Bitcoin Core - Conceptual Architecture

Date: February 9th, 2023

Prepared by:

Daniella Ruisendaal (Team Lead) – 17dar8@queensu.ca

Alina Padoun (Presenter) – 18ap1@queensu.ca

Adam Ciszek (Presenter) – 19ac80@queensu.ca

Aidan Wolfson – 18aaw1@queensu.ca

Camila Izquierdo – 18cni@queensu.ca

Tanner Big Canoe - 19tjb8@queensu.ca

Abstract

Bitcoin Core is the platform upon which the Bitcoin network is built. The system allows users to send and receive bitcoin payments, verifies and validates all transactions, and maintains the security and stability of the bitcoin network. The blockchain network itself is comprised of full nodes, or “peers”, that keep the system decentralized and are responsible for downloading the blockchain when new full nodes are added and broadcasting changes. By owning bitcoin, users are able to create transactions with other users that are then validated by computers tagged as “miners” which solve difficult cryptographic problems before adding the transaction to the chain and delivering funds to the original user. This report explores and presents the conceptual architecture of the Bitcoin Core system, as well as various use cases to effectively describe a user’s actions in the application.

Introduction

Bitcoin Core is a platform created and supported in conjunction with Bitcoin, the cryptocurrency. It provides an easy-to-use user interface where users can mainly send and receive bitcoin amongst other activities by connecting to the network as a full node, storing a copy of the bitcoin blockchain. As the user interacts with the software, the blockchain is altered and added to.

Blockchain is used as a public ledger where all transactions are timestamped and stored, which improves security as it prevents any double spending or modifications to the blockchain. This works closely with the transactions module to record all transactions and ensure they are valid which enables the transferring of funds between users. The transactional module allows for strong security through the blockchain by associating private and public keys with each user. As the keys are used in different processes involving the user, the keys are destroyed and cannot be used again. Some sub-modules include the contracts module which enforces financial agreements between users, the wallet which handles the creation and destruction of the keys, and the payment processing, which handles the actual transfer of funds in a bitcoin send or receive of funds. The Peer-to-Peer network (P2P) is another important aspect of this architecture, which assigns the full nodes as peers and gives them equal importance, thus making the blockchain decentralized. The network allows for the creation and validation of new nodes through user actions, downloading the entire blockchain when new blocks are added, and broadcasting new updates as mining discoveries are made. It also reinforces the security and integrity of the architecture. The P2P network also allows for the use of operating modes which can consist of running a full node, or a Simplified Payment Verification (SPV) which relies on the full nodes to provide blockchain information. Lastly, mining is another part of the system that allows a user to solve cryptographic hash puzzles to verify new blocks and add them to the blockchain ledger. This can be completed solo or in groups, referred to as “pooled” mining. Overall, mining provides users with rewards while also securing the network further by validating the newly added blocks.

Over time, Bitcoin Core has been able to maintain a high level of evolvability, however there has been a documented scalability issue that is common amongst cryptocurrencies that restricts the attainable number of Transactions Per Second (TPS) due to the current architectural model. Financial service providers like Visa can reach an average of 150 million transactions per day whereas Bitcoin processes a maximum of 397,500 transactions per day.

Bitcoin Core leverages parallelism and context switching to support their multi-threaded environment; however, this is hindered by “global locks” around many functions that creates a queue of processes dependent on the completion of others. Bitcoin Core can maintain many concurrent connections due to the decentralized structure of the P2P network, but per user the default settings limit the number of concurrent connections to 125 to control traffic.

Bitcoin Core is an open-source software that can be locally worked upon by any interested user. However, only a small number of people can make commits directly to the main code. This restricted access mitigates any malicious intent that could be present amongst users. Bitcoin Core implements their codebase access through several layers of protection, including commit keys, verify keys, and security checks during pull requests and code releases. The issue tracker in the Bitcoin GitHub allows developers to fix whatever suggestions or bugs they wish by searching through this list. One issue with this method of division is that there is often a fluctuation in the number of developers working on the system at any time. This causes the timeline of new versions and improvements being made to vary greatly, which could be avoided if there were instead committed long-term developers that are well versed in the codebase. Although there is some risk involved with this method of division, it also allows for many developers to interact with the software and also allows them to develop their personal skills.

