Architectural Enhancements

https://quarch.netlify.app/assignments
(The video is linked under our assignments on our site)

Group 19

Introduction

The report proposes an enhancement to combine the P2P and Consensus Validation subsystems of Bitcoin Core's architecture to improve efficiency, with a SAAM analysis, implementation options, potential effects, use cases, risks, and lessons learned discussed.

Proposed Implementations

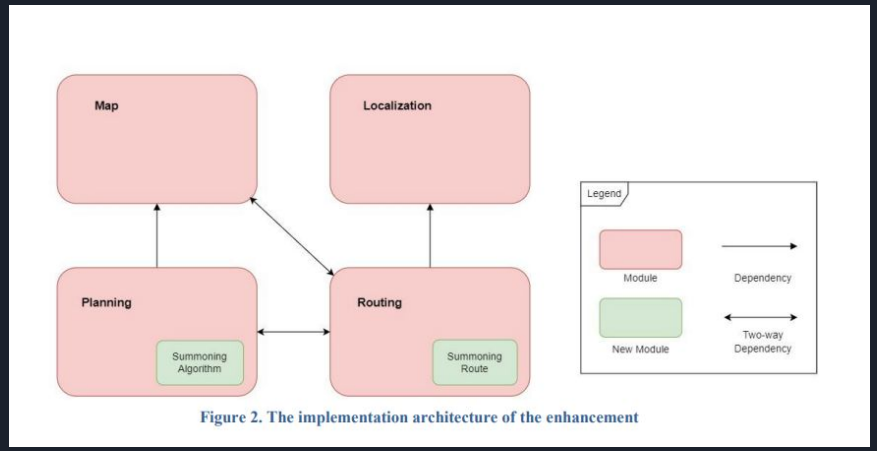


Figure 2. The implementation architecture of the enhancement

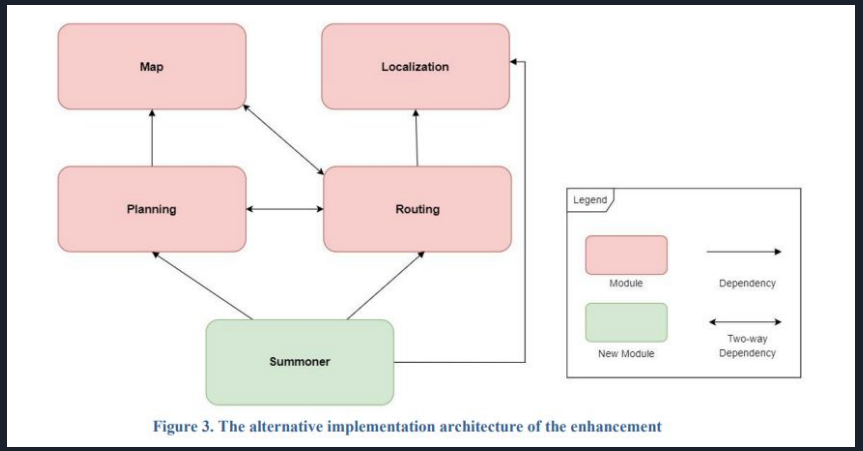


Figure 3. The alternative implementation architecture of the enhancement

Motivation

- Improve Bitcoins network performance, scalability, security, and maintainability
- P2P and Consensus Validation currently operate as separate systems, which leads to inefficiencies and centralization issues.
- Integrating these subsystems will result in more efficient transaction processing, reduced risk of attacks and fraud, and a simpler codebase that is easier to maintain.
- These improvements will ensure the long-term viability of the Bitcoin network, making it a more trustworthy and robust blockchain platform.

SAAM Analysis

- SAAM analysis will be conducted on Bitcoin Core, including stakeholder identification, NFRs, evaluation of enhancement options, and impact assessment.
- The analysis will determine the best enhancement option based on the impact on NFRs and stakeholders.
- Stakeholders
  1. Users
  2. Devs
  3. Investors

Stakeholder	Important NFRs Regarding the Enhancement
User	Safety: The new enhancement maintains the safety that bitcoin core already has integrated in its P2P network. Performance: The application and the network work as smoothly as possible. Usability: Bitcoin core should inform the user of changes, and not impede its usability.
Developers	Testability: The module should be programmed in such a way that it can be tested in a safe environment and/or virtually. The programmers are also looking for a testing environment that is easy to maintain and modify. Maintainability: The module should be easy to maintain, especially when new modules are added that the guardian needs to check.
Investors	Safety: This should ensure that bitcoin investors assets are safe, and will increase in value. Quality: The new approach is improving bitcoin core's systems and investors will keep on funding.

