
Computer Science

Capstone Report - Spring 2021

# An Algorithmic Stablecoin on the Internet Computer

*Oli Chen,*
*Gina Joerger*

Supervised by
Luyao Zhang, Oliver-Gilles Marin, & Yulin Liu

**Abstract**

*As the distributed ledger technology industry grows, there is an increasing demand for fast, open, and secure blockchains that can support a diverse array of protocols. However, current impediments such as long finality time, low scalability, and low throughput have become great inhibitors. Furthermore, Bitcoin in its present state is simply too volatile to be used as a secure financial asset; an alternate solution is required. By building an Ampleforth implementation on the Internet Computer, we provide a price-stable digital asset on another blockchain platform that is not Ethereum. In doing so we increase the protocol's degree of interoperability whilst also creating a way to measure scalability, security, and decentralization on the Internet Computer. In this report, we present our prototype implementation and the tests we conducted upon both Ethereum's Ganache network and the Internet Computer's Sodium network. Our main findings demonstrated that on these local networks the Ampleforth implementation on Ganache executed faster than our Internet Computer prototype. However, our findings also suggested that the Ampleforth implementation is highly unlikely to scale as efficiently (on Ethereum) compared to a deployed version of our prototype on the Internet Computer.*

# 1 Introduction

Even though the emerging industry of Financial Technology (FinTech) has greatly modernized activities and services in finance, the current financial industry is overwhelmed with monopolistic financial intermediaries who not only exploit its clients with high fees and restrictions, but also have an unreliable centralized digital architecture that is susceptible to hacks, data thefts, and security breaches. The drive to tackle these two apparent obstacles of financial exclusion and security vulnerability has led to the rise in popularity of Distributed Ledger Technologies (DLT). With the ability to automate enforceable agreements and enable financial inclusion through what is known as a smart contract [1], the growth of Decentralized Finance (DeFi) apps built upon public blockchains such as Ethereum and Polkadot has been unprecedented.

While there have been some impediments in the efficiency of these platforms such as long finality time, low scalability, and low throughput, a new tamperproof, efficient, and fault-tolerant network protocol has been developed by the DFINITY Foundation, called the Internet Computer. This decentralized computing platform seeks to reduce reliance on big tech for back-end support [2] by "natively hosting hyperscale open internet services, pan-industry platforms, DeFi systems, secure enterprise systems, websites, and all of humanity's software logic and data in smart contracts" [3]. With new and efficient DeFi applications constantly on the rise, the next big innovation in this sector of crypto is to begin developing them on the Internet Computer's independent decentralized servers worldwide to ensure that users have free, unfettered access to their services that are not reliant on an increasingly expensive Ethereum network.

In addition to improving the blockchain platforms, however, there is also a growing need to advance another facet of crypto. Skeptics who are not yet convinced of the potential of cryptocurrencies would point out that Bitcoin in its present state is simply too volatile to ever be used as a secure financial asset. As a solution to that problem, several other cryptocurrencies known as stablecoins have entered the market - one being algorithmic stablecoins. To prevent sharp price fluctuations, algorithmic stablecoins adjust the supply deterministically to move the price of the token in the direction of a price target [4]. Currently, the most popular stablecoin is called Ampleforth, and its protocol operates slightly differently as it attempts to ask the question: can a synthetic commodity have a low correlation with both Bitcoin and traditional asset groups? [5] Ampleforth aims to achieve this through its elastic supply protocol, where once every day it conducts a re-base which adjusts the supply of tokens to the network so that the token price is as close to 1 USD as possible. This algorithmic stablecoin has grown in popularity due to it not being collateralized by a central body and has now become a multi-chain protocol, available on Polkadot, TRON, NEAR, and Ethereum.

As Ethereum expands globally, more and more users are requiring their transactions to be handled. Demand is, unfortunately, miles ahead of supply when it comes to transactional throughput, and as a result, the price of gas on the Ethereum network is rising to ridiculous highs. This gas crisis is resulting in common scenarios where the price of sending the transaction is more than the transacted amount. It is simply inevitable that DeFi users will eventually look towards these other emerging blockchains that have so much more to offer in terms of usability and scalability.

As a potential solution to both of these problems, we have built a prototype of an Ampleforth implementation on the Internet Computer. Within this prototype, there are three main components to the elastic supply protocol:

1. **Oracle:** The protocol must collect the 24-hour volume-weighted average price from market oracles.

2. **Monetary Policy:** Encodes the rule sets on how to read in data from the Oracles and then how to modify the token itself.

3. **Token Canister:** Implements the required token interface and enforces the rebase of tokens within wallets.

