CISC 322/326

Assignment 1: Report

Conceptual Architecture of Bitcoin Core

Saturday February 18, 2023

Group: All Around Average

Jonathan Sumabat *20js30@queensu.ca*

Lukas Boelling *lukas.boelling@queensu.ca*

Nour Mahmoud *17nhm1@queensu.ca*

Caleb Chiu *20cyjc@queensu.ca*

David Kropinski *19dsk3@queensu.ca*

Jeff Jiachuan Li *19jjl6@queensu.ca*

Table of Contents

Abstract:
The purpose of this paper is to analyze the Bitcoin Core software architecture. This paper will detail the overall style, functions, subsystems, data flow, and concurrency of the Bitcoin system. The whole system can be broken down into Peer Discovery, Connection Manager, Transactions, Blocks, Memory Pool, Miner, Validation Engine, Storage Engine, Wallet, and RPC. The paper provides a thorough analysis of each module's functionality, their dependencies, and how they function when integrated. Through our analysis of multiple open source documentations, it was determined that Bitcoin Core is a Peer-to-Peer network with a repository style architecture used in the design of the Blockchain system.

The report contains use cases and sequence diagrams to demonstrate how the Bitcoin Core architecture is applied in practical scenarios. One essential use case is processing a transaction or transfering bitcoins from one person to another.

Understanding Bitcoin's architecture can provide insights on how a decentralized payment system could be designed and operated. It also helps us understand the advantages and disadvantages such a system has from an architecture perspective. Overall, this paper defines the conceptual architecture of Bitcoin Core, and it will serve as a reference model for Bitcoin's concrete architecture in the future.

Introduction and Overview:

For centuries humans have traded with one another with the intention of striking a deal that would benefit them. This may be regarded as the core essence of business which has been practiced throughout history. The finance industry has revolutionized since the 21st century where the unique idea of decentralized banking was introduced. This fought against the traditional system where instead of a single entity being in control of all financial transactions, it would be maintained by a network of users making it accessible to anyone with an internet connection.

Bitcoin Core was initiated in 2009 by Saktoshi Nakamoto, a pseudonym used to represent the creators, with the intention of creating a peer to peer digital cash system that processes transactions without the need of a centralized entity or authority such as a bank. The purpose of this was to create a new type of currency that would be secure, transparent and safe from government control or manipulation which rapidly grew in popularity in the coming decade. This system is an open source project which is made publicly available for anyone to view, modify or use. The software is maintained by a group of volunteer developers who work on developing new updates and bug fixes, however, anyone is welcome to contribute.

Since the first release back in 2009, Bitcoin core has gone through numerous updates enhancing functionality, security and scalability of the software. The Bitcoin Core software provides a complete and comprehensive interpretation of the Bitcoin network and its protocols, including full validations of transactions and blocks, as well as storing a copy of the entire blockchain, which is the record of all validated transactions in the network. As a full node, Bitcoin Core helps to secure the network by

validating deals and blocks and relaying them to other nodes. By running a full node, individuals can contribute to the overall health and decentralization of the Bitcoin network, as well as providing a way to access their own transactions and the network without relying on third- party services. Since its release, Bitcoin Core has continued to evolve and improve, with new features being added regularly to help maintain itself as the primary structure for the Bitcoin network.

This report encapsulates the breakdown and analysis of the architecture style of the Bitcoin core system, using information accumulated from course teachings, in addition to external accredited sources. The breakdown of the architectural analysis is split into multiple sub sections: subsystem breakdown & interaction, Diagrams/Use-cases, evolution, control and flow of data, concurrency, and conclusions. The final section of the report, lessons learned, focuses on any challenges our team may have faced, how we overcame these adversities, and finally an overview of our findings and experiences.

Subsystem Breakdown & Interaction:

The Connection Manager module interacts with known peers to handle requests to/from them. When transactions and blocks are broadcasted from peers, the Connection Manager places them into lists/queues so they can be checked by the Validation Engine.