The functionality of Bitcoin Core can be separated into three use cases. The first use case demonstrates a user A attempting to send bitcoin funds to a user B. The second use case shows a user accessing their transactional history. Lastly, the third use case shows the process of a user starting and stopping a mining session on their local device. These use cases and their diagram representations assume that the essential authentication and authorization processes have been satisfied by the user previously signing in or otherwise.

In addition to the core parts of the system, there are a few external interfaces that are present in Bitcoin Core. Users can store their bitcoin funds in a lightweight, secondary wallet that is separate from the built-in wallet. In addition, Bitcoin Core can be used through a Graphical User Interface (GUI) or the command line, so any necessary information would need to be transferred from the selected platform to the system. There are also other external connections that communicate with the bitcoin P2P network to transmit transactions, validations, and connect to the blockchain.

Architecture

System Functionality and Interactive Parts

Bitcoin Core is the platform in which the bitcoin network is implemented on, thus, the system provides users the chance to take part in the network as a full node – meaning through participation, the software stores a copy of the bitcoin blockchain. This system allows users to send and receive bitcoin payments, to verify and validate all transactions, and to engage in mining – which adds new blocks to the blockchain and consequently makes new payments more secure. All of this relates to the contribution of security and decentralization of the bitcoin network. Bitcoin Core also implements privacy features such as keeping public keys anonymous, and using new key pairs (Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System) to prevent them from being linked to a common owner. Users are also able to monitor the network, ensuring stability and security using information such as the number of connected nodes, the block height, and detailed information about peers such as their IP address. With these functions in place, Bitcoin Core helps ensure the security and stability of the bitcoin network.

Blockchain

The blockchain is the public ledger in which transactions using bitcoin are timestamped and stored. This is essential to protect the security and validity of the currency, as it prevents double spending and/or any other modifications to previous transaction records. Once a transaction is put through (1+ outputs to single node), it is first verified by the nodes in the P2P network, and then is recorded in the block chain. Any future reference of the same key is forbidden – this would be double spending. It is also important to note that the input of one transaction is the output of all previous transactions. This leads to the interaction between transactions and the blockchain, where it is clear to see that the blockchain implements the transactional module, as every single transaction must be recorded in the blockchain. Furthermore, the nodes depicted within the operating modes system implement the blockchain, as this is what validates and verifies the blockchain as a client. Another subsystem that the blockchain implements is the mining procedure, as this adds new blocks to the block chain, enhancing its security. A final module that plays an important part within the blockchain is the P2P network, which allows each full node to communicate, further permitting each node to have a full copy of the blockchain. These full nodes are responsible for verifying every block on the blockchain to help ensure the stability of the entire network, thus, making the P2P network crucial in distributing this work.

Transactions

The transactions module is a significant aspect of the Bitcoin Core software as it is responsible for the transferring of funds. This also includes making sure these transactions are valid. Transactions can be seen as a transfer in ownership, where one wallet receives ownership of n number of bitcoins. It is also important to note that everyone has pairs of public and private keys, each allowing the owner to receive and send funds. This module interacts with many other modules, including the blockchain, the P2P network, wallets, and payment processing. The blockchain is what allows these transactions to be public and this protects the integrity of the currency itself. It provides a record of all transactions, which decentralizes the currency and allows for a transparent and secure way of transferring funds. The P2P network is within this process as it is what allows the different nodes in the network to verify and validate each transaction. This group of modules play a significant part in validating each individual transaction keeping the network stable and secure. Furthermore, the wallet and payment processing modules interact with the transactions module, as the wallet is what stores the public and private keys, and this allows users/nodes to receive and send payments. The payment processing system is simply the module of bitcoin core that is responsible for processing the transactions (i.e., verifying sufficient funds), and ensuring the payment follows the different constraints. All these modules must directly interact with the transactions module to ensure security, stability, and integrity of spenders.

Contracts

Contracts are a sub-module of transactions, as they are transactions themselves (implement the transactions module), but these contracts are more complex as they enforce financial agreements and can be looked at as smart contracts. We can look at it as the following: when pre-conditions are met, the network executes a certain action that was agreed upon (i.e., transfer of funds). There is no limit to the contracts, therefore there can be as many conditions as needed to complete a certain task resulting in a transaction. Contracts interact with the following modules just like transactions do - they interact with the blockchain, the P2P network, wallets, and payment processing.

