Bitcoin Core's Infrastructure
A Conceptual Architecture Report

CISC 322
Software Architectures
Professor Bram Adams

Group 34:
Sweet Home Ala-bram-a

Jean-Philippe Le Blanc - 20157203 - 18jpcl@queensu.ca (Team Leader)
Dallas Doherty - 20220683 - 19ded2@queensu.ca (Presenter)
Ethan Mah - 20224551 - 19ettm@queensu.ca (Presenter)
Ruairí O'Connor Clarke - 20235043 - 20romoc@queensu.ca
Muhammad Ibrahim - 20218324 - 19mi22@queensu.ca
Mayy Mounib - 20230803 - 19mlm15@queensu.ca

Submitted on February 19th, 2023

Abstract

Bitcoin Core is an open-source peer-to-peer software platform written in the C++ programming language. Bitcoin Core is the client software for the Bitcoin network. It uses the Bitcoin protocol and provides fundamental infrastructure required to operate and interact with the Bitcoin network. The platform is designed to support a variety of OS such as Windows, MacOS, and Linux. It also has a GUI for users interacting with the network. The GUI allows users to create and manage Bitcoin wallets that send and receive Bitcoin, and view transaction history. The system uses a peer-to-peer architectural style. The main functions of the platform is to maintain the blockchain, validate and relay transactions, and give access to historical transactional data.

The goal of this system is to facilitate online transactions between two parties without the need of a third party to oversee the transaction. It does this by taking advantage of the fully validating node. This node maintains the entire copy of the blockchain. Bitcoin Core remains one of the most secure transactional clients to breakthrough blockchain technology. This report explores the conceptual architecture of Bitcoin Core and its subsystems, and presents use cases of user interactions with the system. Included is a conclusion exploring our findings as well as lessons learned reporting what we learned while making this report. At the end, is our data dictionary and references.

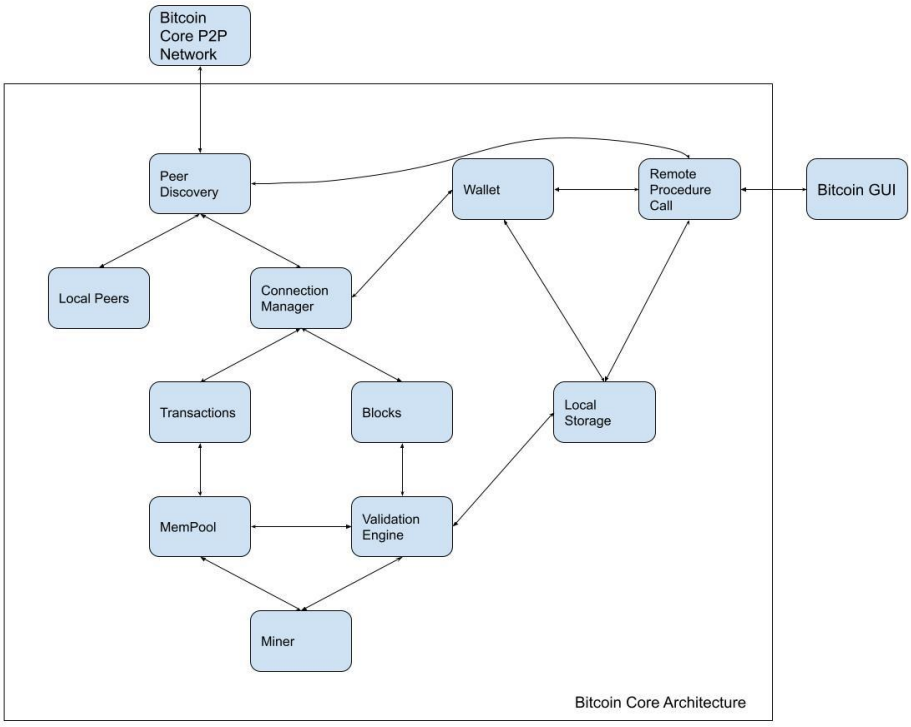
Introduction

In 2008, Satoshi Nakamoto sought to create a decentralized, digital currency that would allow for secure, transparent, and fast peer to peer transactions without a central authority. Satoshi's whitepaper on bitcoin addressed several key issues that plague the traditional fiat systems including a lack of privacy, high transaction fees and the inflationary pressures that can arise from the printing of new currency.

