VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

Blockchain - A Design and Architecture Perspective, with Future Applications

Software Design Document

Project Team Members:

1. Balaji Chandramouli (18MIS1059)
2. Kailash Subramanian (18MIS1074)
3. V.B.Vishnu Balaji (18MIS1069)

Course name: Software Architecture and Design

Course code: SWE2004

Faculty name: Dr. Ilakiyaselvan N

Review 2

Design Documents - Bitcoin

1. **The visualization of a Blockchain**

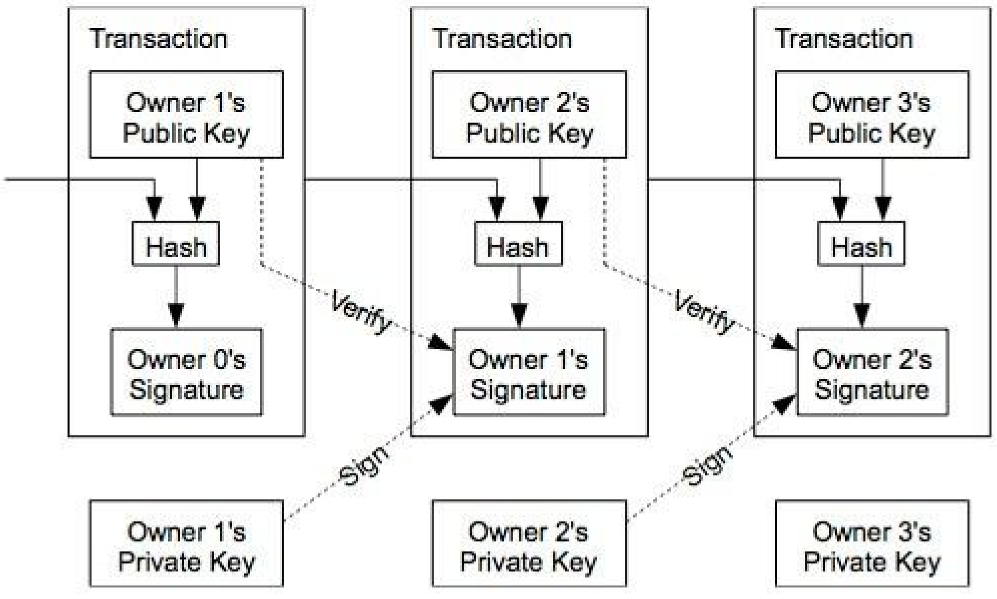
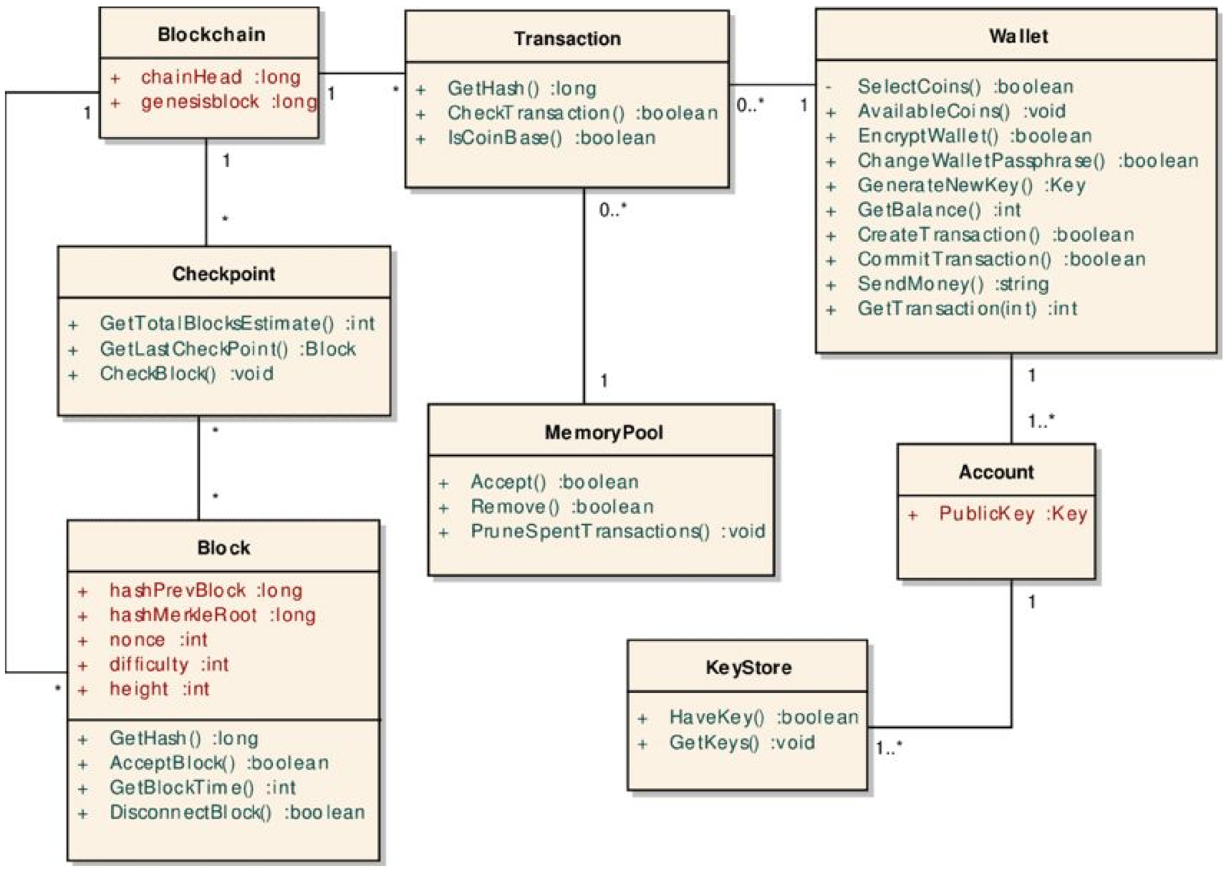


Diagram of a Bitcoin
from *Bitcoin: A Peer-to-Peer Electronic Cash System*, published in 2008 by "Satoshi Nakamoto".

We define an electronic coin as a chain of digital signatures. Each owner transfers the coin to the next by digitally signing a hash of the previous transaction and the public key of the next owner and adding these to the end of the coin. A payee can verify the signatures to verify the chain of ownership.

2. Bitcoin transaction domain model



The above diagram explains the class diagram or object oriented architecture of blockchain. Each time a bitcoin is mined, all the nodes verify the integrity and it is added to the blockchain, thereby extending the genesis block. Since bitcoin mining is a computationally expensive process, individual computers cannot contribute much to the reward pool; hence, the mining pool is created to generate hashes and get rewards.

REVIEW 3

1. INTRODUCTION

1.1 Purpose

This Software Design Document describes the architecture and system design of Blockchain, a technology that powers the first-ever cryptocurrency "Bitcoin" and is still finding new ways to revolutionize technologies to a whole new level. This document can be read by any professional developer to get a basic understanding of the technology, or anybody with interest in Blockchain.

