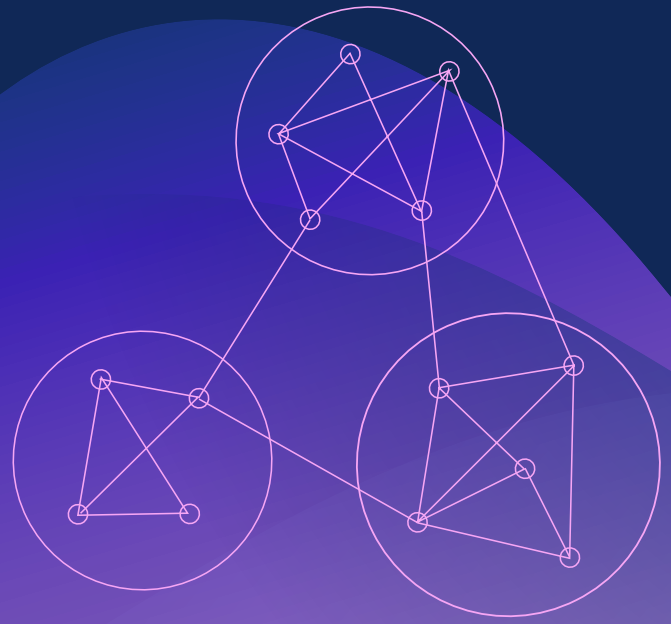
BITCOIN CORE: CONCRETE ARCHITECTURE

A2 – Group 2 – Bit by Bit

https://youtu.be/BCZ4pBmdpnA



PRESENTATION BY:

- Daniella Ruisendaal - Group Leader
- Alina Padoun - Presenter
- Adam Ciszek - Presenter
- Aidan Wolfson
- Camila Izquierdo
- Tanner Big Canoe

CONCEPTUAL ARCHITECTURE

- Blockchain
  - Interacts primarily with P2P network, wallet, transaction, mining and operating mode subsystem
- P2P Network
  - Interacts primarily with blockchain, transaction, mining, and wallet subsystems
- Wallet
  - Interacts primarily with blockchain and transaction subsystem

