CISC 322 - A3 Report: Enhancements of Bitcoin Core

Group 27 - Based

April 2023

Amy Cui (19ayc1@queensu.ca)
David Courtis (20dhc@queensu.ca)
Jagrit Rai (19jr28@queensu.ca)
John Alajaji (18ja19@queensu.ca)
Logan Cantin (logan.cantin@queensu.ca)
Matthew Vandergrift (19mwv1@queensu.ca)

Abstract

In this report we will be discussing a possible enhancement which can be brought to bitcoin core. The enhancement we will be proposing is a way of tackling the deflationary problem present in the bitcoin system. This problem arises firstly due to the fact that once a bitcoin user loses their wallet then the corresponding unspent transactions are lost, and secondly since the total amount of minted bitcoin is capped at a fixed amount. This means that in the long run there will be a decreasing supply of bitcoin, which can lead to deflation and hence disincentivizing spending.

The way we plan on tackling this problem is in one of two ways. Firstly we propose automatically adjusting the value of the amount of bitcoin minted in each new block depending on the rate of transactions in a last cycle (automated adjustment), and secondly we propose using voting system which will allow nodes to vote on the value of the coinbase transaction in minted blocks (voting). For each of these proposed ideas we will be performing a SEI SAAM architectural analysis, which we will use to determine which method of implementing the enhancement is the overall best and hence the one we will be focusing on for the rest of the report.

The enhancement we ended up choosing is the automated adjustment enhancement, which we then used to update our conceptual architecture accordingly. We will then discuss the impacted directories and files of our enhancement, some ways of testing the impact of our enhancement on the bitcoin system, and finally consider the use cases of transaction monitoring/adjustment detection and currency adjustment. Finally, we will conclude with some lessons learned and final thoughts.

1 Introduction

Decentralized currencies such as Bitcoin have some unique properties that make them attractive alternatives to fiat (government-run) currencies such as increased anonymity and cryptographic proof. However, one distinct benefit of fiat currencies is that they can be manipulated by the government for economic reasons. For example, in a recession, governments can decrease interest rates to stimulate spending. As it currently stands, Bitcoin does not have a mechanism for adjusting the inflation rate. Additionally, there are only a fixed number of Bitcoin that will ever be released: 2041 is the year that the last Bitcoin will be created [1]. This has a deflationary effect, which will decrease the utility of Bitcoin as a currency, because it will prompt users to hold on to their coins instead of spending them.

In this paper, two methods for changing the inflation rate of Bitcoin will be explored: an automatic adjustment based on transaction volume, and a decentralized voting system. Both of these methods will grant the network some of the economic benefits of fiat currencies related to inflation rate adjustment, without the drawbacks like centralized control. For the purposes of this paper, when we discuss "changing the inflation rate", we mean to adjust the block reward for mining a new block.

We will then perform a SEI SAAM analysis for both proposed enhancements, which we will use to determine which is the overall best. We ended up choosing this to be the automated adjustment implementation of the enhancement. Using this chosen enhancement we then updated our conceptual architecture to be in line with it, as well as discussed directories and files impacted by these changes to the system. Finally we considered the use cases of use cases of transaction monitoring/adjustment detection and currency adjustment, and concluded with some lessons we learned over the course of this project as well as some final thoughts. Let us begin by providing some further motivation of the importance of our proposed enhancements.

2 Feature Motivation

The bitcoin system is designed so that over time the quantity of newly minted bitcoin decreases until a hard limit of 21 million is reached. [1]. This results in a gradual decrease towards zero in the amount of new bitcoin entering the system. As with any monetary system any addition of new currency (all else equal) will lead to inflation. Since bitcoin gradually adds less currency it's inflation rate will fall accordingly. The relationship between the creation of bitcoin and inflation can be seen in Figure 1.

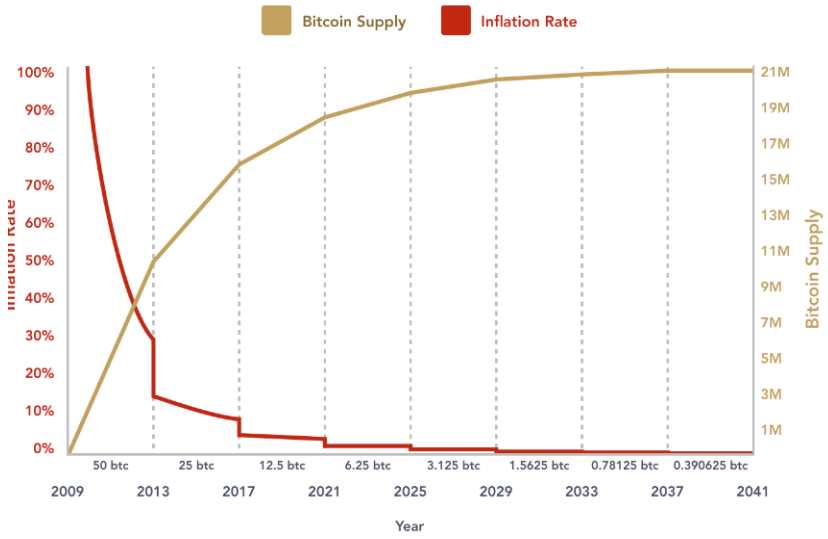


Figure 1: Bitcoin minting vs Inflation. Reference [2]

As this figure showcases, the inflation rate of bitcoin will fall to 0 by year 2041. The inflation rate as of 2023 has already plummeted to approximately 1.4% and will only continue to fall as the final 2 million bitcoin are mined. As a result of this lack of inflation, bitcoin will begin to deflate. Oppositely to inflation deflation occurs when currency is removed from a monetary system. The supply of currency in bitcoin will in practice fall due to currency becoming unavailable or 'lost'. This happens for a variety of factors such as lost wallets, death, or simply hoarding. Current estimates point to 10-20% of the bitcoin supply as currently lost [3], and this will only increase over time.

But why is deflation necessarily bad? Well, according to nobel prize winning economist Pual Krugman "deflation discourages borrowing and spending" [4]. In the case of bitcoin, this is in fact a very serious issue, since similarly as with normal currencies, the corresponding economic system can only be maintained as long as people are continuously spending. What is rather surprising however, is that no mechanism is present in the bitcoin system to handle this issue. This naturally motivates our proposed enhancement, which will allow the bitcoin system to manage its inflation rate and thus combat the deflationary problem. Let us now discuss in the following section, two ways in which this enhancement can be implemented.

3 Proposed Enhancement

Introducing an innovative proposal for an inflationary adjustment system, we aim to revolutionize Bitcoin's monetary policy by controlling the rate of inflation and removing the 21-million hard limit on its supply. This dynamic approach seeks to strike a balance between promoting a stable, usable currency and preserving Bitcoin's core principles of decentralization and security. By implementing a carefully designed algorithm, we can adapt the rate of Bitcoin issuance in response to network demands and economic conditions, ensuring the long-term sustainability and utility.