Bitcoin Core is the original and most widely used software for interacting with the Bitcoin Network, providing a digital wallet that allows users to send, receive, and store bitcoins. Bitcoin Core is also open source, enabling anyone to view, comment, or propose changes to the system. Bitcoin Core is the foundation upon which the entire Bitcoin network is built, providing the underlying technology and infrastructure that enables the secure and decentralized transfer of value, and it has been instrumental in the growth and evolution of the Bitcoin network.

Bitcoin Core operates on a transparent ledger without a central authority, known as the blockchain, which is built on a peer-to-peer software architecture. It allows for secure and transparent peer to peer transactions. In this report we will examine in detail the conceptual architecture, including the system functionality and interacting parts, system evolution, control and data flow, concurrency, and the implications for division of responsibilities among participating developers. Through this exploration, we aim to provide a comprehensive understanding of the conceptual architecture of Bitcoin Core.

Conceptual Architecture



Conceptually, due to the nature and focus of the system, Bitcoin Core architecture implements a peer-to-peer (P2P) architecture; this is further explained in the peer-to-peer network section. The P2P architecture can be broadly divided into four layers for ease of understanding. Each of these layers represents a conceptual aspect of the system.

Network Layer:

This layer is responsible for handling communication between the nodes in the Bitcoin network. It uses the peer-to-peer protocols to transmit transactions and blocks between nodes. Information about transactions, blocks, and other network-related data are shared from node to node until the entire system is aware of the information. This keeps the network in sync and ensures that everyone is up to date with the most recent information

Transaction Layer:

This layer is responsible for managing transactions within the Bitcoin network. It verifies transactions, creates new transactions, and broadcasts transactions to the network. When Bitcoin is sent to another user; the wallet creates a transaction message containing information about the sender, the recipient and the amount being sent. A unique digital signature for this message is generated by mathematically mixing it with the sender's private key. Contracts are also verified through this layer. Bitcoin uses a simple programming language called Bitcoin Script to encode contracts that are used to define the conditions under which Bitcoin can be spent as a form of currency. These contracts allow users to create transactions with other users, giving them the ability to

set their conditions, provide Bitcoin addresses, spread their transaction across multiple parties, and even exchange between different cryptocurrencies.

Blockchain Layer:

The blockchain is a collection of records linked with each other that are strongly resistant to alteration, and protected by complex algorithmic cryptography. Miners validate blocks and add them to the end of the chain. They do this by solving complex mathematical problems using computer hardware in exchange for an amount of newly minted Bitcoin. As the blockchain grows the problems get increasingly more difficult. This allows for a higher level of security within the blockchain, but also requires the miners to put in more work within each new iteration.

Application Layer:

This layer consists of the GUI, wallets and the Bitcoin Core API. The graphical user interface provides a user interface for interacting with the Bitcoin network. Bitcoin wallets are a tool that manages a user's Bitcoin holdings and sent or received transactions. The wallet feature provides users with the ability to manage Bitcoin transactions, view balances and transaction history, manage and create private keys, and keep backups of wallet data. Lastly, the Bitcoin Core API, which allows developers to build applications on top of the Bitcoin network.

Within the Bitcoin Core system there are multiple components that interact with each other to play a critical role in the functionality of the system. These components work together to enable secure, decentralized transfers of value on the Bitcoin network. These components are as follows:

Peer-to-peer network

The peer-to-peer (P2P) network architectural style is a type of decentralized network architecture that allows peers to act as both the client and the server. Peers are equally privileged, equipotent participants and partition workload amongst each other evenly. This means there is a direct communication between nodes, which makes a server facilitating these messages redundant.

In the Bitcoin Core system, all communications occur from node to node using secure protocols in order to ensure security and privacy. Communications occur in order to share information and to validate transactions. Block and transaction exchange are collaboratively maintained by full nodes (peers) on the P2P network, and nodes also store all historic transactions. When a new transaction is initiated, a signal is broadcasted to all the other nodes. These nodes read the signal and start to validate the transaction. Once validated, the transaction is stored in a local memory until the block is ready to be added to the blockchain. The idea is that when a transaction occurs, all the nodes in the network add it to their local blockchain, once the block reaches a certain size, a consensus is reached among peers to decide if all the local blockchains match the others. If the consensus is that it does, the official blockchain is updated and the process starts again. This is the main purpose of a decentralized system as there is not a single entity or user that has the power to alter the blockchain, it is by consensus.

