CISC 322
Assignment 1: Report
Conceptual Architecture of Bitcoin Core
February 19th, 2023

Team BigTime

Azeem Khan (Presenter 1) - 19aik2@queensu.ca
Ben Tomkinson (Presenter 2) - 17bat3@queensu.ca
Lucas Wong (Team Lead) - 20lylw@queensu.ca
Kenny Wong - 20klkw@queensu.ca
Oscar San - 19os14@queensu.ca
Yannik Brunzema - 19ycb@queensu.ca

Table of Contents

Abstract
Introduction
Derivation Process
Overview of Conceptual Architecture
Evolution Process
Implications of Division of Responsibilities
Control and Data Flow
Concurrency
Conclusion
Lessons Learned
Data Dictionary
Naming Conventions
References

Abstract

While the commercial finances industry on the Internet has become almost completely integrated with large financial institutions, Bitcoin Core introduces a completely peer-to-peer way to manage finances, while skipping any sort of financial institution or middle-man. The focus of this paper is to outline and represent in detail, the conceptual architecture of the Bitcoin Core software. Bitcoin Core is an open-source client based on the original Bitcoin Protocol, which allowed a variety of methods to be explored. We proposed a process that outlines each inner-working of the software and goes into a comprehensive analysis of each interaction. This includes the derivation process, and the long evolution process the software has gone through over the years. As long as there are and continue to be constant upgrades and implementations to software security, privacy, efficiency, and scalability, it will remain the sole choice for a majority of enthusiasts. The paper then explores the mechanics and interactions of each key component, by examining how one other interacts with each other in the control and data flow. This directly relates to the issue of concurrency, and how Bitcoin Core tackles that issue through a mix of methods including using peer-to-peer networking, and unique synchronization methods to ensure efficiency and security.

Introduction

Bitcoin Core, created in 2009 by Satoshi Nakamoto is the open-source software implementation of the Bitcoin network, a decentralized digital currency system that allows for peer-to-peer transactions without the need for a middleman or bank service. The conceptual architecture of the Bitcoin Core is designed with the purpose of conducting highly secure transactions over a stable network. The main components of Bitcoin Core consist of both node-based software for validating the blockchain, and a bitcoin wallet. Each node in the Core network runs on an individual version of Bitcoin and connects to the network via thousands of individual nodes. The base of the Bitcoin Core software is to process secured transactions and deals over the blockchain while masking the signatures and identity of the transactions and parties involved.

The introduction of the Bitcoin Core had a significant impact on the cryptocurrency industry and remains the original, most used implementation of the Bitcoin Protocol. It meant a new way to interact and conduct financial transactions, without the need for a Government regulated financial system or bank. Bitcoin Core was responsible for many of the key features identified in several common cryptocurrencies, such as security techniques, and storing transaction data in the blockchain. The introduction of this software quickly spread and created for itself, a community of avid users, developers, and stakeholders. This made it easy to transition the software from a command-line-based program to the inclusive and comprehensive ecosystem it is today.

The goal of this report is to research the conceptual architecture of Bitcoin Core, and the role it plays in working with the Bitcoin network. This includes researching its key design principles, and the inner workings surrounding Bitcoin Core. The report is broken down into multiple subsections, each containing a different part of Bitcoin Core's conceptual architecture;

The Derivation Process, a general overview of the Conceptual Architecture, Bitcoin Core's Evolution Process, the Control and Data flow of the software, as well as a Data Dictionary and general naming conventions. Our closer, "Lessons Learned", encapsulates the collective challenges and inner workings of our team, as well as any valuable lessons learned during the writing of this report.

Derivation Process

Our team's strategy in deriving the conceptual architecture of Bitcoin Core consisted of the following procedure: the topic of research, analysis of structure, solidification of core components, and creation of the final architecture.

Starting out, the team had a collectively weak idea of what the software architecture of a cryptocurrency like Bitcoin would appear as. This lack of info led to an extensive investigation of the new technology, and what its core components would look like. Through this research, a general understanding of the fundamental pieces within crypto, such as blockchain systems and user security, was fully integrated into the team's knowledge as a foundational base.

