CISC 322

Assignment 1 Report

Bitcoin Core: Secure Digital Wallet

Sunday, February 19th, 2023

Group 44 :

Victor Coombes (20vmwc@queensu.ca)

Osamu Adun (19osa2@queensu.ca)

Anton Gudonis (19adg8@queensu.ca)

Selena Kesidis (21smk13@queensu.ca)

Navin Pandey (17np1@queensu.ca)

Kyle Verma (18kv10@queensu.ca)

**Table of Contents**

**Abstract Pg 3**

**Introduction and Overview Pg 3**

**Architecture Overview Pg 5**

**Architectural Style Pg 5**

**Layers Pg 5**

**Component Breakdown Pg 6**

**Interaction Breakdown Pg 7**

**External Interfaces Pg 9**

**Use Cases Pg 9**

**Data Dictionary Pg 10**

**Naming Conventions Pg 11**

**Conclusions Pg 11**

**References Pg 12**

**Abstract**

Conceptual architecture is the abstract design structure of a system or piece of software in which many of the key components are interconnected. Bitcoin Core is an open sourced software that utilizes the bitcoin network in order to provide users with a variety of features including a wallet for sending and receiving bitcoins, mining capabilities, and a sophisticated validation engine. The architectural style of bitcoin core is based on a peer to peer network of users where the functionality of these features will make use of various subsystems and key components. This report will further discuss Bitcoin Core's subsystems, dependencies, use cases, design patterns, and derivation process.

**Introduction and Overview**

Cryptocurrency has become quite mainstream in today’s day and age as the ever-growing adoption towards digital currency and blockchain technology continues. The pioneer of this shift to a decentralized approach to payment was Bitcoin, a peer to peer digital currency system that is free from government regulations and completely independent from the banks. The original Bitcoin codebase is currently maintained under Bitcoin Core which is a software client that fully implements the bitcoin protocol for digital payments. The system is mostly known for being a first party, open sourced wallet solution however this wallet also allows users to mine bitcoin if desired as well as other attributes that will be discussed in this report.

This product strives to solve common problems revolving around Bitcoin such as scalability, security, and efficiency. The scalability of Bitcoin becomes an issue as more transactions occur, and the size of the blockchain increases which can lead to network shutdowns or high transaction fees. Bitcoin also has a lack of security because of the fact it is a decentralized cryptocurrency and therefore, has no real central authority. Bitcoin’s efficiency problems stem from the fact that it requires significant computational power to support a decentralized process like this. This can lead to extremely high energy consumption and long-term effects on the environment. Bitcoin Core has created a stable platform where Bitcoin transactions can take place while limiting these issues.

Bitcoin Core provides the ability to store and manage private keys along with multi-signature transactions, adding an additional layer of protection and permitting more intricate transactions. Users are able to confirm and publish transactions on the network through the full node, open source implementation of the Bitcoin Protocol. The software encourages decentralization in allowing users to participate in the consensus process, further contributing to overall stability and network security.

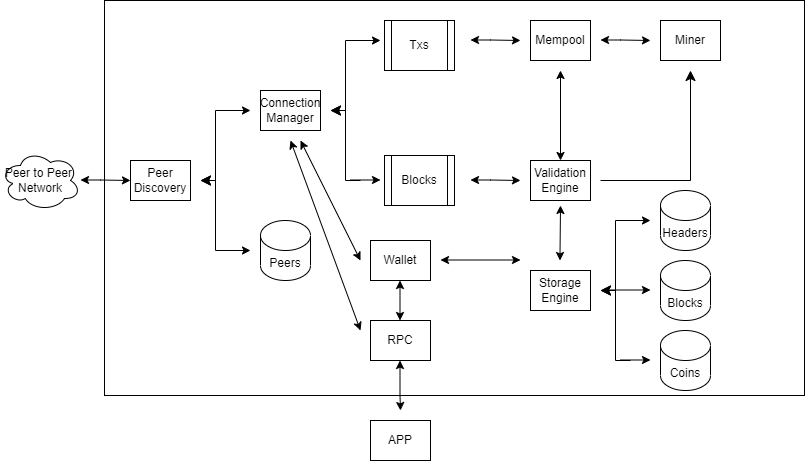
Bitcoin Core offers a slate of features that instills confidence in the user that they’re bitcoin transactions are executed in a safe and efficient manner. This includes a GUI for the wallet management system in which users can store and manage as many wallets as they like. Additionally, Bitcoin Core is able to support both legacy and SegWit addresses as well as newer address formats like Bech32 and Pay to Script Hash. Bitcoin Core offers other useful software-based tools such as, a transaction fee estimator, manual coin selection, and a blockchain explorer. The software is customizable, permitting users to tweak various settings to fit their needs. With the added flexibility of Bitcoin Core, the interface provides an ideal environment for a user’s transactions and other Bitcoin related needs.