Wallet

Wallets are what create public keys (used to receive), and to spend, the corresponding private keys must be used. The difference between a traditional wallet and a crypto wallet is the currency lives on the blockchain, rather than in your actual wallet. It is the private keys that verify ownership of this currency. This shows the significance of the P2P network, which allows users to get information from the blockchain and broadcast new transactions. This means the wallet also interacts with the transactions module. The payment processing module also manages the processing of sent and received funds, hence playing a significant role in the interaction between the wallet, other modules, and other nodes.

Payment Processing

The payment processing module handles the transfer of funds from the payer's wallet to the payee's wallet – which also includes refunds and recurring bills. It outlines the structure and steps necessary for the payments to be processed. Sub-modules under the payment processing module include pricing orders – which takes care of exchange rates as many bitcoin orders are priced in fiat but paid in Satoshi's, requesting payments, verifying payments, issuing refunds, and recurring bills. All these sub-modules make up the payment processing module, and all interact with it. A main component of payment processing is the acquiring of the address from the transactions module, as well as the P2P network and blockchain for recording these transactions and validating these payments.

Operating Modes

The operating mode of a Bitcoin Core client refers to the level of security to verify the blockchain. This includes running a full node, or as an SPV. The difference between the two is a full node has a copy of every transaction recorded on the block chain and can validate transactions, while an SPV client relies on full nodes to in the network to provide information on the blockchain, as they only store a subset of block headers from the blockchain. This makes SPV clients much less secure relative to the full node which fully participates in the blockchain. Significant dependencies include the blockchain, and the P2P network. The blockchain is an essential part of this module as these modes are what the blockchain depends on to validate blocks and keep the network's integrity. What allows for all of this is the P2P network which distributes many processes, but mainly allows for the SPV clients to connect with full nodes and receive necessary information to interact with the blockchain. This network is essential for different clients to run and to communicate amongst each other and the blockchain.

P2P network

The P2P network is an essential part of the architecture of Bitcoin Core, as the full nodes (the peers) are the ones providing the network itself. Hence, this is what makes all full nodes equal – the fact they all share the same burden of providing and maintaining the network to keep the entire system and network decentralized. This network of nodes is responsible for validating all transactions that are added to the blockchain, however, this isn't the only responsibility of the P2P network. It is also responsible for downloading the entire blockchain when new full nodes are added and is also responsible for broadcasting miners' discoveries. The P2P network interacts with many other modules of the architecture, primarily the blockchain, transactions, mining, and wallet. The blockchain and transactions modules both extend to the P2P network because the entire blockchain is stored on the network, and prior to transactions being added to the blockchain they must be verified. Furthermore, the mining architecture must depend on the network for the newly discovered blocks to be broadcasted to the blockchain. All of these tie into the wallet system, as it is responsible for keys and maintaining ownership of the currency (transactions) on the blockchain. The P2P network interacts with all of these modules, and is the reason the entire architecture maintains security, integrity, and decentralization.

Mining

Mining is an essential part of the blockchain, as it is what makes the network more secure, keeping it decentralized. It allows users to solve cryptographic hash puzzles to verify new blocks, which are added to the blockchain ledger. Mining is done in two different ways – solo mining and pooled mining. Solo mining refers to a single user mining for a bitcoin, receiving all the rewards that comes with it, while pooled mining refers to multiple users who 'pool' their resources to solve these hash puzzles. The architecture behind this module is the constructing of these blocks, which are then validated and added to the blockchain, however, it extends to other modules. The mining software must interact with the blockchain and the P2P network, because these two modules allow for these blocks to be validated (through the P2P network), and then added to the blockchain. These are essential for the mining module to be implemented by Bitcoin Core. A less implemented system would include the wallet architecture, as keys are produced through the wallet module, and the block will include transaction history, thus public (and private) keys.

System Evolution

By implementing a P2P architecture, the Bitcoin Core system can maintain a high level of both evolvability and scalability. However, there is a documented scalability issue that prevents Bitcoin and other cryptocurrencies from ever reaching the TPS of businesses like Visa, which handles "an average of 150 million transactions every day" (VISA, 2023) (Approximately 1,700 TPS). Bitcoin, on the other hand, processes 4.6 transactions per second, or 397,440 transactions per day. The current architectural model has emerged as a bottleneck to scalability because increasing the block size or decreasing block generation time won't solve the problem of it capping at 4.6 TPS (Muli, 2019).