In addition to these three components, we have also implemented the following:

1. Creation of local wallets with encrypted keys.
2. Allowing for the exchange of tokens between wallets on the network.
3. Encoding the protocol's rebase mechanism into the token canister so that a rebase occurs across the network.
4. Collecting data for the rebase from certified market oracles.

We will conduct both quantitative and qualitative assessments to confirm this result. The long-term objective of this project would then be to ensure that it is connected with the global chain of token platforms running the Ampleforth supply protocol.

# 2 Related Work

To assess the advantages and disadvantages of building the Ampleforth Protocol on the Internet Computer, one must first understand the performance limitations of certain blockchains. In addition to understanding what constitutes a good blockchain, we must also evaluate what constitutes a good Algorithmic Stablecoin. The combination of both paves the way for us to better design and implement an Ampleforth Protocol on the Internet Computer blockchain.

In this document, our exploration of blockchain standards will be influenced by The Blockchain Trilemma, termed by Ethereum founder Vitalik Buterin [6], which address the three main challenges to building a permissionless Blockchain:

1. **Decentralization:** creating a public blockchain that cannot be controlled by any one entity
2. **Scalability:** creating a public blockchain that can expand its size efficiently without bound
3. **Security:** creating a public blockchain that is secure and can defend itself from attacks

The Blockchain Trilemma states that blockchains are often forced to make trade-offs that make it difficult to be successful in all three fields. For our research, it is important that we survey each Blockchain running the Ampleforth Protocol and compare how certain design choices influence a blockchain's decentralization, scalability, and security.

## 2.1 Degrees of Decentralisation

Decentralization, for many, is an integral part of any blockchain system. At its core blockchain technology is a distributed ledger that holds information about a network. That ledger is replicated across the network and relies on existing nodes to keep track of consistency. Should any single entity control a majority of the network then they would have the power to manipulate that ledger at will. In order to avoid this, blockchains, especially permissionless blockchains must be built with a high degree of decentralization in mind.

### 2.1.1 Network Permission

Currently, there exist two types of network permissions. The most popular type and the one we will be focusing on is the permissionless blockchain. Much like the internet, anyone can start interacting with the blockchain, all they need is an address and an internet connection. Furthermore, there is no single entity that can control the network or make alterations to it. These types of systems are typically decentralized in nature.

The alternative type is permissioned blockchains. These are private systems where each user is identifiable. These are perhaps more suited for larger organizations that need to efficiently

exchange and record transactions but also prefer to keep such information to authorized members only [7]. Differing to permissionless blockchains, in a permissioned blockchain, governance power is restricted to the authorized users of the blockchain.

A requirement of the Ampleforth Protocol is that access to Ample be unrestricted [8] and permissionless, hence all the blockchains that we will be comparing in this paper are permissionless.

An important question to ask is then how do network permission types affect scalability? Mattias Scherer states in [9] that "Permissioned blockchains have the advantage to configuring the network to allow parallelism whereas public blockchain struggles to partition the network to even make it possible for parallel execution. The problem is really complicated when you consider that no single entity can decide how the network should be partitioned".

It is the inherent decentralized nature of permissionless networks that make it so difficult to come to a consensus on issues like the ones mentioned above. Already we can see that there exists a tradeoff between the degree of decentralization on permissionless blockchains and their ability to scale.

## 2.2 Scalability: Overview of Consensus Protocols

Blockchains are driven and maintained by a consensus protocol which ensures for the stable operation of the system. Having a consensus protocol that scales well is extremely important for permissionless blockchains if they want to compete with more mainstream centralized systems.

The role of the consensus protocol is to help nodes in a system come to an agreement over a dispute. "Consensus algorithms have many applications, such as deciding on the validity of distributed transactions in cryptocurrencies like bitcoin, confirming the identity of the leader for a distributed task, ensuring the consistency among state machine replicas, and synchronizing them" [10]. However designing a consensus algorithm that does all of these things well is no easy task.

In the following section we will analyze consensus protocols for Ampleforth-hosted blockchains.

### 2.2.1 Ethereum - Proof of Work (PoW)

Currently the most popular Ampleforth platform is Ethereum with a circulating supply of approximately 300,000,000 AMPLs. Initially, Ethereum utilized the most famous consensus algorithm, the Proof-of-Work method, introduced by Satoshi Nakamoto in 2009 [11]. This type of consensus protocol is best applied to currency cryptography and involves a computer repeatedly executing computations in a race to solve a very labor-intensive mathematical puzzle. The winner of the race receives the right to append a block to the blockchain. The puzzle can be more accurately described as such: 'In PoW, each node of the network is calculating a hash value of the block header. In other words, to reach consensus in the network, miners try to find a hash value equal to or smaller than a certain given value. When one node finds the target value, it would broadcast the block to the whole network and all other nodes should confirm the correctness of the hash value. Hence, if the block is validated, all nodes would append this new block to their own chain' [10].

