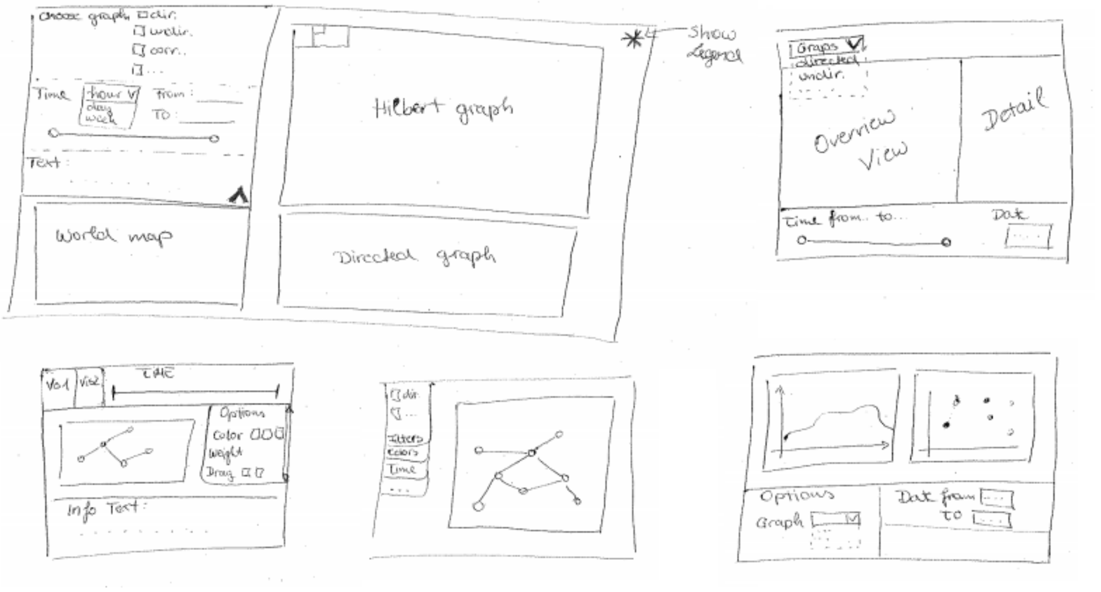


Figure 4.11: First sketches for the prototype and development of the name Plan₿

As soon as the basic concept for the prototype was determined, the name was decided and a final mock-up was drawn (see Figure 4.12). The mock-up is the basis for the frontend and was adopted into the final web-application nearly in total (screenshots of the final application can be found in chapter 6).

4.4 Prototyping

Prototyping is a software development process which puts the main focus on a fast and basic implementation to demonstrate the contemplated functionality. It is a fast-forward way to confirm the capabilities of a system and the understanding of the requirements

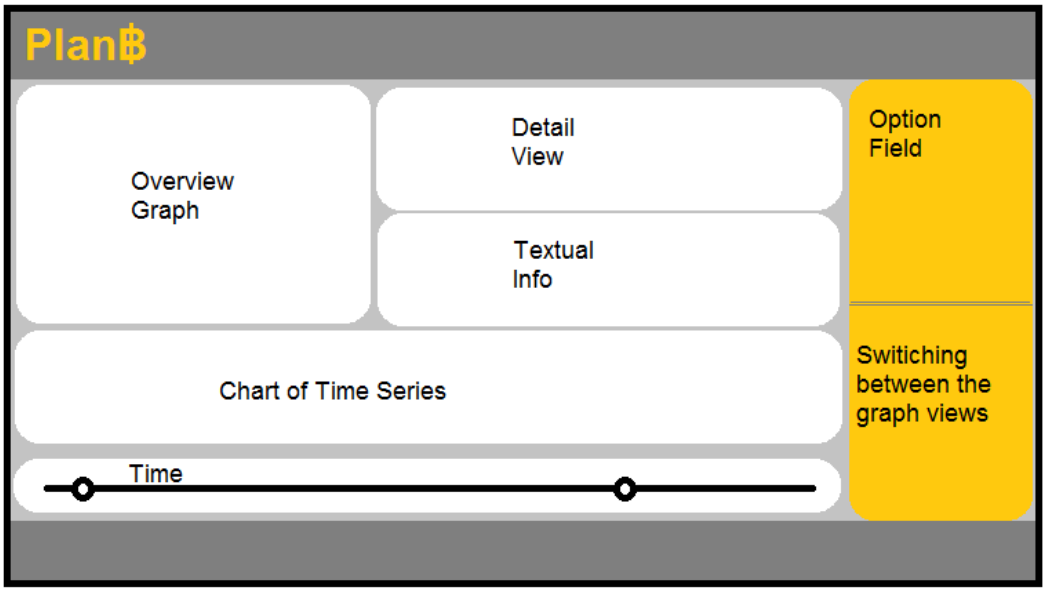


Figure 4.12: Final mock-up for web-application Plan₿

[28]. Hereby it is important to have requirements and design concepts which should be implemented. For our prototype Plan₿ we decided on the following requirements:

- Correlation graph: displays all/the shortest path between to address nodes
- Time-based filter option: displaying only the transactions which were done within a specific period of time
- Multiviews: showing the data in different graphs which are correlated to get more information of the visualized output

CHAPTER 5

Implementation Details and Technologies

A prototype was implemented based on the mock-up and the requirements defined in section 4.4. The prototype provides visualization capabilities for the Bitcoin transaction network. This chapter gives a summary about the implementation and technological details of Plan₿.

5.1 Platform and Architecture

The prototype is implemented on the one hand in Python and on the other hand in webbased languages such as JavaScript. It is split into two parts: a backend and a frontend. The backend was implemented by Matthias Gusenbauer and will be only discussed very shortly to give a better understanding of the main structure (for further details see the master thesis of Matthias Gusenbauer, soon to appear). The database used to store the complete transaction data available in the Blockchain is the graphbased system Neo4J [21]. For the internal temporary storage and processing of the data (e.g.: path queries and aggregation) the Python based Graph-Tool [22] module is used. The command and data channel is implemented via a websocket by Autobahn [2], which provides an open-source implementation of the Web Application Messaging Protocol (WAMP). The connecting link between the backend and the frontend is the Twisted [26] webserver, which is an event-driven networking engine simplifying custom network applications.

The frontend is a Chrome based website, which is implemented with standard web-technologies and the visualization library D3.js (more information can be found in section 5.5). An overview of the general structure of Plan₿ is displayed in Figure 5.1.

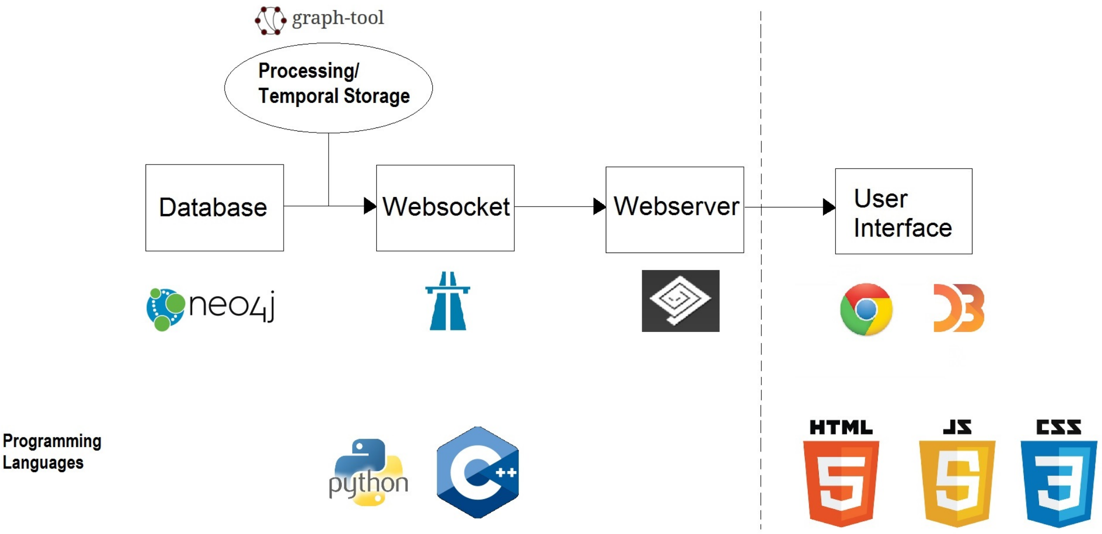


Figure 5.1: Architecture of Plan₿

5.2 Data Format and Structure

The information of each block, transaction or individual Bitcoin Address is stored in the Neo4J database. To get a visual access to the data, they are processed (interrogated and aggregated) and afterwards send via a websocket to the glsui. The data is transported using JavaScript Object Notation (JSON) files [42]. JSON is a text based and language independent format to interchange structured data. Those data structures are organized as a collection of information which are called *objects*. Each object consists of one or more *members* which store the name/value *pairs*. In Figure 5.2 a general JSON file with its typical form is displayed.

