BITCOIN CORE: ARCHITECTURAL ENHANCEMENT

A3 – Group 2 – Bit by Bit

https://youtu.be/gjI4mq7pIZg

PRESENTATION BY:

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

ENHANCEMENT PROPOSAL

Problem: Limited concurrency for many critical paths due to global locks and mutexes.

Solution: Reducing and splitting up these global locks in order to improve modularity and increase parallelism.

CURRENT STATE OF SYSTEM

- Multi-threaded but most critical actions are single-threaded
  - Running wallet tasks
  - Completing transactions
  - Validating blocks
- Limited parallelism due to global locks and mutexes
  - Recursive mutex cs_main, among others
- Potential for additional modularity and parallelism

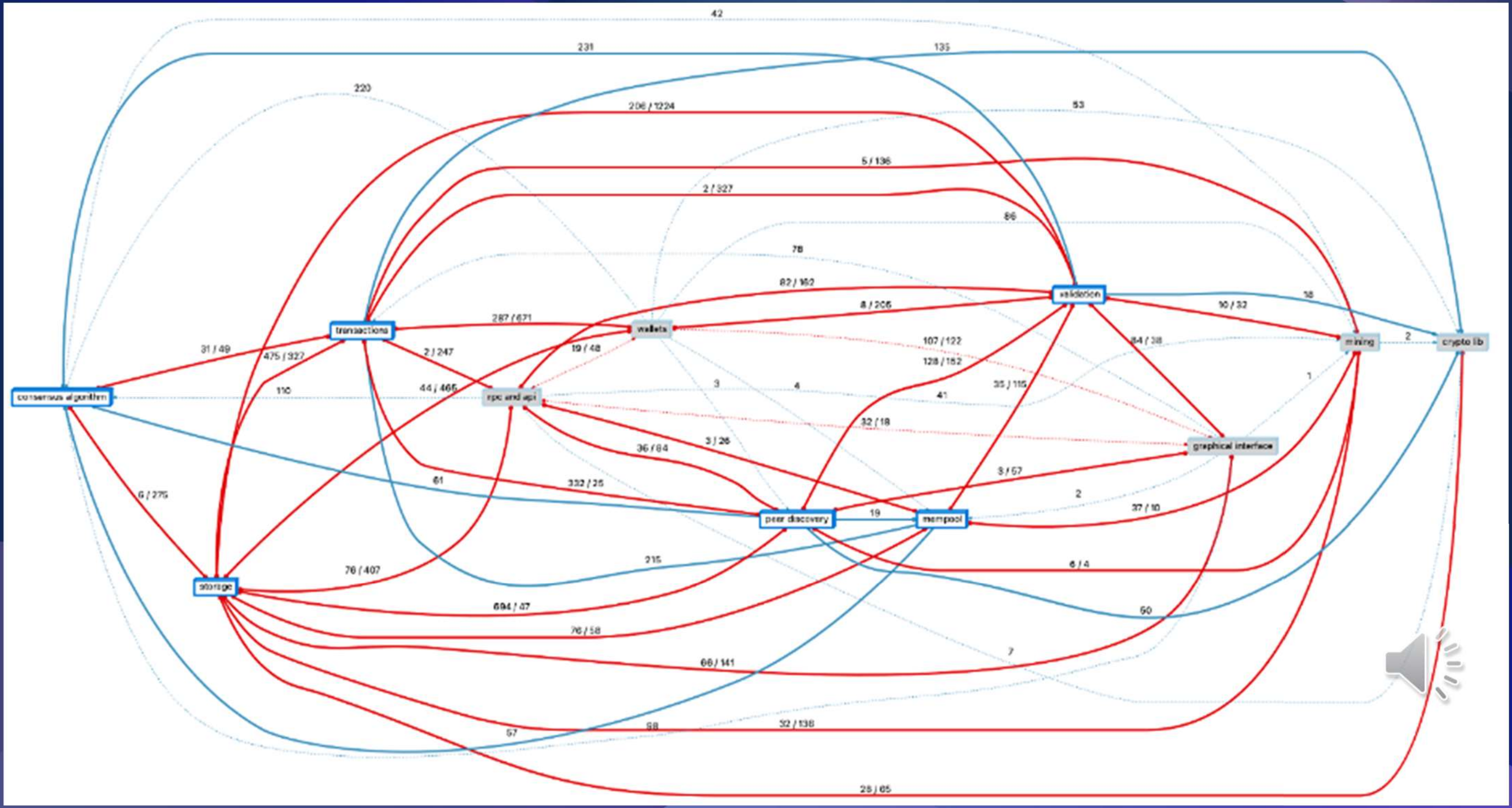
EFFECTS OF ENHANCEMENT

- Maintainability
  - May remain the same
  - Modularity
  - Risk of issues
- Evolvability
  - More concurrency
- Testability
  - Higher workloads and throughput
- Performance
  - More concurrency
  - Higher throughput
  - Faster response time

INTERACTIONS OF ENHANCEMENT

Subsystems	Files
Validation/Consensus Algorithm	*Validation.cpp*, etc.
Transactions	*attributes.h*, *txdb.cpp*, *arith_uint256.cpp*, *policy.cpp*, etc.
Storage	*coins.cpp*, *blockencodings.cpp*, *chain.h*, *bitcoin-tx.cpp*, and other directories such as *leveldb and txdb*
Peer Discovery	*protocol* files and the *connection manager* directories