The Peer Discovery module is responsible for keeping the database of known peers updated.

The Wallet module creates and manages private/public key pairs, which it uses to receive and send transactions. The module can also access the wallet's current balance by checking the Storage Engine for transactions in the blockchain which are directed towards the keys associated with the wallet.

The Storage Engine module stores a local copy of the blockchain. Validated blocks from the Validation Engine are added to the chain here.

The Validation Engine verifies blocks and transactions against the current state of the blockchain which is stored in the Storage Engine.

The RPC module provides an API for external programs to interface with bitcoin core. It exposes methods that get information on the state of the blockchain, mine new blocks, interact with the P2P network, and work with transactions. To achieve this, the RPC module communicates with the Storage Engine, Connection Manager, and Wallet modules.

The Mempool stores transactions that the node has verified as valid, but have not appeared in a block. When the Mempool receives a new broadcasted transaction from the Connection Manager, it first uses the Validation Engine to verify the transaction is valid (i. e. the sender has sufficient funds to complete the transaction, and that the transaction follows the consensus rules of the network).

The Miner module runs calculations to construct proof-of-work to create new blocks containing valid transactions from the Mempool module. Once the miner creates a new block, it adds it to its local blockchain and broadcasts it to the rest of the P2P network.

Diagrams:

Bitcoin Core architecture (Source: Eric Lombrozo) shows the architecture of Bitcoin Core.

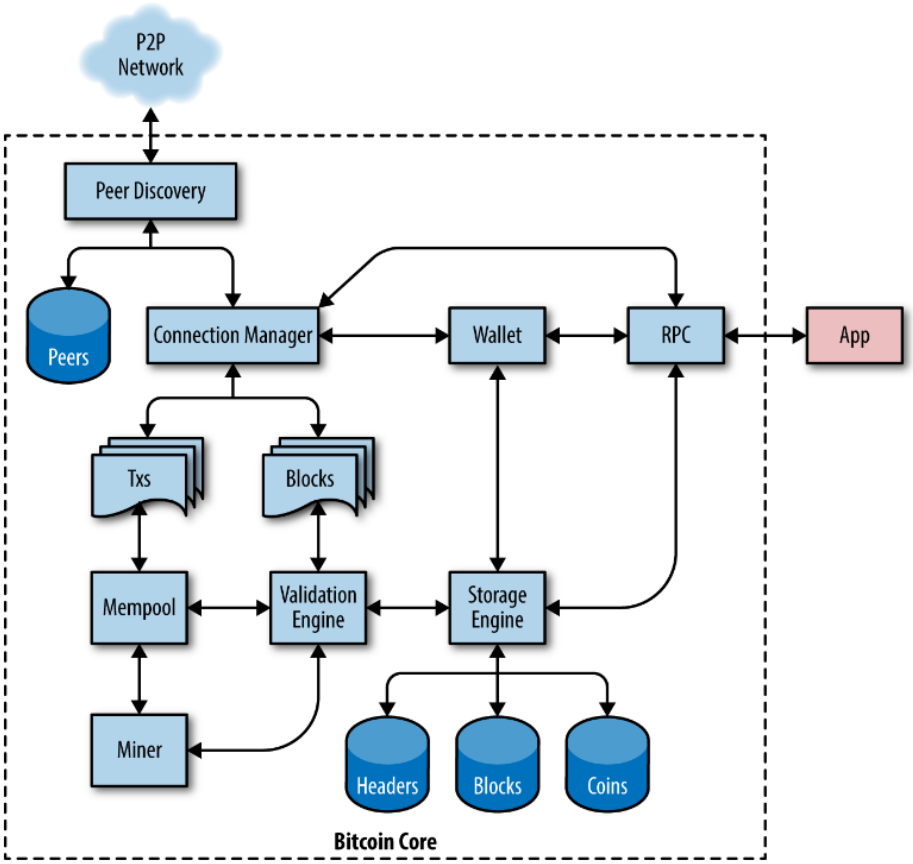

Figure 1. Bitcoin Core architecture (Source: Eric Lombrozo)