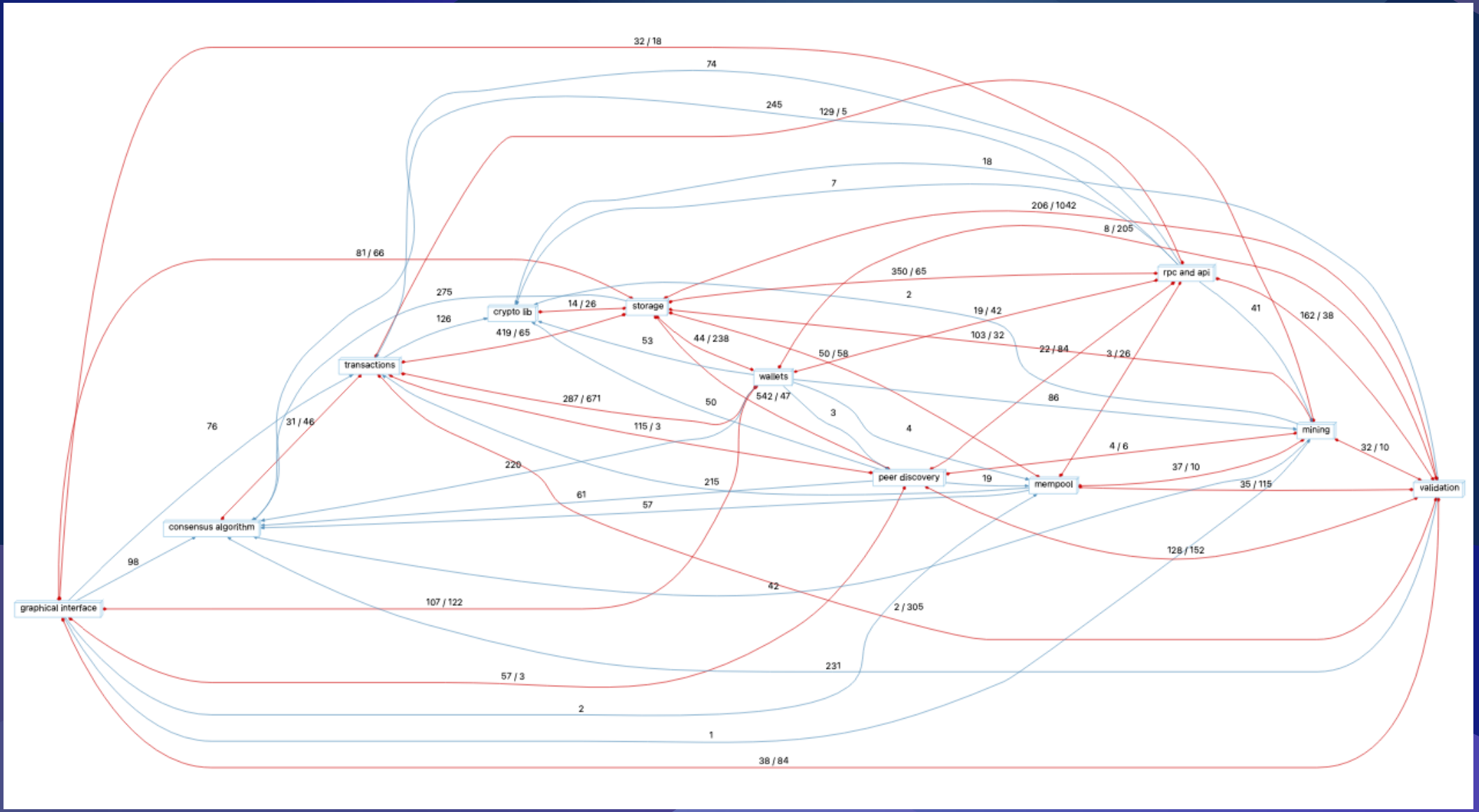
CONCEPTUAL ARCHITECTURE

- Transaction Module
  - Interacts primarily with blockchain, P2P network, wallets, and payment processing subsystem
- Mining
  - Interacts primarily with blockchain and P2P network subsystem
- Payments Processing
  - Interacts primarily with transaction, P2P network and blockchain subsystem
- Operating Mode
  - Interacts primarily with blockchain and P2P network subsystems

HIGH-LEVEL CONCRETE ARCHITECTURE

- Derivation process from source code
- Interactions and dependencies between:
  - Components from conceptual architecture, such as wallets, validation, peer discovery
  - New components, such the API module, and the consensus and crypto libraries