The advantage of using a P2P network versus any other architecturals style is based on the concept and vision of Bitcoin Core. “A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution.”. There are many styles that clearly do not have the capabilities and the structure required for such a system. A client-server system would mean that there is one central governing power that would have control of the blockchain as users would request and receive information directly from one place. This means that there is no guarantee that the blockchain has not been altered for someone’s benefit as they could alter the ledger without the user’s knowledge.

Control and Data Flow

The control and data flow within a P2P system is special, as all peers are both the client and the server, meaning there is no central authority or server controlling the communication between nodes. All peers communicate directly with one another and they all have equal control of the network. This means that each node in the network is responsible for both sending and receiving data, and the control of the system.

Conceptually, the nodes in a decentralized P2P network communicate with each other through a variety of protocols and algorithms to ensure that the data is transferred securely and efficiently. For example, nodes may use a consensus algorithm to agree on the state of the network and ensure that each node has a consistent view of the data. All the data required is distributed amongst everyone, so if something is needed, the request is broadcasted to other nodes in the network, and any node that has a copy of the data can respond to the request and send the data back to the requesting node.

In Bitcoin Core, proof of work (PoW) is the consensus algorithm style that is implemented. This is a cryptographic algorithm that requires miners to perform computationally demanding tasks in order to create new blocks on the blockchain. The first miner that solves the PoW puzzle is rewarded with newly minted bitcoins and the right to create the next block in the blockchain. After a new block is created, as explained above, a signal is broadcasted to the other nodes in the network. Each node then validates the block before adding it to their own copy of the blockchain. This consensus algorithm contributes to the control of the system as it determines who gets to add the next block to the blockchain. The data flow within Bitcoin Core is managed by the P2P network and using the “gossip protocol”. This protocol states that each node propagates information to their neighboring nodes in order to ‘spread the word’. This data flow is done securely using various protocols in order to protect the system.

Overall, the control and data flow in decentralized P2P networks are distributed among the participating nodes, allowing for a more resilient and flexible network that adapts to changes and continues to operate even if there are offline or maliciously behaving nodes. Each node takes care of their own control and data flow by communicating with neighboring nodes when an event occurs.

Payment Processing

The steps to process a payment are generating a bitcoin address, receiving the payment, verifying the incoming transaction, sending the payment, and verifying the outgoing transaction. Bitcoin Addresses can be infinitely generated and each address is associated with a private key, which is used to sign transactions and prove ownership of the address. Bitcoin Core uses a system of unspent transaction outputs (UTXOs) to track the balance of each Bitcoin address. When a transaction sends Bitcoin from one address to another, it creates one or more new UTXOs for the recipient and spends one or more existing UTXOs from the sender. Each UTXO is associated with a specific transaction output and can only be spent once. Bitcoin Core's wallet allows a user to manage their Bitcoin addresses and UTXOs.

The nodes in Bitcoin core are what verify this transaction by making sure the transaction adheres to proper formatting and making sure that the sender of the transaction has enough bitcoin on their profile to transfer the amount. Once verified, it is added to the mempool. To confirm a transaction in the mempool a miner must solve the associated cryptographic puzzle. Doing this confirms the transaction and adds it to the blockchain. Once added to the blockchain the transaction is final.

Operating Modes

A vital part of the Bitcoin Core architecture is the ability to validate the blockchain and there is one main, secure method by which Bitcoin Core achieves this. This highly secure method is called the "full node" method of validation. Also known as a "thick" or "full chain" client, a full node downloads the entire blockchain to validate it by the most recently discovered block's "height" in the chain. A block's height is defined by the number of blocks below it which indicates the computational difficulty of the current block. This is a highly secure method of validation because in order to fool a client into believing a false chain is true, an entirely new chain of greater computational difficulty to the current "true" one would have to be made. This would be extremely difficult and computationally expensive to the point where it's nearly impossible as the chain having the most cumulative proof of work is what defines the "true" chain. This creates a method of transaction verification which is highly secure, especially against sybil attacks since only one real or "honest" network peer is needed to receive and verify the complete true block chain.

System Evolution

In Bitcoin Core there will be two main components that are constantly evolving over time. The first is the system itself. Due to the open source nature and the community that supports cryptocurrencies, the system is constantly evolving, improving and changing over time. There will constantly be new features, functionalities and improvements being made to the system over time as all software systems do. Aspects such as the protocols, the scalability, the privacy, the usability, the consensus and the security are constantly being tested and improved on.

