Enhancement Proposal of Bitcoin Core

Presented by group 14, “Kryptic”

https://youtu.be/TZvMcSLmjL8

Team Kryptic

- Eric Lam – 20229013 (Leader)
- Andrew Zhang – 20210066 (Presenter)
- Dylan Chipun – 20224970 (Presenter)
- Amy Hong – 20219853
- Sueyeon Han – 20217002
- Asher Song – 20112257

New Enhancement

- Allowing users to cancel unmined Txs.
    - Return it to unconfirmed transaction pool (UTXO)

The Issue Being Solved

- Users are not able to cancel transactions even right after transferring money.

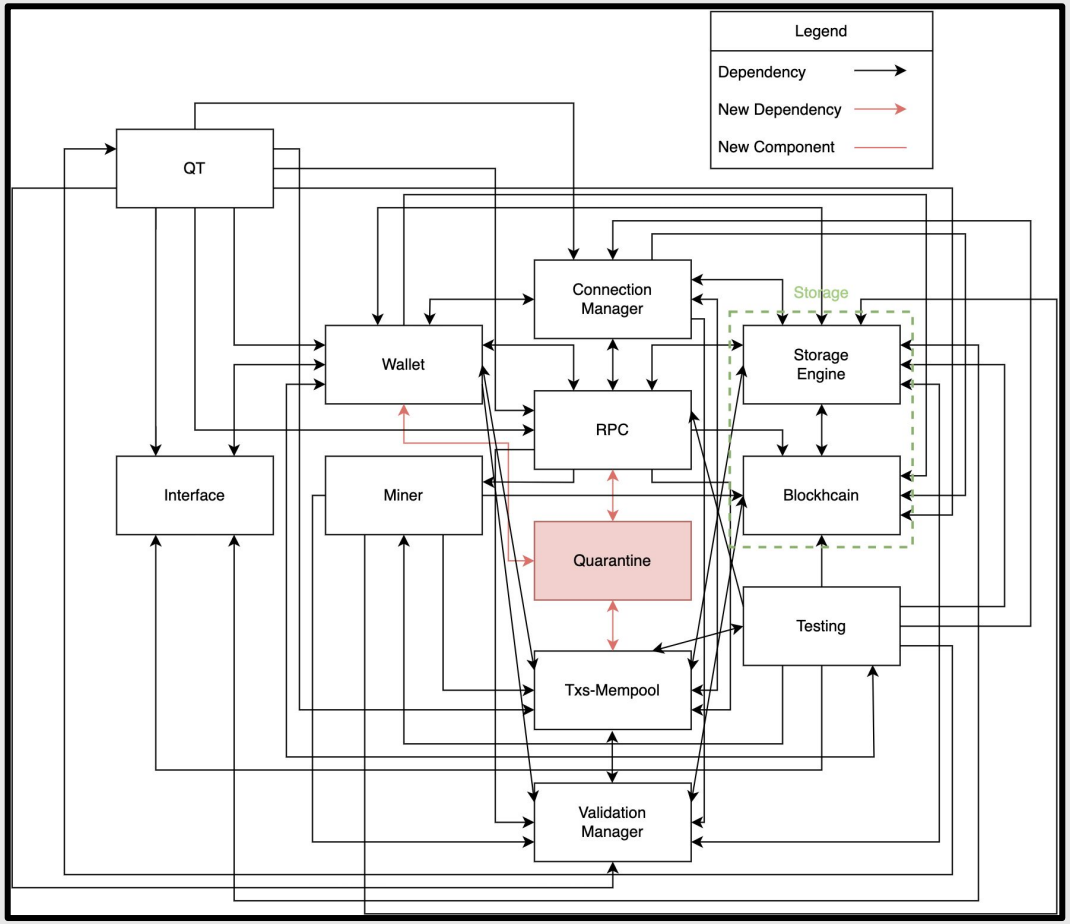
Two Implementations

- Implementation of a new component called “Quarantine”.
- alter the inner components of the components implementation 1 depended on

Implementation 1

- Implementation of a new component called “Quarantine”.
- Invokes the RPC
- Txs & Mempool will verify the locktime and confirmation status
    - If the recipient confirms, the Tx will be reversed to UTXO status
    - otherwise, it will be returned to Txs & Mempool to resume confirmation.
- Publish Subscribe Architecture Style

Implementation 1



Implementation 2

- can be added without adding a new module or component into the system
- alter the inner components of the components implementation 1
- prompting the Mempool for the Tx to be cancelled through their wallet

  causing the Cancel Tx process to be executed by invoking the RPC

SAAM Analysis

Major Stakeholders:

- Bitcoin Users
  - **Usability**: Users want the new feature to be easy to use and learn
  - **Availability**: Be available to use whenever they need it
  - **Security**: Safe to use and will protect user information.
- Bitcoin Core Developers
  - **Performance**: Software to still perform as optimally as possible
  - **Testability**: Test the new feature and all modules that interact with it
  - **Maintainability**: Implement the new feature and have the system sustain itself
- Businesses
  - **Performance**: high in performance to continue operations with consumers
  - **Security/privacy**: Ensure the safety of both company and consumer
  - **Accessibility**: Be accessible to all consumers
  - **Maintainability**: Software to be stable

Implementation 1 Pros & Cons

Pros

- Usability
- Security
- Accessibility

Cons

- Availability
- Testability
- Maintainability

Implementation 2 Pros & Cons

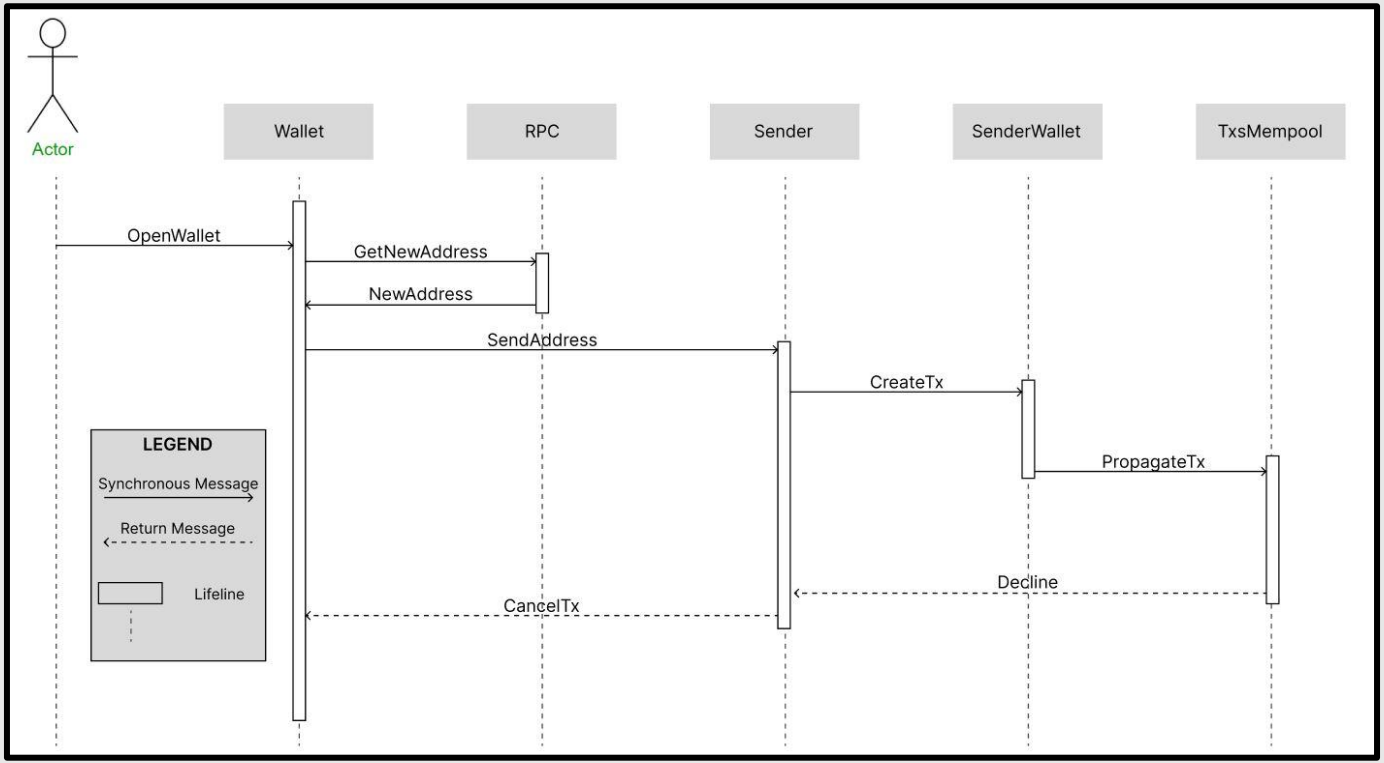
Pros

- Maintainability
- Testability
- Security

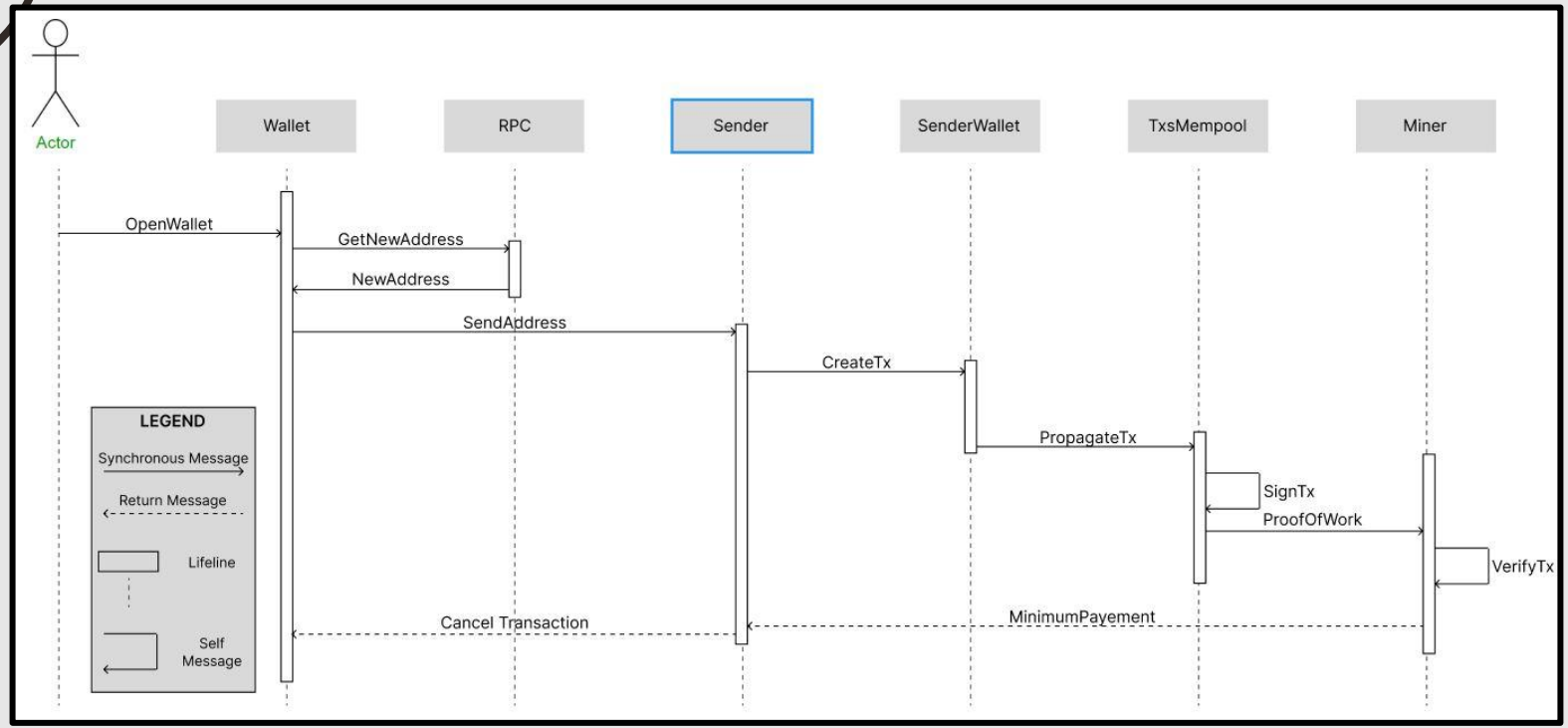
Cons

- Maintainability
- Performance
- Evolvability

Use Case 1


Actor
Wallet
RPC
Sender
SenderWallet
TxsMempool
OpenWallet
GetNewAddress
NewAddress
SendAddress
CreateTx
PropagateTx
Decline
CancelTx
LEGEND
Synchronous Message
Return Message
Lifeline

Use Case 2


Actor
Wallet
RPC
Sender
SenderWallet
TxsMempool
Miner
OpenWallet
GetNewAddress
NewAddress
SendAddress
CreateTx
PropagateTx
SignTx
ProofOfWork
VerifyTx
MinimumPayement
Cancel Transaction
LEGEND
Synchronous Message
Return Message
Lifeline
Self Message

Risks And Limitations

- May compromise the integrity of Bitcoin transactions performed on Bitcoin Core
  - Double spending
  - Complexity to the verification of transactions
  - New forms of DDOS attacks may be possible
- Difficulty in maintainability

Testing

- Integration Testing for the publish-subscribe architecture style
- unit testing of the specific functionalities of the new feature
    - Already mined transactions
    - Invalid transactions

Lessons Learned

- Different processes of building potential parts
- We saw was missing in the base architecture that a lot of users would benefit from
- Easier to consider all the aspects of the feature

Conclusion