After studying the topic of cryptocurrency, we delved into the resources pertaining to the overall structure of Bitcoin to analyze its components and their connections. Satoshi Nakamoto's paper *Bitcoin: A peer-to-peer electronic cash system*, provided us with the core information to understand the overall structure and components that would make Bitcoin run. The source code of Bitcoin Core on GitHub gave us the opportunity to properly pinpoint which of those components were largely relevant in the current architecture. To fully understand the functionality of each component and their interactions with other aspects of the system, the official developer guide was analyzed thoroughly to accomplish this. The GitBook on Bitcoin Core was used to consolidate any loose information regarding the subsystems of the software.

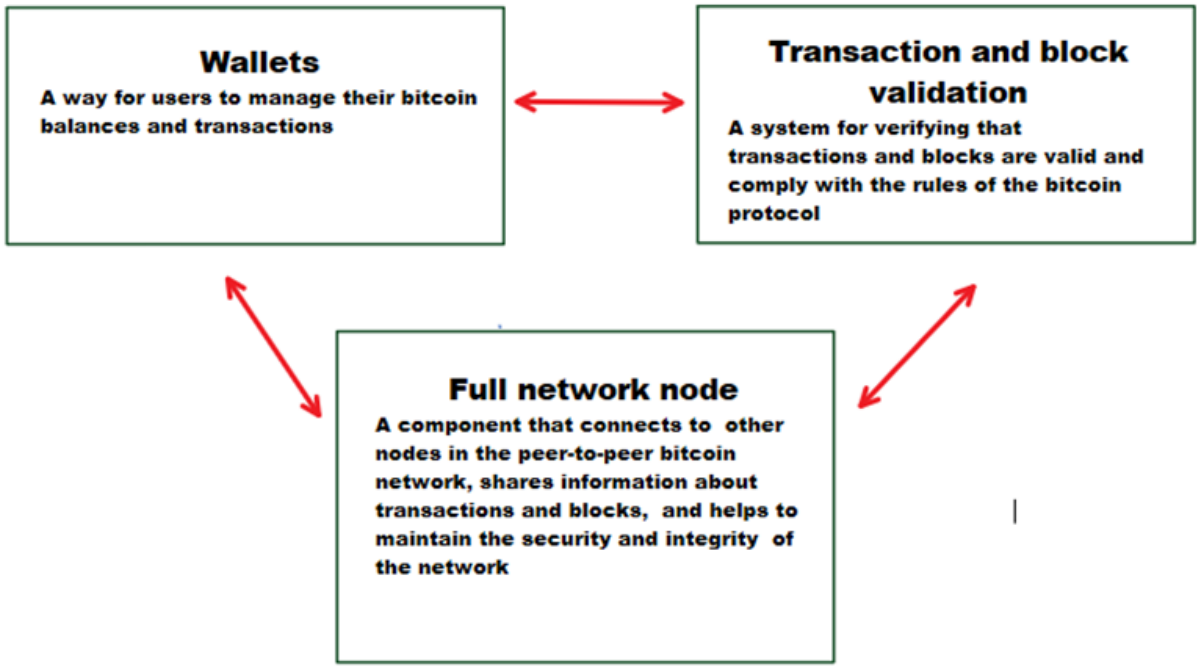
Once the team finished analyzing the overall structure, we used the information gathered to create a rough draft containing every core component present within the overall architecture of Bitcoin Core. The most important components were written down, and the challenge of creating a cohesive system of interactions between them began. After a long discussion on each component's functionality and looking back on research, the team discovered the software followed a peer-to-peer style of architecture with some elements of the implication invocation style. With this knowledge, we created a couple of subsystems modeled around their concepts which contained the numerous components and their connections to one another.

With the completion of the rough draft of subsystems, our work on the creation of the conceptual architecture began. With the help of our rough draft, the final iteration of the architecture is displayed in the diagram below.

Overview of Conceptual Architecture

The Bitcoin Core architecture is designed to be modular, with each component being largely independent of the others. This allows the components to be reused for different

applications, while also making it easier to maintain and update the codebase. Below shows an overview of subcomponents of the Bitcoin core including the wallets, transaction, and block validation, and full network nodes which all interconnect.