We will accomplish this using one of two proposals to modify Bitcoin's monetary policy and address its deflationary nature.

3.1 Enhancement I - Automated Adjustment

The first approach is to dynamically adjust the block reward based on the volume of transactions processed on the network. As transaction volume increases, block rewards decrease, and vice versa. This demand-driven approach ensures that miners' incentives align with network usage, promoting efficient resource allocation and creating a more stable environment for Bitcoin. It is important to also implement flexibility within the automated system for future open-source development, such as the fine-tuning of Bitcoin's issuance rate in response to economic conditions and network demands, fostering a more stable and usable digital currency while preserving the core tenets of decentralization and security. The increased reward

will be distributed to both miners and wallets that create the highest amount of transactions. The reward distributed between wallets will not be high enough to encourage spam spending, but enough to inflate the value of the currency (ie. the reward is always much less than the total cost for the transactions made). Deflationary decisions will decrease mining rewards only.

This approach necessarily introduces a new component, "Demand-Driven Reward Adjustment (DDRA)" responsible for the algorithmic system in controlling the rate of Bitcoin inflation. This component and its interactions can be seen in Figure 3, showing the outer connections with the concrete architecture of Bitcoin Core. This component interacts bidirectionally with the miner and p2p networking interface components in order to receive and propagate up-to-date transaction adjustment information throughout the blockchain. The crypto component may also be required in order to facilitate verifications to determine valid updates.

The 2nd level conceptual architecture of the DDRA component itself is shown in Figure 3 and is composed of "Transaction volume measurement", "Reward adjustment", "Monitoring and reporting", and "Governance and updates" subsystems. The DDRA component must be designed to function within the decentralized nature of the Bitcoin network, requiring a form of pre-acknowledged consensus to govern the operation.

3.1.1 Technical Implementation Details

There are two new data structures required for storing reward adjustment data. The first is within the governance and updates module for tracking algorithm version data in new implementations:

- Algorithm version: 4-byte integer representing the unique version id for the algorithm
- Locktime: a date attribute representing the earliest time that the algorithm takes effect
- Algorithm: a function taking in a set of parameters and producing a dictionary of reward values for miners, and wallets with some transaction volume threshold.

The second is in monitoring and reporting for storing updated coinbase reward. This is required for the API-RPC to make decisions on how to distribute rewards to miners and wallets:

- Miner reward: a 32-byte float for storing the reward given to miners in the coinbase transaction.
- Wallet reward: a 32-byte float for storing the reward ratio given to wallets in the coinbase transaction.
- Wallet threshold: a 32-byte float for storing the minimum threshold required to receive spending rewards in currency adjustment.

3.1.2 2nd Level Conceptual Architecture

There are four modules introduced for the automated adjustment process.

Transaction measurement : A method for accurately measuring the metrics of transactions processed within a specific time frame, such as a moving average or a sliding window, to determine the appropriate block reward. This measurement module will initially only measure transaction volume but will depend on the definition of the Governance and Updates module.

Reward adjustment : Defines the relationship between transaction measurement and block rewards, determining how the block reward should be adjusted based on the measured transaction metrics. This algorithm should be designed to provide a smooth, predictable adjustment in response to changes in transaction volume. It should also define limits on the minimum and maximum block rewards to ensure that the system remains within reasonable bounds, preventing extreme fluctuations in mining incentives that could destabilize the network.

Monitoring and reporting : Tools and mechanisms for monitoring the performance of the DDRA system, reporting on the impact of the adjustments, and providing visibility to network participants. This will provide insight to developers and analysts in determining the stability and trends of the network, as well as information to other components about the state of the rewards.

Governance and updates : A process for updating or refining the DDRA system's parameters and algorithms in response to network performance, technological advancements, or changes in the broader cryptocurrency ecosystem. This may involve community input, expert recommendations, or other forms of governance. This will directly impact, or log changes, to the other subcomponents within the DDRA system by implementing changes to the reward adjustment algorithm.

We then have the following dependencies within the DDRA component. The higher-level conceptual architecture is further detailed in section 5 and Figure 3.

Reward adjustment → Transaction measurement : The Reward adjustment component depends on the metrics attained by the transaction measurement component in order to calculate the updated reward.

Monitoring and reporting → Reward adjustment : The changes made by the reward adjustment mechanism is captured within the Monitoring and reporting component so that it is accessible to other components and agents.

*** → Governance and updates** : All components depend on the governance and updates component in order to incorporate unseen or future changes within the algorithmic calculations for transaction rewards.

Overall, this component relies on the object-oriented nature of the bitcoin core software in order to dynamically interact with neighbouring functionalities. By utilizing an automated system, we can eliminate user biases in reward calculation.

3.2 Enhancement II - Voting

Another way to implement the inflation rate adjustment would be a decentralized voting system. This system would allow nodes on the network to vote for a new inflation rate using computing power. This method has the advantage of allowing for more control over the inflation rate, without requiring a centralized authority.

The process for changing the inflation rate begins with the declaration of a vote. A voting period is a period of time in which the nodes on the network are allowed to submit ballots with proposed changes to the inflation rate of Bitcoin. When a voting period is declared, it is propagated across the network so that all nodes are aware of it. The voting period is two weeks long, starting from the time that the vote is declared. Each voting period is identified by a unique salt (a salt is a random number, which has uses that will be described below), which is randomly generated by the node which declared the voting period. It costs computational power to declare a vote.

The votes themselves are communicated via a new component called the "ballot". The ballot contains the newly proposed block reward, among other things (discussed in more detail below). The validity of votes are protected by proof of work, the same way that the validity of the blockchain is protected. In order to cast a valid vote, it costs a certain amount of time and computing power, which prevents a bad actor from flooding the network with votes. Each node on the network has the ability to generate as many ballots as it wants to. Each node is also responsible for listening for, keeping record of, and re-propagating all ballots it receives. Received ballots are only stored and repropagated if they are valid, which is to say that they: a) have the correct level of difficulty; b) have a salt that corresponds to the current voting period; c) has a unique id; and d) the proposed block reward is within a given range of the current block reward.

The ballot acts both like a transaction and a block. Proof of work is done on ballots, like blocks, to ensure their validity. Ballots are then propagated across the network like transactions, in the sense that they are stored in a transient mempool and propagated on a best-effort basis. Ballots are not put into blocks, and are never stored on the blockchain.

At the end of the voting period, each node will tally up the ballots it has a record of and calculate the newly voted-on block reward. From that point on, until the end of a new voting period, this block reward will be the only one that it accepts for newly minted blocks.