Control and Data Flow

The general overview of the data flow of Bitcoin Core is that a user owns bitcoin as any person owns money. From there, they can request and receive funds to create transactions amongst other Bitcoin users. Transactions begin by sending a request to the blockchain. The blockchain is a distributed database running on a P2P system of thousands of computers across the world. Once the transaction has been received by the blockchain, computers tagged as "miners" verify the transaction by checking the balance of the user's "wallet," and then if approved move it into a chain of blocks with other transactions. Adding the transaction block to the chain of previous blocks requires the miners to solve a large cryptographic problem. This problem is how the Bitcoin Core can maintain a secure data flow, as solving these problems requires a lot of computational power that would be very expensive to take on as an individual trying to falsify transactions. The work of the original miner of the transaction blocks is then verified by other miners (peers) in the network. If the answer is subsequently verified by the other miners in the network, then the transaction has been completed. Once the transaction has been completed it will be added to the blockchain, and the person on the other end of the transaction will receive their funds.

Concurrency

In a concurrent environment, threads or processes share common resources while executing basically independently. Concurrency can be achieved through either context-switching on a single-core environment in which threads appear to run simultaneously, or parallelism on a multi-core environment in which threads do indeed run simultaneously (Abhisek, 2018). In the case of Bitcoin Core, the system is multi-threaded in many aspects (Bitcoin Developer, n.d.). Bitcoin Core achieves such concurrency by using multiple threads and locks guarding shared data structures (bitcoin / bitcoin, 2023). On a multi-core machine, Bitcoin Core is capable of running multiple threads to achieve parallelism; however, this parallelism is hindered by the fact that many functions require a "global lock" to be grabbed for work to be done, forcing other threads to wait for said lock to be released in order for them to resume their own work. This includes vital actions such running wallet tasks, validating blocks, and completing transactions with peers. Additionally, new updates have recently enabled users to have separated wallets running concurrently, which is very useful for use-cases such as individuals having individual wallets for business and personal use (Newbery, 2017).

Moreover, P2P networks such as Bitcoin Core are able to handle a large quantity of concurrent connections without issue due to the decentralized structure of this architecture style (Gautam, n.d.). In Bitcoin Core, peers can maintain many concurrent connections to other peers. To control traffic, the default settings limit the number of concurrent connections to other peers to 125 inbound and outbound connections (Reduce Traffic, 2020) (Grundmann, Baumstark, & Hartenstein, 2022).

Implications of Division

Error! Bookmark not defined. The Bitcoin network is controlled by the users of the network, meaning they all have a vested interest in making it function the best it can (Guardian, 2022). Multiple minds working towards the same goal leads to more, and faster improvements. One advantage of spreading the work out over so may developers is that they all have different ways of thinking about and therefore solving a problem. There will also always be developers out there who want to sabotage instead of helping to improve the software, but because each user can choose what software and which version to run and there are only a select few people who can make direct changes to the code, this isn't very effective. If there was only one central version of the software and more developers with direct code access, malicious changes would be more consequential. There is only a small group of developers who can access the code for Bitcoin directly. This also limits the damage a developer can do to the software, be that with malicious intent or not. The ability to merge new code with the existing code can only be done by a very select number of people who have the title of Project Maintainer (Betz, 2021). These are the only people who have commit access to the Bitcoin core code, the rest of the Bitcoin core developers cannot make any changes themselves to the shared Bitcoin core code. The rest of the developers get to choose what part of the software they would like to work on, be that the documentation or testing of the software or the research and peer review of potential improvements to it (Betz, 2021). There is an issue tracker in the Bitcoin GitHub that developers can search through and find one that they would like to contribute to helping fix as well.

The lack of a dedicated team of developers that get paid to do the work means that there is a lot of fluctuation in the number of developers working on Bitcoin Core at any one time. They come and go with the fluctuations in its price, some may do it just as a hobby and only make a few contributions every now and then, and some others yet may just make a one-time contribution as part of a personal project/goal or as part of a hackathon. This fluctuation in developers also means that there is a fluctuation in knowledge and expertise being applied to the software. Someone who has a very good knowledge of the main coding language used, C++, may only be interested in making a few contributions when they have the time, whereas if they were a paid developer for the software their expertise would always be available to the project. At the same time, though, because this language is very common, it brings in many developers with all levels of knowledge, many of whom will be experts in it and will have the same or similar knowledge as each other.