The Proof of Work method comes with some core advantages. It enables a high degree of decentralization whilst also keeping the blockchain secure. On the other hand, the process of mining and validating blocks is extremely costly and requires huge amounts of electricity to process [10]. Furthermore, in order to mine blocks, specialized and expensive hardware is required. Due to the scalability and environmental issues that Proof-of-Work entails Ethereum is currently in the process of switching to a Proof-Of-Stake consensus algorithm [12]. We will cover the advantages and disadvantages of Proof-Of-Stake in the next subsection.

### 2.2.2 Proof-Of-Stake (PoS)

Polkadot, Tron, and NEAR all use some variation of the Proof-Of-Stake algorithm, but what are the general principles behind it? In this algorithm, unlike Proof-Of-Work, miners are no longer racing to finish the puzzle first - instead, the network selects them at random. Before validators can be selected (on Ethereum), users must first stake 32 ETH [12]. Validators (miners) can then either be chosen at random to create blocks or alternatively, are responsible for maintaining and confirming blocks they don't create. The purpose of the stake is to incentivize desirable behavior from validators [12].

Some of the immediate advantages to Proof-Of-Stake include:

- Lower barrier of entry. Expensive and specialized hardware is no longer required to perform blockchain validation. This allows for greater decentralization and participation as all you need is a valid stake.
- Staking allows for a greater degree of security than Proof-Of-Work when computing concurrent transactions ('sharding' in Ethereum). Sharding in Ethereum is a process designed to increase transactional throughput on its blockchain [12].

However, of course, Proof-Of-stake is not without its own issues. Unlike PoW, PoS has yet to be properly tested as it is still in the early stages. Moreover, it is unlikely such a change is enough to help solve the gas price crisis on Ethereum. PoS may increase transactional throughput, due to incentivized validators, but those gains are marginal when compared to the sheer magnitude of congestion on Ethereum.

### 2.2.3 TRON - Delegated Proof-Of-Stake (DePoS)

Delegated PoS differs from traditional PoS in that it is more closely aligned with a representative democracy as opposed to each user being directly represented [13].

Tron utilizes the most common variation of PoS: Delegated Proof-Of-Stake [13]. The Tron Foundation states in its whitepaper: "The TRON consensus mechanism uses an innovative Delegated Proof of Stake system in which 27 Super Representatives (SRs) produce blocks for the network. Every 6 hours, TRX account holders who freeze their accounts can vote for a selection of SR candidates, with the top 27 candidates deemed the SRs. Voters may choose SRs based on criteria such as projects sponsored by SRs to increase TRX adoption, and rewards distributed to voters. This allows for a more democratized and decentralized ecosystem." [14]

The Delegated PoS allows for better distribution of rewards as the idea is that people will likely vote for the delegate who rewards them the most and is not dependent on which user is the wealthiest. However, because this system is representative-based, it is less protected from certain attacks such as the 51% attack and cartels formation [13].

### 2.2.4 Polkadot - Nominated Proof-Of-Stake (NPoS)

Polkadot's Nominated Proof-Of-Stake is a more specific interpretation of the Delegated Proof-Of-Stake. Polkadot understands staking tokens with a validator as 'nominating'. This is essentially what other networks call 'staking'; however, a Polkadot nomerator can nominate up to 16 validators to help you earn rewards [15]. In addition, NPoS differs from DPoS in that nominators are subject to loss of stake if they nominate a bad validator [15]. In DPoS this is not the case and the delegator will not lose stake for the behavior of the validator. The possibility of losing stake will hopefully incentivize nominators to always select a good validator.

Furthermore, once the validators have been nominated, stake is distributed evenly amongst validators. On their website, Polkadot states 'Polkadot uses tools ranging from election theory to game theory to discrete optimization, to develop an efficient validator selection process that offers

fair representation and security, thus avoiding uneven power and influence among validators' [15]. Whether or not their distribution algorithm significantly improves the speed of their consensus protocol remains to be seen.

### 2.2.5 NEAR - Thresholded Proof-Of-Stake (TPoS)

The Thresholded Proof of Stake is an election consensus mechanism. NEAR co-founder, Illia Polosukhin describes the overview concept as a deterministic way of organizing a large number of participants to work together to maintain the network [16]. The Thresholded Proof-Of-Stack splits up a segment of time into blocks. Each participant or 'witness' is assigned to a seat in a block. And each witness seat price is calculated based on the stake of all the users who stated they wanted to sign blocks.