3.2.1 Technical Implementation Details

There are two new data structures that are introduced for voting purposes, the ballot and the voting period declaration. The ballot is a data structure that represents the vote of a node, and it contains the following information:

- Voting Period Salt: this 4 byte integer stores the salt of the voting period.
- Id: this 4 byte integer uniquely identifies the ballot.
- Proposed Block Reward: this 8 byte integer represents the number of Satoshis that this node proposes should be awarded for mining a new block.
- Nonce: 4 byte integer that doesn't have a meaning other than to influence the hash of the data structure so that it satisfies the difficulty rating.

The voting period declaration is a data structure that represents the beginning of a voting period. It contains the following information:

- Salt: 4 byte integer representing the salt for this voting period
- Voting Period Start Date: 4 byte Unix timestamp that represents the starting time of the voting period
- Nonce: 4 byte integer that doesn't have a meaning other than to influence the hash of the data structure so that it satisfies the difficulty rating.

3.2.2 New conceptual architecture

There are a few new components that are required:

- **Ballot mempool**: stores the ballots of the current voting period. This mempool acts in the same way as the transaction mempool.
- **Ballot generator**: generates new ballots.
- **Scrutineer**: Counts the ballots and determines the new block reward rate.

The newly required dependencies are:

- **Validation engine** → **Scrutineer**: The validation engine is responsible for ensuring the validity of ballots, and it needs to ask the scrutineer what the current block reward is.
- **Scrutineer** → **Ballot Mempool**: The scrutineer needs access to the ballot mempool to determine the results of the vote.
- **Ballot Mempool** → **Validation Engine**: The ballot mempool relies on the validation engine to determine if a ballot is valid or not.
- **Ballot Mempool** ⇌ **P2P Networking Interface**: The P2P networking interface forwards all ballots to the ballot mempool. Sometimes, the P2P network interface will receive a request to get an inventory of some of the ballots in the mempool for populating a neighbour's mempool.
- **Ballot Mempool** → **Ballot Generator**: The ballot mempool collects all of the ballots generated by the ballot generator
- **Ballot Generator** → **UI**: UI allows the user to control how much computational resources should be allocated to voting.
- **Ballot Generator** → **P2P Networking Interface**: Ballot generator submits its ballots to the network for propagation.

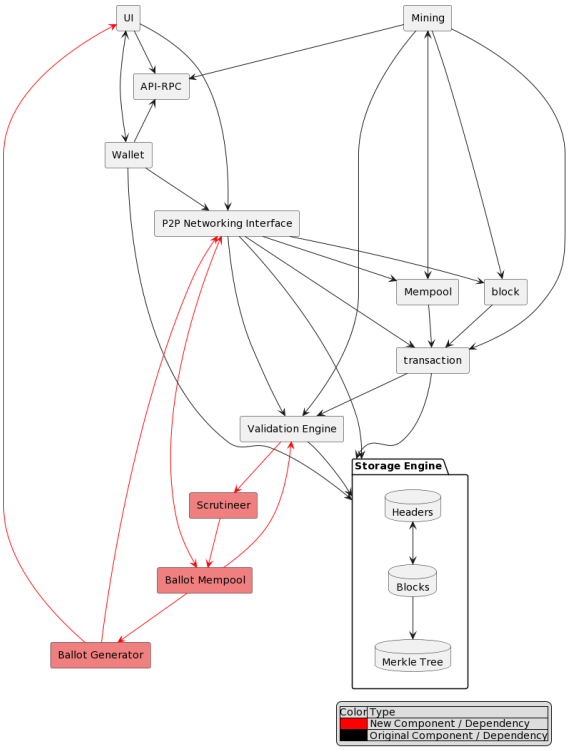


Figure 2: New conceptual architecture for Enhancement II

3.2.3 Discussion

This section will attempt to address some of the potential oppositions to this new enhancement.

A key point that is important to all stakeholders is that that the voting system remains democratic. In other words, that it is infeasible for a small number of people to overpower the will of the network. This is ensured using proof of work, the same mechanism used to protect the blockchain from manipulation. It costs computational power to cast a ballot, and therefore the number of votes a person can cast is proportional to the their computational power. This ensures that the process remains democratic.

Now, voting is not the main purpose of the network, and as such we would like to avoid diverting too much computational power from mining to voting. We want to select a voting difficulty that is relatively easy, but not trivially easy. Therefore, it would make sense to make the required difficulty be an order of magnitude or two less than the required difficulty of mining a block. This makes voting accessible to the average user without diverting too much power from mining, but also is hard enough that it preserves the integrity of the system. Since the difficulty of mining a block changes on a bi-weekly basis depending on the computational power of the network, and the ballot difficulty is relative to the mining difficulty, this system also ensures that the ballot difficulty changes to represent changes in the computational power of the network.

Declaring a vote also requires proof of work. This is to make voting expensive enough that it is only initiated when the network is in dire need of a change of block rate. We propose making it 2-3 orders of magnitude harder than the mining difficulty. This keeps the rate relatively stable by gently disincentivizing starting a voting period.

Also related to keeping the rate stable, the proposed block reward on a ballot must be within a certain range of the current block reward in order for a node to accept the ballot. This ensures the rate does not change too drastically, say a 200% increase, in one voting period. The actual range should be decided by an economist, but an example range might be $\pm$ 5%.

The proposed block reward on the ballot is an 8-byte unsigned integer, which represents the number of satoshis that are earned per new block. The reason that an integer is used, instead of a floating point number representing the number of bitcoin that are earned for a new block, is because of underflow in floating point numbers. If the block reward ever becomes very small or very large, it is conceivable that this limitation would negatively influence the system by slowly corrupting values and losing information.

One of the biggest attack vectors arises from ballot pre-calculation or re-use. This would allow an adversary to subvert the democratic process by supplying ballots that didn't actually require them to do any work. This system accounts for these cases. Firstly, each vote has a unique salt which prevents ballot re-use between voting periods. In cryptography, a salt is random data that is used as an additional input to a hash function. If the salt of an old ballot was changed to match the salt of the new voting period, the hash would not meet the required difficulty and would not be accepted by nodes on the network. This protects against ballot re-use between voting periods. Ballot pre-calculation is also protected against, by the fact that the salt of voting periods is declared at the time of the vote.

Ballot re-use within a voting period is also protected against by the unique ID of the ballot. Ballot mempools won't store multiple ballots with the same id, which ensures that proof of work will have to be done on each ballot individually.

4 SAAM Analysis

4.1 Impacts on Stakeholders

Proposed enhancements to Bitcoin Core have a significant impact on all stakeholders involved. The four main stakeholders are developers, merchants, miners, and users. Developers can be both open-source developers and the Bitcoin Core development team. Merchants are any businesses that accept Bitcoin as a form of payment for their goods or services. Miners are responsible for validating transactions and adding them to the block chain. Users are individuals or businesses who use Bitcoin as a means of payment or store of value.