Wallets rely on the transaction and block validation engine to ensure that transactions are valid before being added to the user's balance, and to ensure that new blocks are valid before being added to the blockchain. The full network node component helps wallets to receive information about transactions and blocks from other nodes on the network and to broadcast new transactions and blocks to the network. Similarly, the transaction and block validation engine relies on the full network node to receive and transmit information about new transactions and blocks, as well as to detect and respond to misbehaving nodes on the network. The transaction and block validation engine also communicates with the wallet component to update user balances and ensure that transactions are properly signed and authorized. The full network node component serves as the backbone of the Bitcoin network, connecting to other nodes in the network and facilitating the exchange of information about transactions and blocks. It relies on both the wallet and transaction and block validation components to ensure the security and integrity of the network.

The architecture of Bitcoin Core itself mainly adheres to a peer-to-peer style. In essence, the Bitcoin network protocol enables peers, acting as full nodes, to collaboratively maintain a network where block exchange occurs. By implementing this style, the main problem of double-spending within a pure peer-to-peer network is solved. Bitcoin's network, serving as the connector between peers, hashes transactions into a continuous chain of proof-of-work, forming a timestamp record that cannot be undone without redoing the related proof-of-work. This chain serves as proof of the sequence of events that occurred coming from the largest pool of CPU

power, and as long as these nodes controlling the CPU power are not cooperating to attack the network, the implementation of this peer-to-peer style will outpace attackers through generating the longest chain. The network also acts on an implication invocation style, where peers can join and leave the network when they want to register for a transaction, while messages are broadcast on a best-effort basis. Using this information, it is easy to see how each of the wallet, full network nodes, and the transaction and block validation subsystems interact with each other and follow both styles of architectural design within the software.

Evolution Process

The Bitcoin Core system has evolved drastically over time, with most of the changes coming from a process of community-driven changes, which are discussed by major developers, users, and stakeholders. Most of the evolution that Bitcoin Core has gone through, has been influenced heavily by both its users and the change in technology over time. While the initial release of Bitcoin Core (released in 2009) only allowed users to send and receive transactions, the software has since caught on to date with its competitors, and other Bitcoin clients.

When it comes to factors such as the increasing capability of technology in this day and age, developers are heavily influenced towards implementing these new technologies, to not only stay up to date with the latest functions but for security purposes as well, as with new advancements in technology, comes new ways to break through systems and firewalls. Another factor that is kept in mind when evolving the system is user demand. While implementing new features and technology is a key evolution process of Bitcoin Core, a large chunk of features and improvements that are implemented are voted on by the community of users, as well as edge cases that are found by users. A majority of Bitcoin Core's key updates came in waves, such as the earliest large update in 2010, which introduced a functional GUI as well as concurrency within the system (multithreading). Like many of the updates and implementations introduced, the introduction of a user interface came as a result of many beginners finding the command line interface confusing, as it was turning away new users, which was dangerous as the software was still in its early stages. 2013s update focused on performance and security, with the introduction of pruning which enabled the system to handle more transactions at a time, and solved issues regarding limited storage, as this new feature would get rid of old transaction data from the storage, allowing for the constant updating of the system, constantly freeing up space for new users. Version 0.8 (announced in 2013) also introduced a new security feature, that would require the verification of multiple transaction signatures in order for a singular transaction to process. This came as a result of advancements in technology, which presented security threats to the system.

In 2017 one of the most major updates was implemented, in Version 0.13, the Segregated Witness (SegWit), which focused on improving the software's scalability and security. This specifically targeted the software's ability to process more transactions, and improve its capacity. It did this by separating the transaction signature from the data, which allowed for faster processing of both facets, leading to an overall improvement in efficiency and scalability. This also allowed for more versatility of the transaction, as separating 2 of the largest components helped implement new security features and space creation, as handling both the signature and transaction data was no longer an issue. Most of the recent features that have caused the system

to evolve have been focused on efficiency, security, or some sort of NFR. While SegWit was one of the more major updates, the years of 2017-2018 introduced multiple important implementations, as updates such as Bech32 and the Lightning Network were both released. Bech32 was introduced as a part of the SegWit soft fork, focused on efficiency and reliability, using shorter addresses to make more space in the blockchain allowing for better error detection in the case of a loss of funds, also meaning more reliable and cheaper transactions. This opened up more possibilities for the software for cases such as QR codes and more. The Lighting Network update also focused on efficiency and scalability, allowing for lower-cost transactions by moving the processing chain off of the network. This meant transactions would be faster and more efficient, as now, not every transaction had to be recorded on the blockchain, whereas before, the entire transaction was recorded and processed on the blockchain, which quickly got expensive and difficult to maintain. This also addressed scalability, as more efficient transactions allowed for more users, without the risk of congesting or crowing the network.

