

Video URL: https://youtu.be/_3lDfUV-NB4

Conceptual Architecture of Bitcoin Core

By: The Foobar Fighters

Roles and Responsibilities

LEADER: Makayla Mcmullin → System & Functionality, How system is broken into interacting parts, External Interfaces

PRESENTER: Daniel Dickson → Slides, Division of Responsibilities, Abstract & Conclusion, Video Presentation Editing

PRESENTER: Aniket Mukherjee → Slides, Control & Data Flow

Maia Domingues → Slides (Concurrency), Intro, Concurrency

Lucas Patoine → Evolution of System, Sequence Diagrams

EVERYONE: Naming Conventions, Data Dictionary, Limitations & Lessons Learned



What is Bitcoin

- Bitcoin is a decentralized cryptocurrency that operates on a trustless basis, unaffected by governments/corporations who would look to control currencies for their own benefits
- Bitcoin operates on a peer to peer architecture to maximize security and anonymity for everyone using the currency
- Due to this enhanced security, the currency can't be manipulated like other previously established currencies (like the Venezuelan bolivar)



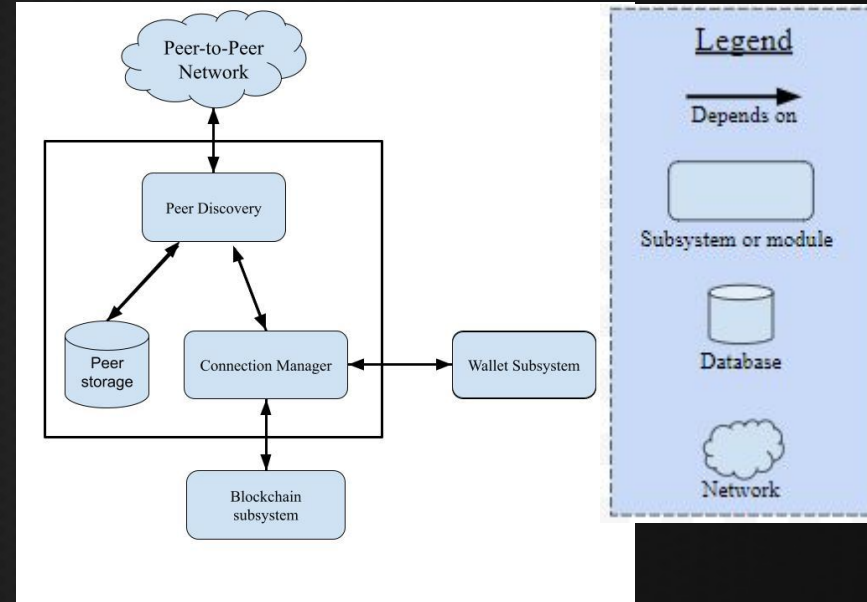
Bitcoin Core Architecture

- Bitcoin Core uses an object-oriented architectural style that acts as a client and a server in the bitcoin peer-to-peer network architecture style.
- The modularity of the system's architectural style supports future changes in the system because changes to one object in the system do not affect the others.
- There are many subsystems at work within Bitcoin Core, such as the network subsystem, the wallet system and blockchain subsystem.



