Conceptual Architecture of Bitcoin Core

Presented by group 14, “Kryptic”

https://www.youtube.com/watch?v=zoCf7vkCBGM&ab_channel=DylanChipun

Team Kryptic

- Eric Lam – 20229013 (Leader, introduction, overview, wallet, tax & mempool)
- Andrew Zhang – 20210066 (Presenter, RPC, miner, data dictionary, conclusion)
- Dylan Chipun – 20224970 (Presenter, introduction, external interfaces, presentation)
- Amy Hong – 20219853 (abstract, peer discovery, connection manager, blockchain)
- Sueyeon Han – 20217002 (use cases, lessons learned)
- Asher Song – 20112257 (validation & storage engine, conclusion)

What is Bitcoin?

How does this lead into Bitcoin Core?

Presentation Outline

01. Architecture Style & Components

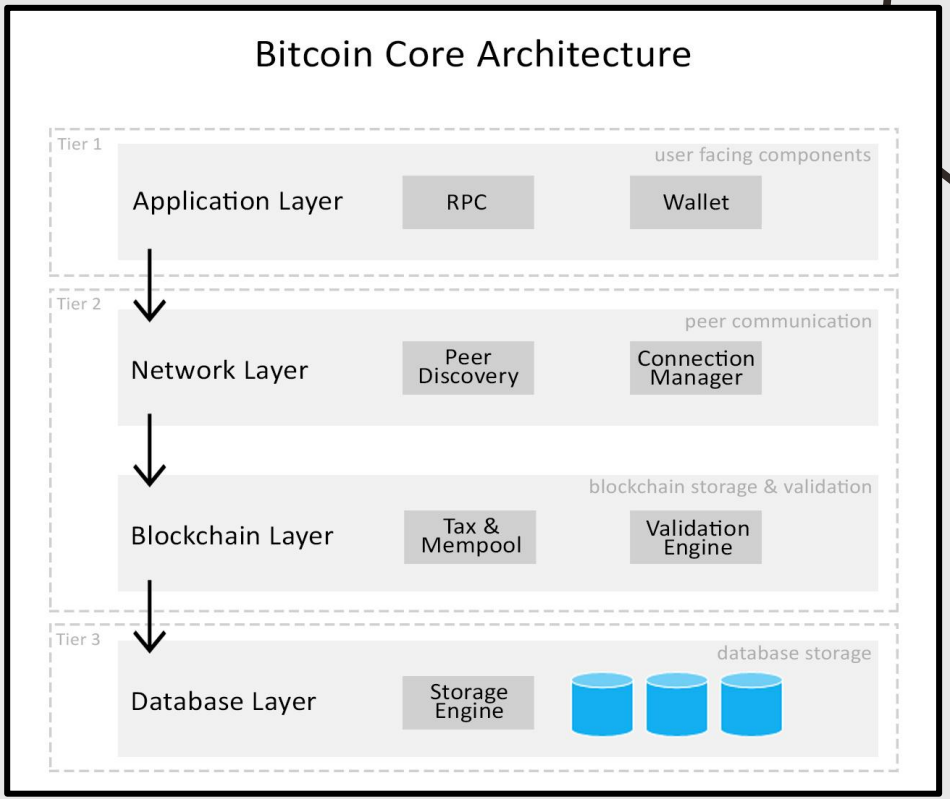
02. Functionalities

03. Component Control & Flow

04. Architecture Analysis

Architecture Style

- Layered
- Each Node is another peer
- No need of a central system



Peer & Peer Discovery

- Represents the current user as a node
- Implements a full node which can discover outbound nodes
- Discover peers through IP address ports
- The Address Manager assists in peer discovery

Connection Manager

- Manages interactions between peer nodes
- Utilizes Semaphore socket threads
- Processes inbound messages
- Ensures security of the system by using Denial-of-Service prevention

Wallet

- Controls user money, manages keys and addresses, tracks balance, and creating and signing transactions
- Uses private/public keys for verification
- Nondeterministic key
  - Key is independently generated
- Deterministic key
  - Derived through a one-way hash function

RPC (Remote Procedure Call)

- Calls procedures to execute
- Eases program runtime
- RPC can be called to other components
    - E.g, the Storage Engine, and Connection Manager

Txs and Mempool

- Encode the transfer value in the bitcoin system
- Most transaction outputs are translated into spendable chunks called UTXO
- Transaction outputs consist of two parts
  - Amount of bitcoin denominated
  - A cryptographic puzzle

Blockchain

- Blockchains validate when a new node is made
- The architecture manages blocks through a block index database
- Bitcoin core stores validation status, the number of transactions saved on the block, and the number of blocks in its chain

Miner

- The process where blocks are added to the blockchain for rewards

- Two ways of mining
  - Solo mining: a single miner creates new blocks for large payment
  - Pooled mining: multiple miners can find blocks together and split payments

- The miner component acts on it’s own

Validation Engine & Storing Engine

- Validation Engine: checks validity of incoming transactions and blocks
- Validated transactions are sent back to the mempool
- Storage Engine: manages the blockchain for the system in a database