With Bitcoin Core having so many layers of protection for the use, sharing, and owning of bitcoin, it comes as no surprise that the security and protection of the code itself would be very secure. This, along with having multiple developers who are mostly unknown, anonymous members of the Bitcoin community, makes security a very important part of Bitcoin Core. There are multiple levels of protection in place to keep the code safe (Lopp, 2018). Commit keys are one layer of this security, they act as fingerprints registered to specific individuals. However, these could easily be used by other people, but it is just one layer in the multilayer security of Bitcoin Core. Another layer being verify-commits, a script that any developer can run that checks the commit key of every merge since December 2015 (Lopp, 2018) and will only successfully complete if each line of code that has been changed or updated since that time has been verified by someone with a key associated with a project maintainer. This is still not enough to be fully secure, though, as keys can be stolen or a maintainer may have malicious intent, it does greatly decrease the chances of an attack on the code. Having layered security also adds to the defense, pull request security and release of code security mean that at every stage of the process, the requested code changes must be verified and checked for integrity.

Finally, the implications of having multiple community developers participating in the upkeep and extension of the Bitcoin Core software can be good for the personal growth and development of the skills of individual developers. They can learn and grow their coding abilities and knowledge of the software's structure in a supported environment. The ability to be part of such a huge project can be daunting, but a great way for developers to hone skills and contribute to something meaningful to them.

Diagrams

The following diagrams represent the use cases explained in the Use Cases section of this report. The first use case involves three parties in the sequence diagram – the user, the user interface that the user interacts with directly, and the Bitcoin wallet. It is assumed in all these cases that the security measures of account authentication and authorization have been completed successfully through signing in before each use case.