HIGH-LEVEL CONCRETE ARCHITECTURE



REFLEXION ON HIGH-LEVEL CONCRETE ARCHITECTURE

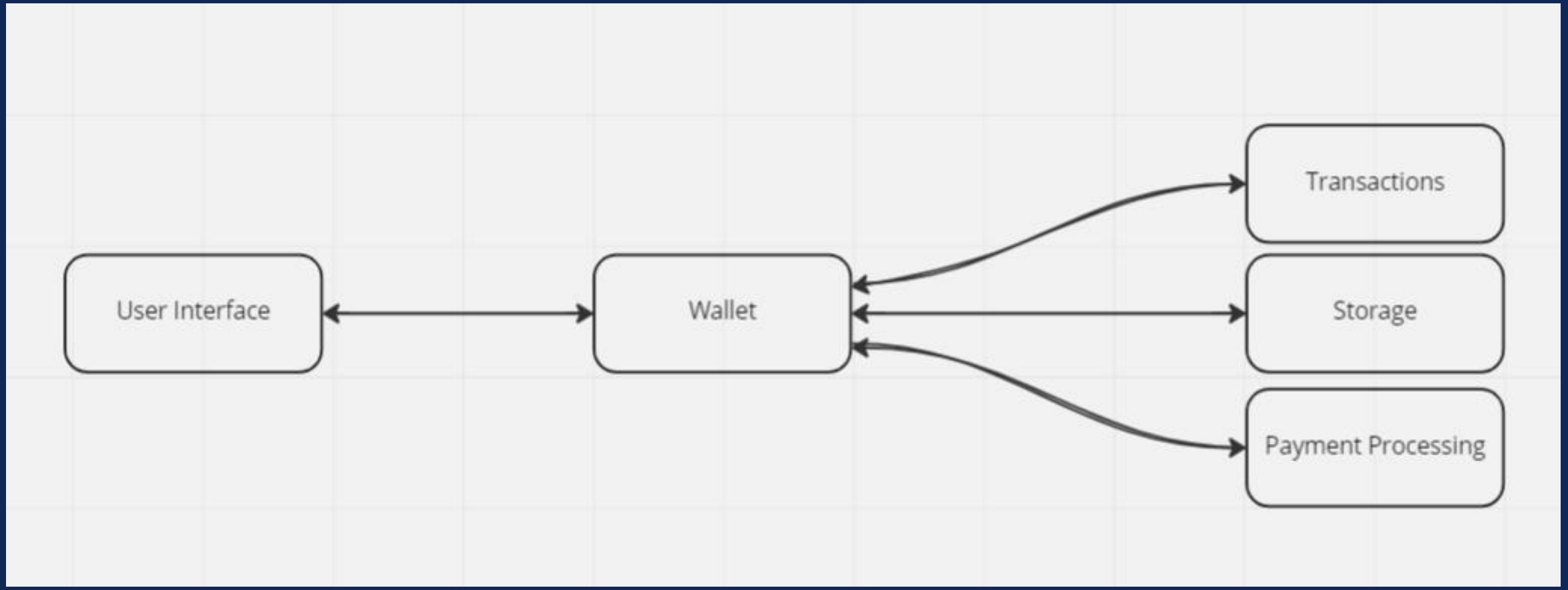
- **New Modules:**
  - Application Programming Interface (API)
  - Consensus Algorithm
  - Cryptographic Libraries
- **Unexpected Dependencies:**
  - Wallet -> RPC
  - Validation -> Consensus
  - Transactions -> Crypto Library

WHAT IS A WALLET SUBSYSTEM?