Mempool	*txmempool.cpp*, *mempool_limits.h*, *mempool_entry.h*, etc.

INTERACTIONS OF ENHANCEMENT



ALTERNATIVES

- Implementation 1:
  - Local lock feature is distributed across the existing module pathways
  - Global lock would become a local lock that exists within each of the modules
  - Advantages:
    - Performance benefits
    - Increase parallelism
    - Secure
    - Improved modularity
  - Disadvantages:
    - Time consuming to implement

- Implementation 2:
  - Remove global lock feature
  - Advantages:
    - Easy to implement
    - Increased parallelism
    - Performance benefits
  - Disadvantages:
    - Not secure

STAKEHOLDERS

Major Stakeholders	Non-functional Requirements
Bitcoin Core Team	Performance, Maintainability, Scalability
Independent Developers	Performance, Maintainability, Scalability
Bitcoin Users	Performance, Safety
Investors	Performance, Safety

SAAM ANALYSIS

Non-functional Requirement	Implementation 1	Implementation 2
Performance	Medium	High
Safety	High	Low
Manageability	Medium	Low
Scalability	Medium	High

TESTING

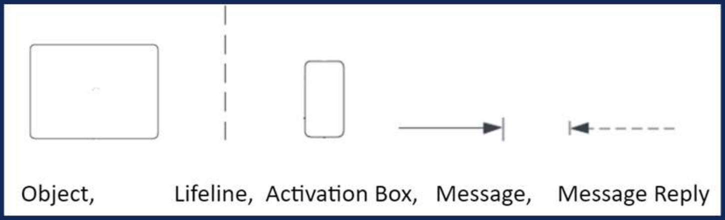
- Process timing
  - Used to test performance before/after implementing the enhancement
  - Can be implemented using chrono library
- Load Testing
  - Used to test load capacity since the enhancement increases the number of threads running simultaneously
  - Tools such as Googletest can be used