1.2 Scope

Bitcoin (and other cryptocurrencies) is a private initiative that has created a virtual currency and a payment system based on cryptography and decentralized management. Bitcoin is considered not only an interesting, but also a disruptive technical innovation by many observers. This online payment system is not controlled by the government or any financial institutions. It sends values directly from one party to another. It is a fast, safe and anonymous way to send and receive money. This report focuses on the technology used in Bitcoin & its applications. Since this network and financial services related to bitcoins are not regulated, customers must take appropriate technical measures to protect their bitcoin holdings. In case of error and fraud, payments are difficult to reverse. Furthermore, the significant exchange rate fluctuations could pose a grave risk to bitcoin owners' wealth and discourage widespread use for monetary purposes. The objective of our project is to give an in-depth analysis of Blockchain technology, its role in Bitcoin and the future scope of applying this technology in various industries.

1.3 Overview

Firstly, we will be dealing with a few basic terminologies used in Blockchain. Then we will dive into the system architecture and provide a bird's eye view of the software process. The system design provides a detailed understanding of the functionalities of individual components. Data design provides an in-depth overview of the data structures and algorithms used to achieve the expected outcome. Then the component design of Blockchain is analysed briefly, followed by the scope of User Interface. Finally, the requirement matrix with every major component of the system is reviewed to estimate the efficiency of the model.

1.4 Reference Material

[1] S. Nakamoto,” Bitcoin: A peer-to-peer Electronic cash system”, Article, 2008.

[2] Ms.Sheetal and Dr.K.A.Venkatesh, “Necessary requirements for Blockchain Technology and its Applications”, International Journal of Computing Science and Information Technology, 2018, ISSN: 2278-9669, April 2018 (http://ijcsit.org)

[3] Israa Alqassem, Davor Svetinovic, “Towards Reference Architecture for Cryptocurrencies: Bitcoin Architectural Analysis”, Conference Paper · September 2014

[4] Dejan Vujičić, Dijana Jagodić, Siniša Ranđić, “ Blockchain Technology, Bitcoin, and Ethereum: A Brief Overview”, Conference Paper · March 2018

[5] Blockchain Data Structure

https://www.linkedin.com/pulse/blockchain-data-structure-ronald-chan

[6] How to generate public and private keys for Blockchain

https://medium.com/@baloian/how-to-generate-public-and-private-keys-for-the-blockchain-db6d057432fb

[7] Bitcoin Mining, Explained

https://www.investopedia.com/terms/b/bitcoin-mining.asp

[8] Blockchain Architecture Basics: Components, Structure, Benefits & Creation

https://medium.com/@MLSDevCom/blockchain-architecture-basics-components-structure-benefits-creation-beace17c8e77

[9] Pseudocode

https://www.researchgate.net/figure/Pseudocode-of-the-Mix-function-of-Mxr-Here-EthTxObj-denotes-the-Ethereum-transaction_fig2_332140774

1.5 Definitions and Acronyms

SHA - 256 : **S**ecure **H**ash **A**lgorithm is a cryptology algorithm that generates a 256 bit key which is used to create a non-reversible hash.

Merkle Tree : Merkle trees, or hash trees as they are known in cryptography, are used to efficiently verify the contents of large data structures in a secure manner. Each leaf node is labelled with the cryptographic hash of a data block, and every non-leaf node is labelled with the cryptographic hash of its child nodes.

Merkle Root : It is a single root which contains all the transactions included in a block, and kept in the block header and used by SPV clients to verify transactions without having to download the whole blockchain.

Bitcoin with a capital '**B**' stands for the protocol, whereas **b**itcoin with a small '**b**' refers to the actual coin itself, as a unit of currency.

SPV (**S**implified **P**ayment **V**erification) : A feature that allows the node to keep a copy of the block headers of the longest chain rather than keeping a full record of transactions.

DB	: Database
P2P	: Peer-to-peer
PoW	: Proof-of-work
PoS	: Proof-of-stake
BTC	**:** Bitcoin

2. System Overview

Here we look at the general functionalities of Blockchain, and Bitcoin, at a high level; this section serves to induct the reader into the context and design principles, which we shall be looking into at depth later on.

Bitcoin can be divided into broad modules, namely -

- Protocol Specification -
    1. Proof of work
    2. Difficulty adjustment
    3. Bloom filter
- Main Components -
    - Transactions
        - Transaction Size
        - Transaction priority
        - Transaction Merkle Tree
        - Fee Policy
    - Memory Pool
    - Wallet and coin selection
    - Blockchain
    - Alerting system
    - File system and database
    - Networking

3. SYSTEM ARCHITECTURE

3.1 Architectural Design

Bitcoin was the end product of the effort of decentralising the ledgers used to record the transactions. The transactions are practically immutable. Hence, the security and cryptography requirements are considerable. In this section, we shall see a high level overview of how the various components are connected. The various components are as follows:

- **Node** — user or computer within the blockchain

- **Transaction** — smallest building block of a blockchain system

- **Block** — a data structure used for keeping a set of transactions which is distributed to all nodes in the network
- **Chain** — a sequence of blocks in a specific order
- **Miners** — specific nodes which perform the block verification process
- **Consensus**— a set of rules and arrangements to carry out blockchain operations.

All the components work together in this simple sample process:

1. A transaction is requested.
2. A block that represents the transaction is created.
3. The block is sent to every node in the network.
4. Nodes validate the transaction.
5. *Nodes receive a small amount of BTC as reward for the Proof-of-Work.
6. The block is added to the existing blockchain.
7. The transaction is complete.

*When computers solve complex math problems on the Bitcoin network, they produce new bitcoin, not unlike when a mining operation extracts gold from the ground. And second, by solving computational math problems, bitcoin miners make the Bitcoin payment network trustworthy and secure, by verifying its transaction information. The mathematical problems can be insanely difficult, to the extent of testing even the most powerful computers. The computers produce 32 byte hashes, which is an alphanumeric cryptographic key, which cannot be easily broken by a normal computer found at people's homes.

The mined hashes are extracted as reward in the mining pools. In a mining pool, every miner has an equal chance of getting a reward; the algorithmic requirements of SHA 256 guarantee this.

Blockchain serves the following requirements -

(1) Facilitates the coordination between network's nodes, to process transactions
(2) Encapsulates the values of PoW and difficulty, as highlighted in the previous section, that are responsible for the security of the network.
(3) Helps in verifying the ownership of transferred coins, and the security aspects of it.

When bitcoin is sent from one account to another, it is called as a transaction. Bitcoin miners clump transactions into 'blocks', and add them to a public record called "Blockchain".