Figure 1: Bitcoin Core architecture

External Interfaces:

Use case 1: Initiating a transaction

In this sequence diagram, the sender initiates a transaction by using the sender wallet to enter the amount of bitcoin and the receiver's address. The sender wallet generates a new transaction that has the bitcoin amount and the receiver's address. It then identifies the sender's private keys to check if he has the appropriate unspent transaction outputs that represent the input amount. If the sender has enough unspent transaction outputs, the sender wallet requests a digital signature (i.e private keys) from the sender to prove ownership of the bitcoins being sent. The signed transaction is then

broadcasted to the Bitcoin network in order to check the validity of the transaction. If the sender does not have the appropriate amount, the sender wallet displays a transaction error.

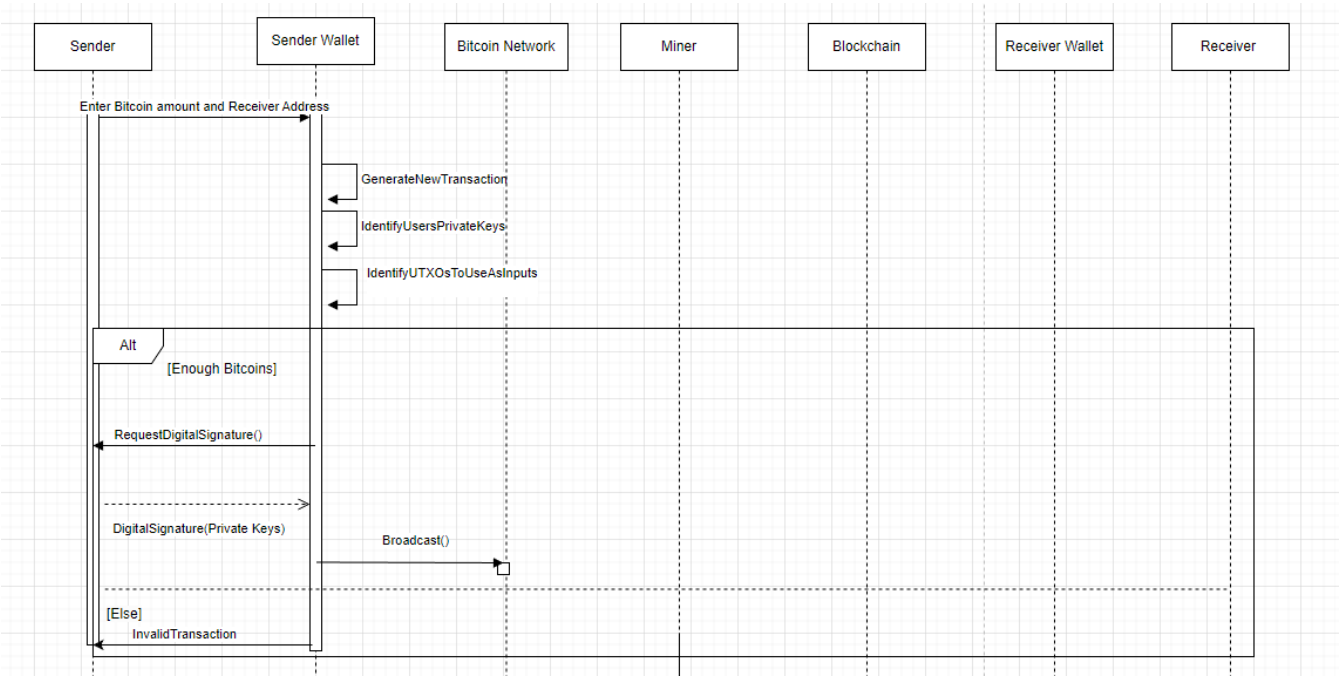

Figure 2: Sequence Diagram for 1st Use Case

Use case 2: Processing a Transaction

This graph shows the interactions between components involved in processing a Bitcoin transaction. The sender initiates a transaction by using the sender wallet to enter the amount of bitcoin and the receiver's address. The sender wallet requests a digital signature from the sender to prove ownership of the bitcoins being sent. The signed transaction is then broadcasted to the Bitcoin network in order to check the validity of the transaction. If the transaction is deemed valid by the network, it's sent to the miner to add it to a block of unconfirmed transactions after using the Proof of Work (PoW) algorithm to solve a mathematical puzzle. The block is then broadcasted to the Bitcoin network in order to check it against the consensus rules of the network and then it gets added to the Blockchain. The receiver's wallet checks the Blockchain to confirm that a transaction has been processed and updates the

receiver's wallet balance to reflect the new amount.