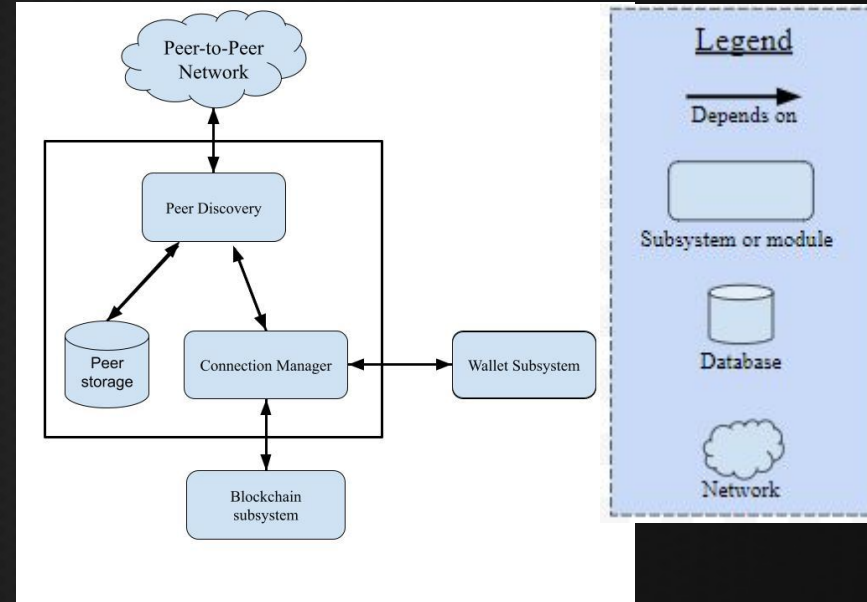
Network Subsystem

- Manages the communication between the Bitcoin Network and Bitcoin Core software system.
- Sets up the initial connection to the peer-to-peer network and maintains its connection through the exchange of peer, blockchain and transaction data.



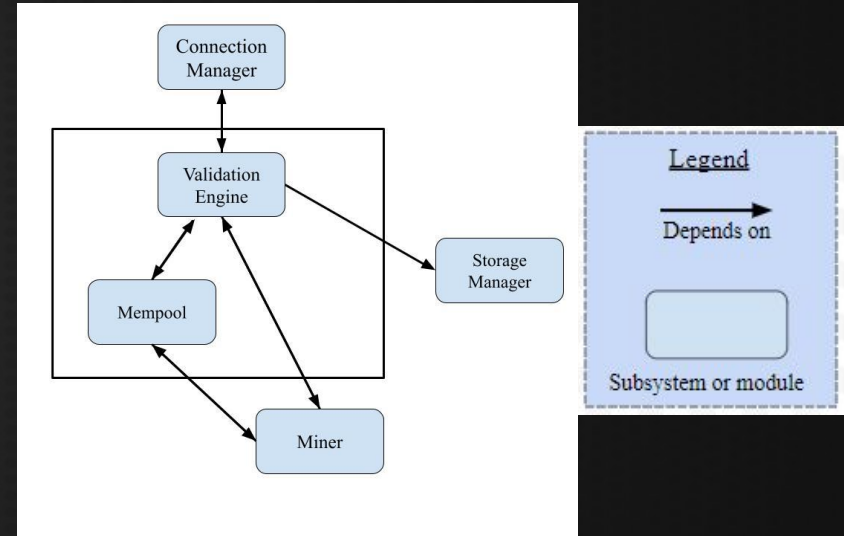
Network Subsystem

- A query will be transmitted to a “DNS seeds” server which contains a list of IP addresses belonging to longstanding Bitcoin nodes.
- Once Peer Discovery finds an established Bitcoin node, the two hosts will set up a TCP connection and the Bitcoin peer will send information about itself as well as a copy of the blockchain and a list of its neighbour peers.



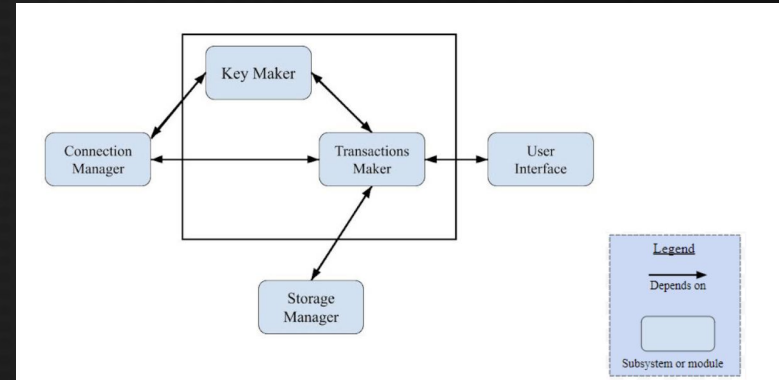
Blockchain Subsystem

- Responsible for maintaining the system's local copy of the blockchain as well as contributing to the network blockchain.
- The subsystem accomplishes this by interacting with the Connection Manager in the Network Subsystem, the Storage Manager, and a Miner module.