The second is the cryptographic puzzles solved by miners. The concept of these puzzles is that they constantly get harder over time in order to regulate the rate at which bitcoins are released into the system. If blocks are being added too quickly, the puzzle becomes more difficult, making it harder for miners to solve and slowing down the rate at which new blocks are added. If blocks are being added too slowly, the puzzle becomes easier, making it easier for miners to solve and speeding up the rate at which new blocks are added.

These two components are crucial to the stability of Bitcoin Core as without this evolution over time, the system will not be able to maintain itself. However, one thing that is certain that won't evolve over time is the system architecture. The implementation of subsystems will change over time but the peer-to-peer nature of Bitcoin Core is here to stay. It is the fundamental building block of the system and is impossible to change without completely restructuring the system. Sub systems within the architecture and the implementation may be improved and updated, but this will be done with the P2P architecture in mind.

Concurrency

The Bitcoin Core system heavily relies on concurrency to process numerous transactions while simultaneously ensuring the security and integrity of the blockchain. In the event of a blockchain failure, the verified transaction ledgers would no longer be reliable and the entire network may become unprotected— causing users to withdraw from the system. As such, the concurrent components in the software are essential in preventing malicious threats.

Through the use of multi-threaded architecture, the Bitcoin Core system is able to execute various different tasks in parallel and efficiently manage incoming and outgoing data. These tasks, which often do not depend on a central authority, include the ability of nodes on the network to receive and validate multiple transactions, and the ability of miners to work on multiple block solutions. With the help of software tools like the Berkeley DB database library and the ZeroMQ messaging library, the system is able to access data through multiple threads and facilitate interprocess communication concurrently.

Concurrency is also achieved in Bitcoin Core through the use of smart contracts, which are self-executing contracts where the terms of an agreement are already embedded into the code of the blockchain. These contracts can be used to prompt transactions based on specific conditions and enable the creation of decentralized applications that can execute on the network.

It is important to note that along with concurrent processes, specific difficulties may occur as a consequence. There is a possibility of race conditions and synchronization issues happening due to the concurrent access to the blockchain data structure. To overcome this, Bitcoin Core implements locking and atomic mechanisms which prevent multiple threads from simultaneously modifying resources and ensure that instructions execute without

interruption. Together, these measures ensure that the network and blockchain are protected and updated correctly.

Division of Responsibilities Amongst Developers

Given that Bitcoin Core is an open-source system that uses P2P architecture, it is challenging to determine exactly how tasks are allocated in the development process. The entire software is available in a GitHub repository, allowing anyone to contribute in any area of development that they wish to focus on.

To understand the division of responsibilities amongst the developers, it is important to note that the team is split into two categories: contributors and maintainers. The contributors are considered to be anyone who contributes code, reviews, tests, or any other valuable improvements to the repository. The maintainers, on the other hand, have permission to change the original source code of the system and are responsible for reviewing and merging pull requests from contributors.

Since there isn't a specific division of responsibilities among the developers, each contributor is able to put their expertise towards the improvement of different aspects of the system. These aspects include but are not limited to, the development of the user interface of the Bitcoin wallet, the development of the security features, and the maintenance of the Bitcoin protocol which governs how its network operates. Any proposed changes and upgrades to the system are peer-reviewed in order to ensure their credibility and correctness. Overall, the open-source nature of the Bitcoin Core system demonstrates an efficient development process that allows its team to collaboratively maintain and improve the network.

External Interfaces

When a user installs the Bitcoin Core application, they are presented with a simple GUI that allows them to interact with the system and perform various actions. Users can send and receive transactions, but they can also view the blockchain and view any previous transactions. The Bitcoin Core stores a copy of everything locally, as to keep the system resilient to any possible harm. Files are encrypted with the most secure encryption algorithms in order to maintain the highest level of security.