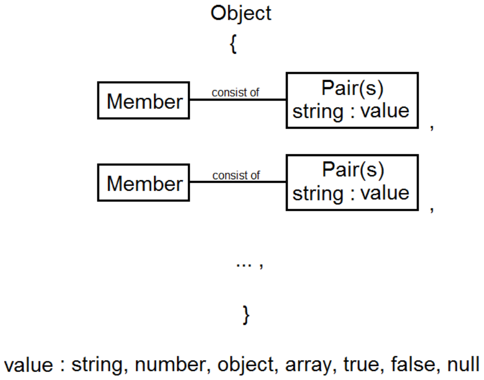


Figure 5.2: JSON general file form

The advantage of JSON in comparisson to XML is a much higher parsing efficiency [31] and its broad usage for web applications. Furthermore it is language independent, however it uses conventions which are familiar to C languages (e.g.: C, C++, C#, Java, Python,...). By using this data format, the frontend and backend are modular and can be exchanged without having to change the whole structure as JSON is supported by all modern programming languages [24]. For future implementations it furthermore gives the user the possibility to store the graph data and simply reload them later on if needed.

Plan₿ has the following two different JSON file structures:

- *Request structure*: Depending on the users input the client assembles a request which is send to the server for processing. Each request consists of at least one member, which is called "general". It contains information about the request type, if the data should be streamed and the data capacities. Optionally aggregation stages and a "query" member, containing details on the requested information, are determined. In Listing 5.1 a cancel request and start request for a simple path search (details on the different visualizations are described in section 5.6) is shown.

```
---------------------------------------------------------
Cancel Request
---------------------------------------------------------

{
  "general": {
    "type": "cancel",
    "streaming": false, //streaming is stopped
    "data": {
      "capacity": 5,
      "used": 2
    },
    "clear_cache": true //cache is cleared
  }
}

---------------------------------------------------------
Start Path Search
---------------------------------------------------------

{
  "general": {
    "type": "new",
    "streaming": true,
    "data": {
      "capacity": 5,
      "used": 2
    },
   {"type":"path",
    "truncate":1000,
    "subgraph_labels":["Address","Transaction"],
    "path_type":"all_shortest",
    "start":{
      "label":"Address",
      "params":
        {"hash"://Fill in hash value
        }
    },
  },
  "node":{
    "var":null,
    "label":"Address",
    "params":{"hash"://Fill in hash value
    }
  },
  "coi":null}
}
```

Listing 5.1: Example of a cancellation and a path finding request

- *Reply structure*: After the request is processed in the backend, the reply is send via the webserver to the client. There are two types of replies, on the one hand a status message and on the other hand a data message. The status message informs the client if the request was recieved correctly and if everything is ok in the backend. Data messages include the requested data for visualization with a header and the corresponding nodes, links and paths of the resulting graph (see Listing 5.2).

```
----------------------------------------------------------
Status Message
----------------------------------------------------------
{
 "header": {
   "message_type": "status",
   "status": "OK",
   "code": 3
   }
}
----------------------------------------------------------
Data Message
----------------------------------------------------------
{
  "header": {
    "message_type": "data",
    "multigraph": true,
    "hierarchy": false,
    "chunk_nr": 0,
    "node_types": ["block", "transaction", "address"]
  },
  "nodes": [{"Node information: id, hash, processed, delete,
     special-type and type"}],
  "links": [{"Link information of source, target, label and
     delete flag"}],
  "paths": [{"Paths which are found"}]
}
```

Listing 5.2: Example of the two reply structures: status and data message

The JSON reply structure is decoded into JavaScript readable strings in the frontend, to visualize the incorporated information. In Listing 5.3 a simplified example, how to listen to incoming messages and parse them, is provided.

```
//Listen for messages
 this.server.get_message = function (event) {
         var message = JSON.parse(event.data);
         if (message.header.message_type == "type") {
            ...
         } else {...}
      };
```

Listing 5.3: Example - Receiving and parsing JSON file

5.3 Communication

The communication between the front and the backend is done by sending and listening to JSON files transported via a twisted webserver. Twisted is an independent server and Python library which is used as the communication interface for our prototype. It supports the streaming of data, which allows us to continuously visualize the received information. Further crucial arguments for the usage of this networking engine are the event-driven and therefore resource preserving request handling and the programming language, Python. As the whole backend is written in Python it seems to be an obvious choice to go with the same language for the communication webserver. The current address of the twisted webserver is on localhost and lies on port 8080. To set up a connection to the webserver the following line of code is used:

```
var socket = new WebSocket('ws://'+ address + ':'+ port + '/'+ path);
```

The backend implementation is using the graph-tool library for the path analyzation. This library is only available in Linux, making it necessary to use a Virtual Machine (VM) for windows application users. The server of the VM is provided by SBA Research on bitvis17 and can be accessed on port 5228. The exact configurations can be found in the master thesis of Matthias Gusenbauer.

5.4 Web Integration

The prototype Plan₿ is implemented as a web-based application in Google Chrome. Although it is possible to access the application from other browsers as well, not all functionality can be guaranteed in them.

Plan₿ references the D3.js library to gain the complete functionality of current visualization technologies within the Google Chrome webbrowser. The web integration in combination with D3.js provides the user with the possibility to access the Blockchain data interactively without being affected by installing further plug-ins as Flash or Java Applets. The integration to the web browser was intuitively performed by the usage of html, javascript and Cascading Style Sheets (CSS) files.

5.5 D3.js

This library was developed by M. Bostock, V. Ogievetsky and J. Heer and was published in 2011 [41]. D3.js is the combination of the acronym of Data-Driven Documents and its implementation language JavaScript. With this library users are able to attach their data to Document Object Model (DOM) elements, which can be directly inspected and manipulated by applying (dynamic) transformations and generating/modifying the data. It is especially a very good choice when handling an enormous amount of data. The advantage of this library is the very minimal overhead, which supports dynamic behaviour and interaction with the data.

D3.js links visualization techniques with the contingency of current web browsers and their standards such as Hypertext Markup Language (HTML), Scalable Vector Graphics (SVG) and CSS. The visual result can be adapted in any desired way by simple styling them in CSS or binding datasets to SVG objects. D3.js provides many possibilities for graph visualization with integrated updating mechanism. Manipulations of the DOM elements are handled similar to jQuery, for example all elements <e>...</e> of the HTML code are selected to change the color as following:

```
d3.selectAll("e").style("color", "red"); //set all elements e to red
```

Furthermore, it still supports the individual manipulation of nodes by simply selecting their tag names, ids, attribute values, or class names.

5.6 Graphical Output

Plan₿ provides four different views for the user to get a better insight into the complex world of Bitcoin. The four views are: the general overview, the detail view, the textual view, and a statistical view. There are two foci of the graphical output, on the one hand, the path visualization between two specific addresses and on the other hand, the neighbourhood search extinguishing from only one address. Both visualizations can be manipulated depending on the period which the user wants to focus on. Both visualizations are implemented as a directed graphs in D3.js [41]. It is very important to take the direction into consideration as it shows the route of the cash flow. The data is streamed continuously depending on the amount of paths found and the available capacity for displacement (this is determined by the client – see the request structure in section 5.2).

The path search determines whether there is a likely connection between two addresses ot not. For each found path, new nodes are added to the graph, depending on the setting (all – transactions, blocks and addresses are displayes; addresses – the addresses are the nodes and the transactions the link between them). In Listing 5.4 the update method can be found, which extends the graphical output with new paths if one/some are found.

```

function update() {

 //...

 //-------------Link - Transaction -------------------------

 // UPDATE
this.links = this.svg.selectAll(".link")
   .data(transactions, function(d) { return d.source.id + "-" +
      d.target.id; });

 // EXIT
this.links.exit().remove();

 // ENTER
this.links.enter().append("line")
   .attr("class", "link")
   .attr(

//...

 //-------------Node - Address -------------------------

 // UPDATE
 this.nodes = this.svg.selectAll(".node")
   .data(address, function (d) { return d.id; })
   .attr("r", this.node_size);

 // EXIT
this.nodes.exit().remove();

 // ENTER
this.nodes = nodes.enter().append("g")
   .attr("class", "node")
   .attr("stroke", this.address_type)
   .attr("stroke", this.address_type)
   .attr("fill", this.address_color)
   .on("mouseover", mouseover)
   .on("mouseout", mouseout)
   .call(force.drag);

//...

};
```

Listing 5.4: Directed graph visualization - simplified version of update method

The start and the end point of the correlation search is marked with a black (start) and blue (end) circle. The transactions are displayed with black lines and an arrow which determines the cash flow direction. An example of a small connection graph can be seen in Figure 5.3.