In addition, Bitcoin Core is an open sourced project, thus a community of volunteer developers help to oversee the ongoing development of the Bitcoin Core software with Wladimir J. van der Laan acting as the lead maintainer. The project was first released by Satoshi Nakamoto in 2009 and has since undergone numerous updates and enhancements with contributions from programmers all over the world.

There are 13 key components within Bitcoin Core that interact with one another in order to authorize transactions and perform mining operations. Users are first greeted by a user interface that provides access to their wallet, this wallet is then reliant on a storage engine to retrieve and store the relevant blockchain data (headers, blocks, coins). Through the validation engine, blockchain data and authenticity of coin transfer is properly verified, leaving the validation engine open to receive unconfirmed transactions via the Mempool before they are added to a block. The miner component picks through transactions stored in the Mempool and will add a block to the blockchain once the validation engine authenticates the transaction. This process is made possible by the connection manager that taps into the peer discovery component to manage the connection of nodes and sharing of blockchain data.

Overall, it is believed that Bitcoin Core succeeds in its goal to provide users with an easy to use, reliable interface that will help them complete their Bitcoin transactions. It deploys various useful features and stores important information by accessing different databases. It is a devoted companion to anyone in the Bitcoin world and it is still constantly evolving and changing due to its open-source nature.

**Architecture Overview**

The conceptual architecture of Bitcoin Core uses a peer to peer network that connects users through search and recommendation to each other. It uses all its other subsystems to perform every other pertinent functionality required of the system. The subsystems interact to offer software suited to share, store, and mine bitcoin.

**Architectural Style**

The Conceptual Architecture of the Bitcoin Core system is a peer to peer style architecture. This style allows for a decentralized system that users and consumers expect from Bitcoin. The appeal of Bitcoin for many is that all aspects of it are decentralized and no one entity has control of the system. Bitcoin Core deals with Bitcoin and must offer the same protections that are native to it. The system is reliant on peers connecting with each other and exchanging information without any central server that stores ledgers or information. Users connect with other users and use the resources of each other.

**Layers**

**Network Layer:** The network layer is responsible for connecting nodes (users) and sending messages between nodes. It uses peer-to-peer communication to facilitate communication between nodes. The network layer uses the bitcoin protocol to send specific messages to nodes to help other features within the architecture. An example of this is “getaddr” where a local client sends a request to a node asking for information to help locate other nodes in the network.

**Blockchain Layer:** The blockchain layer is responsible for maintaining a copy of the blockchain, which is a ledger of all bitcoin transactions. The blockchain layer also holds a consensus mechanism which validates the blockchain by confirming that all active nodes (users) agree on the state of the blockchain. The consensus mechanism is important due to the decentralized nature of the blockchain.

**Transaction Layer:** The transaction layer is responsible for creating and processing bitcoin transactions. When a transaction between two bitcoin addresses is established, the transaction is sent to the network. Miners then include the transaction in a new block and add it to the blockchain.

**Wallet Layer:** The wallet layer is responsible for managing bitcoin addresses and private keys of the local bitcoin core instance. It is also responsible for creating and signing transactions made by the local user.

**RPC Layer:** the RPC (remote procedure call) layer is responsible for hosting a set of API’s that allow developers to interact with the software. API call examples include retrieving data from specific nodes. Although not used by the majority of users it is crucial to development due to the decentralized nature of the blockchain

**Component Breakdown**

In this section we will elaborate on the specific components that compose the different layers of the bitcoin core application. As well as provide additional details on the topics depicted in our architecture overview diagram that were not mentioned in the layers overview.

**App:** The GUI allows users to interact with Bitcoin Core. It allows users to graphically interact with and manage their wallets, send transactions and view transactions, on the network.

**RPC:** The remote procedure call interface enables users to access the API’s provided by bitcoin core to retrieve additional information regarding the blockchain. RPC allows users to interact with the network without having to operate a node. RPC allows users to more easily develop applications that interact with the bitcoin network.

**Wallet:** The wallet component component is responsible for allowing users to manage their bitcoins. It is responsible for creating and managing passwords, keeping track of bitcoin amounts within the wallet, and keeping a record of sent and received transactions to the wallet. The wallet component uses other components within the architecture to synchronize the user’s wallet with the latest blockchain states in order to allow for new transactions to occur.

**Storage Engine:**

The storage engine component of Bitcoin Core is responsible for storing and retrieving Bitcoin blockchain data, such as transaction records, blocks, headers, and other relevant information. It is designed to optimize data storage and retrieval for the entire blockchain and supports various data storage options, such as a database or flat files.