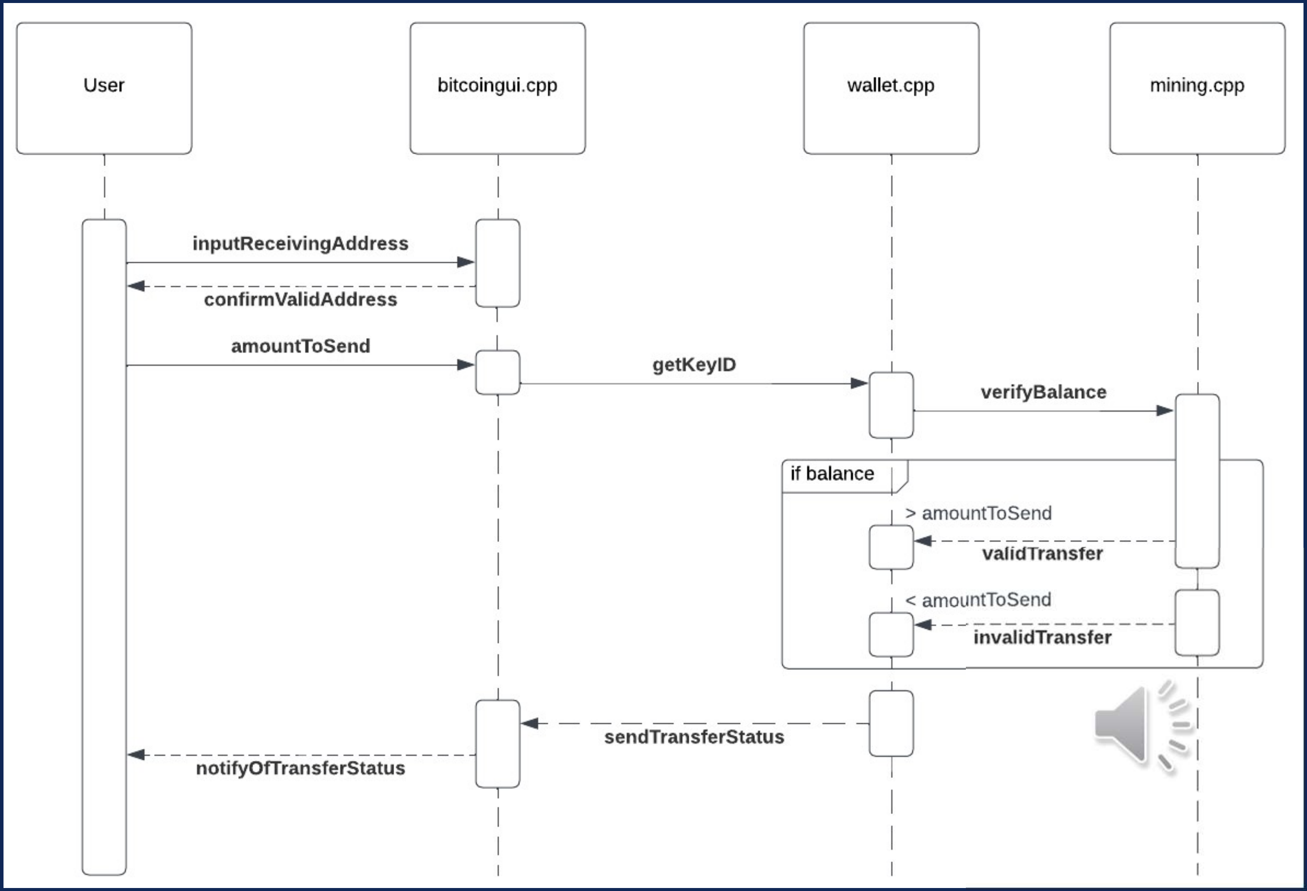
POTENTIAL RISKS

- If mutexes and locks are not removed/split correctly:
  - Runtime and logic errors
  - Data corruptions due to multiple threads accessing/modifying same resources
  - Example: changing the global lock for the transactions module could cause race conditions, with multiple processes trying to access the same transaction data at the same time

SEQUENCE DIAGRAM BEFORE ENHANCEMENT

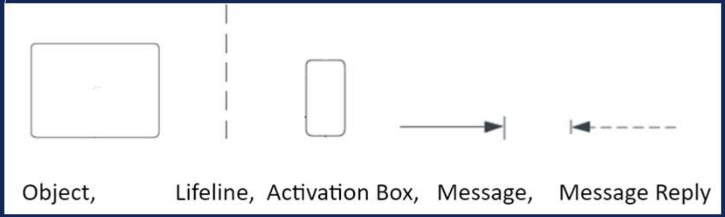
Use Case 1: User A wants to send Bitcoin over to user B.