Stakeholders	Most Important NFRs regarding the enhancement
Developers Responsible for designing and implementing the adjustment system.	*Maintainability*: The system must be easy to maintain, especially during high periods of activity (high transaction period and voting period). *Performance*: The new enhancements should bring better performance in most cases. *Scalability*: The system should have high scalability to handle increases in transaction volume based on enhancement. *Security*: The enhancement should not introduce new security vulnerabilities.
Merchants Ensure that transaction fees are kept low and predictable and that the currency is stable. This makes Bitcoin more attractive to used as a payment method with directly impacts Merchants.	*Reliability*: Want to ensure that the system is reliable and stable after enhancements are added. *Security*: Enhancement introduces higher chance of manipulation which can compromise the trust and reliability of Bitcoin as a form of payment. *Usability*: Enhancement should not introduce any usability issues that would make it difficult to use Bitcoin as a payment method.
Miners Miners are responsible for processing transactions and validating votes.	*Performance*: Enhancements should not negatively impact ability to earn block rewards *Reliability*: Enhancements should not increase the risk of a fork in the network *Scalability*: Enhancement should be able to handle increases in amount of mining at one time.
Users A automated adjustment system would make the network more efficient while a voting system would allow users to have a say in the direction of the Bitcoin Core Network.	*Latency*: System should be optimized to minimize any delays in transaction processing. *Reliability*: Enhancement should be reliable and stable during high periods of use. *Security*: Enhancement should not introduce new attack opportunities to users. *Usability*: There should not be an increase in difficulty for usability of Bitcoin Core due to added enhancement.

Table 1: Stakeholders impacted by the enhancements

4.2 Non-functional Requirement Analysis

Non-functional requirements (NFRs) are critical in Bitcoin Core as they define the characteristics that are necessary for the system's overall performance, reliability, and security. Below are the important NFRs regarding both enhancements suggested. A high rating indicates good compatibility with the NFR, while a low rating indicates that such NFR will be difficult or problematic to implement.

NFRs	Enhancement I	Enhancement II
Latency The time delay between initiating a request and receiving a responce.	*Low*: Latency influences the time it takes for the system to process and adjust the block rewards. A low latency system would be able to adjust the block rewards quicker. This is difficult to achieve as adjustments must be made in real-time, and concurrently between clients to ensure consensus with minimal wasted communication.	*Medium*: The systems voting latency must be optimized to ensure that users receive prompt feedback on the status of their vote and that voting results are calculated quickly. Adjustments must also be made in real-time, but are made considerably less often.
Maintainability The ability of a software system to be easily maintained and modified over time.	*High*: The algorithmic load is low and remains stable over time, as such, no maintenance is required as unexpected situations rarely arise and the system is entirely autonomous and self-preserved.	*Medium*: The voting system must perform well, ensuring that it can handle a high volume of votes without slowing down or crashing. Maintenance of the voting system is fundamental to ensuring a stable function.
Performance How well a system performs in terms of speed, responsiveness, and resource utilization.	*High*: Performance will not be a concern in the automated adjustment system, as the relative computational load is very meagre, and no cyclic network interaction is required.	*Low*: The system will need to be optimized to handle the increased workload efficiently during voting periods, this will directly impact the network congestion and affect the load of the p2p networking component.
Reliability The ability of a system to perform its intended function without failure, errors or breakdowns.	*High*: The system must be reliable the ensure that the block rewards remain stable which is essential for maintaining the incentive structure of the network	*High*: The voting must be available via multiple channels to ensure high availability to ensure that it operates consistently and accurately over time
Security Ensures that the system is secure and protected against cyber threats.	*High*: The automated adjustment system is resilient to attacks as it supports a decentralized and bias-free method of adjusting protocol, and does not accept user input. This makes the system highly secure from attacks.	*Low*: The voting system could potentially introduce a new attack vector. The voting system must be secure so that votes are not tampered with or manipulated, and that there is a low incentive for collusion. The possibility of collusion is a difficult consideration, as a few large mining pools hold much power over the network.
Scalability The ability of a system to handle increasing amounts of work or data in a responsive and efficient manner.	*High*: All clients connected and making transactions generally participate in the calculation of these metrics, thereby, the system must be able to handle a large amount of data and ensure that the data meets consensus and is valid; however, since each client is capable of making independent decisions, scalability is uncomplicated to implement.	*Medium*: If the number of voters grows significantly then the system must be able to handle a large number of votes while maintaining performance and security. Each client must implement an additional layer of communication to achieve this.
Usability The ease with which users can interact with the system to achieve goals efficiently and with satisfaction.	*High*: The dynamic reward adjustment system is only modifiable to developers, thereby, will not require general user interaction. This indicates that the system may not require much user interaction support.	*Low*: The voting system must be user-friendly, ensuring that it is easy to use and understand for all Bitcoin users that wish to vote with minimal user error. This includes the general public.
Robustness The ability of a system to continue operations while handling errors and exceptions without crashing.	*Low*: An automated system will be relatively rigid in decision-making compared to human analysis and voting, thereby, is not as robust to adverse conditions without human updates. The governance and updates module does allow modification to the algorithm but should be rarely updated.	*High*: Adverse conditions can be dynamically handled through the direct decisions of human agents, thereby, allowing the system to adapt to complex scenarios relatively fast and effectively.

Table 2: Important NFRs regarding the enhancement

4.3 Potential Risks

Implementing any enhancement to Bitcoin core may introduce new risks and attack vectors that must be carefully considered and addressed to ensure the stability and security of the network. It is essential to carefully consider and mitigate the potential risks and attack vectors that may be introduced.

4.3.1 Enhancement I - automated adjustment

Some risks involved in implementing this enhancement are latency and robustness. A low latency system may require significant resources to achieve which could increase the cost of operating and maintaining the system. Due to the nature of this enhancement, adjustments must be made in real time between clients to ensure consensus with minimal wasted communication. If the system is unable to adjust block rewards quick enough then it could result in miners leaving the network. Conversely, if adjustments are made too quickly, it could result in inconsistency in the network. Achieving consensus in a distributed system like Bitcoin Core can be difficult, particularly when adjustments must be made in real-time and concurrently between clients.

In terms of robustness, one major risk is that the system may not be as robust to adverse conditions without human intervention. This is because the system's decision-making is based on predetermined rules and algorithms, which may not be able to adapt to unforeseen circumstances. Another risk is that the algorithm should rarely be updated. This can create issues if updates are needed due to changing circumstances, as it may take a long time to make the necessary modifications.

There is the potential risk of inflationary pressure on the currency however, this is solved already in our design of the system as the reward for mining is dished out to wallets that spend the most but not enough to overcome the cost of transactions fees.