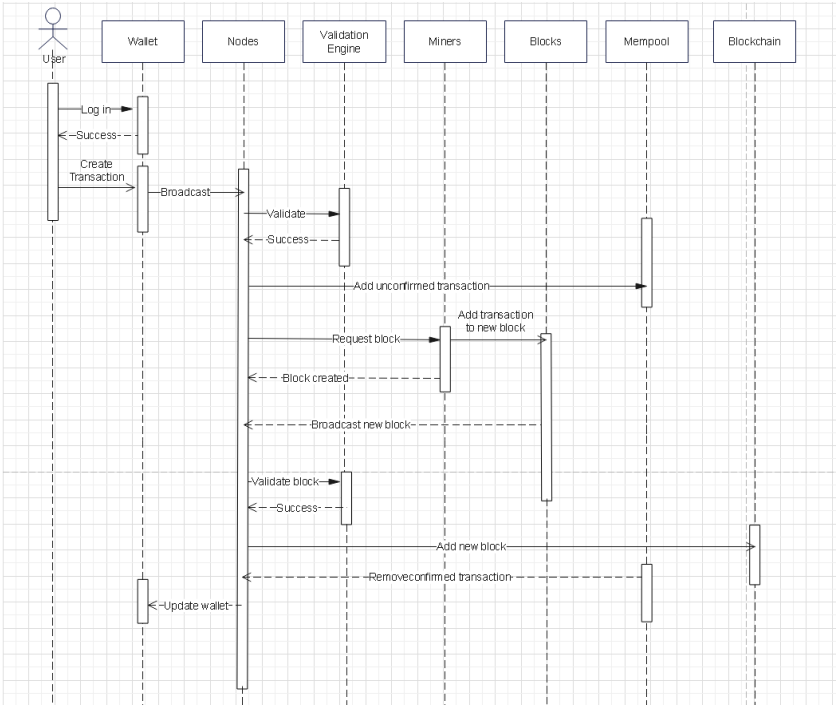
The blockchains historic data is stored on every user's computer. This contains all the historic transactions that have ever occurred, and can be sent to any new users that need to be brought to date on the current blockchain. This ledger is constantly getting updated whenever a new block is added. The P2P network stores the IP addresses of other nodes, transaction data, and blocks of the blockchain. The system keeps some files on hand such as the configuration data and the debugging data. The config data is used to customize Bitcoin Core to use various control settings, such as network connection settings, mining settings, and other system preferences. The debug data is used to troubleshoot and diagnose problems within the system.

When a transaction is created, for example a request for 1 bitcoin from user A to user B, a signal is published to all the neighbouring nodes warning them of the new transaction. These nodes receive data including the amount in the transaction, and both users public keys. The data is passed along the network using the gossip protocol until all nodes have heard about the transaction. Once verified, the transaction is added to the Bitcoin blockchain, which is downloaded and stored on the user's computer. During this whole transaction, data is being propagated throughout the system to every node in order to keep every blockchain up to date.

Use Case Diagrams

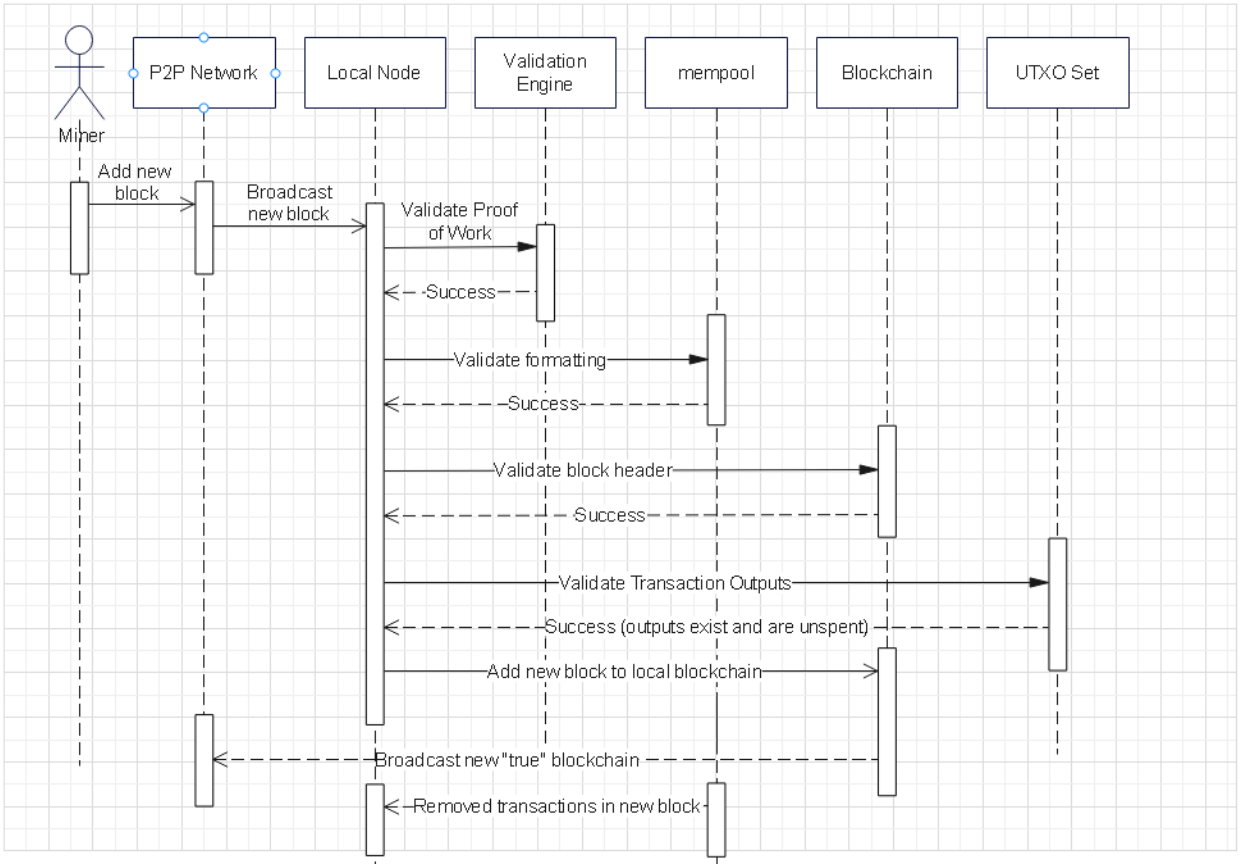
Use Case 1 - Transaction:

Once the user starts up their local Bitcoin Core wallet, they can create a transaction. When creating a transaction the user specifies the amount to send, the receiving address and the fees they are willing to pay. Once a transaction has been created it is broadcast to the nodes in the network. Once a node receives a transaction it verifies that it meets the bitcoin protocol rules then adds it to the mempool, once verified it sends it to the other nodes until the majority have verified the transaction. After the nodes have verified that this is a valid transaction, it must be included in a new block. Miners create these blocks and decide what transactions are included in new blocks. The new block is then broadcast to the node network which then verifies that it is a valid block. If it is declared valid the nodes add it to the blockchain and remove it from the mempool. Finally, the user's wallet will update, reflecting the new transaction and the change in their balance as a result.



Use Case 2 - Adding a block to the blockchain:

When there is a new block to be added to the blockchain it must be validated by the entire peer to peer network. First, it will be broadcast to all nodes in the network from a miner which will check that it has a valid proof of work. Once this has been confirmed the nodes check it against the current mempool which holds a collection of the unconfirmed transaction each node knows about, this includes ensuring that its inputs, outputs and fees are formatted correctly. Once the block has been verified by the mempool it must be verified against the current blockchain to ensure it is being added correctly. To do this the network compares the hash in the header of the previous block with the current one. After the header has been verified the last validation step is to ensure all the transactions in the new block exist in the UTXO set (unspent transaction output) meaning the output from each proposed transaction in the new block exists and has not been spent. Once the new block has passed all validation steps each node will add it to their local blockchain and then broadcast that to the network as the new “true” copy of the blockchain, it will also update the mempool to remove any transactions included in the new block.



Conclusion

In conclusion, Bitcoin Core's architecture is designed to handle a copious amount of transactions while maintaining a decentralized and distributed network. The system's control and data flow is commanded by the peer-to-peer network, with nodes communicating using defined protocols and algorithms, ensuring that data is transferred securely and efficiently. The system was created with evolution in mind. Changes to the platform are made through a consensus process and open source development. Concurrency is also a fundamental aspect of Bitcoin Core's architecture. This allows for multiple transactions to be processed simultaneously, crucial to the scalability of Bitcoin Core. Lastly, the open-source development has helped Bitcoin core maintain a high level of security and decentralization.

Lessons Learned

Throughout the completion of this report, our team had many positive interactions that allowed us to collaborate effectively as a group. After splitting up our responsibilities and working independently, we learned how to effectively communicate with each other in order to set achievable deadlines and share progress. We also learned the importance of allocating enough time to work on different aspects of the report to ensure our presenters had enough time to film their presentation. Additionally, completing this report gave our team a lot of insight into the practical considerations of the Bitcoin Core architecture. We were each able to familiarize ourselves with an aspect of the development of the Bitcoin network, which has prepared us for our second report regarding the concrete architecture.