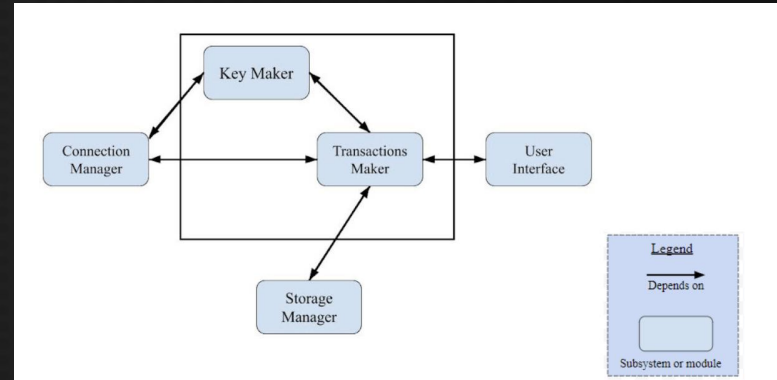
Wallet Subsystem

- Responsible for making & distributing private and public keys, updating storage system, and sent transactions to network
- Key Maker creates private keys and determines if keys are from the same root “seed”
- Only seed needs to be copied if user switches wallets
- Encrypts private keys to public keys, hashing derives a Bitcoin Address
- To transaction maker to send Bitcoin or user interface to request Bitcoin
- Always sent to Connection Manager to store incoming transactions



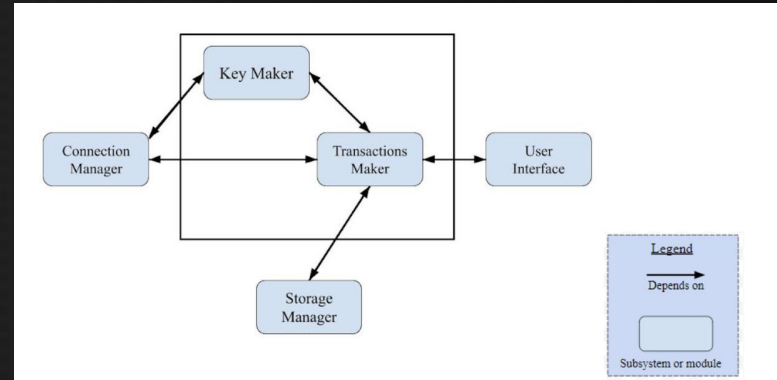
Wallet Subsystem

- Transaction Maker works with other systems to compile and send transactions to be propagated and received into Bitcoin Network
- Message containing transaction info sent from interface to Transaction Maker, which sends to Storage Manager
- Storage Manager picks Unspent Transaction Outputs equal to or greater than the transaction amount; error if not possible, if possible send UTXO to Transaction Maker



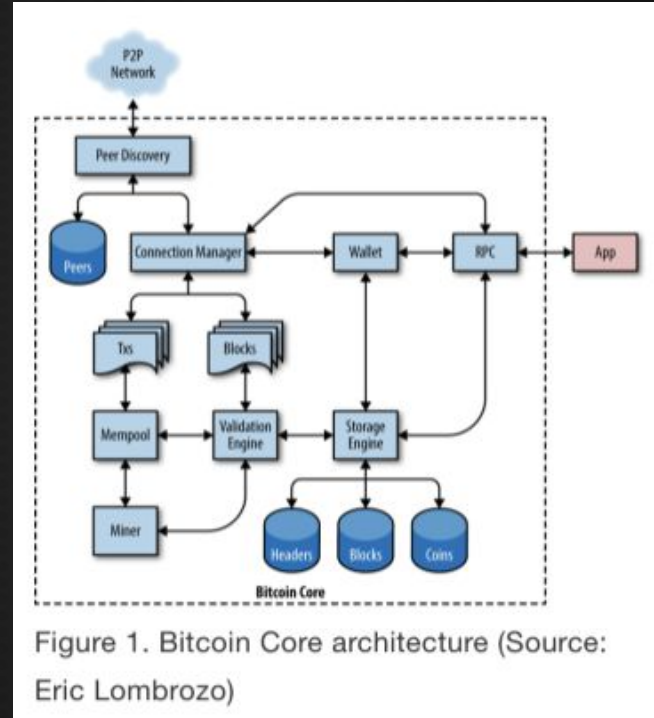
Wallet Subsystem

- Requests Key Maker to make a new key in the previously mentioned process; get user confirmation
- Sent to Peer Discovery and Bitcoin Network, broadcast by nodes and arrives at recipient's node/wallet
- Receiving: Key Maker generates new key, send address to interface, make new transaction and sent to network to be received by user