Some new attack vectors are Sybil attacks, spam attacks, double-spending attacks. Sybil and spam attacks involve the attacker flooding the network with a high volume of transactions (through false identity and multiple low-value transactions) to inflate the block rewards. Attackers could also double-spend coins by exploiting the automated adjustment system to create multiple blocks within conflicting transactions.

4.3.2 Enhancement II - Voting

Since the decentralized voting process calls for proof of work on ballots, this version of the alternative has a layer of protection from Sybil attacks. However, one of the potential risks associated with implementing the proposed enhancement is that it increases the incentive of a 51% attack on the network. This is because the voting power in the system is directly proportional to the amount of compute contributed by the miners. If a user or a group of users controls the majority of the network's compute power, they would also have the majority of the voting power. This would allow them to unilaterally decide on the inflation rate, which could be detrimental to the entire network.

A voting system also opens the network up to the risk of a secession in the form of a fork, if different groups of nodes disagree on the outcome of a vote. Such a disagreement could lead to different groups choosing to adopt the fork with their preferred block reward, leading to a split in the network and a loss of population.

One other potential issue with a voting system that is propogated across the network after being initiated is coordination– It may be possible for nodes that are too great a degree away from the original initiating node to "miss" the voting period, thus causing users with few neighbours to have a higher chance of being left out of the vote and have no say in the vote, despite having the ability to contribute a great amount of compute power to the network.

Finally, a major risk of a proof-of-work based voting system is gridlock in the system. During a contentious voting period, the network may focus on generating many ballots. If nodes decide to spend the majority of their compute power on generating ballots, there will be a lack of work being done on mining. This means that there will be fewer blocks being minted during a voting period. To balance this, we considered lowering the variable difficulty for mining during voting periods to allow for the rate blocks being minted to stay consistent while nodes are dedicating compute power to generating ballots. However, this does not solve the problem, as during a particularly contentious vote, nodes may choose to *only* generate ballots and forego mining during the voting period all together, which can lead to serious negative implications for the network.

4.4 Chosen Enhancement

The proposed enhancement for the Bitcoin core is the implementation of an automated adjustment system. Among the Non-Functional Requirements (NFRs) listed in Table 2, the most important ones are maintainability, performance, reliability, security, scalability, and usability, all of which have a high degree of compatibility with the chosen enhancement.

Compared to the voting enhancement, the automated adjustment system offers a significant advantage in terms of autonomy and self-preservation. The system operates with a low algorithmic load and does not require any maintenance of the voting system, ensuring stable functionality.

While both enhancements address the NFRs, there are differences in their approaches. Notably, the automated adjustment system does not require cyclic network interaction, making performance a non-issue. The system is also more resilient

to attacks since it does not accept user input and provides a decentralized and bias-free method of adjusting the protocol. There is also a scalability advantage to an automated adjustment system as there is no live propagation of a large load that is not usual for the network like in the voting system. The system must be able to handle a large amount of data and ensure that the data meets consensus and is valid; however, since each client is capable of making independent decisions. Thus, scalability is uncomplicated to implement in comparison to the voting system.

On the other hand, the voting system introduced in Enhancement II requires optimization to handle the increased workload during voting periods, potentially introducing new attack vectors through vote tampering and manipulation. Furthermore, the usability of the automated adjustment system is limited to developers, while voting must be user-friendly for both users and the general public.

The automated adjustment system however does have two major flaws. The first being the need to make adjustments frequently to achieve low latency. Enhancement 2 does need adjustments in real time too however, it is considerably less in comparison to Enhancement I. The second flaw being the robustness of an automated system. As the voting system is reliant on human agents, all adverse conditions can be dynamically handled by humans. On the other hand, the automated system is not as robust to adverse conditions. The governance and updates module does allow modification to the algorithm but should be rarely updated.

In conclusion, the automated adjustment system offers a more robust solution to Bitcoin's core NFRs, ensuring better maintainability, reliability, and security without compromising performance. It is a viable enhancement that preserves the core tenets of decentralization while offering a more stable and usable digital currency.

5 Updated Architecture

In this section we will discuss how we updated our conceptual architecture to account for our chosen enhancement (automated adjustment). We will begin by presenting an updated conceptual architecture diagram. Note that the subsystems which make up the new component are highlighted instead of being abstracted since they are focus of this report.

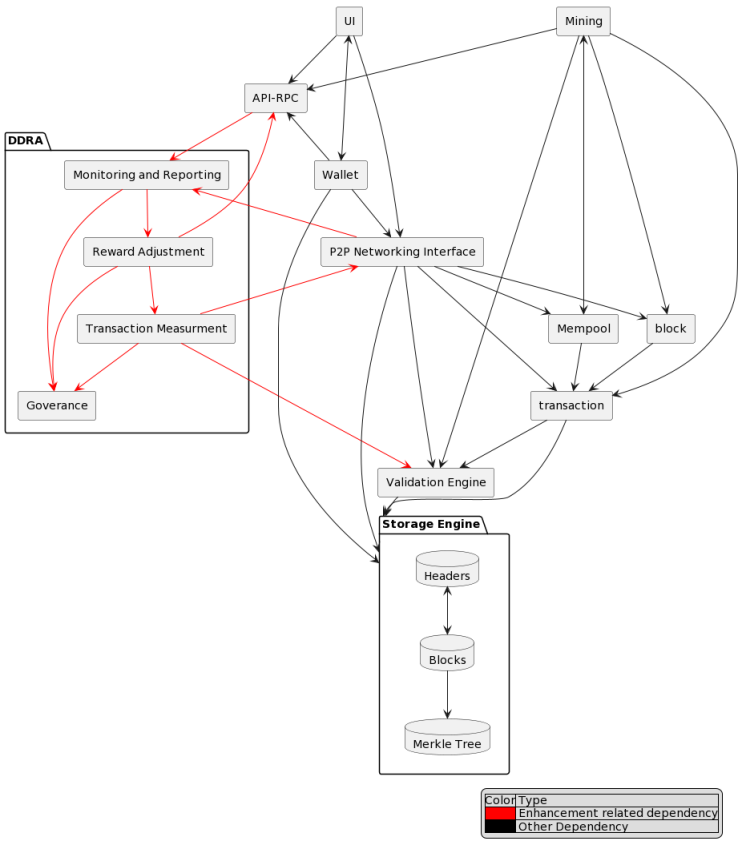


Figure 3: Updated Conceptual Architecture

The "Demand-Driven Reward Adjustment (DDRA)" and it's subcomponents are expanded upon in section (3). We now will expand upon the new dependencies present in our system:

API-RPC : This module usually handles the reward information and instructions for coinbase transactions. This will need to be modified to support coinbase automated adjustment to miners and wallets.