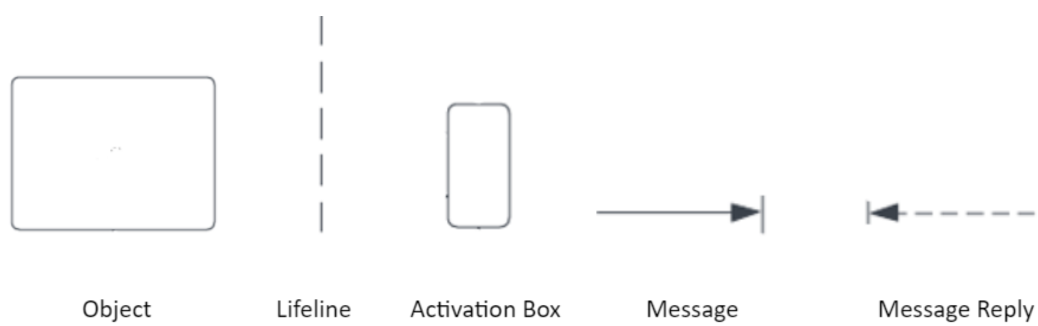

Figure 1: Sequence Diagram Legend

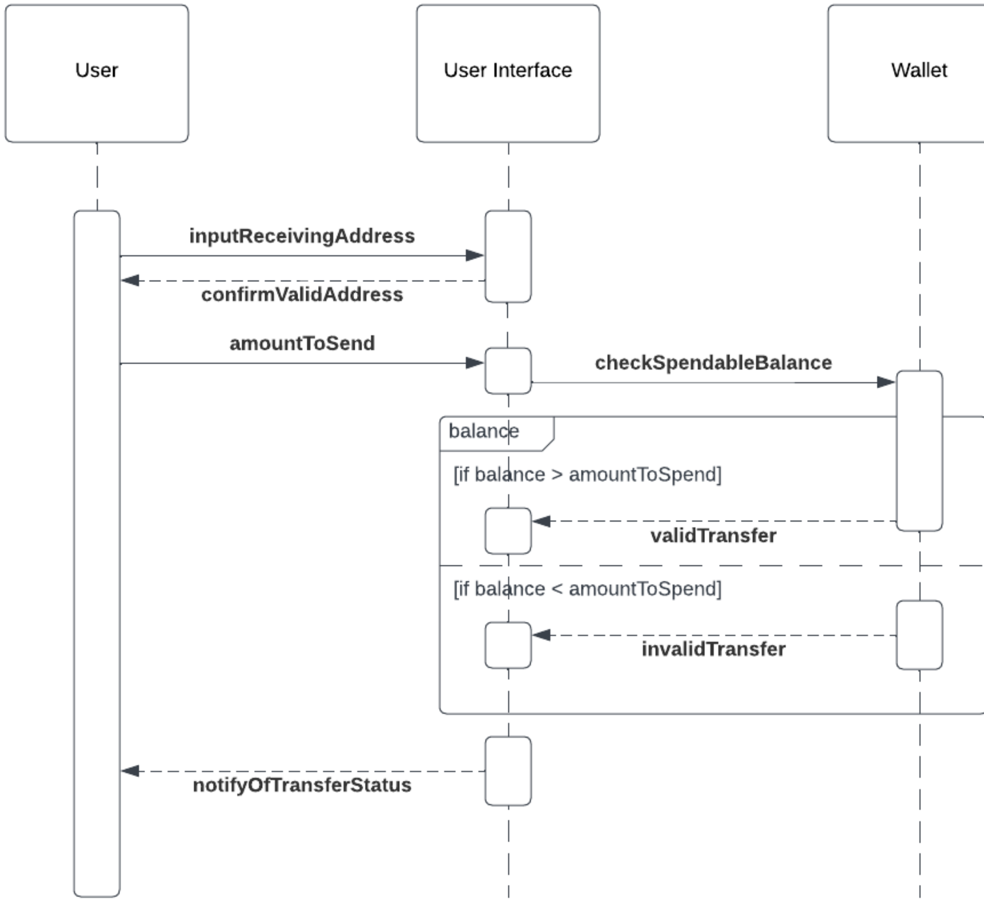

Figure 2: A sequence diagram depicting User A sending money to a User B if their spendable balance allows it (use case 1).

The next sequence diagram shows the process for use case 2, in which a user requests their account details to be shown, which includes their transaction history and their spendable balance. This diagram differs from the first slightly in that the Bitcoin wallet is not the third party being accessed now, it is instead the account database, which is assumed to store all user account information.

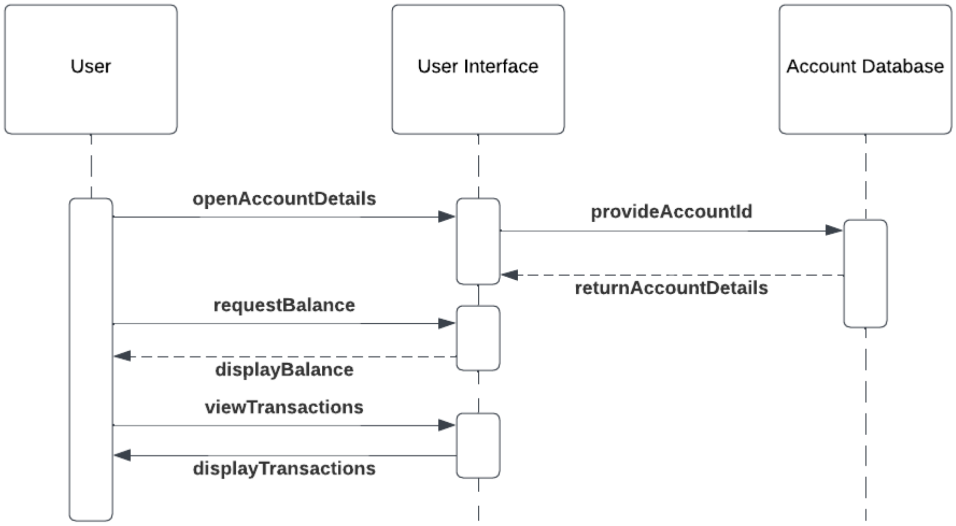

Figure 3: A sequence diagram depicting a user requesting their account details (use case 2).

Lastly, the sequence diagram below represents use case 3 in which the user decides to locally mine bitcoin. The Console is accessible through the account details tab in Bitcoin Core and interacts directly with the user's device to mine.