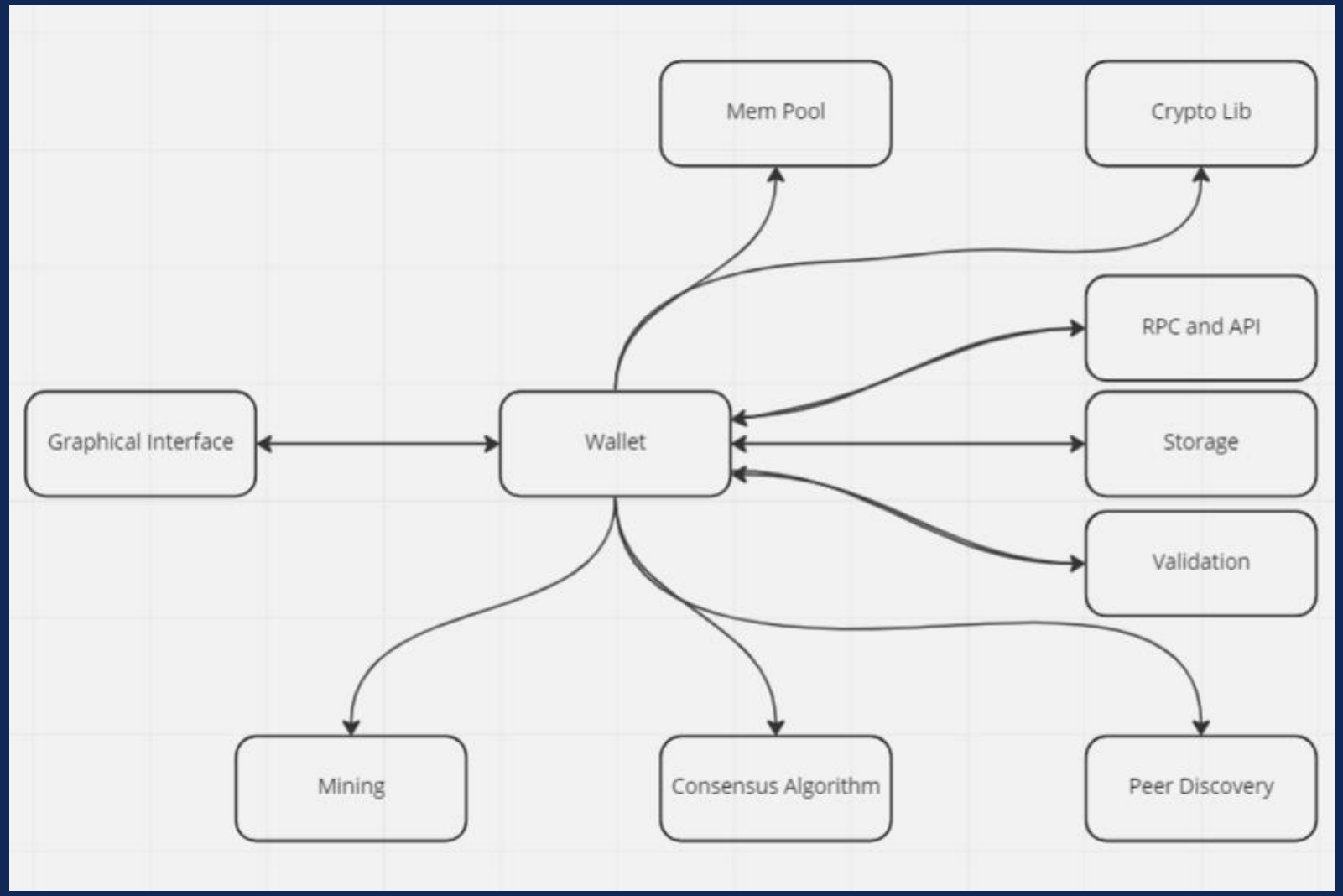
User's perspective; managing keys and addresses, accessing money, tracking balance, and