**Headers, Blocks, and Coins:**

The Headers, Blocks, and Coins components are responsible for storing and managing the blockchain data. Headers contain metadata about each block, such as its hash, the time of creation, and the difficulty level. Blocks contain the transaction data and are used to verify the integrity of the blockchain. Coins are the units of value that are transferred between Bitcoin addresses, and each coin is associated with a transaction output.

**Validation Engine:**

The Validation Engine component of Bitcoin Core is responsible for ensuring the integrity of the blockchain data. It checks the authenticity of each transaction, verifies the signatures of the transaction inputs, and ensures that all transactions are correctly formatted and adhere to the rules of the Bitcoin network.

**Mempool:**

The Mempool component of Bitcoin Core is responsible for storing the unconfirmed transactions that have not yet been included in a block. Transactions are first added to the Mempool before they are verified and added to a block by miners.

**Miner:**

The Miner component of Bitcoin Core is responsible for creating new blocks in the blockchain. It performs the cryptographic calculations required to add a new block to the blockchain and as a result produces the newly minted bitcoins for each block that it adds.

**Connection Manager and Peer Discovery:**

The Connection Manager and Peer Discovery components are responsible for managing the network connections to other nodes in the Bitcoin network. They help the Bitcoin Core application discover other nodes in the network and manage the flow of data between the Bitcoin Core node and other nodes in the network.

**Interaction Breakdown**

App -> RPC: The Bitcoin Core application provides the user interface for interacting with the RPC interface, which enables programmatic interaction with the Bitcoin Core software.

App -> Wallet: The Bitcoin Core application allows users to create and manage their Bitcoin wallets, which are used to store private keys and other information related to their Bitcoin transactions.

Wallet -> Storage Engine: The wallet component relies on the storage engine to store and retrieve blockchain data, including the transaction records, blocks, headers, and other relevant information.

Storage Engine -> Headers: The storage engine component stores the header information associated with each block in the blockchain.

Storage Engine -> Blocks: The storage engine component stores the full transaction data for each block in the blockchain.

Storage Engine -> Coins: The storage engine component stores information about the individual coins that are transferred between Bitcoin addresses.

Validation Engine -> Blocks: The validation engine component is responsible for ensuring that the blockchain data stored in the blocks is authentic and correctly formatted.

Validation Engine -> Coins: The validation engine component verifies the authenticity and validity of each coin transfer.

Mempool -> Validation Engine: The Mempool component sends unconfirmed transactions to the validation engine component for verification before they are included in a block.

Miner -> Mempool: The miner component selects transactions from the Mempool to create a new block to add to the blockchain.

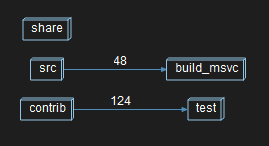
Miner -> Validation Engine: The miner component must validate the authenticity and format of the transactions it includes in a new block before adding them to the blockchain.

Connection Manager -> Peer Discovery: The connection manager component discovers other nodes in the Bitcoin network through the peer discovery component.

Connection Manager -> Peers: The connection manager component manages the network connections to other nodes in the Bitcoin network.

Peers -> P2P Network: The peers component enables the Bitcoin Core application to connect to other nodes in the network and share blockchain data with other nodes to maintain a consistent and up-to-date view of the blockchain.

Dependency graph from the source code:



**External Interfaces**

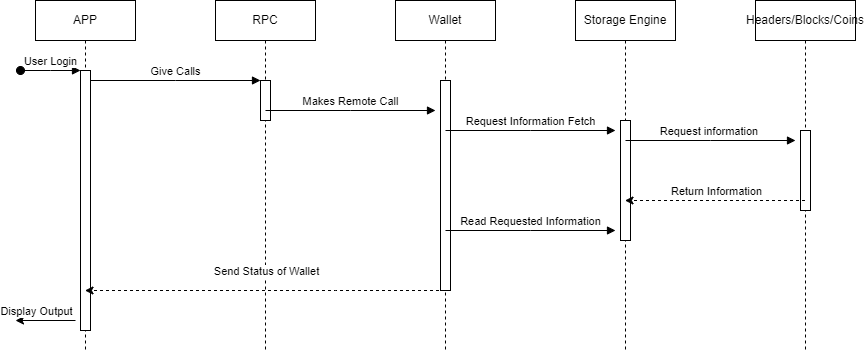
One external interface present in the Bitcoin Core is what connects the wallet management service to the internet and the overall Bitcoin network. Within Bitcoin Core, the peer to peer structure of the architecture allows for the peer discovery node to connect with other devices in the network. There is an external service that is implemented for this to take place. The peer discovery node connects to a service that allows for communication and transfer of information between two nodes, essentially functioning as an external interface for the wallet. This specific architecture implementation means that most crucial information from the system about Bitcoin is transferred to different peers.