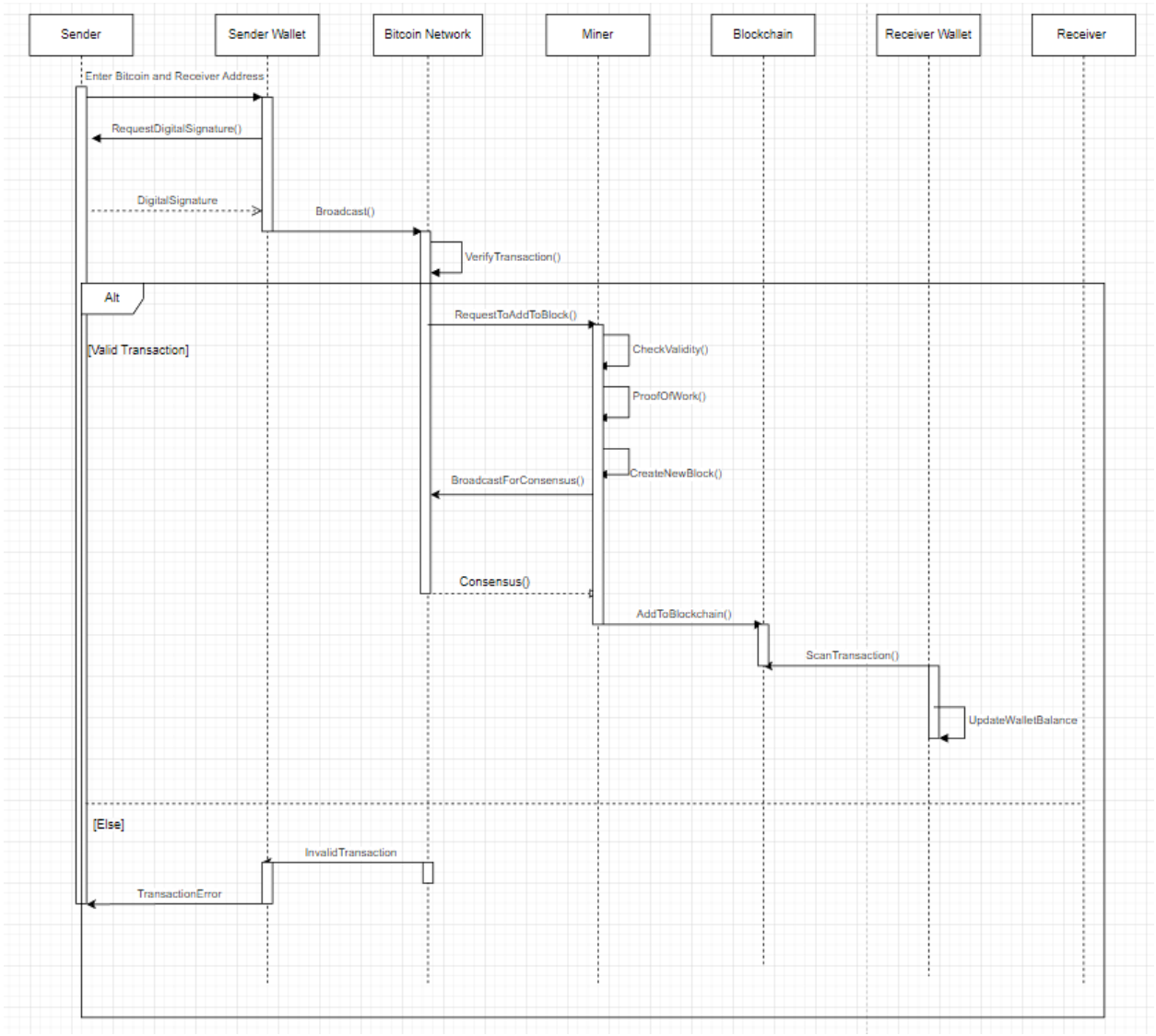

Figure 3: Sequence Diagram for 2nd Use Case

Evolution:

Bitcoin core has undergone numerous updates since its initial release back in late 2008 whose creator(s), Satoshi Nakamoto, served as the original maintainer until late 2010. A series of other individuals would take over this role after his disappearance. The role of the maintainer since has been handed down on a voluntary basis, relying heavily on the trust of the community. Bitcoin Core is the reference implementation of the Bitcoin protocol, and a software widely used for interacting with the Bitcoin network. In this section, we will examine the evolution of Bitcoin Core over the years, from its initial release to the present day version.

The initial release of Bitcoin core in 2009, version 0.1, provided a single command-line interface for sending and receiving bitcoin transactions. The system was designed to be simple and easy to use but lacked many functional and security features which are now available in modern day versions. It was the third bitcoin client developed in the world after Satoshis “Bitcoin” wallet, an open-source system and Hal Finney’s wallet.

Bitcoin core would face several crucial updates in the following years which improved functionality and security. Version 0.3 would be released in 2010 introducing a graphical user interface (GUI) and introduced multi-threading which enhanced system performance. The single command-line interface could prove to be confusing and difficult for users who were not familiar with it, in addition to restricting the amount of information that can be displayed. With the addition of the GUI, the visual interface made it easier to navigate and interact with Bitcoin core. Another major component of