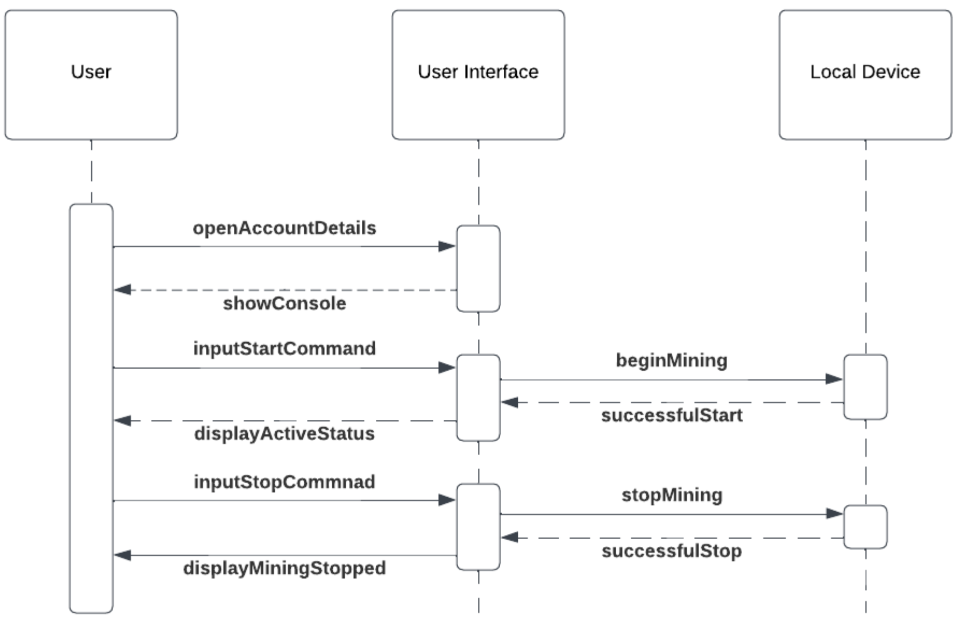

Figure 4: A sequence diagram depicting a user beginning and stopping a bitcoin mining session using their local device (use case 3).

External Interfaces

A number of external interfaces are present in the Bitcoin Core system. Lightweight wallets are secondary wallets for users to store their bitcoin funds in aside from the primary built-in Bitcoin

Core wallet. Connections to these lightweight wallets have similar security and privacy functions as connections to the user's Bitcoin Core. Lightweight wallets are usually connected to several random full nodes to send and receive all their data but can be connected to a user's Bitcoin Core as well (a trusted peer). Bitcoin Core can be used with either a graphical user interface (GUI) or command line modes, meaning information will be transmitted from the GUI to the system (mostly functions). External connections to the Bitcoin P2P network also transmit transactions, do validations, and connect to the blockchain.

Use Cases

There were three use cases identified while reviewing the functionality of Bitcoin Core that can be used to effectively describe a user's actions in the application.

The first use case describes a user A that wants to send bitcoin over to a user B. In order to send bitcoin to another user, a Bitcoin address must be referenced if it already exists or created. Once the address is acquired, Bitcoin Core can check user A's spendable balance and send the funds if their balance allows it. Then, the Bitcoin address that has just been used should be destroyed.

The next use case identified in Bitcoin Core is a user checking their balance or transactions. This would see the user navigate to their Bitcoin wallet and submit a request to return their spendable balance or view their transaction history, which includes past and pending transactions.

Lastly, another large portion of Bitcoin Core is mining. In this final use case, a user would begin mining bitcoin on their device. This would first require that the user opens their wallet in Bitcoin Core, and then navigate to the Console tab. To initiate mining, the user can enter a simple console command to begin. The user then terminates the mining when they wish by entering another console command (Henn, 2022).

Data Dictionary

Mining: Users solve cryptographic hash puzzles to verify new blocks, which are then added to the blockchain ledger.

Fiat: a currency issued by the government that is not backed by a commodity such as gold (CAD, USD, etc.)

Satoshi's: a currency equivalent to 100 millionth of a bitcoin, in order to allow for smaller transactions (smallest denomination)

Naming Conventions

P2P: Peer to Peer architectural style

TPS: Transactions per second

SPV: Simplified Payment Verification

GUI: Graphical User Interface

Conclusion

This examination of Bitcoin Core conceptual architecture led to many findings regarding its functionality, interactive parts, data flow, concurrency, and more. The P2P network architecture style at the heart of Bitcoin Core has many implications on the system; benefits such as increased adaptability and security, and downsides such as a bottleneck to scalability. Concepts such as a public blockchain, wallets with unique keys, and mining, work together to create a decentralized network of peers that are able to send and receive transactions while ensuring validity, integrity, and security. Furthermore, the open-source nature of Bitcoin Core enables further collaboration and innovation within the community. Overall, this in-depth analysis of Bitcoin Core's architecture revealed an ever-evolving, sophisticated system, that nevertheless has its downsides that offer opportunity for improvement. These areas for growth include building upon parallelism capabilities by minimizing functions that require global locks, and much more, which we plan to examine further in future reports.