Transaction measurement → P2P Networking Interface : The P2P Networking Interface component is responsible for managing connections and communication between nodes in the decentralized Bitcoin network. Transaction measurement depends on this component to access up-to-date and accurate information about transaction volume. By leveraging the P2P Networking Interface component, the DDRA system can collect data on transactions being propagated and verified across the network, ensuring that block reward adjustments are based on a comprehensive view of network activity.

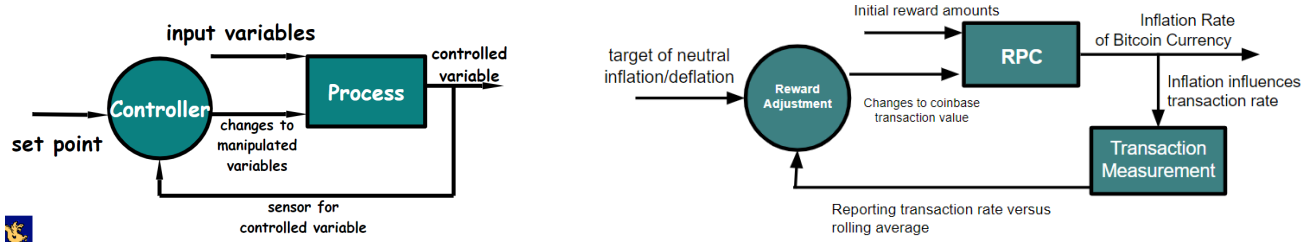
Transaction measurement → Validation Engine : Transaction measurement may depend on the Validation Engine component to verify the authenticity and validity of transactions, ensuring that only legitimate transactions are considered in the measurement process.

P2P Networking Interface → Monitoring and reporting : This component might depend on Monitoring and Reporting to stay informed about the reward rate's state, which can be useful for nodes to understand the network's overall status and make informed decisions about resource allocation, transaction fees, or even mining participation.

API/RPC → Monitoring and reporting: As discussed previously, the method by which the automated adjustment method handles managing the interest rate is by increasing or decreasing the value of the coinbase transactions in freshly minted blocks. That is, to inflate the currency additional the value of the coinbase transaction is raised, and similarly decreased to deflate the currency. Note that during the mining process, miners utilize the getBlockTemplate rpc (remote-procedure call) to obtain relevant information needed to construct a block. One such piece of information provided is the information necessary to construct the coinbase transaction. Therefore, the API/RPC component will depend on the Monitoring and reporting subsystem of DDRA, since the Monitoring and reporting subsystem holds the current coinbase transaction value.

Reward Adjustment → API-RPC: Within the current system of bitcoin core the reward for minting a new block comes from the value of a coin-base transaction. This means that to adjust the reward, the value of a coin-base transaction will need to be modified. The value of the coin base transaction is provided by API-RPC, through the getBlockTemplate. This means that to adjust the value of the coinbase, the reward adjustment component must make a call to this file within the API-RPC component.

Note that the overall peer to peer architecture style of the bitcoin core system remains constant, along with the node specific object oriented programming style. This enchantment would introduce a feedback control system architectural style to bitcoin core. This sub-style is centered around the new and modified components within the wider system. Figure (4) demonstartes the similarities between the feedback control system seen in the week 3 course content, and the interaction of modified sub-systems.



(a) Feedback Control System from Week 3 Slides

(b) Architecture of Enhancement

Figure 4: Comparing enhancement to Feedback Control

These similarites make sense given the intended point of feedback control systems, which is to continously adjust real time systems to ensure some target state. In this case the target state is one where inflation and deflation is not a problem for the currency. This leads naturally to the set point of 0% inflation. The point of the enhancement is to control deflation (negative inflation) in the system. The minting of new currency is what causes inflation and hence the value of coin base transactions is the manipulation variable for controlling inflation. One immediate difference between the enhancement and a regular feedback control system is that the sensor does not monitor the control variable directly. The sensor instead monitors the transaction rate. We know by the basic economic theory discussed in section (2) that inflation and deflation will lead to a change in transaction rate. We hence treat transaction rate as proxy for inflation or deflation. This is what is monitored to guide future decisions. Another difference is that the sensor in bitcoin core is itself a component called transaction management. This component is discussed in the proposal section (3.1) Based on the transaction rate the

reward adjustment component will adjust the manipulation variable which is quintessentially the role of a controller. The RPC then is what puts the coinbase transaction value into place and is hence the process which will impact the whole of bitcoin core. It is clear from this description that feedback control is a sub-architectural style of the enhanced bitcoin core system.

6 Impacted directories and files

- Modification of mining.cpp residing inside the rpc folder. As mentioned previously getBlockTemplate() gets the value of the coin-base transaction. Since the enhancement alters the value of the coin-base transaction this file will either need to make function calls to some other file to determine the value or will need to have it's own internal logic modified.
- Modification of policy.h file within the policy folder in the validation engine. Transaction measurement depends on the validation engine as discussed in the previous section. We want to only count valid transactions, hence a function to quickly determine if a transaction should be counted or not should be added. This is for performance and functionality purposes.
- The consensus folder within the validation engine would be impacted by the need for a defined formula for calculating reward change based on transaction rate. This would need to be a function with parameters based on transaction rate hence adding a function call to the transaction management component.

7 Testing enhancement's impact on system

Our enhancement centers around controlling the system to ensure it stays in an ideal place regarding inflation rate. Therefore the method for testing it would be to simulate specific environments to determine how the enhancement would begin to interact with other components. A test environment for bitcoin core would entail setting up a small variety of nodes and imposing some condition on a block chain which they would all adopt, in essence a private fork of the real block-chain which would be dropped after testing. The agreed upon test chain could be either have specific features for testing, or could be manipulated by the tester who owns the nodes following it for testing purposes. We now outline three test environments.

7.0.1 Transaction shortage

In this scenario, the amount of transactions being done would plummet drastically. This could simply be accomplished by running a test system normally (i.e nodes doing random transactions which each other to model human behaviour) and then at a moment lowering each node's transaction rate by 90%. The enhancement should react by dramatically increasing the reward for mining. The first test would be to check if each node is able to record the same drop in transaction rate as was induced in the network. This ensures that the transaction measurement component can properly interface with the network. Then the getBlockTemplate data from the nodes API-RPC component could be examined to determine if the reward adjustment component is attempting to make the desired increase in reward. Finally other nodes on the test network could begin to mine and determine if increased rewards are being given. This would ensure that changes to the reward rate are being consistently propagated throughout the network.

7.0.2 Transaction Storm

The converse to a shortage of transactions would be a large increase in the transaction rate over a short period of time. Creating such a test environment up would follow similar methodology however instead of decreasing the transaction rate for all nodes it would increase. Similar tests could be performed for network propagation and transaction measurement. Additionally it would be helpful to test the degree of impact on nodes conducting large amount of mining. More transactions imply more transaction fees for miners, however the enhancement lowers the reward for mining. Determining if there is a point of equilibrium and how far the enhancement is from it could provide interesting test data for calibrating the particular adjustment rates before deploying such a change to bitcoin core.