this update was the implementation of multithreading, enabling multiple task processes at a time. With the implementation of multithreading, Bitcoin Core was able to perform multiple tasks simultaneously, which increased its efficiency and reduced time required to complete tasks. Bitcoin Core would be able to verify multiple transactions or blocks simultaneously, meaning the software can process a larger volume of transactions in a shorter amount of time. Multithreading would also improve the performance of other tasks within the system, such as syncing with the blockchain, downloading and verifying blocks, and broadcasting transactions to the network. By improving the performance of these tasks, multithreading has made Bitcoin Core more reliable, efficient, and user-friendly.

Version 0.8 would release on 2013 and introduced multisignature transactions which required multiple signatures in order for a transaction to proceed. Block verification speed would also be improved as a new synchronization method would be introduced, this made it significantly easier for users to get up and running on the bitcoin network. Shortly after update 0.15 would introduce block chain pruning, which helped manage the size of the blockchain by removing old or irrelevant data. Bitcoin core would store a copy of the entire history of transactions that occurred on the network. As more transactions were made, the size of the database would increase making it difficult for nodes with limited storage to keep up with the latest data being processed. Pruning enables the system to old or irrelevant data from the database to free up storage without negatively affecting the integrity of the blockchain since the pruned data is not required for the current state of the network.