creating/signing transactions.

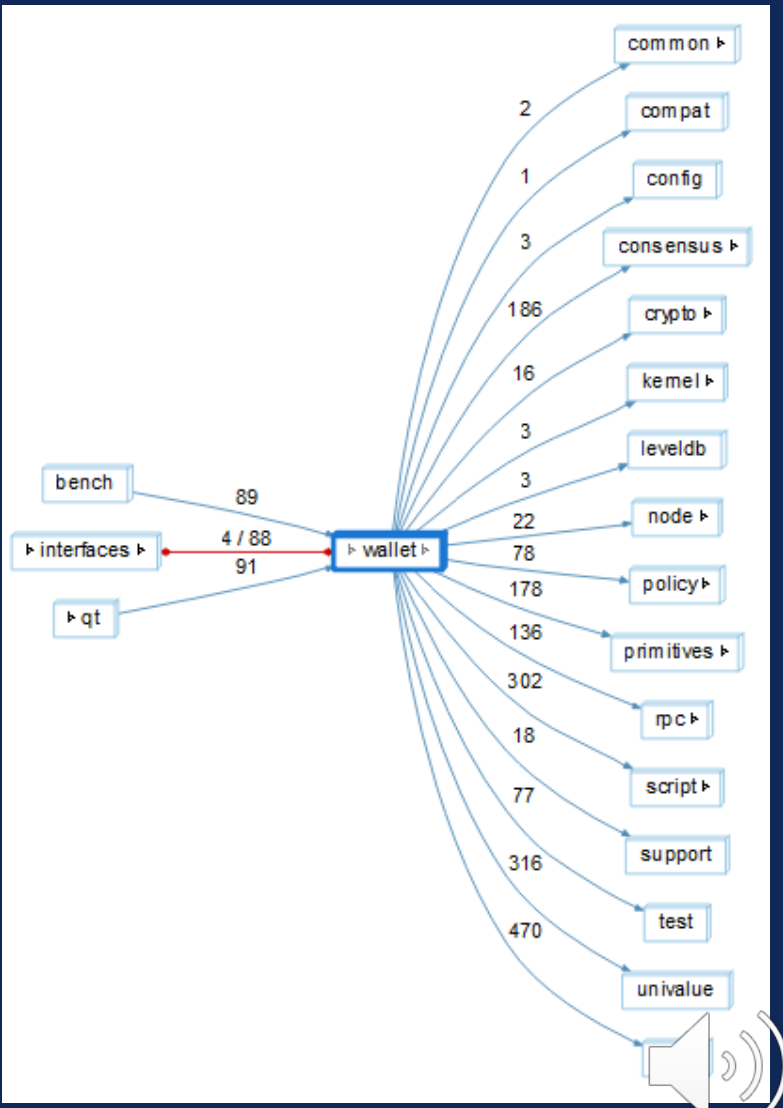
Programmer's perspective; data structure used to store and manage keys.