The blockchain, SPV and BTC connection is explained in the below diagram

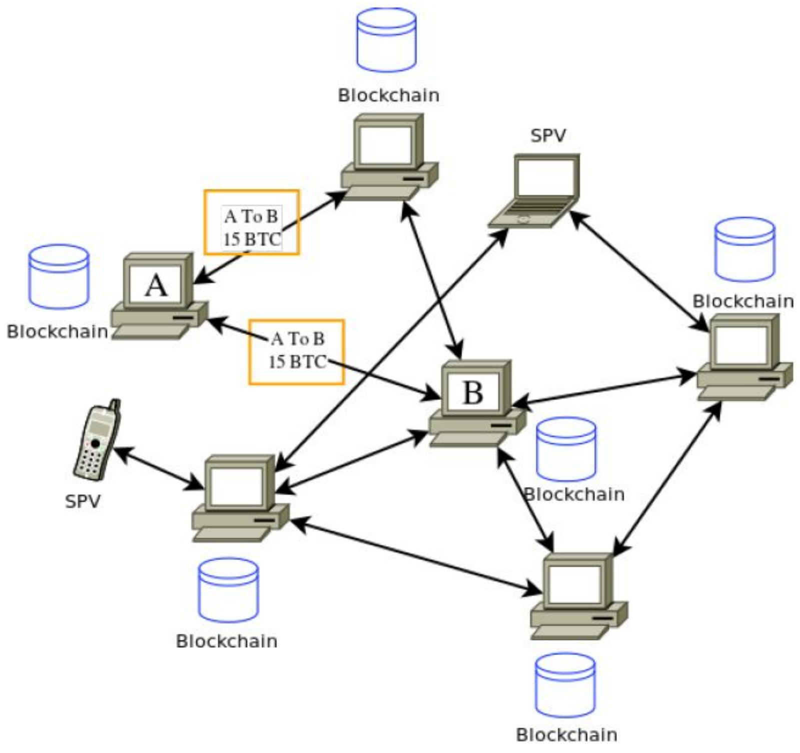


Fig.1 [3]

3.2 Decomposition Description

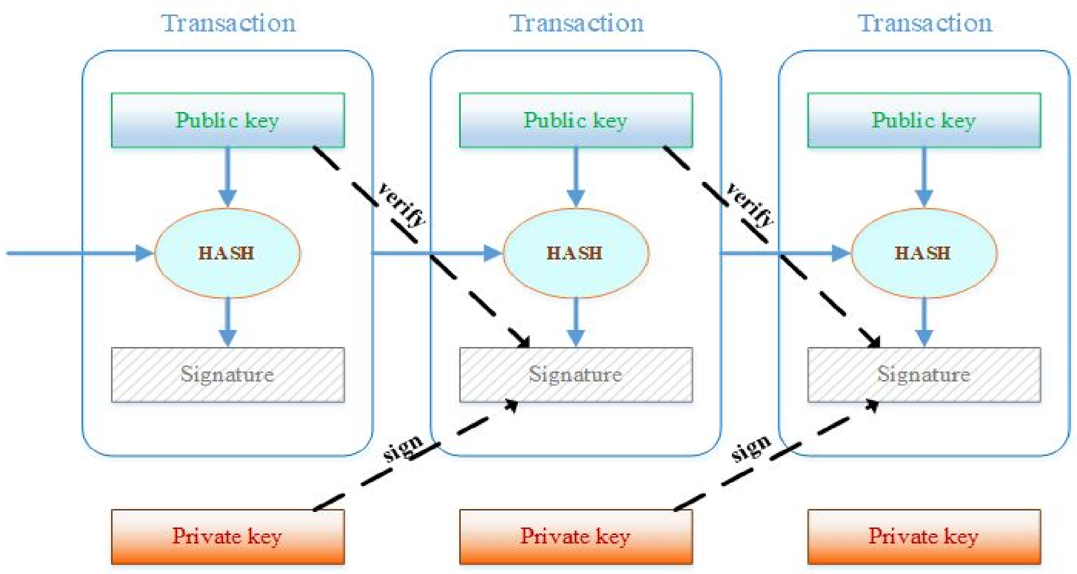


Fig.2 Image Source - Google

When the transaction is initiated, we notice that the public key and the private key are hashed in order to produce the signature of the next block. In other words, the signature of the block depends on the private and public keys of the previous block, as we can see from the above diagram. Thereby even if one of the hashes are corrupted, the rest of the blockchain is rendered useless. Hence, the integrity of the blockchain is intact.

Let us look at the flowchart which explains the initialisation of Bitcoin.

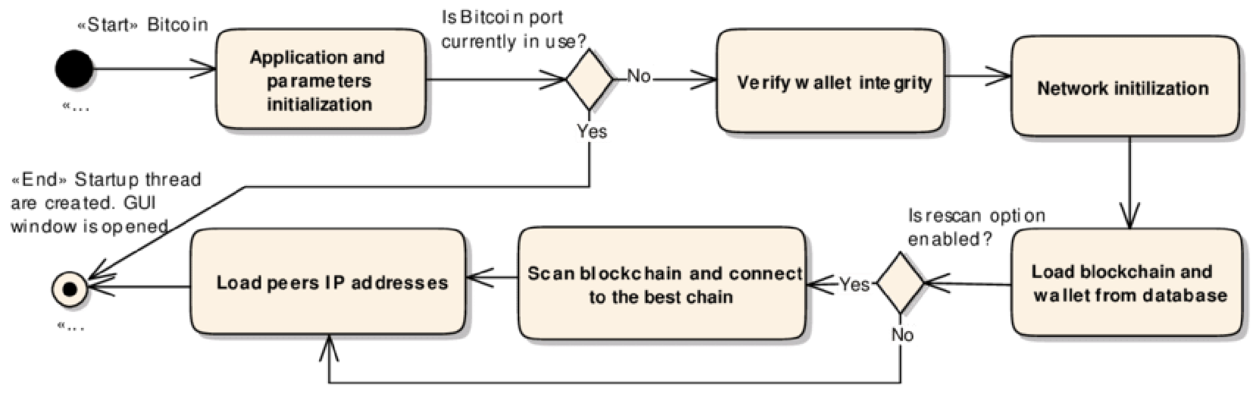

«Start» Bitcoin
«...
Application and parameters initialization
Is Bitcoin port currently in use?
No
Yes
Verify wallet integrity
Network initilization
«End» Startup thread are created. GUI window is opened
«...
Load peers IP addresses
Scan blockchain and connect to the best chain
Yes
Is rescan option enabled?
No
Load blockchain and wallet from database

Fig.3 [3]

First, the parameters of the function are initialised, such as the difficulty in the mining function. After the hashes for the bitcoin are generated, we check if the Bitcoin port is being used currently. If it is being used, we come to the end of the flowchart, by starting the GUI and initialising the thread. If not, we verify wallet integrity, initialise the network, load the blockchain and wallet from DB. If the rescan option is enabled, we scan the block and connect it to the best blockchain. If not, we directly load the IP address of the peers and *then* end the process, by starting GUI and initialising the thread.

3.3 Design Rationale

The architectural style that is used to implement the transactions on a macro level is **Peer-To-Peer Style**. The rationale behind opting for P2P network is as follows:

- The decentralised nature of Blockchain implies that there is no central authority or trusted third party that is involved in the transactions. Hence, users can interact with each other directly in the same way two independent nodes are connected.
- P2P is preferred over the more common Client-Server model because the roles and responsibilities of each node in Blockchain is identical.
- The structured P2P network called Distributed Hash Table provides consistent hashing which is used to assign ownership of each file to a particular peer. This enables peers to search for all copies of the blockchain for discrepancies.
- The decentralized nature of P2P networks increases robustness because it removes the single point of failure that can be inherent in a client-server based system.