Formally, if X is the seat price and Wi are stakes by each individual participant and P is total number of participants:

$$max\, X \in N\ s.t.\ \Sigma_{W_i \geq X}\, floor(\frac{W_i}{X}) \geq P$$

Figure 1: Formula to identify single witness seat threshold

The threshold for becoming a witness is calculated algorithmically and is set at the lowest level possible throughout the 'epoch' period ($\frac{1}{2}$ day), to allow for the most decentralized participation possible [17].

The main advantage of this type of threshold consensus is that pooling is no longer necessary. Token rewards are directly proportional to the amount a witness stakes. However, there are also certain disadvantages:

- The identity of participating witnesses is known early on in the process and hence someone with malicious intentions could target individuals in a DDoS attack (Distributed Denial of Service)
- Another disadvantage that applies to most Proof-Of-Stake consensus algorithms is that the system of dividing reward amongst current participants does not incentivize users to include new witnesses. However, at least in the case of NEAR, this is handled differently. Extra incentives are utilized to encourage the inclusion of new participants. [16]

### 2.2.6 Internet Computer - Threshold Relay (PoW + PoS)

Dfinity's novel Threshold Relay consensus algorithm aims to make the user experience as fast as possible. It is comprised of the following four layers:

- **Identity Layer:** Active participants on the Dfinity platform are called clients. The responsibility of the first layer is to provide open membership to register new clients by stake deposit which is then followed by a lock up period [18].
- **Random Beacon Layer:** This layer is a verifiable random function that is calculated by already registered clients. "This is a key technology of the Dfinity system which relies on a threshold signature scheme with the properties of uniqueness and non-interactivity." [18]
- **Blockchain Layer:** This layer uses a PSP (Probabilistic Slot Protocol) to rank clients for each length of chain. This is determined by the verifiable random function (random beacon) output for that length. [18]

  Advantages to Probabilistic Slot Protocol:

  - The first advantage of the PSP protocol is that the ranking is available instantaneously, allowing for a more constant block time which is also predictable.
  - The second advantage is that there is always a single highest-ranked client which allows for a homogeneous network bandwidth utilization.

- **Notary Layer:** "Dfinity deploys the novel technique of block notarization in its fourth layer to speed up finality. Notarization is a threshold signature under a block created jointly by registered clients. Only notarized blocks can be included in a chain. Of all the block candidates that are presented to a client for notarization, the client only notarizes the highest-ranked one with respect to a publicly verifiable ranking algorithm driven by the random beacon. It is important to emphasize that notarization is not consensus because it is possible, due to adverse timing, for more than one block to get notarized at a given height. This is explicitly tolerated and an important difference to other proof-of-stake proposals that apply full Byzantine agreement at every block." [18]

Dfinity is able to get such high speeds and short block times because notarization is not the same as a full traditional consensus. In the Dfinity white paper notarization is described as 'optimistic consensus' [18].

Another important aspect of the Threshold Relay is that both the Random Beacon and Notary layers are delegated to a *committee* - "A committee is a randomly sampled subset of all registered clients that deploys a threshold mechanism (for safety) that is moreover non-interactive (for efficiency). On the Internet Computer, the active committee is constantly changing. After having temporarily executed the protocol on behalf of all clients, the committee relays the execution to another pre-configured committee hence why this method is called the 'threshold Relay'" [18]. The goal of this technique is to allow for the Internet Computer to be scaled to potentially millions of users, however there are still some obvious tradeoffs. Dfinity's consensus algorithm does not follow traditional Proof-Of-Stake algorithms that apply full Byzantine agreement for every block. Whether or not an 'optimistic consensus' is worth the seemingly unbounded scalability has yet to be determined.

## 2.3 Consensus Protocol Evaluation Criteria

Before assessing each of these platforms we must first ask, by which evaluation criteria will we judge each consensus algorithm? S.M.H. Bamakan, A. Motavali, and A. Babaei Bondarti use the following criteria to measure the effectiveness of consensus algorithms [10]:

- Algorithms Throughput
  - Transactions per Second
  - Block Creation / Latency
  - Verification Time
  - Block Size
- Degree of Decentralization
  - Blockchain Governance
  - Permission Model
- Security and Vulnerabilities
  - 51% Attack
  - Double Spending Attack
  - Criminal Smart Contracts

For this section, we will be focusing on algorithmic throughput and some other metrics that directly affect a blockchain's ability to scale. Evaluation criteria for Security will be explored in later sections.

### 2.3.1 Algorithmic Throughput

In a decentralized system such as blockchain, transactions are verified by a consensus algorithm. How well a consensus algorithm performs and scales is very much dependent on its maximum transactional throughput. The throughput of a consensus algorithm is the maximum rate at which it can verify a transaction [14].