Other major features implementations added include Schnorr Signatures in 2019, and Taproot, a highly anticipated security-focused upgrade in 2021. These both made significant changes and improvements to Bitcoin Core's efficiency, privacy, and security. Taproot introduced a new smart contract system, allowing for more complex and secure transactions. Although there have been many upgrades and version releases in Bitcoin Core's history, it is widely considered that the more recent ones have been the most significant ones, with technology getting significantly better by the year. Taproot's implementation of the technology Merkle Tree allowed it to execute transactions off-chain, and mask the identity of the user who controlled it. It did this by enabling the ability to need multiple parties to sign off a singular transaction, increasing privacy and making it difficult to discern one transaction from another. This was a previous implementation of 2019's Schnorr Signatures, which introduced a better, alternative method of signature verification in the network. Requiring multiple signatures to be allocated into a singular one allowed for more efficient transactions, as it actively reduced the amount of needed data to be processed, clearing out the blockchain in real-time. While not all suggested features are implemented in Bitcoin Core, the goal of the system's evolution remains the same - to maintain the core principles of what Bitcoin is, continually make improvements, and implement features and upgrades proposed by the community. Overall, the goal of Bitcoin Core's implementation process is to evolve the system in a way that primes the software for the latest technology, and features, which includes ongoing goals such as the improvement of traits such as scalability, security, usability, and privacy. Eventually, the goal for Bitcoin Core is to become a completely reliable alternative to traditional financial systems, that is based on the idea of secure, decentralized transactions that is constantly evolving, being upgraded with new features and implementations with the help of the community, developers, and stakeholders.

Implications of Division of Responsibilities

Bitcoin Core serves as an open-source project, managed by a community-like group of developers and enthusiasts who as a collective, upgrade and manage Bitcoin Core in hopes that it will grow even bigger. Currently, Bitcoin Core consists of 2 software divisions, a "full-node" software used for validating the blockchain, as well as a secure wallet software. Because the software is community-based, the developer community is constantly researching, peer editing, and testing, although there are 2 main core groups: maintainers and contributors. Because the