Reasons for not choosing Client-Server model :

- By having a server, the very idea of “decentralised” is contradicted. By installing a server, either remote or local, we have to centralise it in order to have access to the data.
- Servers are more prone to attacks, compared to a P2P network. Especially with a blockchain, the system is secure as long as

honest nodes collectively control more CPU power than any cooperating group of attacker nodes [1].

- The collective aim of blockchain and bitcoin is to eliminate the need for a trusted third party. If the data generated by a blockchain increases, we are bound to have a space crunch on one server. Thus, we naturally have to outsource it at one point, thereby involving a third party, which can be avoided by means of a P2P network.

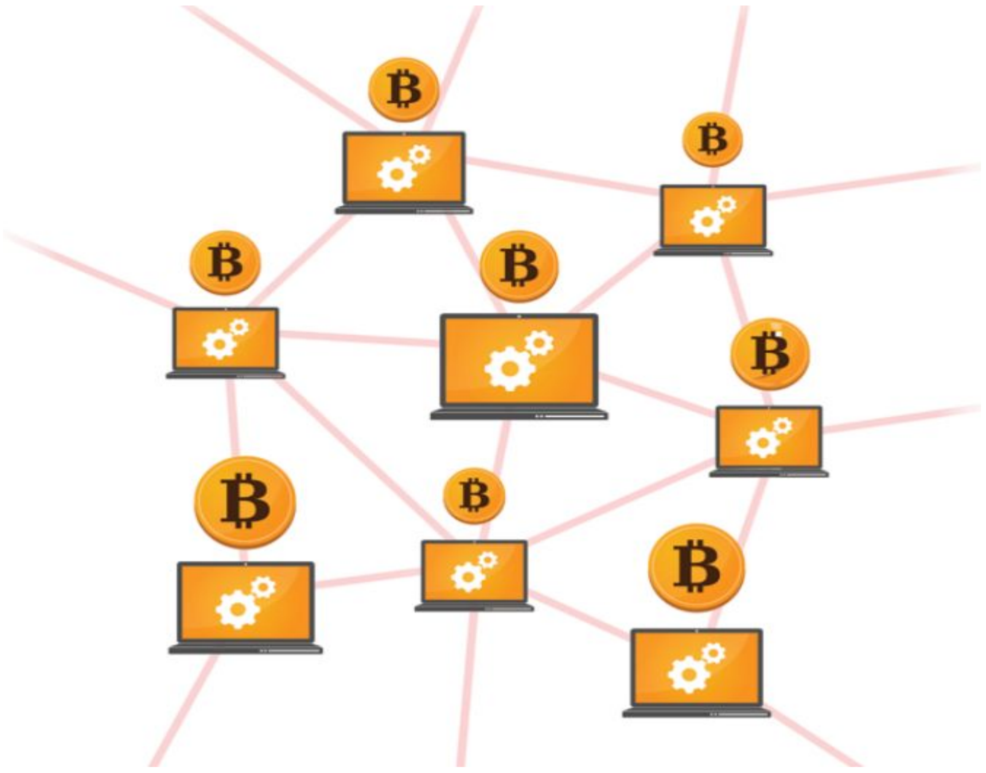


Fig.4 - Peer to Peer Networking in Bitcoin, Image source- Google

4. DATA DESIGN

4.1 Data Description

The Blockchain data structure is made of a back-linked list of blocks of transactions, which are ordered. It can be stored as a flat file or in a simple database. Each block is identifiable by a hash, generated using the SHA256 cryptographic hash algorithm on the header of the block. Each block in the network references the previous block also known as the "parent block".

Each block contains the hash of its parent inside its own header. There lies a chain going all the way back to the first block created, also known as the genesis block, linked together by a sequence of hashes. The “previous block hash” field is inside the block header and thereby the current block hash is dependent on the parent block hash. The child’s own identity changes if the parent’s identity changes. When the parent is modified in any way, the parent’s hash changes. The parent’s changed hash necessitates an alteration in the “previous block hash” pointer of the child. This in turn causes the child’s hash to mutate, which requires a change in the pointer of the grandchild, which in turn alters the grandchild and so on.

This cascading effect ensures that, once a block has many generations succeeding it, it cannot be changed without consequently forcing a recalculation of all the subsequent blocks. Because such a recalculation would require an enormous amount of computation, the existence of a long chain of blocks fortifies the Blockchain’s deep history to be immutable; a key feature of blockchain technology security.[5]

Let us now focus on the design of the file systems and the types of file formats/extensions that we are going to be using. We shall be using a LevelDB database.

File	Description
blocks/blk*.dat	The block chain in network format, blk*.dat files are Berkeley Database files and these files store the

	blockchain itself.
blocks/index/*.sst	A block LevelDB database index that improves/speeds up block information retrieval at the cost of more storage space usage. Indexes provide rapid lookups and efficient access to a database table without searching the whole table on every single access.
chainstate/*.sst	A LevelDB database stores data about unspent transactions, and can be generated from the block data using the reindex command line option. Needed for new incoming blocks and transactions validation, without this database validation should be done through a full blockchain scan
blocks/rev*.dat	Used in case of block chain reorganization for reversing/rolling back the chainstate.
peers.dat	A database of peers IP addresses and their connection time
wallet.dat	Used to keep records of user's accounts, addresses and their associated public and private keypairs, and the bitcoins that the owner of the wallet has spent or received.

4.2 Data Dictionary

A block/ledger in a Blockchain can contain as many as 4000 transactions. The block also encapsulates other block members such as public key/digital signature, hash of the previous block and proof-of-work

- Hash of the previous block - SHA256 bit hashing
- Private key - 32 bytes
- Public key - 33 bytes
- Transactions - 4000 transactions / 250 bytes (size of a single transaction) and as a whole a block is of size 1MB

5. Component Design

The description of each component is as follows :

Blockchain - When all the verified blocks of the node are added, and the length of the block increases from the genesis block to a very long immutable chain, it is called a blockchain. Each node has to verify for the block, in order to achieve trust. It consists of a chainHead, and a genesisBlock.

Block - The hash value of previous block, merkle root value, difficulty and nonce are taken into account while defining a block.

Node - Each computer in the blockchain network is called as a node; they are required to individually.

Difficulty - It is an integer value assigned to the mining function of BTC. At the time of writing this report, the difficulty level of ethereum is around 3000 Terahashes. (A ballpark estimate)

Account - It is analogical to the bank account. We can send/receive cryptocurrency from/to our account. It is encrypted with the public key.

Wallet - Based on the sender's public and private key, we can receive money from our wallet.