The speed at which a consensus algorithm can execute transactions is determined by several factors. The first that we will explore is the number of transactions per second or TPS.

#### Transactions Per Second

TPS is perhaps best described as the number of transactions that are processed in one second throughout a blockchain. Traditional Proof-Of-Work consensus algorithms often perform very poorly in this field [15]. For example, Ethereum is far behind other platforms in this domain. At only 19.5 TPS this is a significant weakness for Ethereum and certainly something that their competitors will aim to capitalize on.

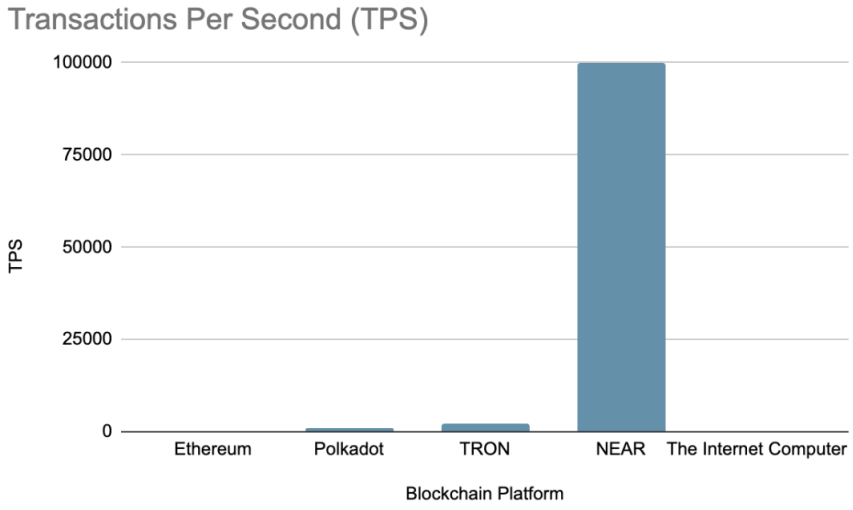


Figure 2: Transactions Per Second (TPS)

In the middle range bracket we have the Polkadot and TRON networks, each running at 1000 TPS and 2000 TPS respectively. However, the clear winner in this category is the NEAR network with a claimed TPS speed of 100,000 TPS.

While Ampleforth's usage as debt collateral may largely be unaffected by transactions per second, if Ampleforth is ever to be used as an everyday unit of exchange, transactions must be fast enough to compete with transactions on a centralized network. In order for this to happen Ampleforth must be expanded cross-chain to platforms such as NEAR that claim to support the necessary TPS rates.

#### Block Time

Block time, also known as block latency is the delay when verifying transactions and placing them inside of a block. Block latency in more precise terms is the time taken from when a value is presented to the chain, until when a consensus has been reached [13]. Ultimately the lower the block time the faster the blockchain can append blocks to itself.

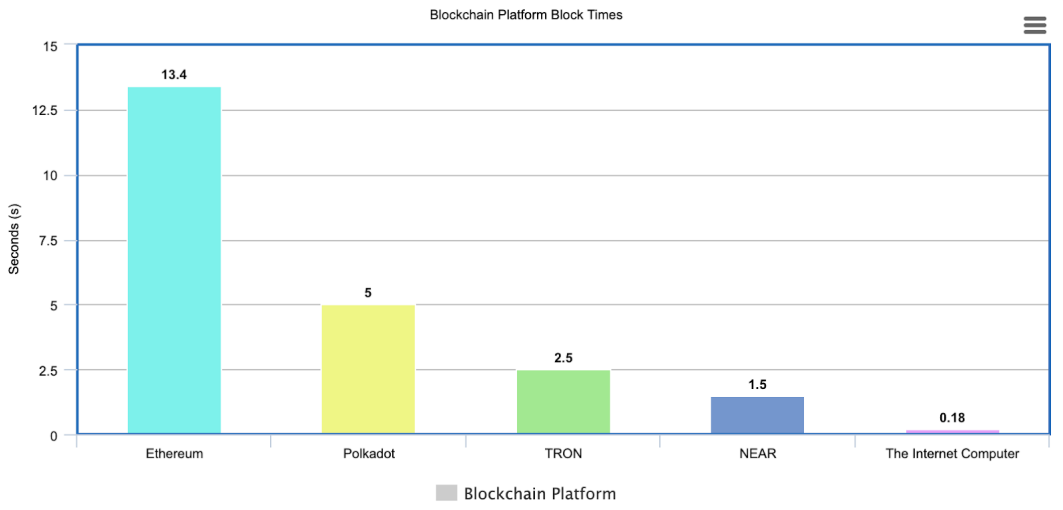


Figure 3: Block Time

From the data graphed above we can see that once again Ethereum lags behind with a block time of around 13.4 seconds. Next we have Polkadot around the 5 second benchmark. The two blockchains that can boast a block time of under 5 seconds are TRON at 2-3 seconds, NEAR at 1-2 seconds, and the Internet Computer at 0.18 seconds.