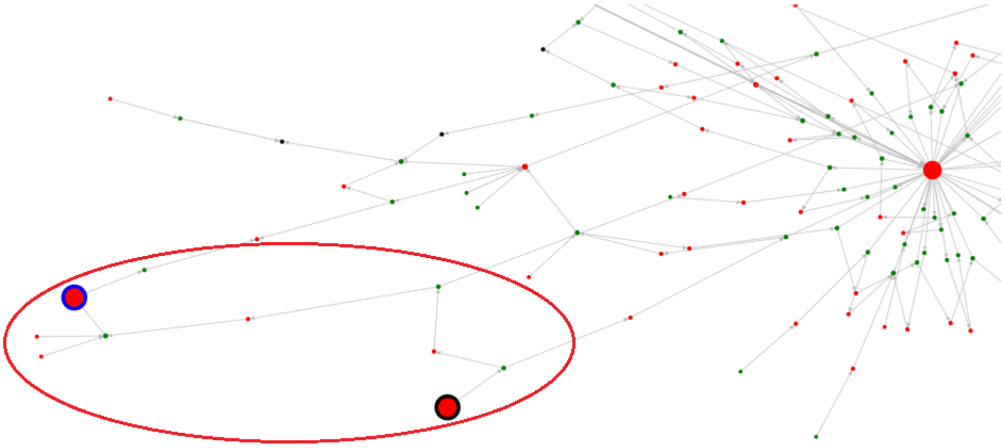


Figure 5.3: Example of connection graph – with a black start and blue end address

To ensure a high interactivity and responsiveness of the prototype the path-search is interrupted after 5 seconds, when no result is found. In this case the user is informed by a dialogue field which states: “No direct path within a plausible distance between the two addresses was found”.

The neighbourhood search searches all the connections starting from a user defined address. Hereby the starting address is marked with a blue circle (see Figure 5.4.

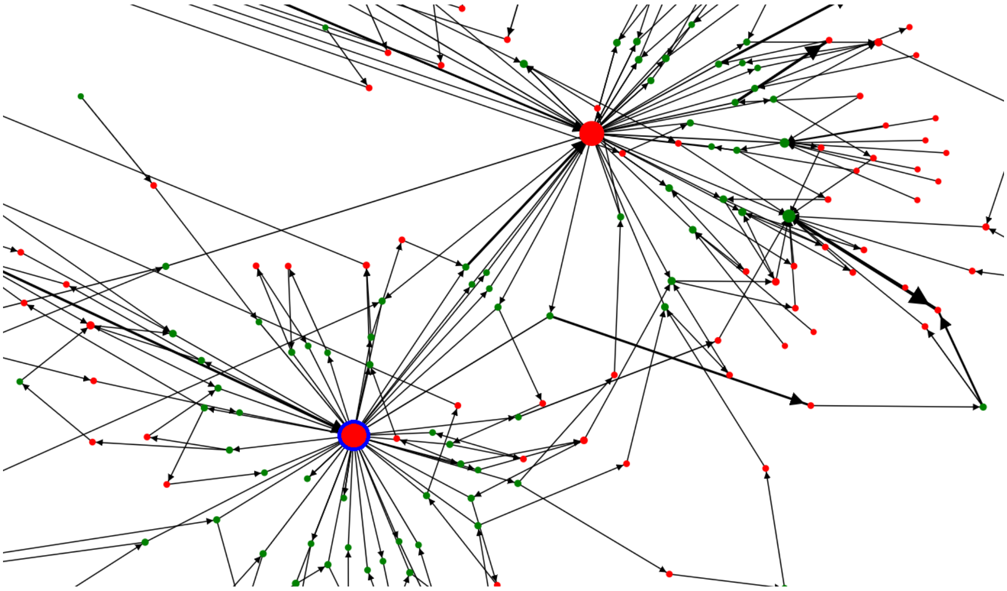


Figure 5.4: Example of neighbourhood graph

Ihe user can toggle between showing all Blockchain data (including the transactions, the blocks and the addresses) and the visual displacement of only the addresses with the transactions as links. To ensure a good look on the nodes and their links a force is applied which structures the graph in an even way. As the force is applied asynchronously it has to be handled separately (one after the other) due to dragging. The user is enable to select nodes and drag them within the view-space to get a better insight in structures which are too close together.

In Listing 5.5 the shortened code for enabling the dragging and simultaneously using a force is shown. It is important to keep in mind that the force is asynchronous and has to be disabled for the complete graph and afterwards applied again to the all the nodes except the dragged one.

```

//give node the dragging behaviour
var nodes = svg.append("g")
    .attr("class", "nodes")
   .selectAll("circle")
   .data(graph.nodes)
   .enter().append("circle")
    .attr("r", 5)
    .attr("fill", function(d) {
         return color(d.group); }) //depending if address, transaction or block
    .call(d3.drag()
       .on("start", dragstart)
       .on("drag", dragging)
       .on("end", dragend));


   function dragstart(d, i) {
      force.stop() // stops the force (no positioning of graph)
   }

   function dragging(d, i) {
      d.px += d3.event.dx;
      d.py += d3.event.dy;
      d.x += d3.event.dx;
      d.y += d3.event.dy;
      tick(); // function call for force
   }

   function dragend(d, i) {
      d.fixed = true; // dragged node should not move
      tick();
      force.resume();
   }

//------------Force------------------

var forces = d3.forceSimulation()
   .force("link", d3.forceLink().id(function(d) { return d.id; }))
   .force("charge", d3.forceManyBody())
   .force("center", d3.forceCenter(width / 2, height / 2));

  forces.nodes(graph.nodes)
    .on("tick", tick);

 forces.force("link")
    .links(svg.links);

```

```
//...

  //function of force
   function tick() {
   link
      .attr("x1", function(d) { return d.source.x; })
      .attr("y1", function(d) { return d.source.y; })
      .attr("x2", function(d) { return d.target.x; })
      .attr("y2", function(d) { return d.target.y; });

   node
      .attr("cx", function(d) { return d.x; })
      .attr("cy", function(d) { return d.y; });
}
```

Listing 5.5: Code snippet of dragging nodes and applying a force

CHAPTER 6

Results

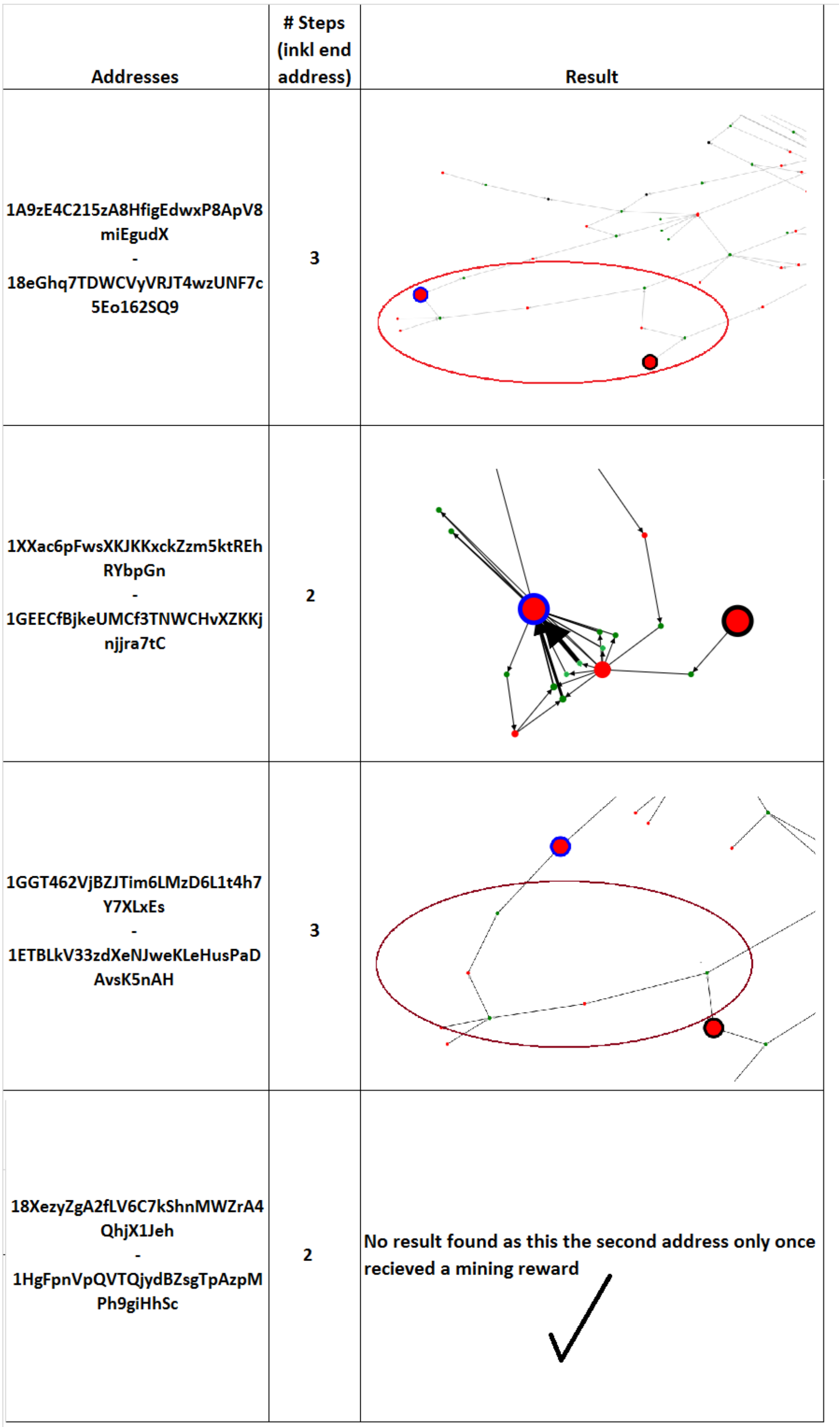
The visualization approaches (section 5.6), as mentioned before, were implemented as a web-based prototype using D3.js [41]. In order to provide the user with meaningful and easy to interpret results, detail views and different visualization techniques were tested and added/replaced during the development process. This iterative process helped to improve the prototype continuously. To proof the visual outputs, small test scenarios were set up too.