External Interfaces

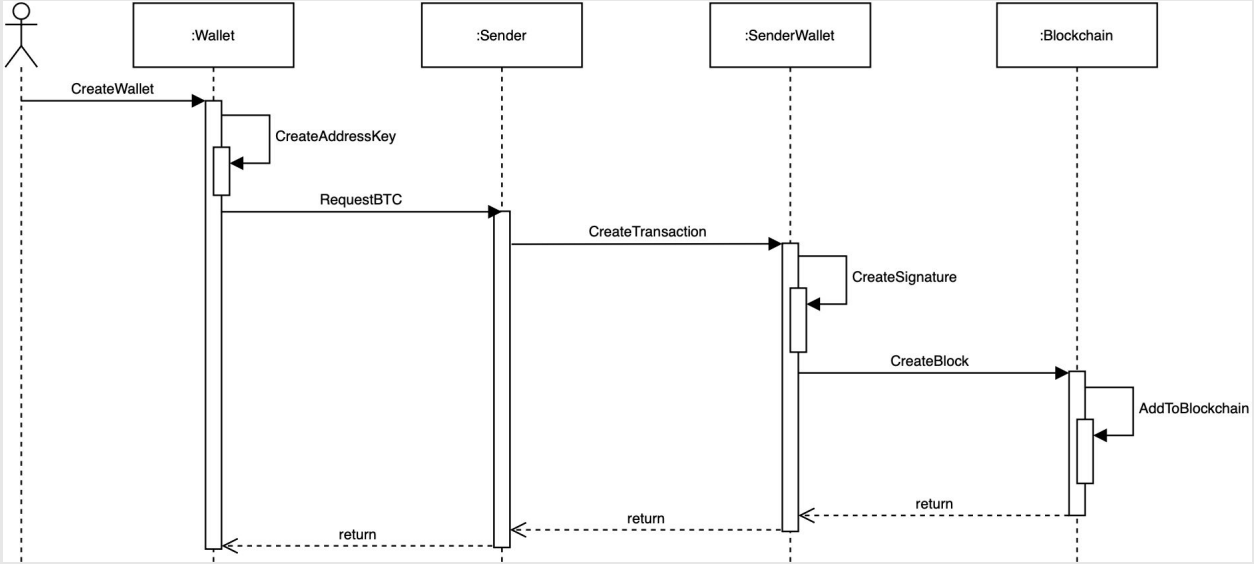
Some external interfaces include

- The Graphical User Interface (GUI)
- The Database
- Application Programming Interfaces (APIs)

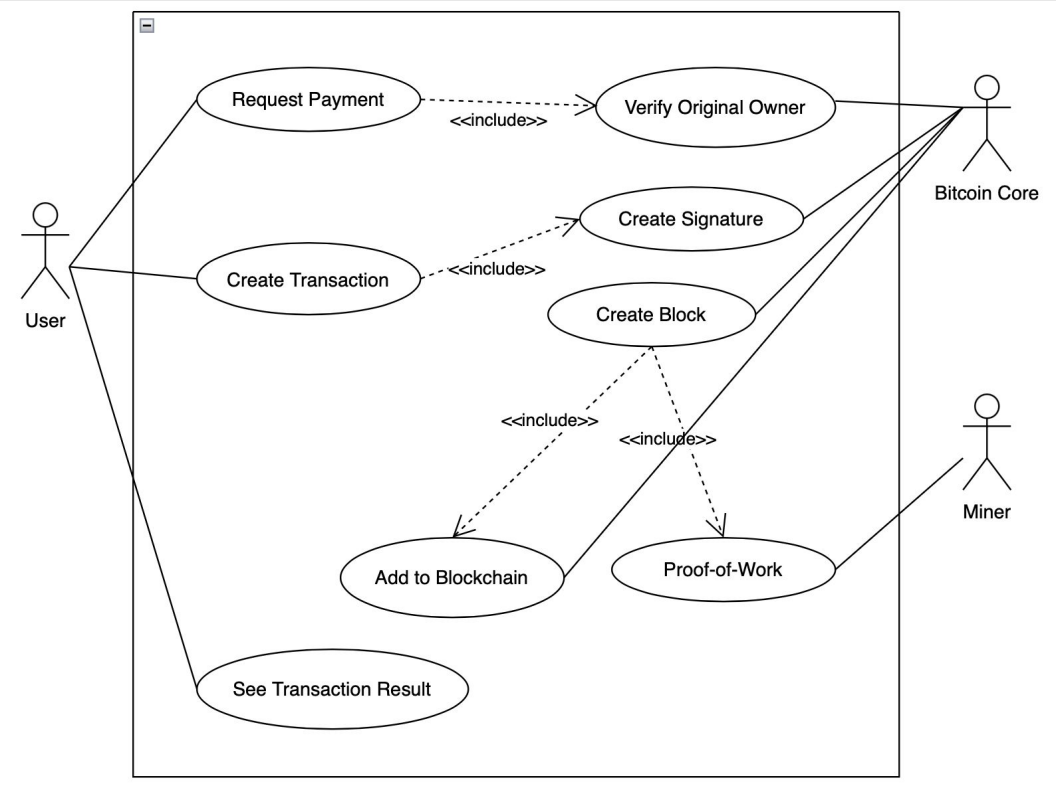
Lessons Learned

- Decentralized nature of Bitcoin
- Higher-level concepts surrounding Bitcoin
- Finding the architecture style late

Use Case #1


:Wallet
:Sender
:SenderWallet
:Blockchain
CreateWallet
CreateAddressKey
RequestBTC
CreateTransaction
CreateSignature
CreateBlock
AddToBlockchain
return
return
return

Use Case #2


Request Payment
<<include>>
Verify Original Owner
Bitcoin Core
Create Signature
User
Create Transaction
<<include>>
Create Block
<<include>>
<<include>>
Miner
Add to Blockchain
Proof-of-Work
See Transaction Result

Conclusion