Lessons Learned

Key lessons learned include the interaction of vital aspects of functionality such as blockchains, transactions, contracts, wallets, and mining. The structure of Bitcoin Core allows for its' strong security through the creation and validation of blocks in the blockchain, which users can get directly rewarded from. Bitcoin Core's P2P network takes advantage of a simple yet highly developed decentralized and peer supported system but may suffer from performance and scalability related issues as a result. Regarding the concurrency of the system, Bitcoin Core leverages both parallelism and context switching through multiple threads and locks guarding shared data structures, but the use of global locks limit the possibility of simultaneous activity. It is also important to note that Bitcoin Core is all about collaboration; its upkeep and continuous improvement rely solely on multiple people with all different types and levels of knowledge to work together.

References

(2018, December 29). *Concurrency, Parallelism, Threads, Processes, Async, and Sync — Related?* Retrieved from Medium: https://medium.com/swift-india/concurrency-parallelism-threads-processes-async-and-sync-related-39fd951bc61d#:~:text=In%20a%20single%20core%20environment,context%20switched%20between%20one%20another.

Betz, B. (2021, December 12). *2 prominent Bitcoin Core contributors step away from their roles*. Retrieved from Coin Desk: https://www.coindesk.com/business/2021/12/10/2-prominent-bitcoin-core-contributors-step-away-from-their-roles/

bitcoin / bitcoin. (2023, February 6). Retrieved from Github: https://github.com/bitcoin/bitcoin/tree/master/src/init.cpp

Bitcoin Developer. (n.d.). *Developer Guides*. Retrieved February 9, 2023, from https://developer.bitcoin.org/devguide/index.html

Chapter 3: 'Bitcoin Core: The Reference Implementation'. (n.d.). Retrieved February 9, 2023, from https://cypherpunks-core.github.io/bitcoinbook/ch03.html

Coinbase. (n.d.). *Crypto basics - What is a crypto wallet?* Retrieved February 9, 2023, from https://www.coinbase.com/learn/crypto-basics/what-is-a-crypto-wallet

Gautam, S. (n.d.). *Peer to Peer (P2P) Architecture*. Retrieved from Enjoy Algorithms: https://www.enjoyalgorithms.com/blog/peer-to-peer-networks

Guardian, N. (2022, July 19). *Who can change Bitcoin's core protocol?* Retrieved from The Guardian: https://guardian.ng/saturday-magazine/brand-intelligence/who-can-change-bitcoins-core-protocol/

Henn, P. (2022, June 15). *What is Bitcoin Core? A simple guide*. Retrieved from currency.com: https://currency.com/what-is-bitcoin-core-a-simple-guide

IBM. (n.d.). *What are smart contracts on blockchain?* Retrieved February 9, 2023, from https://www.ibm.com/topics/smart-contracts

Lopp, J. (2018, December 15). *Who Controls Bitcoin Core?* Retrieved from Cypherpunk Cogitations: Who Controls Bitcoin Core?

Muli, K. (2019, January 30). *The Blockchain Scalability Problem & the Race for Visa-Like Transaction Speed.* Retrieved from Towards Data Science: https://towardsdatascience.com/the-blockchain-scalability-problem-the-race-for-visa-like-transaction-speed-5cce48f9d44

Nakamoto, S. (n.d.). *Bitcoin: A Peer-to-Peer Electronic Cash System.*

Newbery, J. (2017, July 21). *What did Core contributors ever do for us?* Retrieved from Medium: https://medium.com/@jfnewbery/what-did-bitcoin-core-contributors-ever-do-for-us-39fc2fedb5ef

Reduce Traffic. (2020, July 10). Retrieved from Github: https://github.com/bitcoin/bitcoin/blob/master/doc/reduce-traffic.md

VISA. (2023, February). *Small Business Tools*. Retrieved from VISA USA: https://usa.visa.com/run-your-business/small-business-tools/retail.html