In the beginning the test scenarios were set up according to information which can be found on Blockchain.info [15]. Next those scenarios were tested with our prototype to proof the functioning work-flow and the correctness of the results. Afterwards multi-views were set up to provide a closer look into specific details of the result(s). The last feature of Plan₿ is the time-based parallel coordinate plot to get a completely novel view of the Bitcoin block data. The following sections describe how the prototype was tested and its final appearance with a prototype walkthrough.

6.1 Testing

The test scenarios were set up to provide feedback of the correct functionality of Plan₿. Several small tests were performed which could be verified manually. Following a short description of one test scenario is provided to give a insight in the testing process.

To get a meaningful path test the start address was selected radomly and the end address was determined by research. The research was performed on Blockchain.info [15], by following a short path manually, with a predefined length (between two to 5 nodes in between), leading to the test end address.



Addresses	# Steps (inkl end address)	Result
1A9zE4C215zA8HfigEdwxP8ApV8miEgudX - 18eGhq7TDWCVyVRJT4wzUNF7c5Eo162SQ9	3	
1XXac6pFwsXKJKKxckZzm5ktREhRYbpGn - 1GEECfBjkeUMCf3TNWCHvXZKKjnjjra7tC	2	
1GGT462VjBZJTim6LMzD6L1t4h7Y7XLxEs - 1ETBLkV33zdXeNJweKLeHusPaDAvsK5nAH	3	
18XezyZgA2fLV6C7kShnMWZrA4QhjX1Jeh - 1HgFpnVpQVTQjydBZsgTpAzpMPh9giHhSc	2	No result found as this the second address only once recieved a mining reward ✓

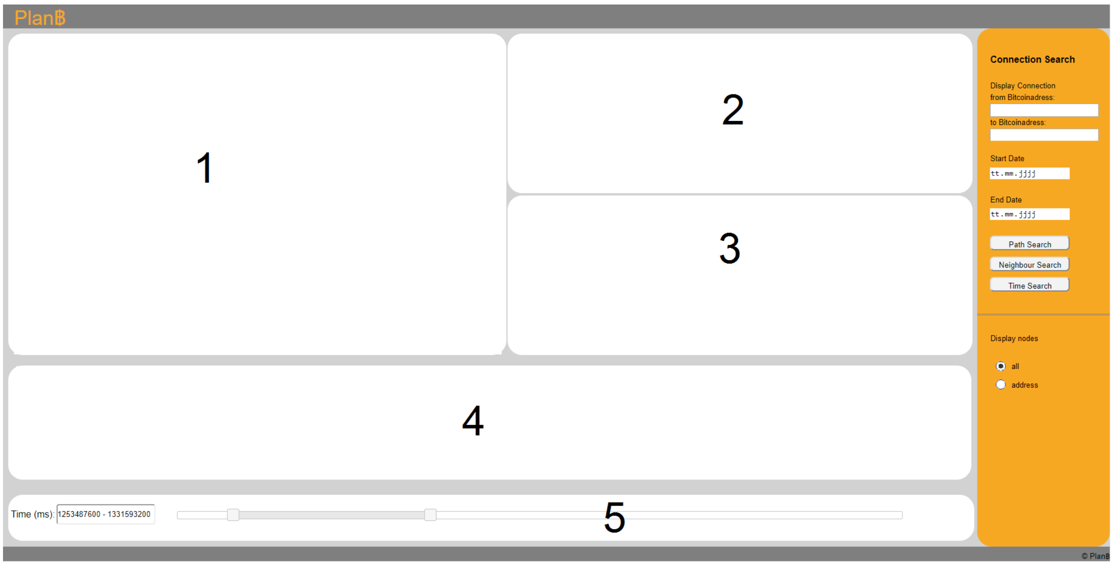
Figure 6.1: 4 Test results of testing the path finding between two addresses

The two addresses have to be entered into the system to get the visual output, which should contain the path between the two addresses. In Figure 6.1 the results of the visual output, with various example addresses, are displayed. In the first version tests the red dots show the addresses and the green ones the transactions. By hovering over the different nodes their information is displayed in the textual information field in PlanɃ and can be verified. As it is easily visible the paths, with a predefined number of addresses between the start and the end node, were correctly found.

6.2 Prototype Walkthrough

In this section the two visualization options are shown and a description of the correct usage of the web interface is provided.

PlanɃ's main window consists of four different view windows. In Figure 6.2 a screenshot of prototype is visible. Field 1 provides the user with the main view on the graph, were he/she is able to move the graph in various directions. The field with number 2 is activated as soon as the user selects a certain area from the overview graph to get a detail view of it. Beneath the 3. field displays information of the graph as soon as the user hovers with the mouse over a node of interest. The field number 4 gives the user a line chart with the number of transactions made within the certain period. In the 5. field the time slider enables the user to limit the search results within time. On the right hand side, marked with number 6 is the control panel for the user.



Time (ms): 1253487600 - 1331593200

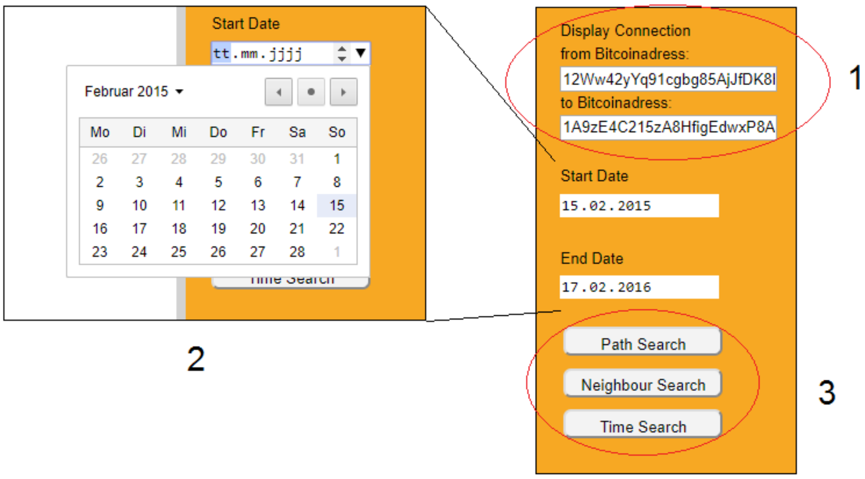
Display nodes

● all

○ address

Figure 6.2: Homescreen of PlanɃ

A closer look at the control panel can be seen in Figure 6.3. First the user has the option to decide if he/she wants to have a closer look at certain addresses. Than there is the possibility to restrict the time frame in which the path of interest should lie. If there is no date inserted the whole Blockchain data is taken into account. In the end there are three options to search. The path search simply searches for all paths available between the inserted addresses. Neighbourhood search simply takes the “from address” and grows a graph with this address as the middle. The last search displays all transactions performed within a certain period of time, which the user can determine. This type of connection search should help the user if he/she does not know where to start and only has a idea about the period of time.