7.0.3 Idle System

It would be possible to record bitcoin over two randomly chosen months since all transactions are public and to recreate the system on a small scale. This would entail programming nodes to follow simple rules about when to make transactions and with whom based on observed data from the chosen window. Such programming would create a system which left running on the currently implemented version of bitcoin core would mimic reality. This behaviour could be mapped to a test network running with the automated adjustment enhancement version of bitcoin core. The main question here would be to test how distributive to the network the proposed enhancement would be. If the behaviour of this test system began

to deviate significantly from the real world sample this would raise concerns and would need to be further examined. The enhancement should improve the system, however it is not intended to completely overhaul how most nodes function most of the time. It should impact the system to large degrees in the aforementioned extreme circumstances, but not during average operation.

8 Use Cases

There are exactly two notable use cases within our diagram, monitoring transactions and detecting the need for currency adjustment based on network metrics, and the adjustment of the currency itself through inflationary or deflationary tactics. Although the interaction between these two cases seems linear, there are deep intricacies to be considered in both.

8.1 Transaction Monitoring and Adjustment Detection

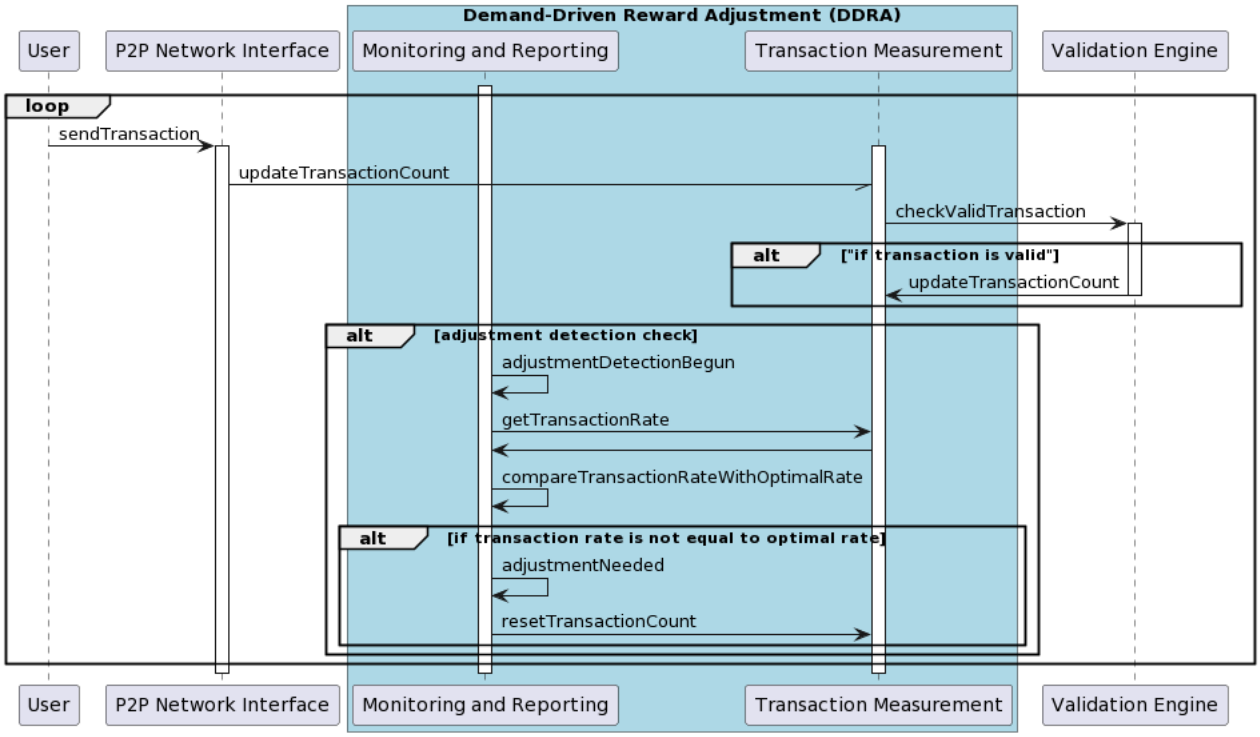


Figure 5: Sequence diagram for Transaction Monitoring and Adjustment Detection

Let us begin by exploring the use-case of the system monitoring transactions and detecting when an adjustment to the value of the coinbase transaction is required. Note that the period for detecting an adjustment will only occur every 2 weeks and will be initiated by the Monitoring and Reporting subsystem. It will utilize the transaction count maintained by the Transaction Measurement subsystem to compute the transaction rate over the last 2 weeks. If the transaction rate is not equal to the optimal transaction rate then an adjustment will have been detected. Once the detection check is complete (whether an adjustment is necessary or not) the transaction count held by the Transaction Measurement component is then reset, so that it can start keeping track of the number of transactions made over the next 2 weeks.

However note that in order for adjustment detection to be performed it is of course key that a transaction count is always maintained. To accomplish this, whenever a user sends a transaction via the P2P Network Interface, a call to update the transaction count will immediately be done asynchronously to the Transaction Measurement component. The latter will then verify that the transaction is valid using the validation engine, and then will update the transaction count accordingly.

8.2 Currency Adjustment

The currency adjustment system will Utilize the DDRA component's reward adjustment, and monitoring and reporting sub-components to calculate and propagate transaction reward updates throughout the local hardware and the network.