Data Dictionary

Peer-to-peer (P2P): a computer network in which each computer can act as a server for the others, allowing shared access to files and peripherals without the need for a central server.

Proof-of-work (PoW): A consensus mechanism where users must solve a cryptographic puzzle to get a reward.

Unspent Transaction Output (UTXO): is the technical term for the amount of digital currency that remains after a cryptocurrency transaction.

Graphical User Interface (GUI): A graphical way to interact with a system.

References

CoinMarketCap. (2021, December 15). *Who are Bitcoin Core's developers?: CoinMarketCap.* CoinMarketCap Alexandria. Retrieved February 17, 2023, from https://coinmarketcap.com/alexandria/article/who-are-bitcoin-cores-developers

Developer guides. Bitcoin. (n.d.). Retrieved February 17, 2023, from https://developer.bitcoin.org/devguide/index.html

Henn, P., & Henn, P. (2022, June 15). *What is Bitcoin Core?: What is it and how to use it.* Retrieved February 17, 2023, from https://currency.com/what-is-bitcoin-core-a-simple-guide

Müller, P., Bergsträßer, S., Rizk, A., & Steinmetz, R. (1970, January 1). *The bitcoin universe: An architectural overview of the bitcoin blockchain.* Startseite - Digitale Bibliothek - Gesellschaft für Informatik e.V. Retrieved February 17, 2023, from https://dl.gi.de/handle/20.500.12116/16570

What are smart contracts on Blockchain? IBM. (n.d.). Retrieved February 17, 2023, from https://www.ibm.com/topics/smart-contracts

What is Bitcoin Core?: River Learn - Bitcoin Technology. River Financial. (n.d.). Retrieved February 17, 2023, from https://river.com/learn/what-is-bitcoin-core/#:~:text=Bitcoin%20Core%20is%20the%20most,with%20consensus%20from%20the%20network.

YouTube. (2021). *Understanding Bitcoin Core: The Reference Implementation.* Retrieved February 17, 2023, from https://www.youtube.com/watch?v=wLYdcH37phE&ab_channel=UnderstandingCrypto.

(n.d.). The Bitcoin Universe: An Architectural Overview of the Bitcoin Blockchain. Retrieved February 18, 2023, from https://dl.gi.de/bitstream/handle/20.500.12116/16570/DFN-Forum-Proceedings-001.pdf?sequence=1&isAllowed=y

Bitcoin Transactions - from "Send" to "Receive". (2017, September 21). YouTube. Retrieved February 18, 2023, from https://www.youtube.com/watch?v=ZPFL6R-voW0

Double spend (problem) - Glossary | CSRC. (n.d.). NIST Computer Security Resource Center. Retrieved February 18, 2023, from https://csrc.nist.gov/glossary/term/double_spend_problem

How does Bitcoin mining work? - BBC Newsnight. (2018, January 24). YouTube. Retrieved February 18, 2023, from https://www.youtube.com/watch?v=cRxL2GKDU5E

What is Bitcoin? Bitcoin Explained Simply for Dummies. (2018, April 4). YouTube. Retrieved February 18, 2023, from https://www.youtube.com/watch?v=41JCpzvnn_0

Blockchain And Cryptocurrency Explained In 10 Minutes | Blockchain And Cryptocurrency | Simplilearn. - Simplilearn, (2022, March 29). Youtube. Retrieved February 19, 2023, https://youtu.be/MFw8Ax0p7dA

The Bitcoin Blockchain Explained. - IEEE Spectrum. (2016 July 6). Retrieved February 19, 2023, https://youtu.be/M_zCjjy59cg

Nicholas Roth. "An Architectural Assessment of Bitcoin: Using the Systems Modeling Language." Procedia Computer Science, Elsevier, 16 Mar. 2015, https://www.sciencedirect.com/science/article/pii/S1877050915003026?ref=pdf_download&fr=RR-2&rr=79c2b2869c54a220.

Ethereum. "Proof-of-work (POW)" https://ethereum.org/en/developers/docs/consensus-mechanisms/pow/

Vitor Mesk. "Gossip Protocol" https://academy.binance.com/en/glossary/gossip-protocol