SAAM Analysis Continued

- Attributes
  a. Performance
  b. Safety
  c. Testability
  d. Quality
  e. Maintainability

Attributes	First Approach: Combine P2P and Consensus	Second Approach: Create a new subsystem
Performance:	High: this approach guarantees that the performance will still be high since this new bigger subsystem already has its previous performances, and may be slightly different since they have been merged.	High: The performance of this new subsystem can improve upon the issues that stemmed from these two systems being separated. This new subsystem that will replace both of them will probably remove unnecessary dependencies and increase efficiency.
Safety: This approach needs to identify errors,	High: The safety from the previous subsystems will be carried over. This is a high level of safety for user's transactions so it is high.	Low/Moderate: The safety for the transactions of this new system is low since some things may have been overlooked in development, on the other one the safety carries over.
Maintainability: This enhancement should be easily maintained by developers	High: The maintainability is high since there is a lot of documentation of these two subsystems that we are merging. All the resources to maintain will not need to be overhauled.	Moderate: The maintainability for this completely redesigned subsystem may be slightly more difficult than previous. Since it has new functions and dependencies than the first approach it will be reasonably difficult to maintain at first.
Quality: This approach must work properly as bitcoin is advertised to investors and users.	High: The quality for combining the two subsystems will definitely be high since we will be merging the two subsystems that have been working perfectly for the system.	High: The quality of this new subsystem may vary. Some functions may encounter bugs or issues upon deployment. This is a lesser quality since the previous subsystem that is combined, the components have been endlessly maintained and tested unlike this one. The quality may still be high since it is removing many issues of the subsystems.
Testability: In this approach, every scenario should be easily testable.	Moderate: This approach is a large system since it is merging those two, but the previous testing code could be used for this approach, making the testability as good as it originally was. Maybe some new tests have to be created.	High: This approach has to test large systems of bitcoin core or at least adjust the testing since they will be merged. More resources will be needed since this will be a pretty large system, so it will be more difficult to test.

Effects of enhancement

- Performance
  - The proposed enhancement aims to improve Bitcoin's performance by combining the P2P networking and consensus validation subsystems, reducing the internal communication during transaction verification and improving speed and scalability for a large number of users.
- Scalability
  - To improve Bitcoin's scalability, the proposed enhancement aims to combine the P2P networking and consensus validation subsystems to reduce transaction propagation time and increase the rate of block processing, which will allow for a higher throughput of transactions and support more users.

Effects continued

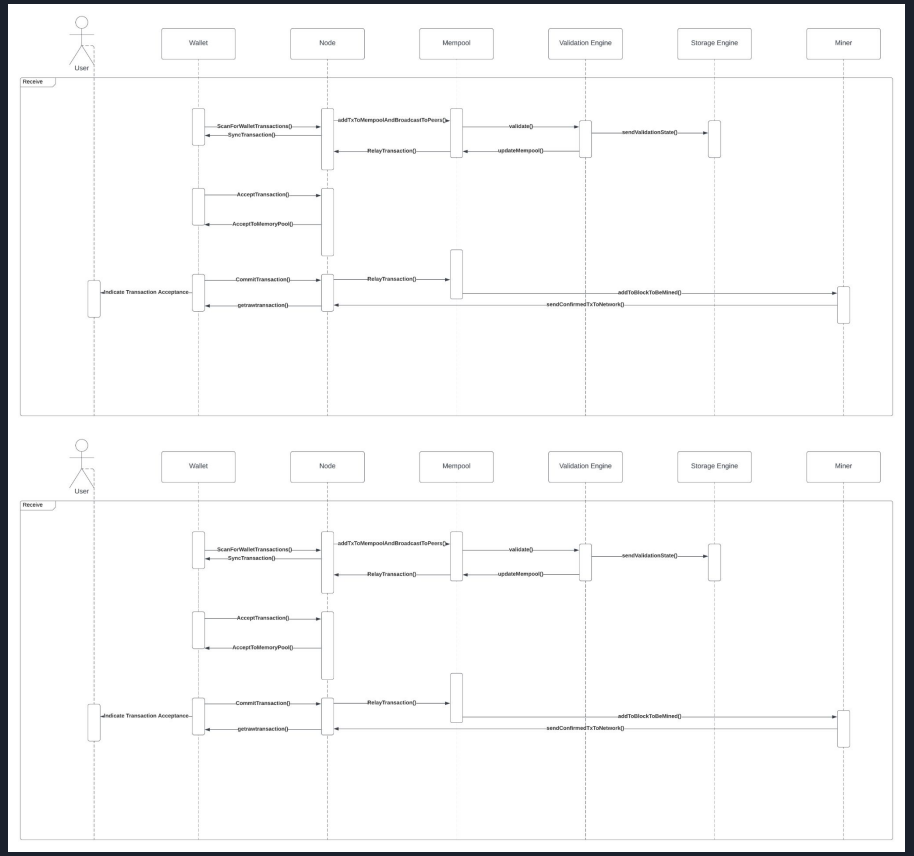
- Security
  - The proposed enhancement will increase Bitcoin's security by reducing transmission between components during the validation process, making it difficult for attackers to intercept and interfere with the network. The new validation at the networking layer will quickly detect and reject invalid transactions, reducing the risk of double-spending attacks.
- Maintainability
  - The proposed enhancement will simplify Bitcoin Core's codebase, making it easier to maintain, update, and document, and allowing for more modular development. A simpler codebase will also reduce the time and resources needed for testing and debugging, improving the software overall.