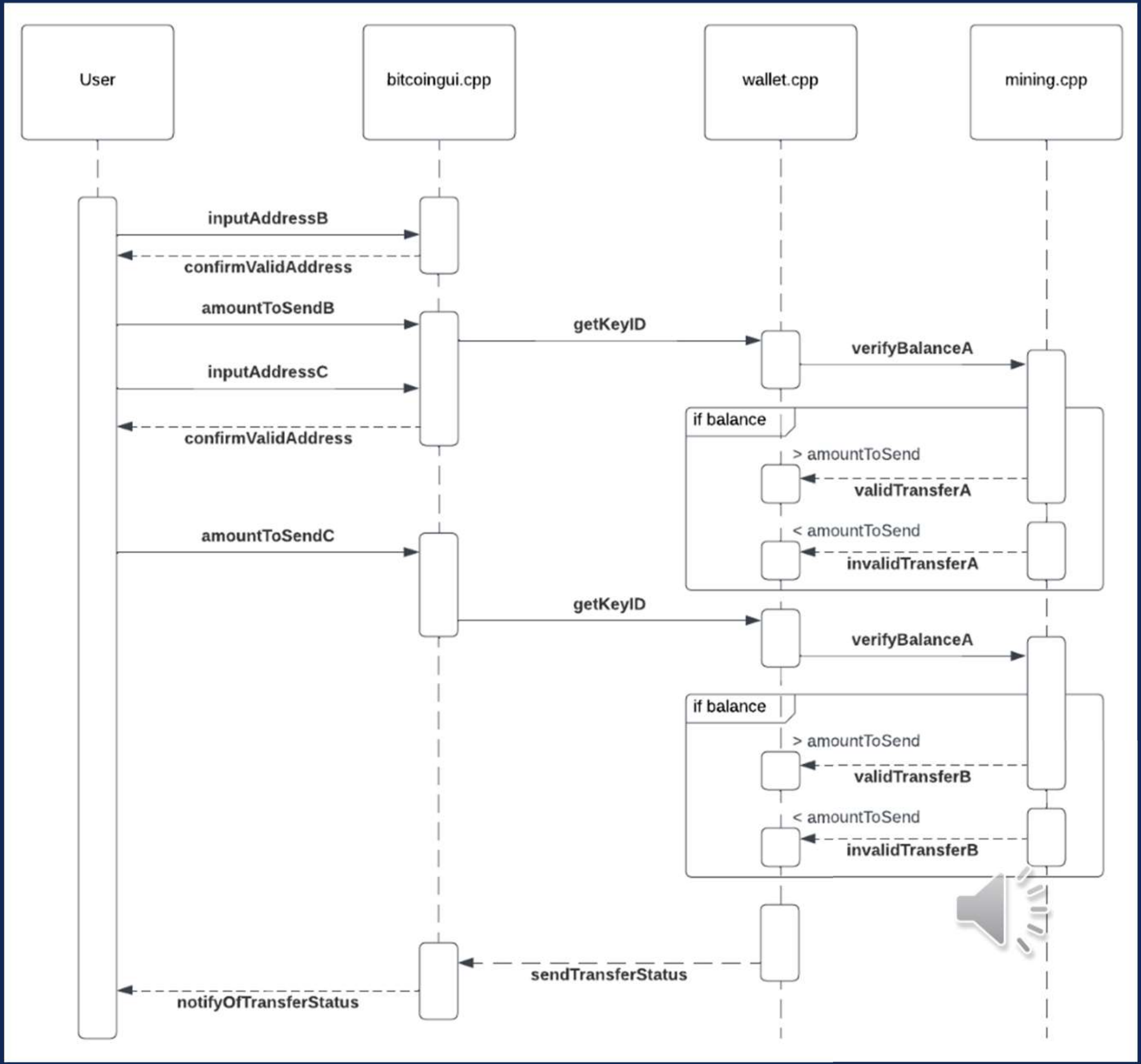




SEQUENCE DIAGRAM AFTER ENHANCEMENT

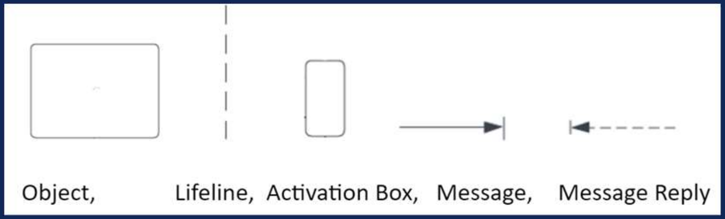
Use Case 1: User A wants to send Bitcoin over to user B.