It is also important to note, a protocol with a shorter block time needs more confirmations for the same level of security as a slower protocol, such as Proof-of-Work. Here we can see an example of the expected tradeoff between scalability factors like TPS and block time with the security of the blockchain.

### 2.3.2 Transaction Fee

Miners (validators) are constantly verifying transactions on the blockchain however they're not doing so for free. There is a transaction fee that must be paid to miners before their transaction is accepted and processed. It is generally expected that a platform's transaction fees will increase as more people join and use the blockchain [10].

#### Fees on Ethereum

A perfect example of this is the Ethereum network. As usage has skyrocketed over the last few years, so too have gas fees.

Gas is the unit that measures the amount of computational effort required to execute specific operations on the Ethereum network [19]. Fees are necessary for transactions as they prevent malicious actors from spamming the network.


Figure 4: The price of gas on the Ethereum network in the past year

As you can see in the chart above, over the last year, the price of gas on the Ethereum network has gone from 14.20 Gwei to its current level of 140.39 Gwei (Gwei is a subunit of ether). That is an increase of approximately 888.6% since a year ago [20]. Clearly, this exponential fee increase is neither sustainable nor practical for scalability.

### Fees on Tron

Currently TRON is ahead of the race for the lowest transaction fee, boasting a near $0 transaction fee. While it is of course not completely 0, TRON is able to get close to nothing due the fact that: "transaction fees are calculated via Bandwidth and Energy, with each user allowed to make about 15 transfers free of charge. In subsequent transfers, the transaction fee is paid according to the Bandwidth used on each transaction" [21]. As we can see, the Tron network allows users to make a number of transactions for free. If the user spends all their free transactions before the end of the epoch, then an additional fee is required. Here Tron allows for very low transaction fees whilst also preventing malicious actors from spamming the network past a given amount.

### Fees on Polkadot

Polkadot does things slightly differently as the transaction fee is calculated from the following parameters:

- **A per-byte fee (as known as "length fee"):** the product of a constant per-byte fee and the size of the transaction in bytes. [22]
- **A weight fee:** weights are a constant value designed to manage the time to verify a block. Each transaction has two kinds of weight:
  - **Base weight:** accounts for time spent on a transaction's signature verification.
  - **Dispatch weight:** accounts for the duration of time that it took to execute the transaction.
- **A tip (optional):** increased incentive to make that transaction a higher priority

Combined, these three costs produce the inclusion fee. This fee is subtracted from a user's account before a transaction is executed. It is important to note that once the total is calculated 20% of the fee will go to the block validators and 80% to the Polkadot treasury [22].

The Polkadot fee can adjust how it is calculated allowing for a more representative transaction fee. On their project website Polkadot states "Polkadot uses a slow-adjusting fee mechanism with tips to balance these two considerations. In addition to block *limits*, Polkadot also has a block fullness target. Fees increase or decrease for the next block based on the fullness of the current block relative to the target. The per-weight fee can change up to 30% in a 24 hour period. This rate captures long-term trends in demand, but not short-term spikes. To consider short term spikes, Polkadot uses tips on top of the length and weight fees" [22]. This type of dynamic fee calculation allows for a more reasoned approach as fees are more representative of the true state of the network.

#### Fees on the Internet Computer

The Internet Computer uses a dynamic exchange rate between ICP and cycles (cycles being the equivalent to gas on Ethereum), meaning that the cost of cycles used to deploy and run canisters (a smart contract evolution) should remain relatively stable.

The Internet Computer allows canisters to be "pre-loaded" with cycles, such that consumers (even anonymous users) can use the canister-based application without incurring a cost. This also means that having a user interact with your dapp is as easy as sending a URL with the canister ID. Compare this to Ethereum's model where transactions need to be originated and paid for by a human user.

The "pre-loaded" canister model is a concept that the other blockchains have yet to make use of and is certainly a key aspect to Dfinity's Internet Computer that will allow for more mainstream adoption and scalability.

### 2.4 Interoperability

The boom in new blockchain platforms has both its benefits and hindrances. On the one hand, each new blockchain aims to offer a novel feature or solution to a common blockchain problem. However, having so many unconnected blockchains can also lead to fragmentation [23]. Peter Wegner defined the term as such "interoperability is the ability of two or more software components to cooperate despite differences in language, interface, and execution platform"[24]. In the case of blockchains, it is the ability for a source blockchain to initiate a transaction that is executed on a target blockchain despite the obvious component differences [23].