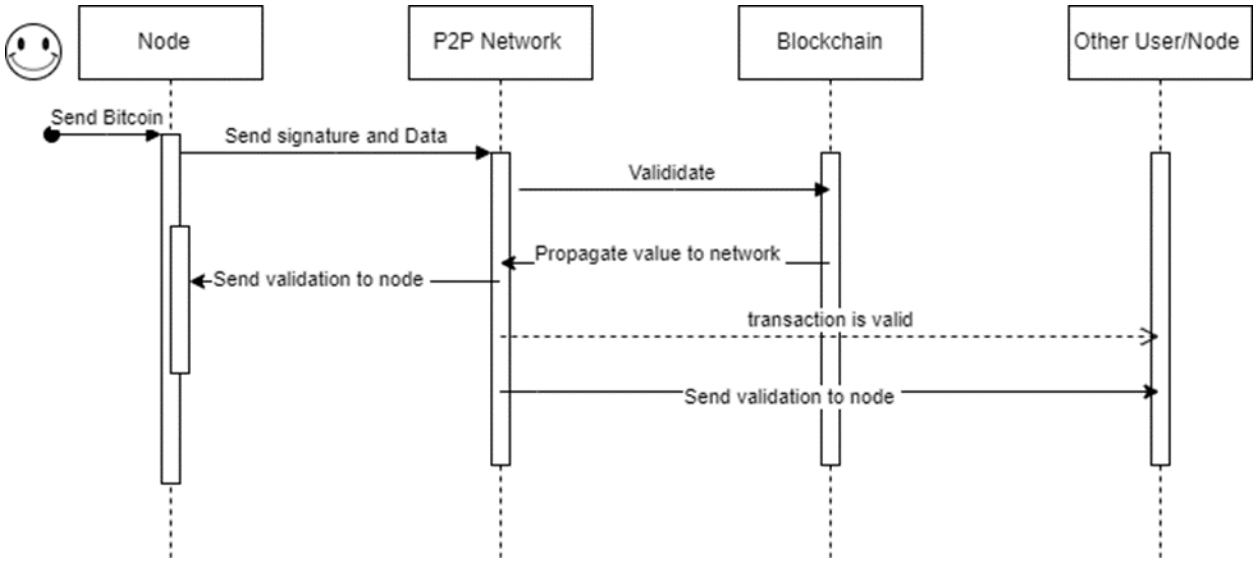
software is open-source, contributors are free to propose changes, as well as comment and make pull requests. Therefore, if you have ever made a comment, participated in any translation, or proposed a change, you are considered a contributor. A comprehensive list of contributors can be found here. While having such a large community of contributors can be healthy for the long-term growth of Bitcoin Core, it can also be a detriment, as not every change can be considered healthy and positive. That is why there remains a group of maintainers, who are tasked with implementing and merging patches and pushing pull requests from contributors. They are tasked as a final line (or a "janitorial role"), to ensure that the changes that go out are healthy for the future of Bitcoin Core, and contribute to the growth of the system. Each maintainer is responsible for a different aspect of Bitcoin Core, such as core development, wallet development, testing and quality assurance, documentation, and more. Now, as of 2023, only 5 key maintainers remain left, with the previous lead maintainer, Wladimir Jasper van der Laan, voluntarily resigning his access, citing burnout and health issues. As only the 2nd successor to Satoshi Nakamoto, he was one of the few individuals with final commit privileges and held them for more than 9 years. Van der Laan is the third maintainer to step down, within the last 2 years. The core development team now consists of Hennadii Stepanov, Micheal Ford, Andrew Chow, Marko Falke, and Gloria Zhao. Stepanov being responsible for the network's GUI, Ford responsible for the overall system build, Zhao is responsible for writing and reviewing code, as well as community outreach, Chow is responsible for wallet development, and Falke focuses on testing the system and managing changes.

Control and Data Flow

.

TRANSACTIONS

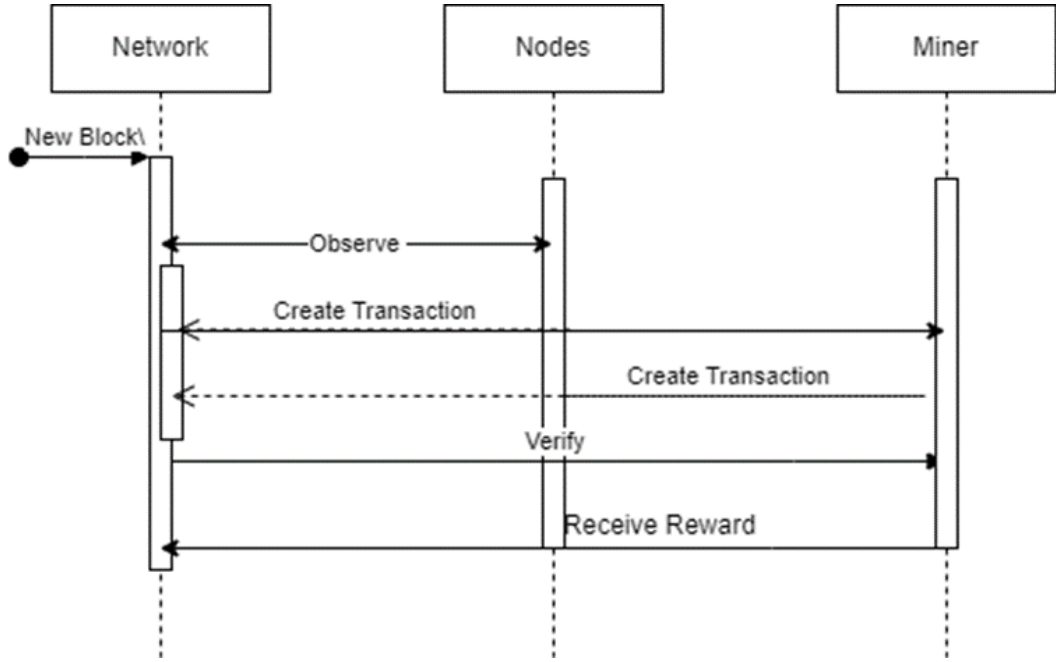
Transactions are one of the main ways bitcoin facilitates the control and flow of data. In simple terms, a bitcoin transaction tells the network that an owner of some bitcoin value authorized a transfer of that value to another bitcoin user. This means that transactions allow bitcoin users to send and receive data and control said data. Transactions are able to transfer control because each transaction contains proof of ownership by assigning a new owner address to the bitcoin value. Transactions have inputs and outputs. Inputs refer to the debit to a bitcoin account, and output refers to the credit to a bitcoin account. Input and output are what allow transactions to move value from one account to another. For example a simple payment from one address to another. The input is a reference to a previous transaction's output which shows where the value is coming from. So there will be an input instance from the address paying, and then an output that detonates a specific value to a new owner's bitcoin address.