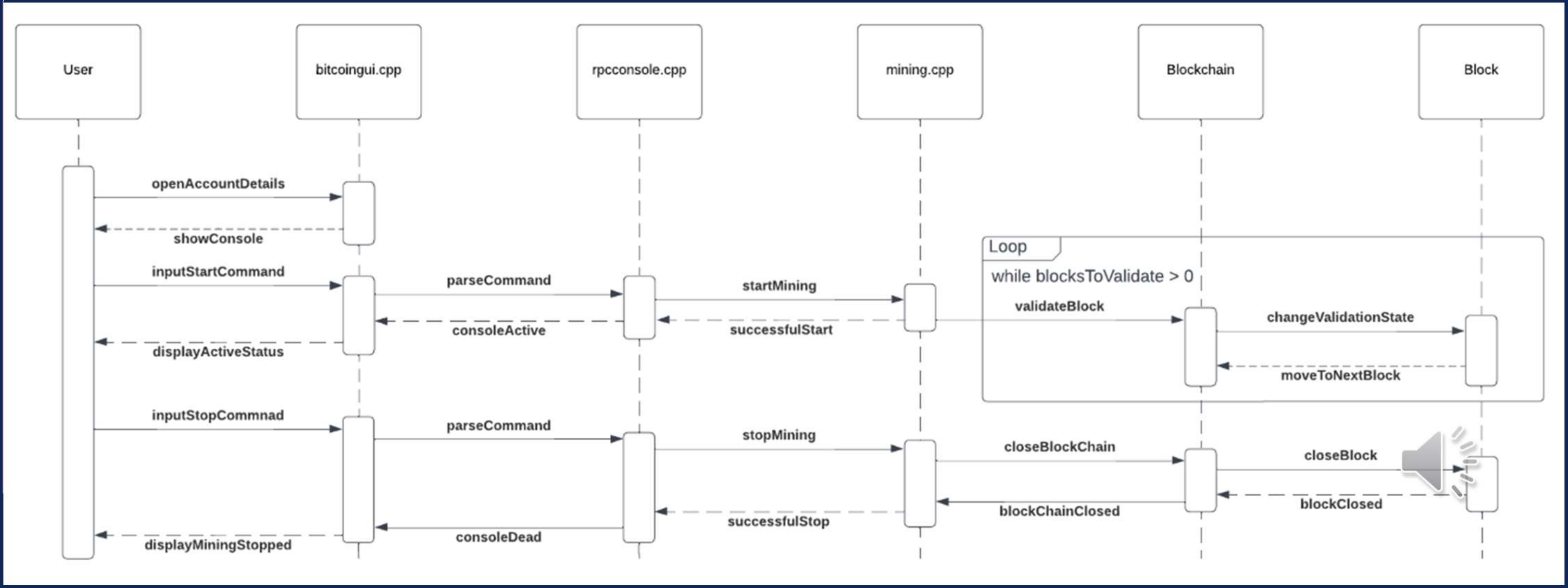




SEQUENCE DIAGRAM BEFORE ENHANCEMENT

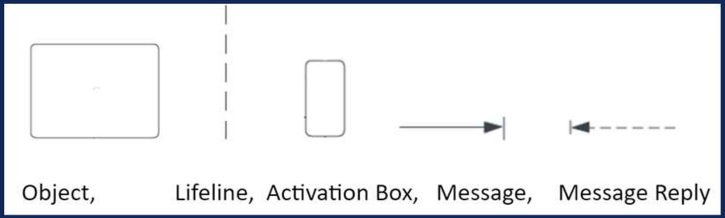
Use Case 2: *User begins and ends a mining session via the console*