WALLET SUBSYSTEM CONCEPTUAL ARCHITECTURE:



WALLET SUBSYSTEM CONCRETE ARCHITECTURE:





REFLEXION ON WALLET SUBSYSTEM

- New Modules:
  - Bench
  - Qt
- Unexpected Dependencies:
  - Bench -> Wallet
  - Qt -> Wallet

CONCURRENCY

- Concrete architecture shows Bitcoin Core uses various **threads, mutexes, and global locks** to achieve concurrency while guarding shared data structures
- Much of the initialization and management of this occurs throughout https://github.com/bitcoin/bitcoin/blob/master/src
- System faces issues regarding **deadlocks**, and offers tools to assist with debugging such deadlocks

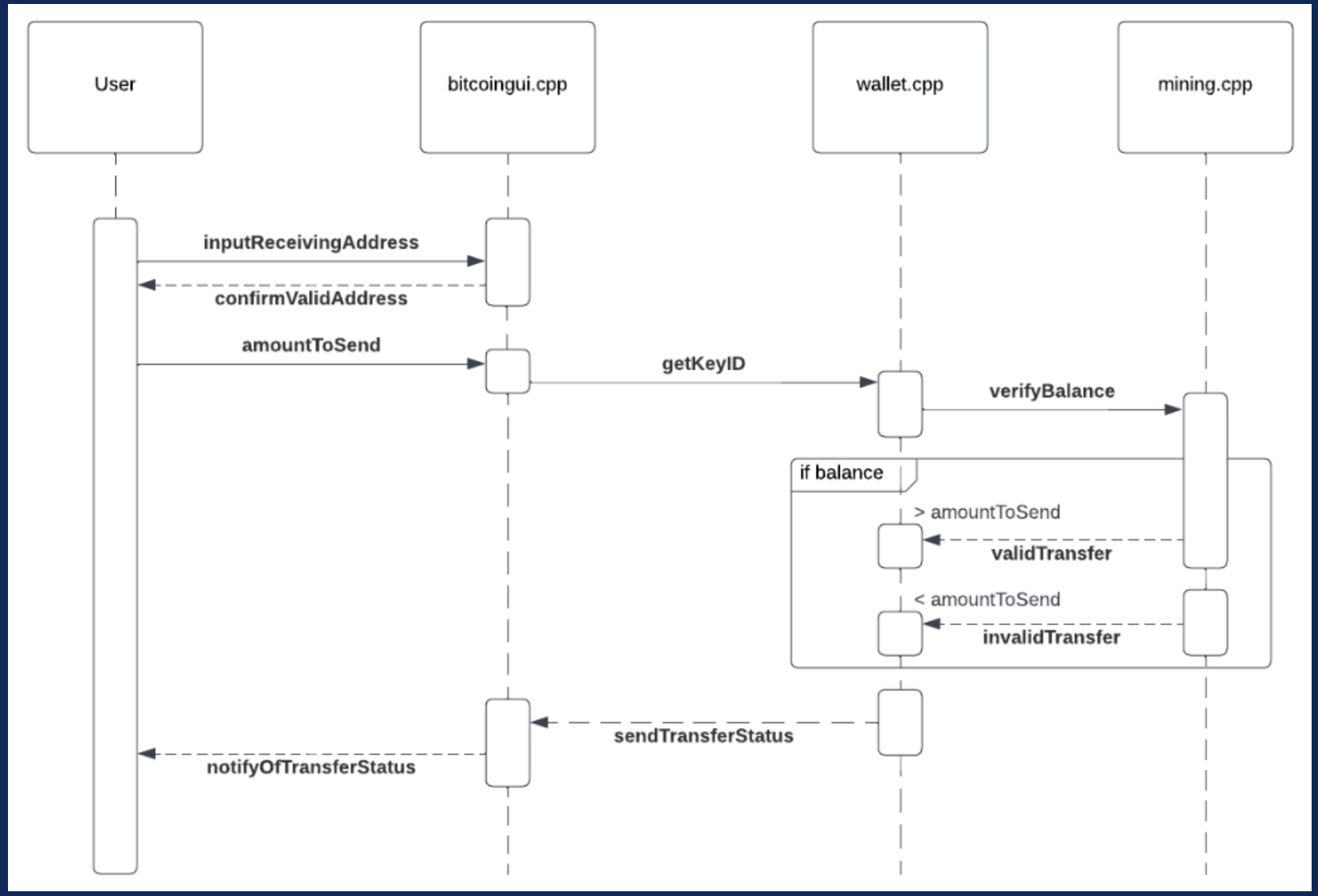
TEAM ISSUES AND IMPLICATIONS OF DIVISION