Version 0.12 of bitcoin core would show some improvements in the security sector with a new validating system being introduced. This improved security by making it more difficult to create fake transactions by introducing a new signature verification "SIGHASH_FORKID”. This was added as a new bitcoin protocol which provides additional support against transactions malleability, which can occur when an attacker modifies a transaction such that the transaction ID is changed but still remains valid. This signature verification tackles this issue by introducing a new transaction algorithm that includes a unique fork ID making it much more difficult for attackers to modify the transaction ID without invalidating the signature. The future update 0.13 would continue to show upgrades in bitcoin core’s security by introducing support for the SegWit fork (segregated witness), improving

efficiency and security for the network. This protocol focuses on how transaction data is stored and restructures it. SegWit sifts through data and separates the digital signature data from the transaction data and stores them accordingly. By removing the signature data, which is oftentimes the largest part of the transaction, from the transaction data, allows more transactions to be included in each block increasing the overall capacity of the bitcoin network. Segwit also helps in dealing with transaction malleability by separating signature and transaction data, as well as improving network security by implementing new technologies such as the lighting network, allowing for faster and cheaper transactions. Overall with updates 0.12 and 0.13 there have been significant improvements in security, functionality and scalability of the network.

One of the most recent major updates however would be version 0.21, released in 2021 which added support for "taproot", a bitcoin protocol that improves privacy and functionality. Taproot is a protocol that introduced a new type of signature called Schnorr signatures, an alternative to ECDSA signatures used in bitcoin. One key benefit that taproot offers is it allows for the creation of complex multi signature transactions that appear as standard transactions which makes it difficult for outsiders to observe and determine the nature of the transactions such as who is involved. This is a significant security improvement as it makes it harder for third parties to track and identify individual users. Another key feature to note is the introduction of maternodes, enabling more complex and smart contract functionality. This works by allowing bitcoin users to create a single public key that represents a set of complex conditions which can then be used to create more complex transactions which require multiple signatures or conditions to be met. This not only expands the range of smart contracts that can be created on the bitcoin network but opens up new possibilities for decentralized applications.

Over the years bitcoin core has seen many new improvements that increase security, performance and functionality. With a strong community behind this system, more updates and versions are introduced every year with the goal of improving user experiences. There are many versions of bitcoin core since its release, however, the versions comprised above mark some of the most significant and major evolutions of bitcoin core up to date, with many more that are sure to come.

Control and Flow of Data:

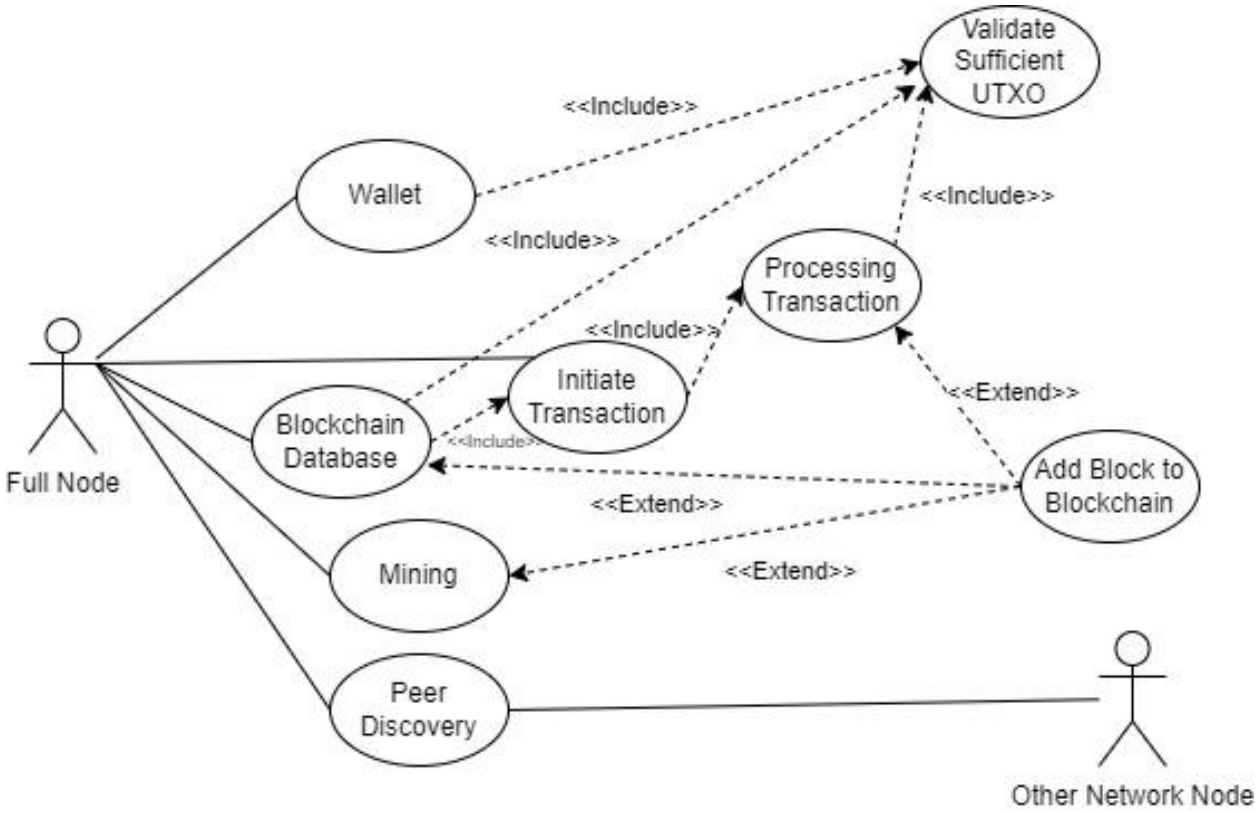

Figure 4: Use Case Diagram of Data Flow

There are four main functions that a Bitcoin Core node can have; wallet, mining, network routing, and full blockchain database. All of these functions work together to ensure the most fundamental system of Bitcoin works, which are transactions.

Wallets can create private and public keys, distribute public keys, get outputs spent to those public keys, create, sign and transmit transactions. When a user checks their wallet, the bitcoin "balance" that is displayed is the sum of all UTXO that the user's wallet can spend and which are usually spread across hundreds of transactions and blocks on the blockchain.The mining function is used to validate new transactions in order to add new blocks to the block chain and receive a reward. The network routing function is used to maintain the peer to peer network and allow the transmission of new information such as transactions and block validations. When new nodes are created they must first discover other bitcoin nodes on the network to participate. This process is kickstarted when a new node discovers an already existing node on the network and connects to it. Some nodes may be just connected to a single server and that server is connected to the larger network. This can happen in mining pools where many mining nodes work for a central mining pool server. The full blockchain database is used to store a copy of the current valid blockchain. This allows that node to verify transactions without external help. All nodes need some network routing functionality in order to be a part of the system and may have some or all of the other functions. Some common node types include Bitcoin core client, full node client, SPV wallet, edge routers/blockchain nodes, solo miners, pool protocol servers, and pool

miners. The Bitcoin core client has all four functions. The full node client does not have mining but has the other three. The SPV wallet just has the wallet and networking functions. The SPV wallet uses validated blocks from other nodes with a blockchain database in order to verify transactions. Edge routers just have the blockchain database and networking. These are mostly used as connection points and for wallets to validate transactions off of. Solo miners have all but the wallet function. Pool protocol servers act as gateways between the larger network and their pool miners which are not connected to the network. Pool miners like solo miners do not have wallet functionalities.