A diagrammatic overview -

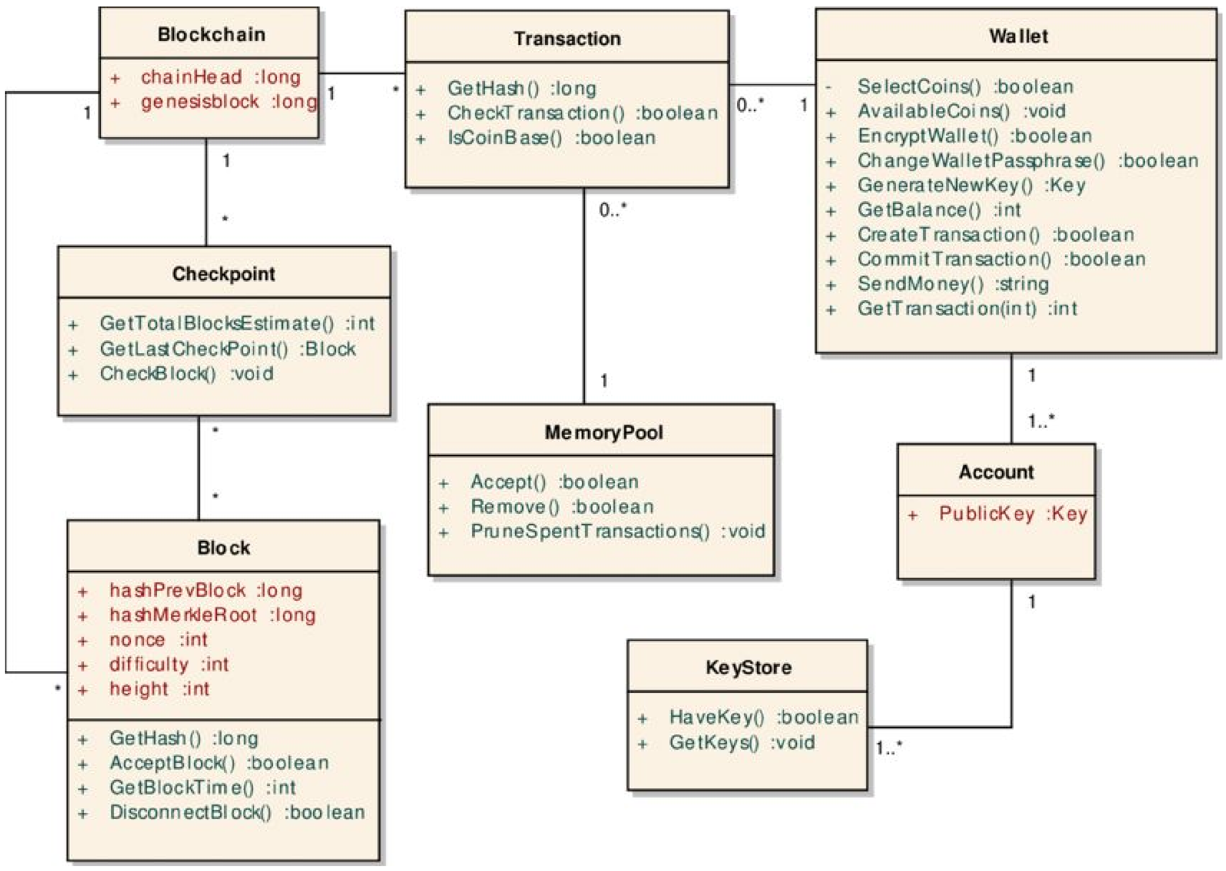
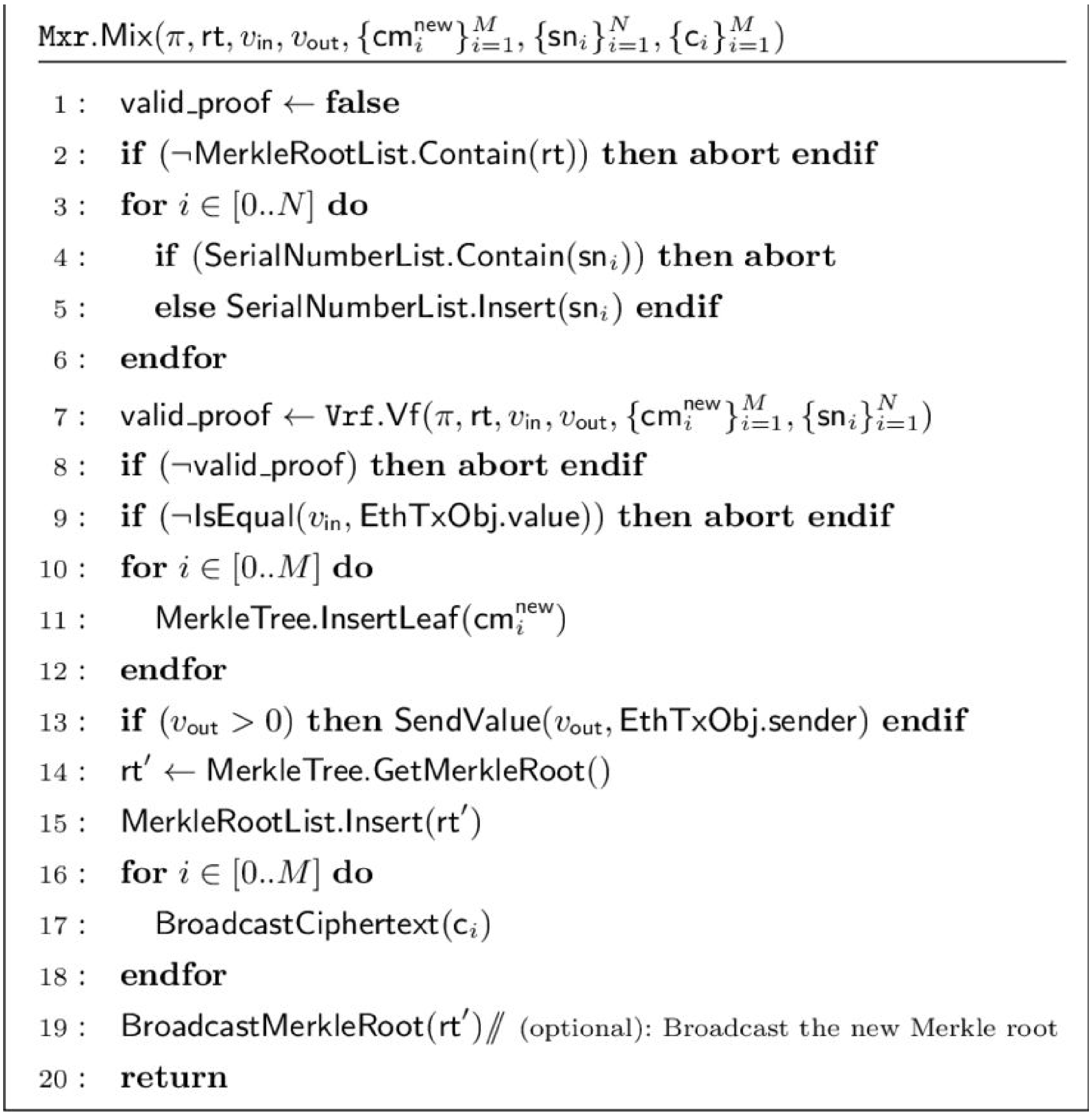