- Same observations as for the conceptual architecture
- Open-source and all about collaboration
- Small group of developers who can access the main branch of code for Bitcoin directly, limiting damage
- Fluctuation in number of developers working on Bitcoin Core
- Multiple levels of protection in place to keep the code safe:
  - Commit keys
  - Verify-commits

SEQUENCE DIAGRAM

Use Case 1:

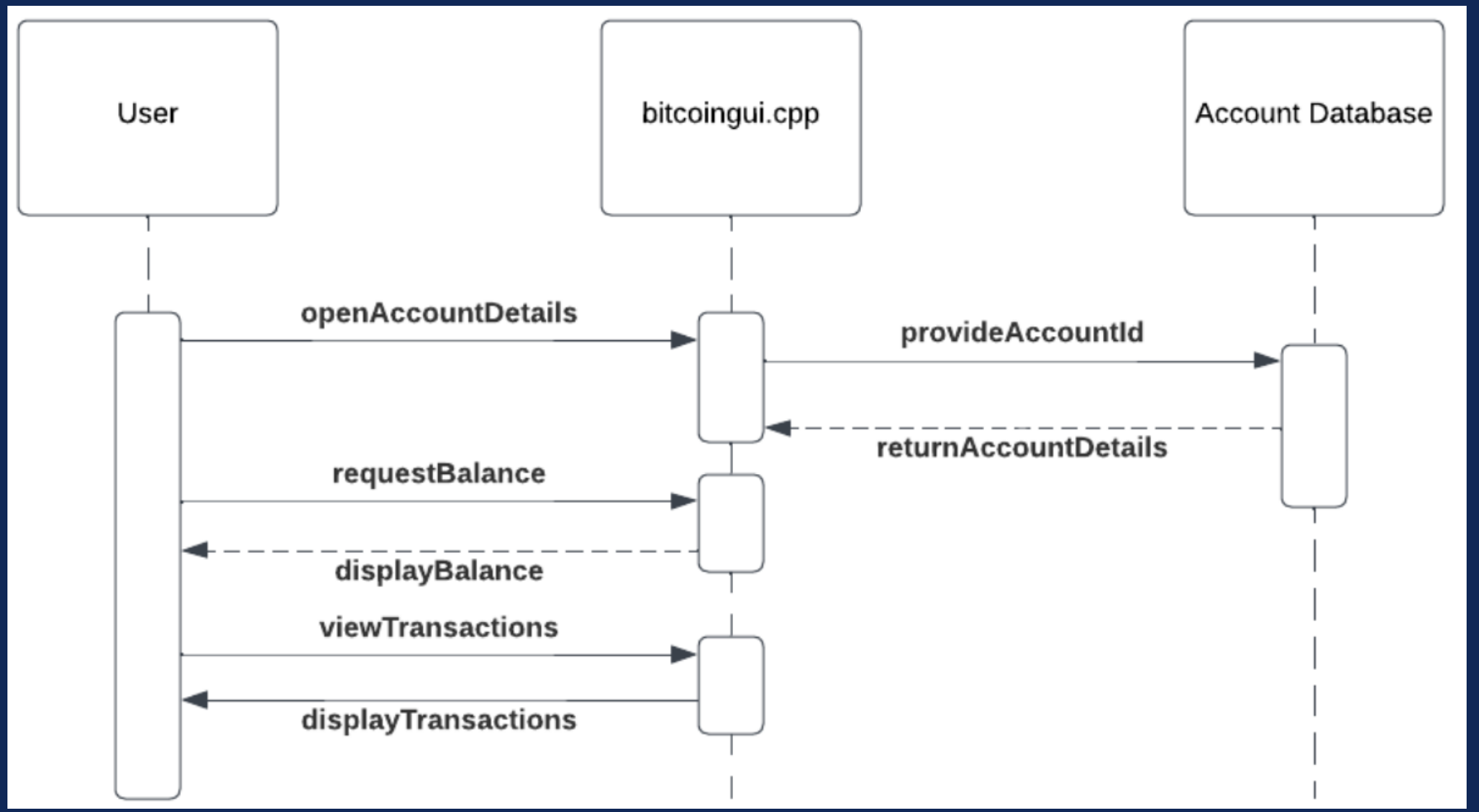
User A wants to send Bitcoin over to user B.

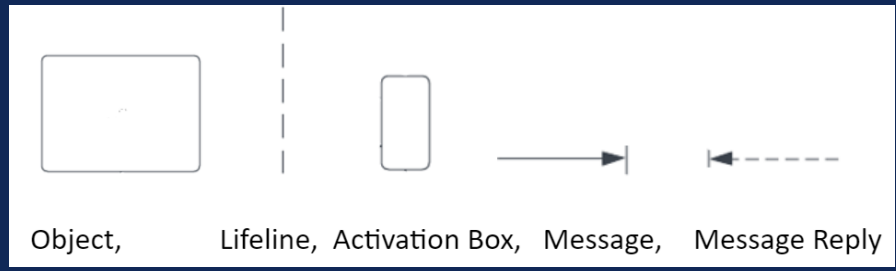


SEQUENCE DIAGRAM

Use Case 2:

User navigates to their settings and views their transaction history and current balance.