Start Date
tt.mm.jjjj

Februar 2015

Mo	Di	Mi	Do	Fr	Sa	So
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	1

Display Connection
from Bitcoinadress: 12Ww42yYq91cgbg85AjJfDK8l
to Bitcoinadress: 1A9zE4C215zA8HfigEdwxP8A
Start Date 15.02.2015
End Date 17.02.2016

Figure 6.3: Close-up of the control panel.

The major focus of this thesis is on the path finding between two addresses. In the following screenshot (Figure 6.4) the result is shown, when a user enters two different addresses and presses on “Path Search”. The result shows all possible connections between the requested Bitcoin addresses including transaction (green), block (black), and address (red) nodes.

If the other option “address” was selected for the same input. In Figure 6.5 the output of this request can be seen. The major difference in representation is, that now the transactions are the arrows between the address nodes.

The neighbourhood search should help the user to get a better feeling for the connections an address has in the Bitcoin system. In Figure 6.6 a neighbourhood search was performed. Here the extinguishing address is marked with a blue circle and all its connections are loaded continuously. Another neighbourhood search was performed with the option of only displaying the addresses as nodes, visible in Figure 6.7.

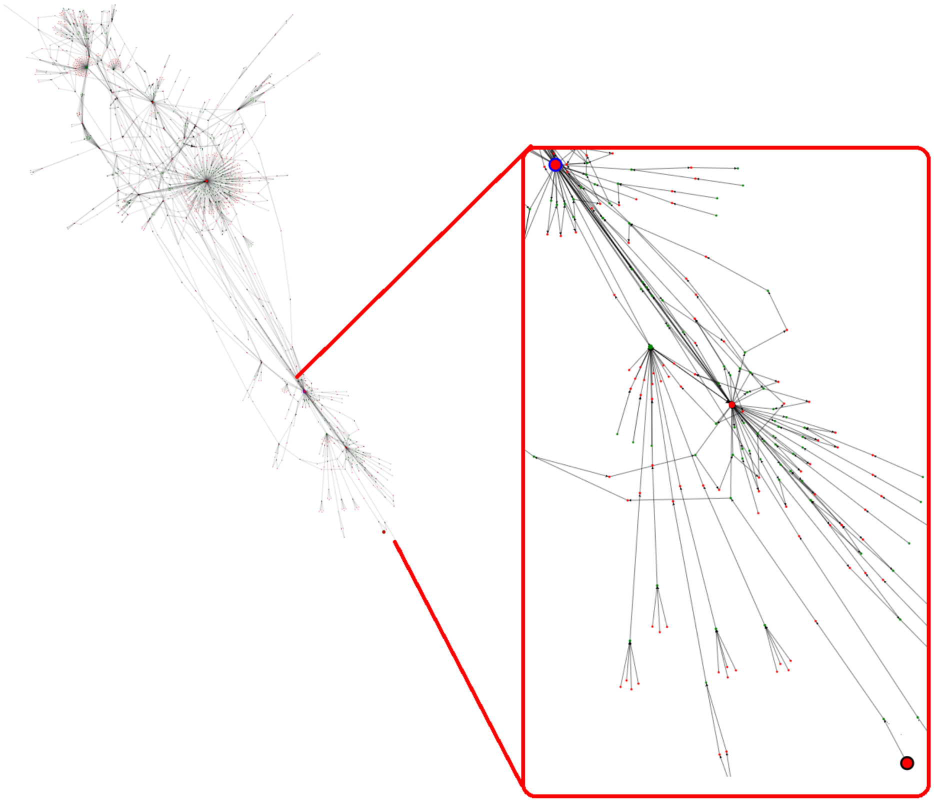


Figure 6.4: Path search result with all transactions, blocks, and addresses displayed as nodes)

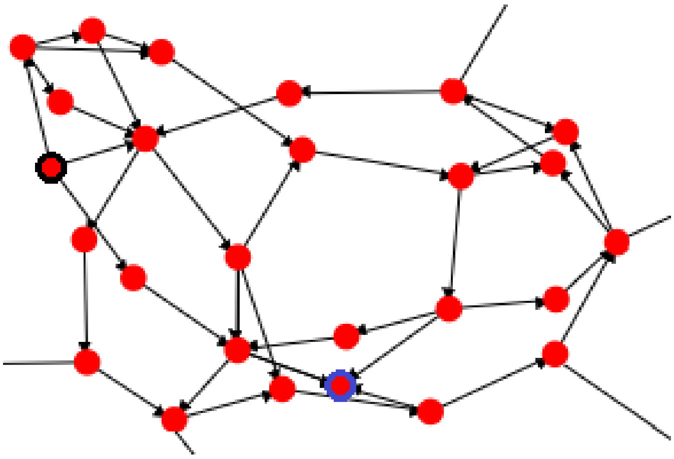
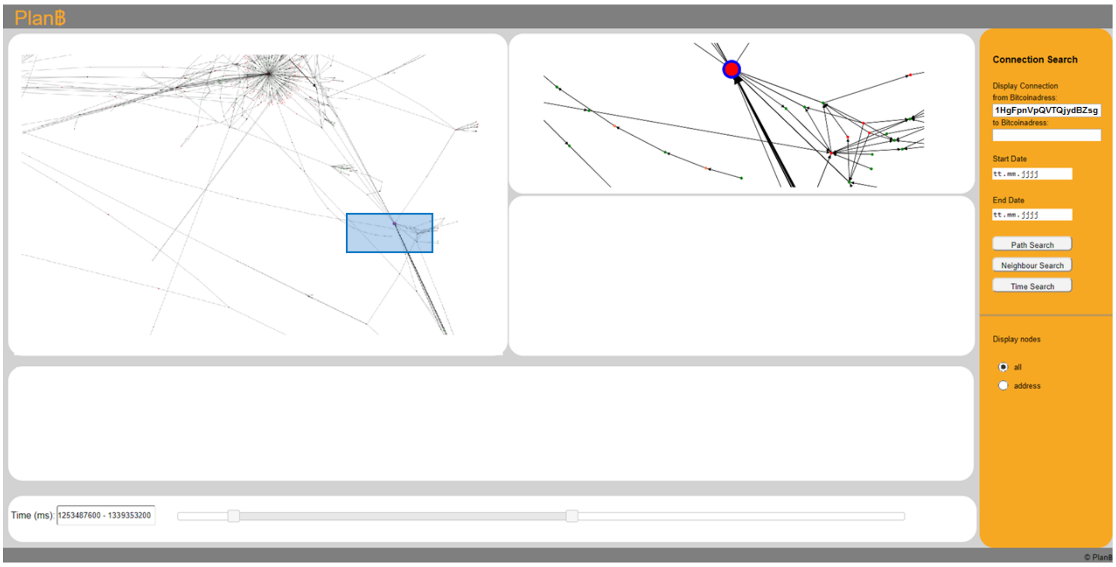


Figure 6.5: Path search result with addresses only



Connection Search

Display Connection
from Bitcoinadress: 1HgFpnVpQVTQjydBZsg
to Bitcoinadress:

Start Date: tt.mm.jjjj

End Date: tt.mm.jjjj

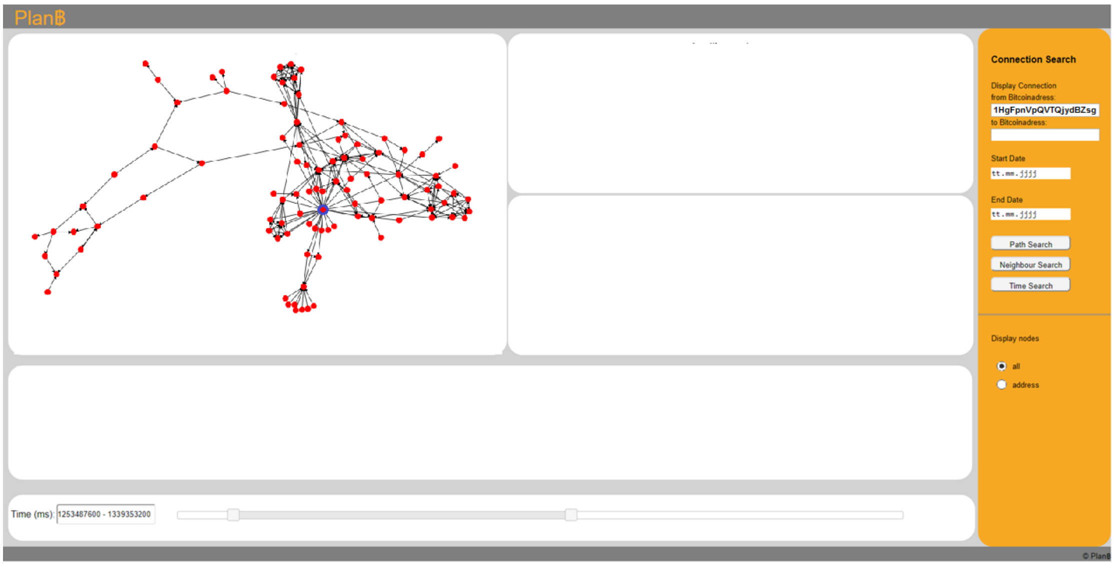
Display nodes

● all

○ address

Time (ms): 1253487600 - 1339353200

Figure 6.6: Neighbourhood search result with all nodes displayed



Connection Search

Display Connection
from Bitcoinadress: 1HgFpnVpQVTQjydBZsg
to Bitcoinadress:

Start Date: tt.mm.jjjj

End Date: tt.mm.jjjj

Display nodes

● all

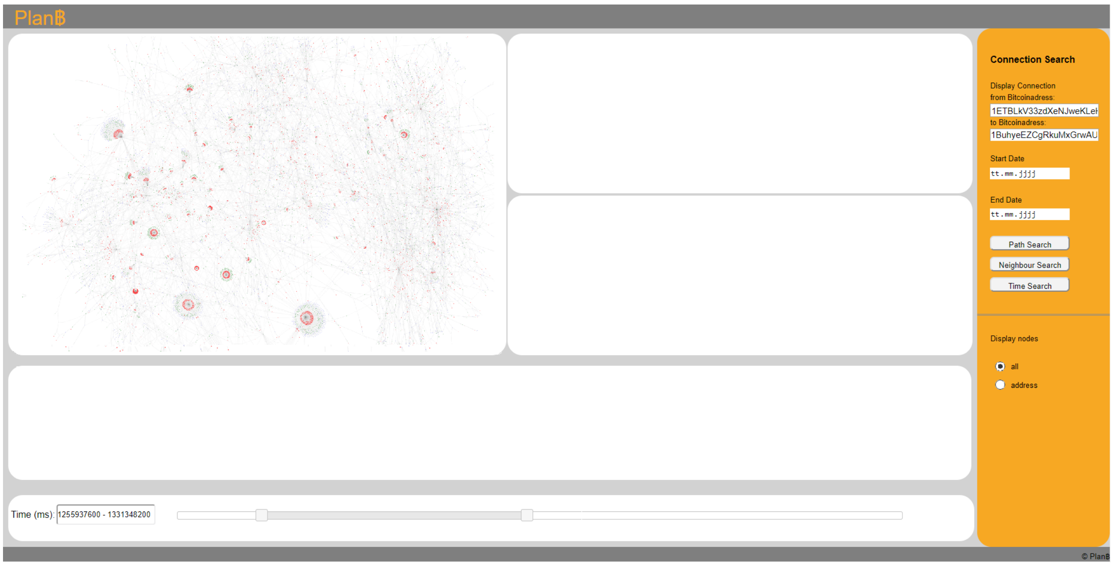
○ address

Time (ms): 1253487600 - 1339353200

Figure 6.7: Neighbourhood search result with addresses as nodes and transactions as arrows

6.2.1 Use Case

A security/forensic analyst gets a case where he/she has to have a look at a suspected person of shifting money to a known address abroad (in this case the address is random – not known for abroad connection) without paying his/her taxes. In the beginning the analysts performes a path research to see whether a connection exists between the two addresses. The starting and end address lie far apart in the results (see Figure 6.8), which means that there is somehow a connection however it is a very distant one and therefore very unlikely.



Display Connection
from Bitcoinadress: 1ETBLkV33zdXeNJweKLe
to Bitcoinadress: 1BuhyeEZCgRkuMxGrwAU

Start Date: tt.mm.jjjj

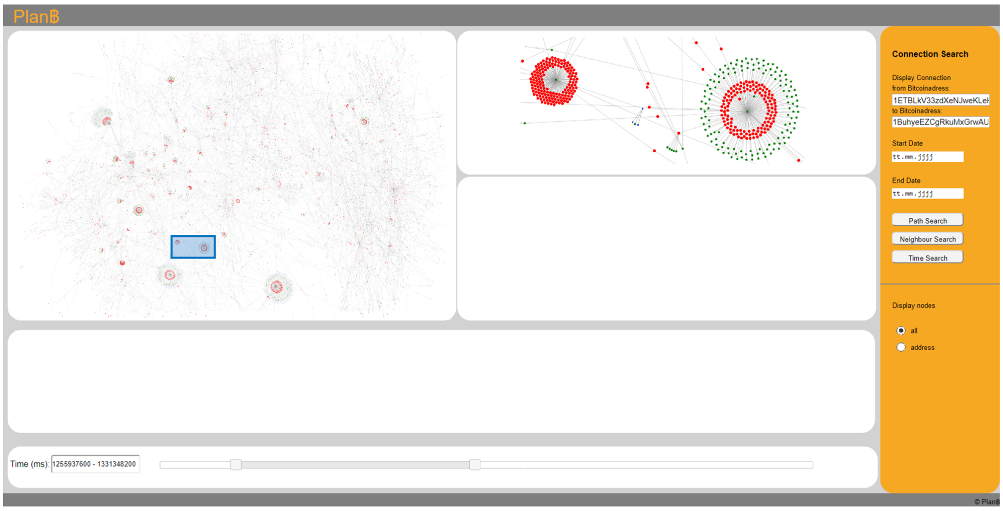
End Date: tt.mm.jjjj

Display nodes
● all
○ address

Time (ms): 1255937600 - 1331348200

Figure 6.8: Use case screenshot – first step

Afterwards the analyst wants to inspect certain areas of the overview. He marks an area of interest and it get displayed in the detail view (see Figure 6.9). Here two transactions are displayed with loads of addresses involved. By hovering over a node some textual information is displayed.



Display Connection
from Bitcoinadress: 1ETBLkV33zdXeNJweKLe
to Bitcoinadress: 1BuhyeEZCgRkuMxGrwAU

Start Date: tt.mm.jjjj

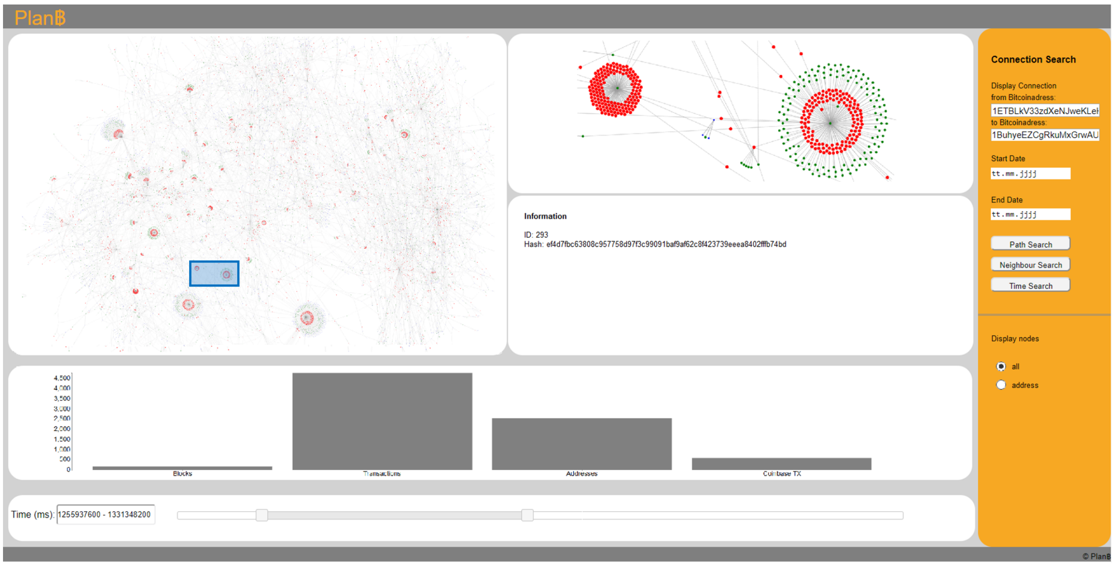
End Date: tt.mm.jjjj

Display nodes
● all
○ address

Time (ms): 1255937600 - 1331348200

Figure 6.9: Use case screenshot – second step

In the end the analyst wants to get a better idea of the amount of data displayed in the overview. He/She clicks therefore in the overview field and selects “Info”. In the lowest field a statistical bar graph is displayed stating how many blocks, transactions, addresses and coinbase (simple mining rewards) transactions are currently visible (see Figure 6.10).