Fig.5 [3]

Pseudocode of the Mix function of Mxr



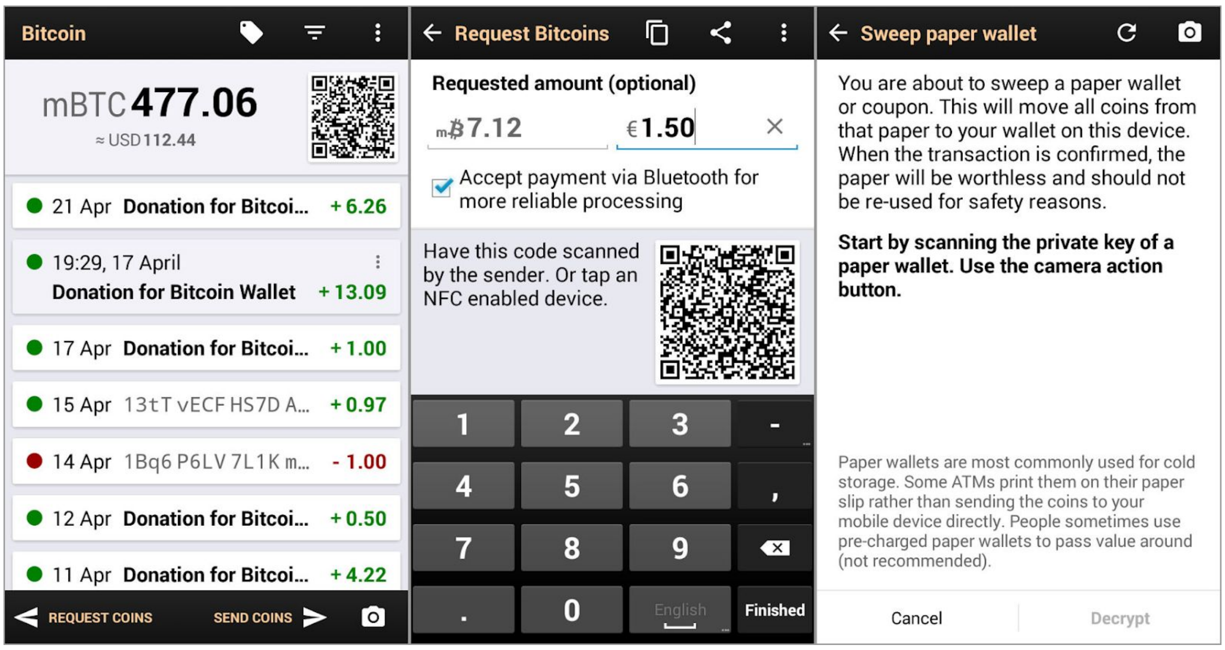
```
$\texttt{Mxr}.\mathsf{Mix}(\pi, \mathsf{rt}, v_{\mathsf{in}}, v_{\mathsf{out}}, \{\mathsf{cm}_i^{\mathsf{new}}\}_{i=1}^{M}, \{\mathsf{sn}_i\}_{i=1}^{N}, \{\mathsf{c}_i\}_{i=1}^{M})$
1 :  valid_proof $\leftarrow$ false
2 :  if ($\neg$MerkleRootList.Contain(rt)) then abort endif
3 :  for $i \in [0..N]$ do
4 :    if (SerialNumberList.Contain($\mathsf{sn}_i$)) then abort
5 :    else SerialNumberList.Insert($\mathsf{sn}_i$) endif
6 :  endfor
7 :  valid_proof $\leftarrow$ $\texttt{Vrf}.\mathsf{Vf}(\pi, \mathsf{rt}, v_{\mathsf{in}}, v_{\mathsf{out}}, \{\mathsf{cm}_i^{\mathsf{new}}\}_{i=1}^{M}, \{\mathsf{sn}_i\}_{i=1}^{N})$
8 :  if ($\neg$valid_proof) then abort endif
9 :  if ($\neg$IsEqual($v_{\mathsf{in}}$, EthTxObj.value)) then abort endif
10 : for $i \in [0..M]$ do
11 :   MerkleTree.InsertLeaf($\mathsf{cm}_i^{\mathsf{new}}$)
12 : endfor
13 : if ($v_{\mathsf{out}} > 0$) then SendValue($v_{\mathsf{out}}$, EthTxObj.sender) endif
14 : $\mathsf{rt}' \leftarrow$ MerkleTree.GetMerkleRoot()
15 : MerkleRootList.Insert($\mathsf{rt}'$)
16 : for $i \in [0..M]$ do
17 :   BroadcastCiphertext($\mathsf{c}_i$)
18 : endfor
19 : BroadcastMerkleRoot($\mathsf{rt}'$)// (optional): Broadcast the new Merkle root
20 : return
```

Fig.6 [9]

6. USER INTERFACE DESIGN

6.1 Overview of User Interface

In order to track the ownership and manage the public and private keys in your account, developers made a service/program called as a cryptocurrency wallet (For both Android and iOS).



Bitcoin

mBTC 477.06

≈ USD 112.44

● 21 Apr Donation for Bitcoi… + 6.26

● 19:29, 17 April Donation for Bitcoin Wallet + 13.09

● 17 Apr Donation for Bitcoi… + 1.00

● 15 Apr 13tT vECF HS7D A… + 0.97

● 14 Apr 1Bq6 P6LV 7L1K m… - 1.00

● 12 Apr Donation for Bitcoi… + 0.50

● 11 Apr Donation for Bitcoi… + 4.22

REQUEST COINS SEND COINS

Request Bitcoins

Requested amount (optional)

m฿ 7.12 € 1.50

✓ Accept payment via Bluetooth for more reliable processing

Have this code scanned by the sender. Or tap an NFC enabled device.

Sweep paper wallet

You are about to sweep a paper wallet or coupon. This will move all coins from that paper to your wallet on this device. When the transaction is confirmed, the paper will be worthless and should not be re-used for safety reasons.

Start by scanning the private key of a paper wallet. Use the camera action button.

Paper wallets are most commonly used for cold storage. Some ATMs print them on their paper slip rather than sending the coins to your mobile device directly. People sometimes use pre-charged paper wallets to pass value around (not recommended).