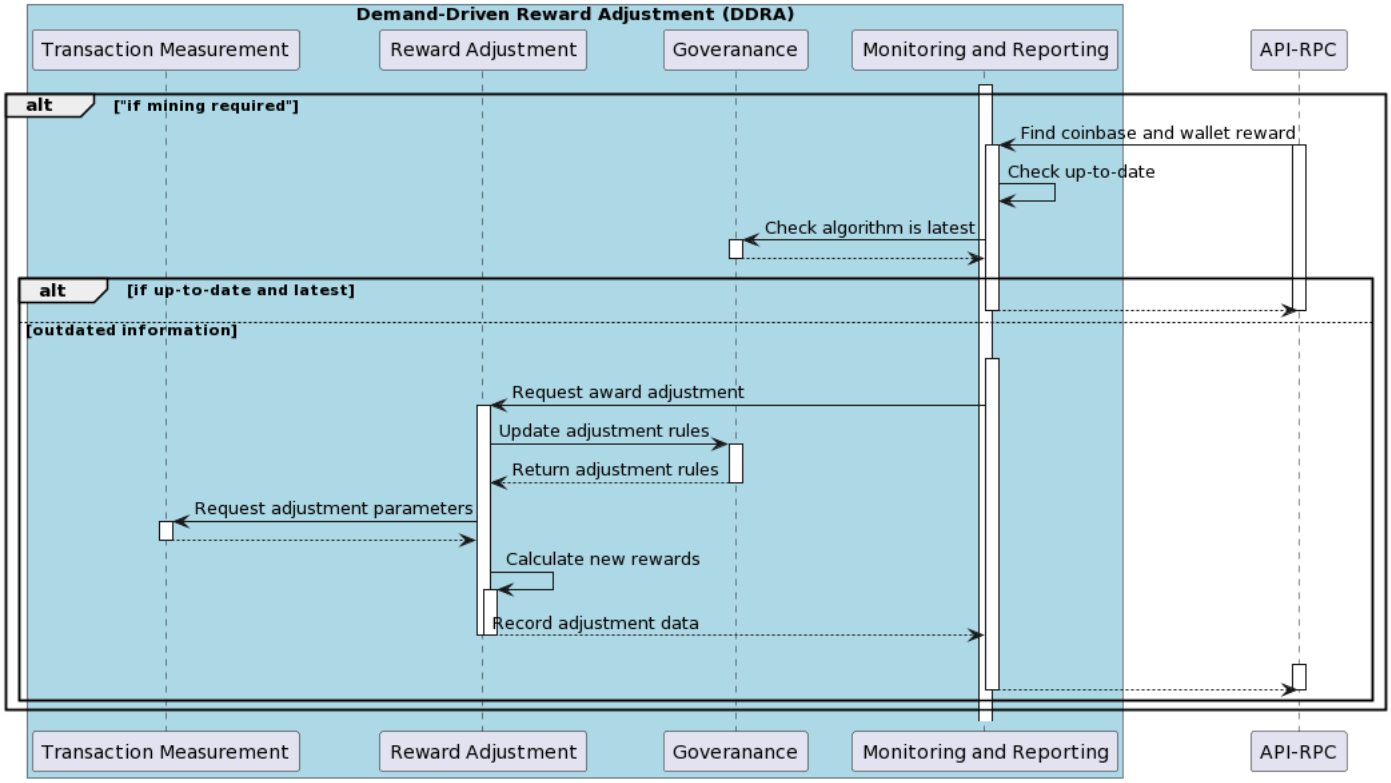


Figure 6: Sequence diagram for Adjustment Execution and Updating Reward Data

API-RPC is the source of coin base reward data for the mining program, thereby, will require update checks to the monitoring and reporting system to ensure up-to-date coinbase reward information. The monitoring and reporting module seeks consensus with the governance module to determine if the algorithm used to calculate the reward is outdated, and with itself to determine if the reward information is out-of-date. The frequency of outdating such data can be configured as desired through economist/developer preferences in each official update.

The API-RPC will receive information instantly if the reward is up-to-date. If not, the monitoring and reporting module will call the reward adjustment module to generate new reward values. The reward adjustment module will ensure its algorithm is up-to-date before requesting the required parameters from the transaction measurement module. After retrieving the required data, reward values are calculated and returned to the monitoring and reporting module for further propagation. The monitoring and reporting module will supply the API-RPC request with this updated information. Note that this call is client-side only and does not require heavy computation, thereby, is accomplished very quickly.

9 Lessons Learned and Limitations

9.1 Lessons Learned

In this report, we were able to apply lessons we learned from previous assignments in order to divide work up between the group members and efficiently complete this assignment. This included having frequent meetings to discuss topics that were necessary for the entire group to understand.

From this report, we were able to learn the importance of interdisciplinary skills amongst computer scientists. Our enhancement required knowledge of economics relating to inflation, and even political knowledge relating to the value of a democratic voting system versus a meritocracy. Since none of our team members had specialized knowledge of these non-technical realms, we found it important to have a team that is versatile in their knowledge base.

9.2 Limitations

One significant limitation was that we had expertise in computer science, but were lacking knowledge of fields like economics. As a result, deciding on certain related metrics, such as an “ideal inflation rate” to be maintained by the system, was difficult. Although we attempted to take into account economic factors involved, such as the incentive structure for Bitcoin miners, we recognized that our understanding of economics is limited. A more in-depth analysis of the specific economic implications of our enhancement would be necessary to ensure we maintain non functional requirements such as usability and scaleability.

Another limitation we found was that testing is difficult as it is not possible to test Bitcoin Core as a standalone system as the behavior of the network depends on the nodes that are currently active on the network– thus, the nature of the system means that many of the impacts we can consider are largely theoretical. Additionally, there are a vast number of variables that can affect the behavior of the system, making it difficult to predict how the network will react under different conditions. Furthermore, the needs and behaviors of the nodes on the network are continuously changing, and it is impossible to anticipate all possible scenarios.

10 Conclusions

In conclusion, the proposed enhancement to the Bitcoin Core Network aims to introduce an automated inflation adjustment system to combat deflation and ensure the long-term utility of Bitcoin. The enhancement will have a significant impact on the system's maintainability, scalability, performance, reliability, security, and usability.

The most important non-functional requirements (NFRs) for the system include maintainability, performance, scalability, reliability, security, and usability. Each stakeholder in the system, including developers, merchants, miners, and users, has specific NFRs that are crucial to their operations and concerns. Developers are concerned with maintainability, performance, scalability, and security. Merchants are concerned with reliability, security, and usability. Miners are concerned with performance, reliability, and scalability. Users are concerned with latency, reliability, security, and usability.

However, there are some risks associated with the proposed enhancement that should be considered. The implementation of the Bitcoin Core Network's enhancement to combat deflation poses risks in terms of latency and robustness. Achieving consensus in a distributed system like Bitcoin Core can be challenging, especially when adjustments must be made in real-time and concurrently between clients. The system's decision-making is based on predetermined rules and algorithms, which may not be able to adapt to unforeseen circumstances. Updating the algorithm may also take a long time to make the necessary modifications. However, we believe that with careful planning and implementation, these risks can be mitigated and the Bitcoin Core network can be strengthened for the future.

References

1. Writing team, River Financial. "Can Bitcoin's Hard Cap of 21 Million Be Changed?: River Learn - Bitcoin Basics." River Financial, https://river.com/learn/can-bitcoins-hard-cap-of-21-million-be-changed.

2. Writing team, River Financial. "Who Creates New Bitcoin?: River Learn - Bitcoin Basics." River Financial, https://river.com/learn/who-creates-new-bitcoin/.

3. Hendy, James. "How Much Bitcoin Is Lost Forever?" HedgewithCrypto, 3 Apr. 2023, https://www.hedgewithcrypto.com/how-much-bitcoin-is-lost/#:%20:text=to.

4. Krugman, Paul. "Fear of a Quagmire?" The New York Times, 24 May 2003, https://www.nytimes.com/2003/05/24/opinion/fear-of-a-quagmire.html.

5. Week Three Lecture Slides, Bram Adams 2023.