Concurrency:

Concurrency occurs constantly in the Bitcoin network as it is made up of many independent nodes performing its own tasks. One such task is the validation of new transactions or "mining". Mining nodes will group new transactions into blocks and try to solve a cryptographic hash for the new block and add it to the chain of previously validated blocks. Since many nodes are doing this at the same time there can be multiple blocks that could become the next accepted block to add to the chain. In this instance the most agreed upon block among other nodes will be accepted. The node that first made the new block will receive a small amount of currency as a reward. There is a method of collaboration called pool mining which allows multiple nodes to merge their resources. This works by having all the independent nodes working on the new block and if an answer is found the reward is shared among the pool. For most pools the reward is split based on who did the most work. The purpose of doing this is to increase the consistency of a single node getting some reward. Working for a mining pool allows a node to get a reward more often without increasing its own computational power, though the reward is less due to sharing.

Incorporating concurrent processes within the Bitcoin network ensures that the system functions fluidly and correctly. By concurrently processing data of various transactions and minings the blockchain will be able to stay up to date with the most secure and validated blocks of information. Making sure each node is up to date with the correct data ensures the proper functionality of Bitcoin's peer-to-peer network.

Conclusions:

Architecture design plays an important role in the Bitcoin Core system. The decentralization of control is a core design principle of the system that is maintained by the decentralized Bitcoin Network Architecture. The Bitcoin system is structured as a peer-to-peer network where nodes interact with each other without the need of a centralized server or intermediaries such as banks or payment processors. The system consists of several modules or subsystems, including Peer Discovery, Connection Manager, Transactions, Blocks, Memory Pool, Miner, Validation Engine, Storage Engine, Wallet, and RPC. The subsystems interact with each other to communicate with peer-to-peer networks, process and validate transactions, and maintain consensus by adding newly created blocks to the Blockchain.

The Blockchain architecture is more of a repository-style architecture. The Blockchain represents the central data storage and each node maintains a local copy of it and is kept in sync through the consensus protocol that ensures that all nodes agree on the current state of the Blockchain. The repository based approach used in the Blockchain system provides security benefits as the Blockchain contains the ledger of all transactions that have occurred in the network, and this ledger is replicated across all nodes in the network. If one node attempts to introduce fraudulent transactions into the ledger, the other nodes would quickly detect the inconsistency in the modified copy.

Although the system is highly secure, there are still issues with security that need to be addressed such as the 51% attack problem. In such a problem, an attacker who gains control of a miner node that controls more than 50% of the network's computing power has the ability to alter the blockchain. When solving the cryptographic puzzle of the network, the attacker can introduce new fraudulent transactions before broadcasting the block to the network to become part of the new Blockchain.

Lessons Learned:

During the collaboration of this assignment, we ran into difficulties and limitations regarding the architecture of Bitcoin Core, our understanding of how blockchain's work, and organization. A large portion of us spent a large amount of time on comprehending how blockchains work, many of us inexperienced on the topic. This made it far more difficult to break down the architecture of Bitcoin Core. Our organization also suffered due to the amount of tests and assignments during the duration of this report.

Despite the challenges we face, our team learned a lot from working together on this project. Our main solution to the problems we faced were scheduling daily meetings at a consistent time while essentially creating a checklist of tasks we needed to complete by a certain date, similar to an agile sprint. This not only made it possible for us to divide the tasks and schedule our time effectively, but allowed us to moderate our progress and make sure that everyone was on the same page. As a team, we learned how blockchain works and how Bitcoin Core Architecture works as a series of subsystems combined together. Moving forward, we intend to keep everyone on track with the understanding of the architecture, and continue to do consistent daily meetings as it was an effective strategy to overcome our concerns.

References:

https://bitcoin.org/bitcoin.pdf
https://developer.bitcoin.org/devguide/index.html#
https://cypherpunks-core.github.io/bitcoinbook/ch03.html
https://river.com/learn/what-is-bitcoin-core/
https://www.youtube.com/watch?v=L_sI_tXmy2U&t=1s
https://www.investopedia.com/terms/1/51-attack.asp#:~:text=for%20financial%20brands.-,What%20Is%20a%2051%25%20Attack%3F,power%20to%20alter%20the%20blockchain.