Cancel Decrypt

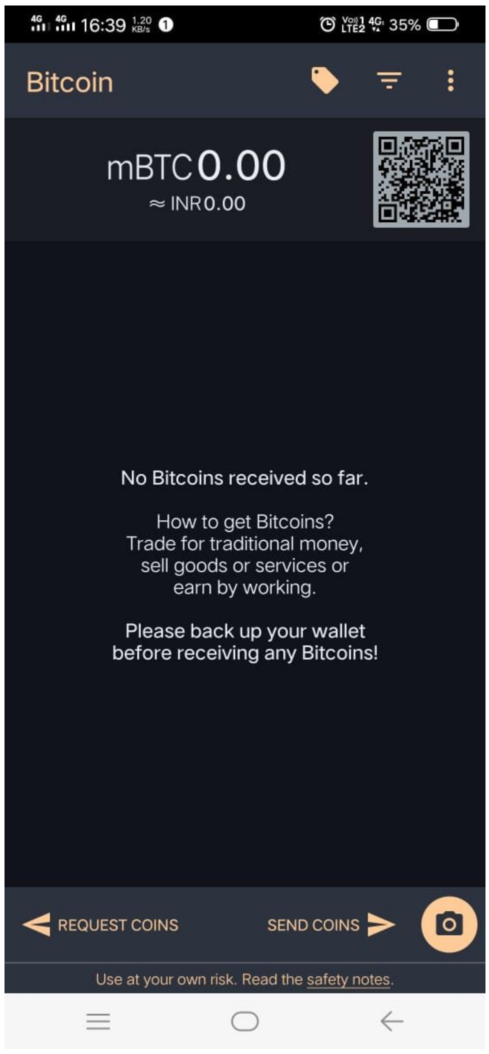
Fig.7 Image source - Mobile App

This cryptocurrency wallet / wallet helps you to manage your transactions or payments made at BTC accepted stores hassle free.

To download the app, visit the respective app store and download the Bitcoin app.

The above screenshots were taken from the Android play store, so for simplicity purposes we will be taking a look at the interfaces of the app on Android.

On launching the app, you will notice the following screen as shown below.
(Note : The following interface was captured on Bitcoin Wallet version 8.02)

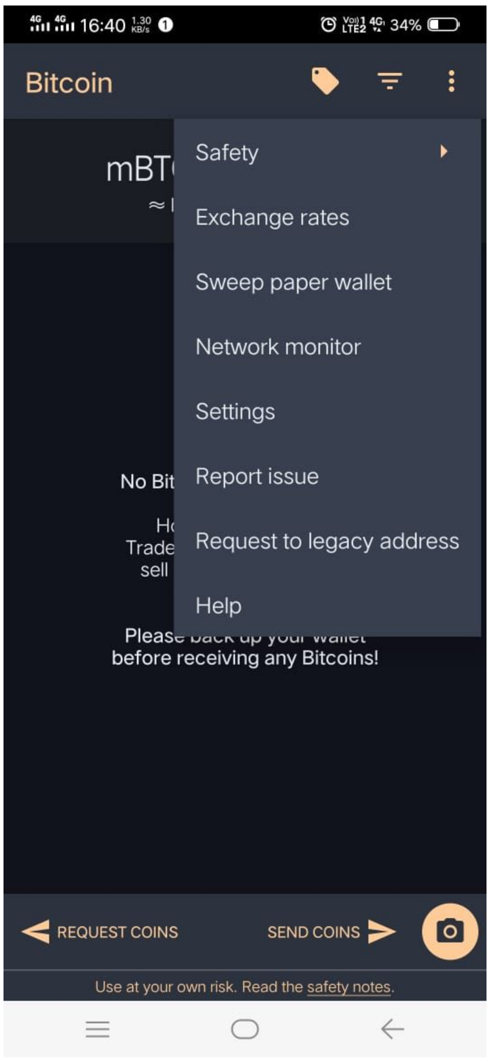


No Bitcoins received so far.

How to get Bitcoins?
Trade for traditional money,
sell goods or services or
earn by working.

Please back up your wallet
before receiving any Bitcoins!

(1)



(2)

Image sources for both images - Mobile app

You will be greeted with a home screen (image on the left), that displays your current BTC balance. Note that the BTC balance is displayed as mBTC (millibitcoin), which means one thousandth of a **BTC** (1 **mBTC** = 0.001 **BTC**).

The QR code present on the top right corner is the address for receiving payments, much like an account number in your bank. This QR code is unique and no two exist the same.

The bottom section has three buttons to perform three unique actions.

1. Requests coins
2. Send coins
3. Camera

- **<u>Request coins</u>**

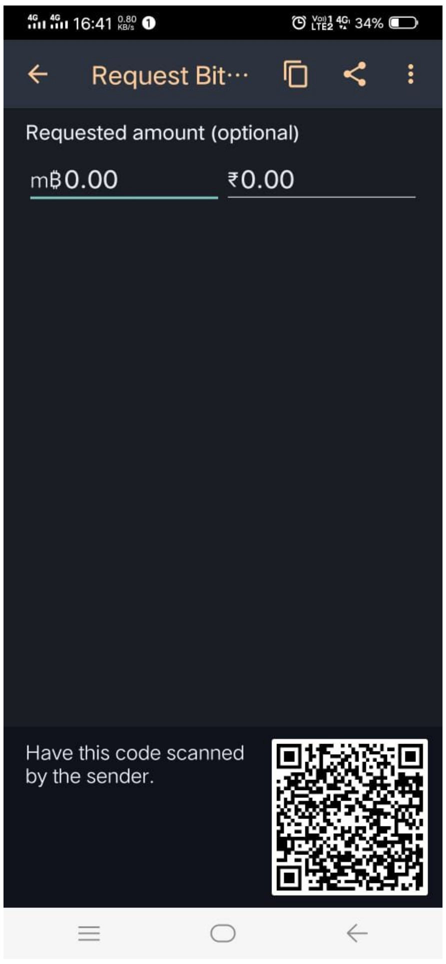
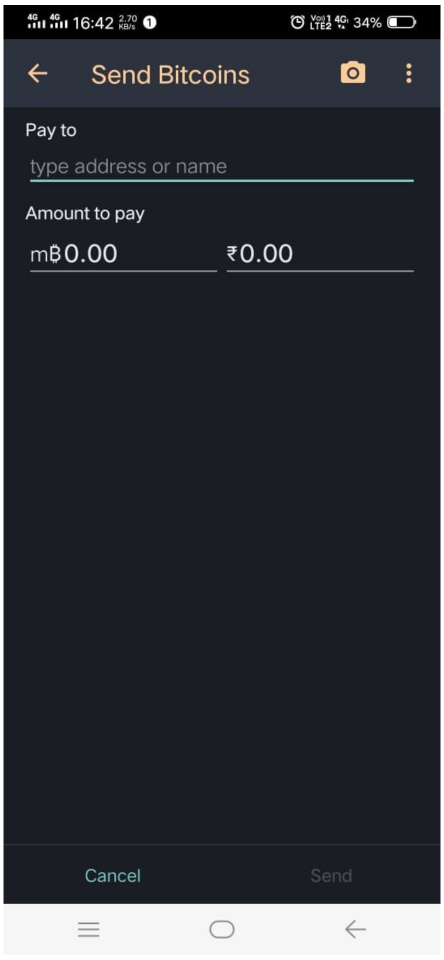

16:41
Request Bit···
Requested amount (optional)
m฿0.00
₹0.00
Have this code scanned by the sender.
(3)

Image source - mobile app

Request coins are used to receive coins from another user's wallet to your wallet.
Once the **Requests coins** button is clicked it will take you to the above screen where you can notice a **request amount** textbox along with the equivalent conversion to our desired currency. Once the desired amount is typed, the other person who is going to transfer the amount can simply just scan the QR code. As simple as that!