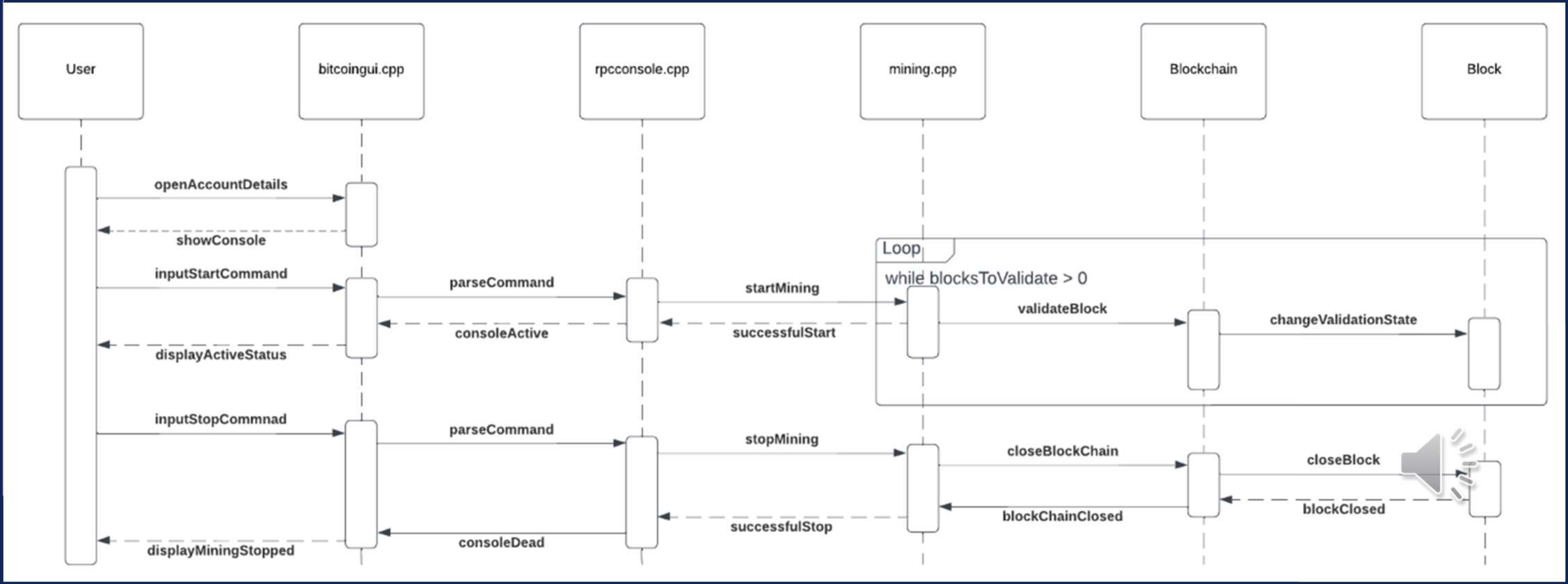




SEQUENCE DIAGRAM AFTER ENHANCEMENT

Use Case 2: *User begins and ends a mining session via the console*





TEAM ISSUES AND IMPLICATIONS OF DIVISION

- 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

LIMITATIONS

- No ability to speak directly with developers (and others) with questions and concerns related to this potential enhancement
- Removal of a global lock, while providing many benefits, may negatively affect the system more than it helps
  - Needs proper planning, testing and risk mitigation

LESSONS LEARNED

- Splitting mutexes can help improve modularity and increase parallelism
- Many subsystems/modules will be impacted
- Many potential side effects that require rigorous planning and testing

CONCLUSION