Information

ID: 293
Hash: ef4d7fbc63808c957758d97f3c99091baf9af62c8f423739eeea8402fffb74bd

Connection Search

Display Connection
from Bitcoinadress: 1ETBLkV33zdXeNJweKLeh
to Bitcoinadress: 1BuhyeEZCgRkuMxGrwAU

Start Date: tt.mm.jjjj

End Date: tt.mm.jjjj

Display nodes

● all
○ address

Time (ms): 1255937600 - 1331348200

Figure 6.10: Use case screenshot – third step

CHAPTER 7

Discussion and Conclusion

The following chapter summarizes the results of our approach. It critically reflects the outcome and discusses the limitations. Some final thoughts and planned future extensions of the prototype are provided at the end of this chapter.

7.1 Critical Reflection

The introduced prototype is a novel approach for visualizing the connection of transactions within the network of Bitcoin. Plan₿ is a web-based prototype (Google Chrome), making it easy to access and to extend for further usages. Its interactivity enables the user to take a closer look at regions of interest and get statistical outputs on the displayed graph. To guarantee an optimal graphical output for the user the prototype gives the user the opportunity to drag the nodes of the graphs. Furthermore, textual information is provided when hovering over the nodes.

The advantage of continuously streaming the data, is the avoidance of long waiting periods for the user. The streamed data is immediately shown in the user interface and is updated continuously. On the opposite site, however is the time which has to be waited until all the data is loaded. Furthermore there is no visual output, how much information is already loaded and how much is left, as it is hard to determine.

The backend is correntiy unfinished leading to some method lacks (sometimes). As it is continuously updated and expanded (request, aggregation and unfinished Blockchain data-loading) the frontend often also needs to be updated for the new features. The complete backend is implemented in a Linux environment making it necessary to connect to it via a virtual machine. A stable version of the prototype will be released approximately in the beginning of summer as an open source project on github.

7.2 Limitations

Although Plan₿ provides a good basis to get an insight of the connections between addresses, it has some limitations. The graph representation of the path search provides the user with multiple arrows, displaying the cash flow. Multiple transactions between two nodes are represented as one path arrow (overlaying each other). Therefore, it is hard to determine how many transaction have been made between two addresses, however in favour of a better (not too crowded) visual output we merged them.

Another shortage of the prototype is the textual output in address display mode, it only provides information about the node (address). The transaction information is neglected, however in the view displaying all nodes the information is accessible. In general the textual output is under inspection, to determine better ways for the representation.

The path search has two major limitations, on the one hand the limited time and on the other hand the "likeliest" path is not determined. As we fixed a time limit for the path search it can happen that no path is shown although there would be a connection. The likelihood of a found path is not determined and therefore only the cash flow is taken into consideration if a path is displayed or not.

7.3 Future Work

There are still many possibilities to utilize the Blockchain network in new ways. Therefore we plan to further improve and expand the prototype Plan₿.

The first feature extension we plan to implement is a likelihood analysis with more parameters. Hereby, the transaction amounts, the transaction fees and the distance should be taken into account. Ongoing we plan to implement a machine learning algorithm for pattern detection which semi-automatically detects mixing services and/or suspicious activities.

As mentioned above a stable release version of the complete prototype is planned to be published by the beginning of summer as an open source project. This open source version should give Bitcoin users the opportunity to see the pseudo-anonymity of their cryptocurrency usage. In the best case it can be used from security experts as well as non-expert users to inspect the Bitcoin system and take it as a basis for further research.

7.4 Conclusion

Cryptocurrencies became very popular within the recent years, especially because of the tremendous ups and downs of Bitcoin. The increase in Many people still believe in the anonymity of this cryptocurrency [57]. Although there are many concepts and prototypes for visualizing the Bitcoin network, none of them focuses on the automatic path finding and the connections of addresses.

With our prototype Plan₿ we want to show that Bitcoin is only pseudo-anonymous and can be tracked back until the first transaction. It combines visual and textual information about the data stored in the Blockchain. The data can be visualized and restricted to a period of interest from the user. The goal of this prototype is to provide a basis for further investigations of the paths and suspicious activities within the transaction network of Bitcoin.

Acronyms

altcoins alternative bitcoins. 20

CSS Cascading Style Sheets. 43

DOM Document Object Model. 42

HTML Hypertext Markup Language. 43

JSON JavaScript Object Notation. 40

P2PKH Pay-to-Public-Key-Hash. 14

P2SH Pay-to-Script-Hash. 14

SDLC System Development Life Cycle. 27, 35

SVG Scalable Vector Graphics. 43

UI User Interface. 35

UTXO Unspent Transaction Output. 12

WAMP Web Application Messaging Protocol. 39

Bibliography

[1] 7 bitcoin scams you need to be aware of. https://btcmanager.com/7-bitcoin-scams-you-need-to-be-aware-of/. Accessed: 2017-12-14.

[2] Autobahn. https://crossbar.io/autobahn/. Accessed: 2017-12-09.

[3] Bip16. https://github.com/bitcoin/bips/blob/master/bip-0016.mediawiki. Accessed: 2017-10-15.

[4] Bip30. https://github.com/bitcoin/bips/blob/master/bip-0030.mediawiki. Accessed: 2017-10-15.

[5] Bip34. https://github.com/bitcoin/bips/blob/master/bip-0034.mediawiki. Accessed: 2017-10-15.

[6] Bip50. https://github.com/bitcoin/bips/blob/master/bip-0050.mediawiki. Accessed: 2017-10-15.

[7] Bitcoin - developer guide. https://bitcoin.org/en/developer-guide. Accessed: 2017-10-21.

[8] Bitcoin currency and gpu mining performance comparison. https://www.pcper.com/reviews/Graphics-Cards/Bitcoin-Currency-and-GPU-Mining-Performance-Comparison/Bitcoin-Mining. Accessed: 2017-10-24.

[9] Bitcoin legal status. https://www.slideshare.net/earthsite/bitcoin-101-the-currency-the-network-the-community. Accessed: 2017-11-08.

[10] Bitcoin transaction types. https://www.cryptocompare.com/coins/guides/what-are-the-bitcoin-transaction-types/. Accessed: 2017-10-15.

[11] Bitcoin (usd) price. https://www.coindesk.com/price/. Accessed: 2017-09-02.

[12] Bitcoin wallet. https://en.bitcoinwiki.org/wiki/Bitcoin_wallet?mobileaction=toggle_view_desktop. Accessed: 2017-10-29.

[13] Bitcoin: Why is it prone to criminal activity? https://lawstreetmedia.com/news/bitcoin-why-is-it-prone-to-criminal-activity/. Accessed: 2017-12-14.

[14] Block size limit controversy. https://en.bitcoin.it/wiki/Block_size_limit_controversy. Accessed: 2017-10-15.

[15] Blockchain.info. https://blockchain.info/. Accessed: 2017-10-24.

[16] Coinjoin. https://en.bitcoin.it/wiki/CoinJoin. Accessed: 2017-28-08.

[17] Cryptocurrency market capitalizations. https://coinmarketcap.com/coins/views/all/. Accessed: 2017-12-18.

[18] The free dictionary - visualization. https://www.thefreedictionary.com/visualization. Accessed: 2017-12-10.

[19] Genesis block. https://de.bitcoin.it/wiki/Genesis_Block. Accessed: 2017-11-08.

[20] Global bitcoin nodes distribution. https://bitnodes.21.co/?__hstc=210446858.27ba77eb7687b824ac934c5ce03566ce.1485428753060.1485428753060.1485428753060.1&__hssc=210446858.2.1485428753060&__hsfp=2180031853 and https://bitnodes.21.co/nodes/live-map/visited10th. Accessed: 2017-08-07.

[21] Graph database. https://neo4j.com/. Accessed: 2017-11-09.

[22] Graph-tool. https://graph-tool.skewed.de/. Accessed: 2017-12-09.

[23] Historical price data bitoin. https://www.coindesk.com/price/. Accessed: 2017-22-08.

[24] Introducing json. https://json.org/. Accessed: 2017-12-09.

[25] One of the largest bitcoin mixing services closes its doors. https://news.bitcoin.com/one-of-the-largest-bitcoin-mixing-services-closes-its-doors Accessed: 2017-12-14.