Control and Data Flow

- Once the app is started up, it initializes both the user's wallet and storage engine through remote procedure calls.
- The storage engine stores both the coins in the wallet, as well as a copy of the verified blockchain, to append to when it gets updated.
- The validation engine is what performs the verification of the mempool that miners operate on, as well as the network version of the blockchain.



Evolution of System

- The initial framework for what would eventually become Bitcoin Core, along with the specifications of the peer-to-peer Bitcoin network itself was outlined in the whitepaper for Bitcoin, created by Satoshi Nakamoto: a pseudonym used by Bitcoin's creator
 - September 1st, 2009 → Satoshi Nakamoto uploads version 0.1 of Bitcoin.
 - Later called Bitcoin-Qt when QT GUI implemented
 - At the time, Bitcoin was only available for Windows operating systems
 - Would remain that way until later that year when version 0.2 was released, adding support for Linux along with allowing for the use of multi-core processors in mining



Evolution of System

- v0.3.9 → Nakamoto leaves project & disappears?
- v0.4.0 → optional wallet encryption & private keys
- v0.5.0 → QT name official, many more GUI improvements, wallet encryption, RPC commands
- v0.6.0 → backup wallets, QR Code address sharing
- v0.7.0 → getmemory, getlocktemplate, submitblock
- v0.8.0 → redesigned to use LevelDB, decreased blockchain synchronization time
- v0.9.0 → BitcoinCore!



Concurrency

- Bitcoin Core employs a number of methods to handle concurrent transactions to ensure security and reliability of the Bitcoin blockchain:
- To allow concurrent processing of transactions, Bitcoin Core utilizes a memory pool, which is accessible by all nodes in the network, to store transactions before they are confirmed
 - Multiple blocks can be generated concurrently and enter the memory pool before confirmation
 - Transactions must be properly signed and require a number of confirmations before being fully confirmed
 - If a transaction is not confirmed within a certain amount of time, it is taken out of the memory pool
- Replication of the Bitcoin blockchain across multiple nodes on the network ensures that there is no single point of failure



Concurrency


- Any changes or alterations to the Bitcoin blockchain requires a majority consensus to be reached amongst the network nodes before changes can take place, further reducing the chances of errors occurring due to concurrent transaction processing
- Priority Queuing: Transactions with a higher priority are processed sooner than lower priority transactions
 - Transaction fees: Applied by a Bitcoin miner to a transaction. The higher the fee, the more likely the transaction will be included in the next block added to the blockchain
 - Factors such as transaction age and size can also affect the position of a transaction in the queue



Divisions of Responsibilities


- GitHub used facilitate collaboration
- Separate GitHub repos used for separate parts of projects
- Large team with differing roles and responsibilities
 - 61.7% of contributors only contributed to one single repository
- Version control used to keep track of changes made and released
- Archiving ensures older architecture may still be referenced if need be

	gal	hw01	hw02-gal-gui										hw03-brown-brown-misc-assets-models										gala_101a	packaging-brown-of-brown-10-cr-10-2-uniform-gal-gui-brown-of-dots										apple-sold-cases										hw04-std																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																														
frankauk	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT	CONT



[Product](#)
[Solutions](#)
[Open Source](#)
[Pricing](#)

[Sign in](#)
[Sign up](#)



Bitcoin Core

[11,295 followers](#)
<https://bitcoincore.org>
[@bitcoincoreorg](#)




[Overview](#)
[Repositories 25](#)
[Projects](#)
[Packages](#)
[People 15](#)

Popular repositories

[secp256k1](#)

Public




Customized C library for EC operations on curve secp256k1


 1,165
  905

[bitcoincore.org](#)

Public

Bitcoin Core project website


 489
  564

Repositories

Type




Language

Sort

[gui](#)