Use Case 1

Receiving money with wallet

The sequence diagram shows how a user receives money in their Bitcoin Wallet and includes multiple sub-objects such as Node, Mempool, Validation Engine, Storage Engine, and Miner to provide more detailed information on the process.

Before (top) and After (bottom)

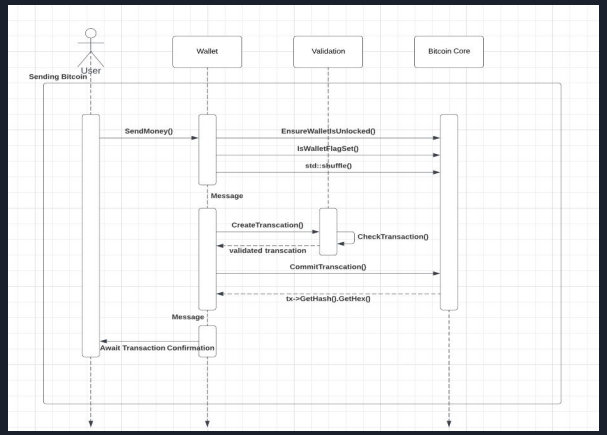


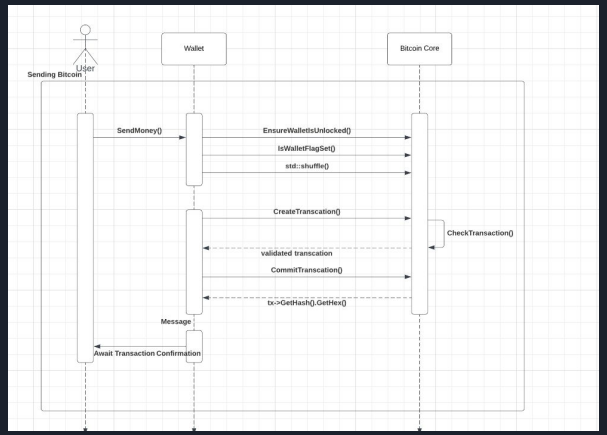
Use Case 2

Sending money with the wallet

The sequence diagram here shows receiving bitcoin from a user from before we added our enhancement onto it. We can see that this is a bit different from our original as we provided more detail to better depict what is happening with validation. This meant the inclusion of the Validation Engine object, where after the transaction is created it will be checked to see if it is a real transaction. After it will allow for the transaction to be committed.

Before (top) and After (bottom)





Testing

- Testnet will be used to evaluate the integration, performance, and security of combining P2P and consensus validation components.
- Small-scale testing in Testnet will be used to eliminate the possibility of data breaches and theft while simulating common blockchain network attacks.
- Logging and analytics of transactions will be conducted to assess the impact of the update, including transaction throughput and network latency.
- The primary goal is to evaluate whether the integration of components provides a more optimized use of bandwidth.

Potential Risks

- The proposed enhancements for Bitcoin Core platform's center pose risks, especially in transferring and refactoring prior dependencies. Testing can mitigate this risk, but edge cases remain problematic.
- The combination of subsystems could provide greater performance but also presents a risk of bottlenecking. A larger allocation of resources should compensate for this risk.
- The proposed enhancement raises concerns regarding availability and redundancy, as it would have a wider impact if it were to fail. Currently, the same issues exist with the two dedicated subsystems, but the enhancement increases the risk of issues if the combined subsystem were to fail.

Limitations and Lessons Learned

- Limitations:
  - Size of the source code
  - Not having much knowledge or experience performing improvements for large systems
  - Building sequence diagrams
  - Time since there are many things we could have looked at in the documentation/source code to come up with a more detailed enhancement.
- Lessons Learned:
  - The importance of teamwork and group collaboration
  - Organizing group meetings asynchronously is an asset
  - How great a resource source code documentation can be
  - It is difficult to come up with enhancement ideas on large scale professionally created systems