- **<u>Send coins</u>**



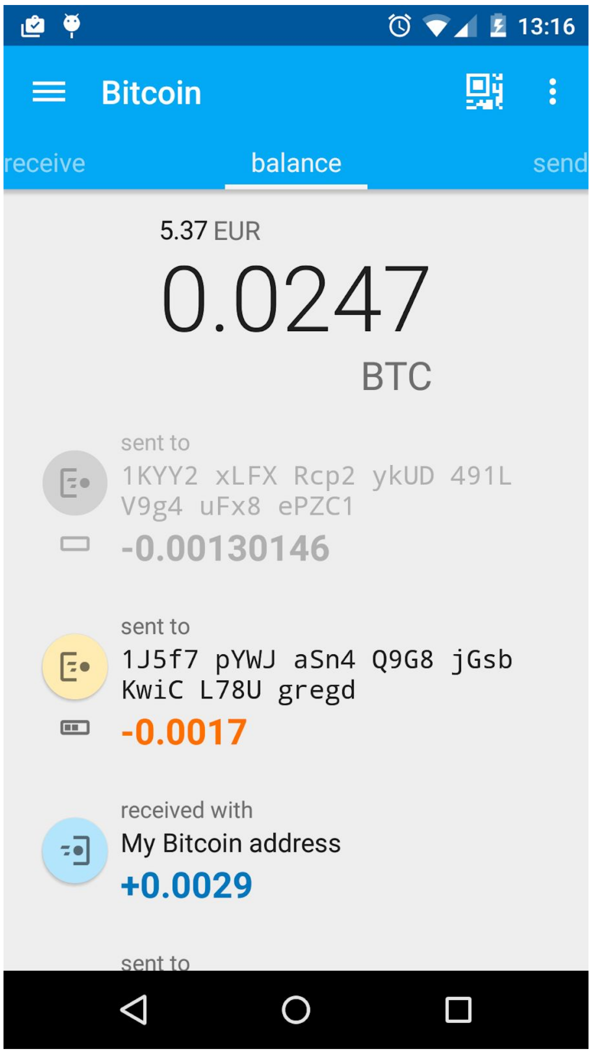
(4)

The send bitcoins follows the same process as the request coins.
You need to fill the address/pay to textbox and the amount to pay or you could just use the QR code scanner, scan the QR code of the person who requested it and get the job done as easy as possible.

There are a bunch of other features as well like, the developers have provided the users an address book to store the address of the sender of previous transactions, so that if you want to perform another transaction to the same user you could do it without having the hassle to provide the address of the sender again.

Other bunch of features in the three-dotted icon at the top-right corner of the home screen are as shown in the image above(2)

6.2 Screen Images


Bitcoin

receive balance send

5.37 EUR

0.0247 BTC

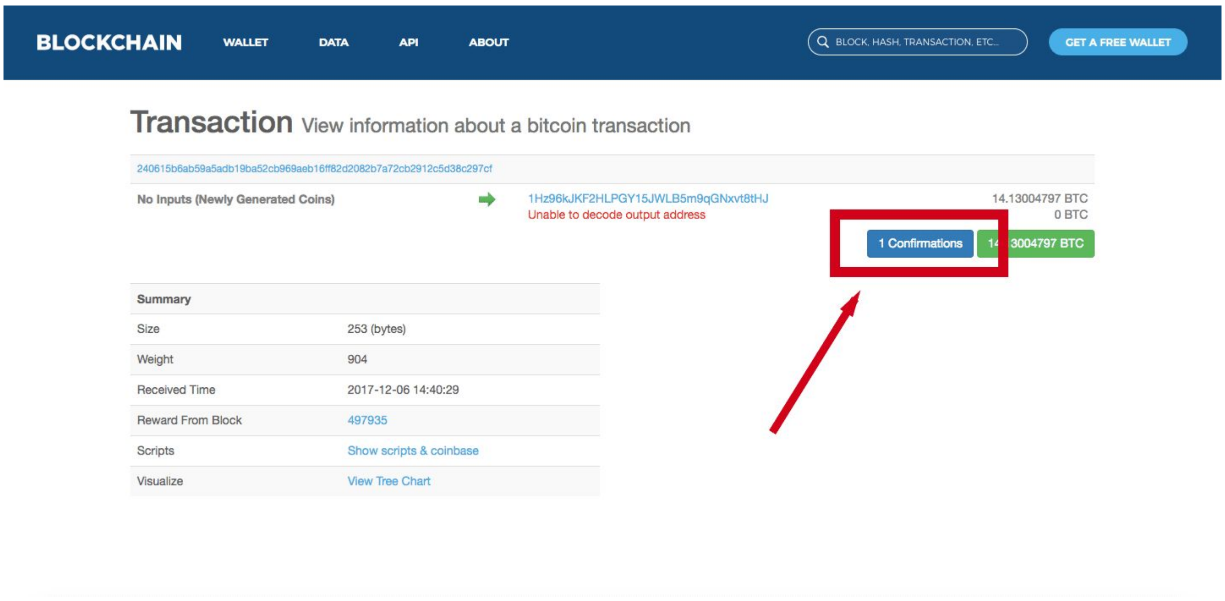
sent to
1KYY2 xLFX Rcp2 ykUD 491L V9g4 uFx8 ePZC1
-0.00130146

sent to
1J5f7 pYWJ aSn4 Q9G8 jGsb KwiC L78U gregd
-0.0017

received with
My Bitcoin address
+0.0029

sent to

Fig.8 Image source- google



Transaction View information about a bitcoin transaction

240615b6ab59a5adb19ba52cb969aeb16ff82d2082b7a72cb2912c5d38c297cf

No Inputs (Newly Generated Coins) 1Hz96kJKF2HLPGY15JWLB5m9qGNxvt8tHJ 14.13004797 BTC
Unable to decode output address 0 BTC

1 Confirmations 14.13004797 BTC

Summary	
Size	253 (bytes)
Weight	904
Received Time	2017-12-06 14:40:29
Reward From Block	497935
Scripts	Show scripts & coinbase
Visualize	View Tree Chart

Fig.9 Image source - google

6.3 Screen Objects and Actions

Objects on the screen include -

- A QR Code which can be used to scan and access the payment module of the wallet, unique to the QR Code,
- Buttons, with the option to send or receive coins,
- A text box, which can get numeric input values, for further transaction and processing,
- And lastly, a sweep option to clear your wallet.

7. Requirements Matrix

	Transaction	Block	Chain	Miners	Consensus	Node
Software	-	-	-	+	+	-
Transaction Ledger	+	+	-	-	-	+
Cryptographic Algorithm for Hashing	-	+	+	+	+	-
Verifying Transaction	+	+	-	+	+	+
Network	+	+	+	+	+	+

8. Conclusion:

In this report, we have analysed all the essential aspects of Blockchain technology, from a software architecture and design perspective. We have also presented some of the real-world applications for the technology, such as Bitcoin and its underlying design and its operating principles. Further study can be done by the reader, through the references cited in section 1.4, page 5.