Additionally, the individual information for each bitcoin transaction is transmitted to and from the system. This includes the content for each block and coin that is exchanged or sent from the user’s wallet. This data includes the history of all of the transactions for that specific block.

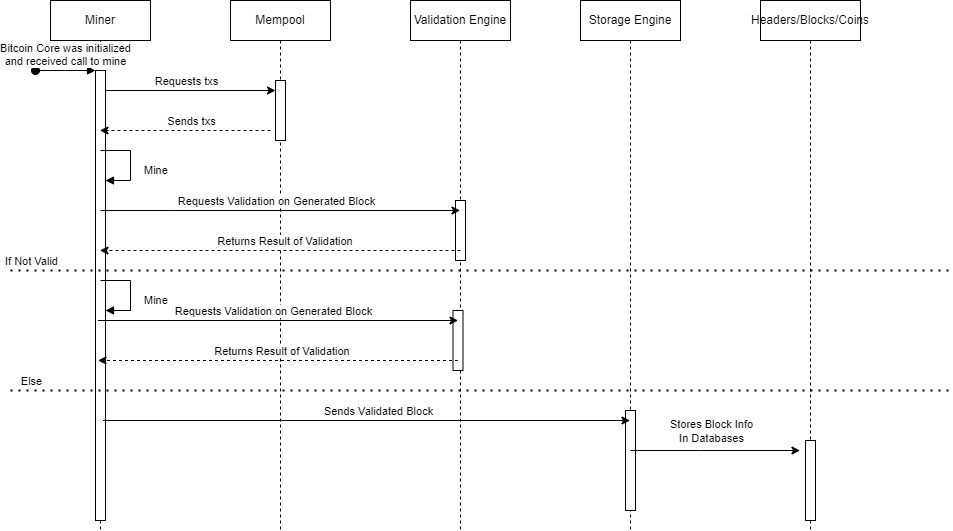
**Use Cases + Sequence Diagrams**

The two use cases we will be investigating are:

1. A user wants to log in and access their wallet and view the information inside.



1. A user wants to mine and mint a new Bitcoin.



**Data Dictionary**

Bitcoin: A digital currency that is entirely independent of a central bank. It uses the blockchain to support peer to peer interactions.  
Decentralized: When the control of decision making and information does not have a central entity, and is instead is given to a distributed network  
Blockchain: The system that acts as a record or ledger for transactions  
Peer-To-Peer: A system in which each device acts as a server for other devices, allowing access of files.  
RPC: When a device makes a process execute in a diffperent address space from itself

**Naming Conventions**

In our diagram, “Txs” stands for transactions. Additionally, “RPC” stands for remote procedure call. P2P is the abbreviation for Peer to Peer and GUI is Graphical User Interface.

**Conclusions**

Our Conceptual Architecture of Bitcoin Core is an accurate representation of how the system works and how each of the fundamental components interact to create the functional system and all of its processes. As outlined in the introduction of the report, Bitcoin Core uses a peer to peer style architecture. This is necessary for Bitcoin reliant software as it is a decentralized currency and itself works on a peer to peer style, so a companion to the cryptocurrency should also be decentralized.

The system does not have any one component that is central to its fundamental operations, it instead uses all of its components to perform all fundamental processes. It accesses the databases of peers, headers, coins, and blocks to store and access important information. The P2P network and Peer Discovery are what handles the interactions between users in the peer to peer network. The Storage and Validation Engines handle the integrity and security of blockchain interactions. Through the creation and extrapolation of the Bitcoin Core system, we have gained a much deeper understanding of the system and are excited to create a concrete architecture in the future.

**References**

Bitcoin Core Website:

<https://bitcoin.org/en/bitcoin-core/>

What is Bitcoin Core? By River Financial: <https://river.com/learn/what-is-bitcoin-core/#:~:text=Bitcoin%20Core%20is%20open%20source%20code.,versions%20of%20the%20Bitcoin%20software>.

Review of Bitcoin Core wallet:

<https://wallets.com/bitcoin-core-review/>

Conceptual Architecture Reference:

<https://cypherpunks-core.github.io/bitcoinbook/ch03.html>

Bitcoin Architectural Overview:

<https://dl.gi.de/bitstream/handle/20.500.12116/16570/DFN-Forum-Proceedings-001.pdf?sequence=1&isAllowed=y>