[26] Twisted. https://twistedmatrix.com/trac/. Accessed: 2017-12-09.

[27] Visualization of bitcoin transactions. http://dailyblockchain.github.io/?__hstc=210446858.27ba77eb7687b824ac934c5ce03566ce.1485428753060.1485428753060.1485428753060.1&__hssc=210446858.2.1485428753060&__hsfp=2180031853. Accessed: 2017-08-07.

[28] What is a prototyping methodology? http://www.information-management-architect.com/prototyping-methodology.html. Accessed: 2017-11-09.

[29] What is blockchain technology. https://blockgeeks.com/guides/what-is-blockchain-technology/. Accessed: 2017-10-21.

[30] X11. https://dashpay.atlassian.net/wiki/spaces/DOC/pages/1146918/X11. Accessed: 2017-28-08.

[31] Kereshmeh Afsari, Charles M Eastman, and Daniel Castro-Lacouture. Javascript object notation (json) data serialization for ifc schema in web-based bim data exchange. *Automation in Construction*, 77:24–51, 2017.

[32] ShaikShakeel Ahamad, Madhusoodhnan Nair, and Biju Varghese. A survey on crypto currencies. In *4th International Conference on Advances in Computer Science, AETACS*, pages 42–48, 2013.

[33] Brais Alvarez-Pereira, Matthew Ayres, Ana María Gomez Lopez, Shai Gorsky, Sean Hayes, Zhi Qiao, and Jessica Santana. Network and conversation analyses of bitcoin.

[34] Elli Androulaki, Ghassan O Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun. Evaluating user privacy in bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 34–51. Springer, 2013.

[35] Marcella Atzori. Blockchain technology and decentralized governance: Is the state still necessary? 2015.

[36] Asaph Azaria, Ariel Ekblaw, Thiago Vieira, and Andrew Lippman. Medrec: Using blockchain for medical data access and permission management. In *Open and Big Data (OBD), International Conference on*, pages 25–30. IEEE, 2016.

[37] Annika Baumann, Benjamin Fabian, and Matthias Lischke. Exploring the bitcoin network. In *WEBIST (1)*, pages 369–374, 2014.

[38] Martí Berini Sarrias. Bitcoin network simulator data explotation. Master's thesis, Universitat Oberta de Catalunya, 2015.

[39] Stefano Bistarelli and Francesco Santini. Go with the-bitcoin-flow, with visual analytics. In *Proceedings of the 12th International Conference on Availability, Reliability and Security*, page 38. ACM, 2017.

[40] Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A Kroll, and Edward W Felten. Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. In *Security and Privacy (SP), 2015 IEEE Symposium on*, pages 104–121. IEEE, 2015.

[41] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D3: Data-driven documents. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2011.

[42] Tim Bray. The javascript object notation (json) data interchange format. 2017.

[43] Remco Chang, Mohammad Ghoniem, Robert Kosara, William Ribarsky, Jing Yang, Evan Suma, Caroline Ziemkiewicz, Daniel Kern, and Agus Sudjianto. Wirevis: Visualization of categorical, time-varying data from financial transactions. In *Visual Analytics Science and Technology, 2007. VAST 2007. IEEE Symposium on*, pages 155–162. IEEE, 2007.

[44] Usman W Chohan. A history of bitcoin. 2017.

[45] David Lee Kuo Chuen. *Handbook of digital currency: Bitcoin, innovation, financial instruments, and big data.* Academic Press, 2015.

[46] Michael Crosby, Pradan Pattanayak, Sanjeev Verma, and Vignesh Kalyanaraman. Blockchain technology: Beyond bitcoin. *Applied Innovation*, 2:6–10, 2016.

[47] Giuseppe Di Battista, Valentino Di Donato, Maurizio Patrignani, Maurizio Pizzonia, Vincenzo Roselli, and Roberto Tamassia. Bitconeview: visualization of flows in the bitcoin transaction graph. In *Visualization for Cyber Security (VizSec), 2015 IEEE Symposium on*, pages 1–8. IEEE, 2015.

[48] Evan Duffield and Kyle Hagan. Darkcoin: Peertopeer cryptocurrency with anonymous blockchain transactions and an improved proofofwork system. *Mar-2014 [Online]. Available: https://www. dash. org/wpcontent/uploads/2014/09/DarkcoinWhitepaper. pdf.[Accessed: 21-Sep-2016]*, 2014.

[49] Danno Ferrin. A preliminary field guide for bitcoin transaction patterns. In *Proc. Texas Bitcoin Conf*, 2015.

[50] Florian Glaser, Kai Zimmermann, Martin Haferkorn, Moritz Christian Weber, and Michael Siering. Bitcoin-asset or currency? revealing users' hidden intentions. 2014.

[51] Alex Greaves and Benjamin Au. Using the bitcoin transaction graph to predict the price of bitcoin. 2015.

[52] Jason Hirshman, Yifei Huang, and Stephen Macke. Unsupervised approaches to detecting anomalous behavior in the bitcoin transaction network. Technical report, Technical report, Stanford University, 2013.

[53] Christian Jaag and Christian Bach. Blockchain technology and cryptocurrencies: Opportunities for postal financial services. In *The Changing Postal and Delivery Sector*, pages 205–221. Springer, 2017.

[54] Aljosha Judmayer, Nicholas Stifter, Katharina Krombholz, and Edgar Weippl. Blocks and chains: Introduction to bitcoin, cryptocurrencies, and their consensus mechanisms. *Synthesis Lectures on Information Security, Privacy, & Trust*, 9(1):1–123, 2017.

[55] Irni Eliana Khairuddin, Corina Sas, Sarah Clinch, and Nigel Davies. Exploring motivations for bitcoin technology usage. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 2872–2878. ACM, 2016.

[56] Christoph Kinkeldey, Jean-Daniel Fekete, and Petra Isenberg. Bitconduite: Visualizing and analyzing activity on the bitcoin network. In *EuroVis 2017-Eurographics Conference on Visualization, Posters Track*, page 3, 2017.

[57] Katharina Krombholz, Aljosha Judmayer, Matthias Gusenbauer, and Edgar Weippl. The other side of the coin: User experiences with bitcoin security and privacy. In *International Conference on Financial Cryptography and Data Security*, pages 555–580. Springer, 2016.

[58] Hiroki Kuzuno and Christian Karam. Blockchain explorer: An analytical process and investigation environment for bitcoin.

[59] Matthias Lischke and Benjamin Fabian. Analyzing the bitcoin network: The first four years. *Future Internet*, 8(1):7, 2016.

[60] Felix Konstantin Maurer. A survey on approaches to anonymity in bitcoin and other cryptocurrencies. In *GI-Jahrestagung*, pages 2145–2150, 2016.

[61] Dan McGinn, David Birch, David Akroyd, Miguel Molina-Solana, Yike Guo, and William J Knottenbelt. Visualizing dynamic bitcoin transaction patterns. *Big data*, 4(2):109–119, 2016.

[62] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M Voelker, and Stefan Savage. A fistful of bitcoins: characterizing payments among men with no names. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 127–140. ACM, 2013.

[63] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[64] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, 2016.

[65] ANDRÉ NORDBØ. Data visualization for discovery, analysis and presentation of digital evidence. *Project report in digital forensics II*, 2013.

[66] Fergal Reid and Martin Harrigan. An analysis of anonymity in the bitcoin system. In *Security and privacy in social networks*, pages 197–223. Springer, 2013.

[67] Loıs Saublet, Petra Isenberg, and Tobias Isenberg. Bitcoin visualization. 2015.

[68] Anthony Serapiglia, Constance P Serapiglia, and Joshua McIntyre. Crypto currencies: core information technology and information system fundamentals enabling currency without borders. *Information Systems Education Journal*, 13(3):43, 2015.

[69] Michele Spagnuolo. Bitiodine: extracting intelligence from the bitcoin network. 2013.

[70] Michele Spagnuolo, Federico Maggi, and Stefano Zanero. Bitiodine: Extracting intelligence from the bitcoin network. In *International Conference on Financial Cryptography and Data Security*, pages 457–468. Springer, 2014.

[71] Tri Sundara. Study on blockchain visualization. *JOIV: International Journal on Informatics Visualization*, 1(3):76–82, 2017.

[72] Kentaroh Toyoda, P Takis Mathiopoulos, Iwao Sasase, and Tomoaki Ohtsuki. A novel blockchain-based product ownership management system (poms) for anti-counterfeits in the post supply chain. *IEEE Access*, 2017.

[73] Hans Van der Heijden. Factors affecting the successful introduction of mobile payment systems. *BLED 2002 proceedings*, page 20, 2002.

[74] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 151, 2014.