Public




Bitcoin Core GUI starging repository


 410
  248

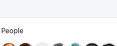
[bitobd](#)

Public






Bitcoin Script Debugger


 276
  158

People



Top languages

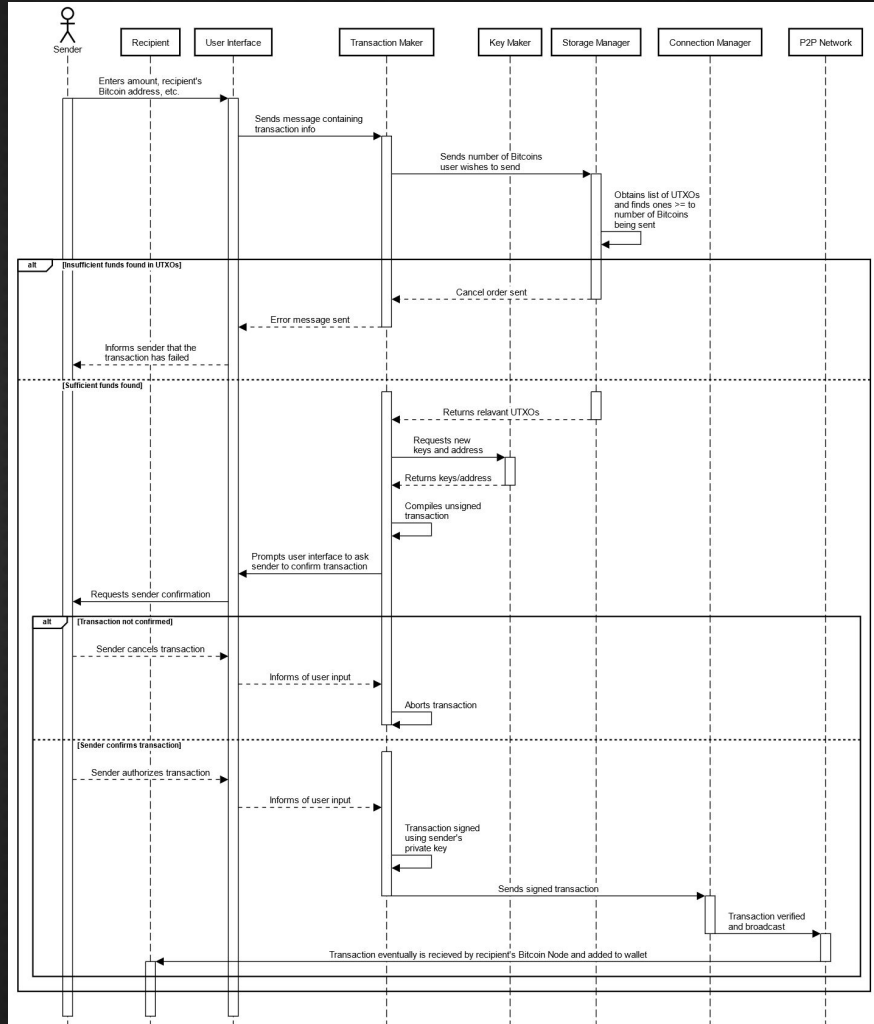






External Interfaces

- The Bitcoin Core system interacts with two external interfaces: the Bitcoin Network and the user interface. All input and output are sent through these interfaces.
- The Bitcoin Network interface is how the system connects to its Bitcoin peers. Transactions, blocks and peer information are sent from the Bitcoin Network to the Bitcoin Core system and vice versa
- The user interface manages input and output regarding user transactions and bitcoin balance. There are four main functions the user interface fulfils: sending bitcoin, Bitcoin address requests, checking a user's bitcoin balance and viewing a user's past transactions. When the user wants to send bitcoin, they input the amount of bitcoin and the Bitcoin address of the recipient into the user interface.