The Internet Computer has been designed specifically with this in mind. The Internet Computer has been constructed to make external calls to other chains, including Ethereum. Therefore, transactions can be created and signed via the Internet Computer and sent to the target chain. This allows for users to build settlement logic on the target chain, whilst also hosting the application's front-end on the Internet Computer, instead of hosting on third-party services like Amazon Web Service [25]. Interoperability is a key feature that will almost certainly define the future of emerging blockchains. The Internet Computer has taken this into account and is setting itself up to excel in this field.

Having the capability to communicate cross-chain is perhaps the single most important factor with regards to a blockchain's scalability, as it allows for the exchange of functionality across systems. Without it, traditional blockchains such as Ethereum, are in danger of being left behind as interoperable blockchains look to patch themselves together.

### 2.5 Security

As we have already seen, blockchain security is very much tied to a blockchain's ability to scale. For example, shorter block times must be paired with a higher rate of confirmation if Proof-Of-Stake protocols are to maintain the same degree of security as more traditional consensus mechanisms like Proof-Of-Work. Moreover, we have seen that the Internet Computer is able to

scale to millions of users in large part because of its novel Threshold Relay protocol which utilizes an 'optimistic consensus'. Whether or not the security compromise is worth the ability to scale is certainly a tradeoff that will have to be taken into consideration.

We will assess the degree of security for a given blockchain by comparing its ability to manage the following blockchain attacks and vulnerabilities:

- 51% Attack
- Criminal Smart Contracts

### 2.5.1 51% Attack

Avoiding the 51% attack relies on a robust consensus algorithm and a high degree of decentralization. In the case of Proof-Of-Work, this attack can occur when a single miner accounts for more than 50% of the hashing power on the network. In Proof-Of-Stake the vulnerability exposes itself if an individual owns more than half the total supply of tokens on the network. With this majority power an individual could potentially implement the following attacks:

- Undo transactions and allow for double-spending attack (repeated non-atomic transactions)
- Disrupt operations of other miners
- Severely impede the confirmation process for normal transactions [26]

We saw that Tron's Delegated Proof-Of-Stake is less protected against the 51% attack as it utilizes a representative-based system where the top 27 candidates represent the entire network. Should 14 of these representatives act unanimously as one body then they could potentially exploit the 51% vulnerability on the Tron network. This is of course easier to do in a representative system that consists of fewer moving parts.

Another disadvantage that applies to most Proof-Of-Stake algorithms is that the deterministic algorithm that distributes the rewards to staking participants does not incentivize the inclusion of new participants. While this may not lead to a complete 51% attack, it makes it harder to protect against as it limits the degree of decentralization in the network. Better performing participants hold on to their positions meaning the same key individuals maintain the network. Polkadot has specifically targeted this Proof-Of-Stake shortcoming by including a tip parameter that can be added when calculating transaction fees on Polkadot. In doing so Polkadot empowers new validators by allowing them to incentivize network nominators.

Ultimately the larger a blockchain becomes, the better. With more people maintaining and growing a blockchain coordinating a 51% attack becomes increasingly difficult. Most modern blockchain protocols are reasonably equipped to handle such an attack. However, as we mentioned earlier, at least among Ampleforth blockchains, the Proof-Of-Stake consensus protocol is still in its infancy and is yet to be sufficiently battle-tested.

### 2.5.2 Criminal Smart Contracts

Another attack that perhaps best applies to Ethereum and smart contract blockchains like the Internet Computer, is a criminal smart contract, for example, a smart contract with malicious or criminal intent. It is possible for criminal smart contracts to facilitate the leakage of confidential information, loss and theft of cryptographic keys, and various real-world crimes such as murder, arson, and terrorism. [26]

This threat is an even greater issue when you consider that many of these blockchains follow the "code is law" ethos meaning that once a canister (contract) is written to the blockchain, it is very difficult to take it down [26]. The Internet Computer's founder Dominic Williams designed

a novel governance system, namely the Network Nervous System, whose functionality includes resolving disputes such as malicious smart contracts running on the blockchain.

The Network Nervous system is designed to allow for authoritative decisions over the network. Decisions are made by distributed voters who hold governance tokens. Williams uses the example of a human "trafficking system" hosted on the internet computer to explain how the NNS might work: "Now, of course, this trafficking system would be tamperproof and replicated across node machines that reveal nothing more than encrypted bytes if they are opened — however, the logic of Internet Computer nodes could be upgraded so that if the NNS adopted such a proposal, they would respond by encrypting the relevant data to the public key of the investigating agency and then making it available for export. Once this action had been taken, the ethics committees involved in supporting the adoption of the proposal would publish why they provided support in the interests of transparency" [26]. Here Williams details how, in extreme cases such as the one mentioned above, the Network Nervous System (NNS) can act as an ethics committee and will have the ability to execute proposed alterations to the blockchain.