Once the transaction is valid, it is propagated into the bitcoin network. However, for it to become part of the blockchain it will have to be verified and included in a block by a process called mining. Mining is usually associated with the bitcoin creation incentive accompanied by a creation of a block, but the most important role of mining is actually the validation and clearance of transactions. Mining validates a batch of transactions and bundles them into a block, this is how transactions become part of the blockchain.

BLOCKS

Blocks are in essence a collection of verified transactions. Copies of each transaction are hashed and paired repeatedly until a single root, the Merkle root remains. This root is stored in the block header, each block also stores the hash of the previous block's header which creates the chain. The point of chaining the blocks together is so that a transaction cannot be modified without modifying the block that records it and all the following blocks. In a data context, blocks and blockchains are basically ordered and timestamped, and the purpose of these blocks and the reason they're chained together is so that the bitcoin values are protected. Because the blocks are chained it makes it very hard to modify any previous transactions, so blockchains are simply a way to record any bitcoin transactions and add security to them.



Here is a simple sequence diagram of how bitcoin value transactions flow.

P2P NETWORK

Briefly insinuated in the transaction section, the bitcoin network is a peer-to-peer network. This means that all computers that participate in the system are peers and have no specific hierarchy. In the bitcoin network context, we have nodes of equal level that communicate with other notes to be able to send and receive information about blocks, and transactions. Although these nodes are of the same level, they could have different functionalities. A network node could hold up to 4 functionalities, wallet, miner, full blockchain database, and network routing. The P2P network is integral for the transference of data for transactions and blocks.

WALLETS

One of the functions that a network node can have is a wallet, which is a software component that enables users to manage their Bitcoin balances and transactions. When a user creates a Bitcoin address, the wallet generates a public key and a private key. The public key is used to receive Bitcoin, while the private key is used to sign transactions and spend the Bitcoin associated with the address. The wallet in Bitcoin Core stores these private keys and allows the user to manage their Bitcoin transactions. It keeps track of the balance of each address and enables the user to send Bitcoin to other addresses. Since the Bitcoin network is peer-to-peer, each node can have a wallet functionality to manage their Bitcoin balances and transactions. This allows users to control their own funds without relying on a centralized authority. This wallet system is essentially an easier way for users to parse transactions, allowing users to spend and receive satoshis and the data is then transferred to the blockchain.

Concurrency

Concurrency exists in multiple areas of Bitcoin Core. To begin, each node in the peer-to-peer network collects new transactions into a block, and new transactions are broadcast to all other nodes in the network. Since every node in the network has the newest transaction data, not only the validation process for different transactions can happen concurrently, but different nodes in the network can compete to validate blocks, and the first node to validate a block that is accepted by the network is rewarded via a process called mining. This newly validated block is finally added to the existing blockchain and is public to all nodes in the network. Furthermore, since the transaction validation is concurrent, the process of sending payments from one address is also concurrentFurthermore, there is a process called pool mining in which different nodes in the network combine their computational power in order to complete the proof of work faster than if the validation were to be done by a single node in the network. Upon successfully validating the block, the reward for the mining is split between the nodes who solved the hash, according to the amount of work that was processed by each node. Concurrent processes are crucial for a P2P payment processing architecture such as Bitcoin to function effectively. The payment and validation processes are constantly executing concurrently, and nodes in the network can even collaborate to achieve their goals faster.