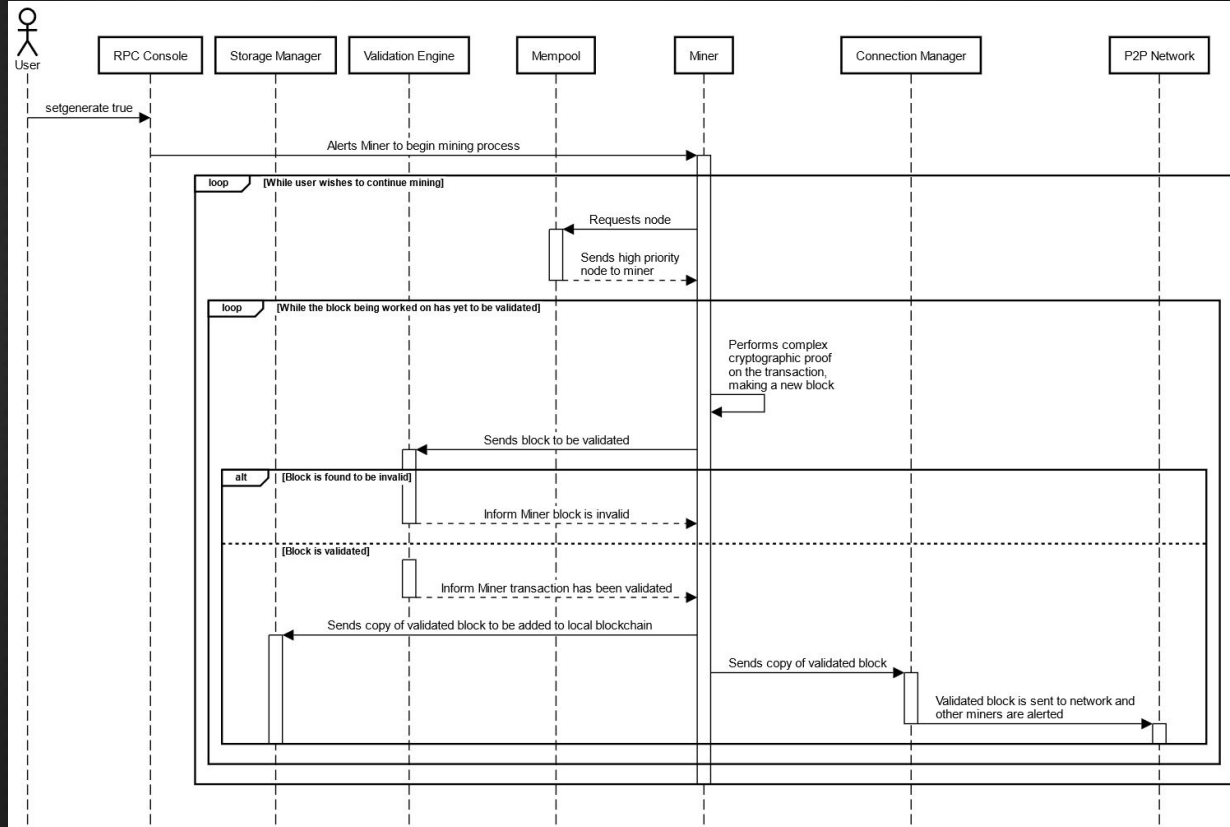
Use Cases



Use Cases



Use Cases



Conclusion

- OOP + P2P = Client-Server Interaction
- Many subsystems & interfaces that make up Bitcoin Core
- “Decentralization” allows for easier user interaction and independence
- Started out very “barebones,” evolved into a much more complex system
- Separation of components and version control aid in project development



Limitations and Lessons Learned

- A time limitation was present, and resulted in us rushing through parts of our presentation and report just a little bit
- We didn't account for the amount of time it would take us to research and summarize our findings from various different sources
- We also underestimated the complexity of the conceptual software we were tasked with analyzing, and because of that, it took us a longer time to completely understand how things were supposed to work
- Software development is fast-paced and constantly changing
- Past versions were harder to use for analysis & understanding



A large, faint, circular watermark of the Bitcoin logo is centered in the background. It features the Bitcoin symbol (a stylized 'B' with two vertical bars) in the center, surrounded by concentric circles. The outermost ring of the watermark contains the text "BITCOIN DIGITAL DECENTRALIZED PEER TO PEER" in a circular arrangement. The entire background is dark with a subtle pattern of small, light-colored dots.

Thank you
for listening!