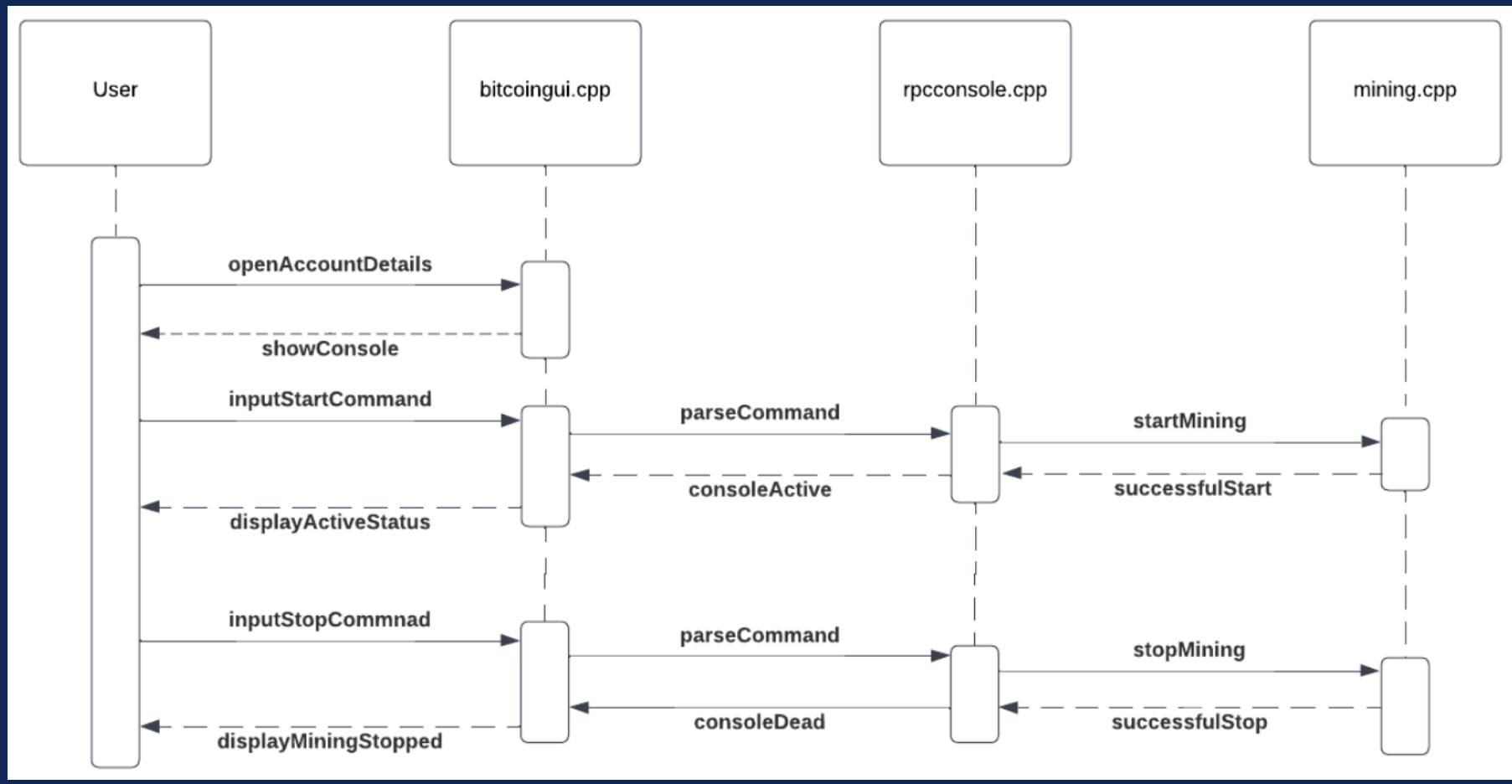


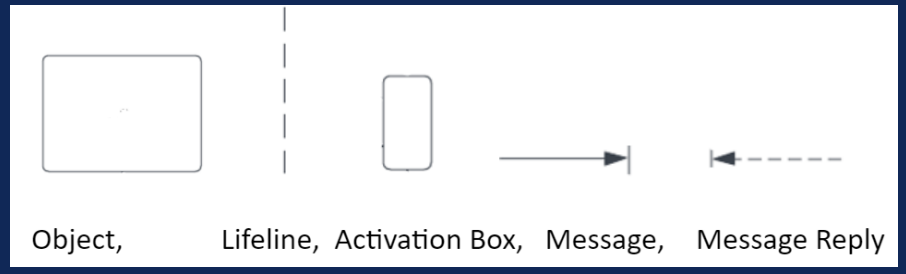


SEQUENCE DIAGRAM

Use Case 3:

User begins and ends a mining session via the console





LIMITATIONS AND ALTERNATIVES

- No ability to speak directly with developers (and others) to achieve conceptual architecture
- Vast quantity of files and lines of code
  - Increased potential for gaps in concrete architecture
- Minimal commenting in certain files
- Subsytems within subsytems vs independent subsytems

LESSONS LEARNED

CONCLUSION