Conclusion

Bitcoin core is a peer-to-peer software architecture that runs concurrency and provides people a means to send deregulated currency in the form of Bitcoins. It is a modular architecture where information is kept hidden from other parts. It can be broken down into the following: wallets, the means by which people manage their money; nodes, which are used to share information about transactions and finally transaction/block validation, which checks that transactions and blocks are valid and comply with the protocol. The software had evolved multiple times to improve its scalability, efficiency, and security with implements such as Bech2, the Lightning Network, and Taproot. The architecture flows in a clear pipeline of data where new blocks are mined by individual users and validated by the blockchain. The same is done for transactions where the information is sent over the network processes and then validated. In conclusion, Bitcoin Core is a unique software with an interesting and in-depth architecture.

Lessons Learned

After the completion of this report, the team has learned many lessons, as well as faced many problems and challenges. We've all gained a greater insight into the inner workings of cryptocurrency, and the blockchain and an in-depth look at how peer-to-peer software architecture works. We learned about the importance of consistency throughout an academic paper, as well as consistent resources. Additionally, we quickly learned how important good time management is, as well as a healthy distribution of work (something we could learn from the maintainers of Bitcoin Core). Healthy teamwork was key in the process of making this report, as we found that it was easy to get overwhelmed, especially when tackling so many different parts

at once. Breaking down the report into parts for each team member to do, made a large daunting report manageable for everyone.

Data Dictionary

Architecture: The organization of a system, its subsystems, and their behavior in software.

Blockchain: A record of transactions maintained as data across computers linked in a peer-to-peer network.

Block: A set of transactions on Bitcoin from a certain time period

Lightning Network: Bitcoin blockchain's second layer that allows off-chain transactions between parties not on the network.

Node: A point serving as an intersection or connection within a data communication network.

Schnorr Signature: Unique digital signatures used to move bitcoin on the blockchain

Segregated Witness: An implemented soft fork change in Bitcoin's transaction format involving the removal of witness information

Taproot: A Bitcoin upgrade making transactions easier and faster to verify on its network by batching multiple signatures and transactions together

Merkle Tree: Also known as binary has trees, they are a type of data structure technology that allows for efficient, secure encryption of blockchain data

Decentralization: Giving the responsibility and authority to a public collective, rather than giving Government powers full authority

Naming Conventions

Central Processing Unit (CPU): electronic circuitry executing instructions from programs that allows a computer or other devices to perform its tasks.

Graphical User Interface (GUI): a form of a user interface allowing users to interact with the system through the display of graphical icons along with audio indicators.

Non-Functional Requirement (NFR): specifies the quality attributes of a system that attempt to improve its functionality.

Peer-to-Peer Network (P2P): A decentralized communications model where either party initiates a communication session, with both parties having the same capabilities.

References

Antonopoulos, A. (n.d.). *Mastering Bitcoin*. Introduction · GitBook. Retrieved February 18, 2023, from https://cypherpunks-core.github.io/bitcoinbook/

Bitcoin. (n.d.). *Bitcoin Core Source Code Repository*. GitHub. Retrieved February 18, 2023, from https://github.com/bitcoin/bitcoin

CoinMarketCap. (2021, December 15). *Who are Bitcoin Core's developers?: CoinMarketCap*. CoinMarketCap Alexandria. Retrieved February 19, 2023, from https://coinmarketcap.com/alexandria/article/who-are-bitcoin-cores-developers

Developer Guides. Bitcoin. (n.d.). Retrieved February 18, 2023, from https://developer.bitcoin.org/devguide/index.html#

Nakamoto, S. (n.d.). *Bitcoin: A Peer-to-Peer Electronic Cash System*. Bitcoin. Retrieved February 18, 2023, from https://bitcoin.org/bitcoin.pdf%20

Investopedia. (n.d.). Investopedia. Retrieved February 19, 2023, from https://www.investopedia.com/

Mutunkei, J. (2023, February 17). *Bitcoin Core has only 5 developers left as key maintainer departs*. crypto.news. Retrieved February 19, 2023, from https://crypto.news/bitcoin-core-has-only-5-developers-left-as-key-maintainer-departs/

https://www.youtube.com/watch?v=iQ33cjeC6mo