The governance system of the Internet Computer is certainly a defining factor when compared to other Ampleforth hosted blockchains. Again, assessment of the NNS is limited at this time as we wait for the minting of the very first ICP genesis block.

## 2.6 Algorithmic Stablecoin Metrics and Features

With existing blockchains and their metrics thoroughly examined, it is now necessary to evaluate and understand what constitutes a good algorithmic stablecoin. To pursue this goal, we explored three well-known algorithmic stablecoins, in addition to Ampleforth (AMPL) - Empty Set Dollar (ESD), Basis Cash (BAC), and FRAX (FRAX).

To summarize, the following algorithmic stablecoin metrics and features will be methodically explored:

- Token-based Governance
- Elastic Supply
- Volatility

### 2.6.1 Token-based Governance

Governance within cryptocurrencies greatly varies, with some organizations more directly involved with internal decision-makings than others. Interestingly, all the algorithmic stablecoins examined in this section were governed by tokens.

So what is token-based governance? Simply put, decisions concerning governance of the coin, such as "the core protocol, product or feature roadmap, hiring and staffing, and changes to governance parameters" [27] are all determined by those who are holders of the coin itself. Every proposal or decision must be voted on and approved to be enforced, and the rules vary depending on the platform.

When looking at Empty Set Dollar (ESD), one can see it is governed by an On-Chain Protocol Governance. There are certain thresholds (see Figure 5) that must be met by voters for certain changes to be enacted, such as the proposal threshold being 0.5% ownership of the Decentralized Autonomous Organization (DAO), or the vote length lasting for 9 Epochs [28]. This is to prevent any form of manipulation or process gaming.

- **Proposal threshold:** 0.5% ownership of the DAO
- **Vote length:** A vote lasts for 9 Epochs
- **Minimum Quorum:** 20% of the DAO must vote on the proposal
- **Locking:** If you vote your tokens will be locked for the duration of the vote
- **Expiry:** Successful votes must be committed within 3 epochs.

Figure 5: Threshholds that need to be met by voters for the Empty Set Dollar (ESD)

Basis Cash (BAC) also operates as a DAO, allowing those who have a share of this cryptocurrency to influence the developmental direction of the protocol. For this process to begin, a Basis Cash Improvement Proposal (BIP) would be introduced and discussed within the community, and when deemed worthy, voted upon to be enacted or rejected [29].

A similar process can also be observed with FRAX Finance (FRAX) and Ampleforth (AMPL), where both platforms are operating as a DAO and the same proposal process can be observed. For FRAX, "Once a proposal has been submitted, it begins an active voting period of 3 days, where it may be voted for or against. If a majority is in support of the proposal at the end of the period, and the proposal has a minimum of 4,000,000 FXS in favor, the change is queued into the Timelock where it can be implemented after 2 days." [30] The Ampleforth Governance Protocol also requires a proposal submission, along with other steps that can be observed in Figure 6 [31].


Figure 6: The Step-by-Step Process for the Ampleforth Governance Protocol

With these results in mind, for our implementation we believe that it will be best to implement a DAO as well, as it will provide the system and users the "transparency in decision making, independence of funding for development initiatives, and adequate representation of key user groups were key points of contention in early discussions". [32]

### 2.6.2 Elastic Supply

What makes Ampleforth and many other algorithmic stablecoins unique is that they have an elastic supply. While we will soon observe that this does not apply to all coins, many incorporate this unique system. "An elastic supply (or rebase) token works in a way that the circulating supply expands or contracts due to changes in token price. This increase or decrease in supply works

with a mechanism called rebasing. When a rebase occurs, the supply of the token is increased or decreased algorithmically, based on the current price of each token." [33]

Though the rebasing parameters could differ for every coin, the process and theory are all extremely similar. ESD, BAC, and AMPL all conduct a rebase to balance out the supply and demand of the coin. The only coin that doesn't fit this exact mold is FRAX. If FRAX $>$ \$1, users can mint new FRAX by depositing \$1 worth of USDC/FXS and sell newly minted FRAX in the market. If FRAX $<$ \$1, users buy cheap FRAX and redeem them for \$1 worth of USDC/FXS.

This leads to the question - is it better for Ampleforth to have an elastic supply instead of other systems like FRAX and Bitcoin? "Since demand shocks are relatively common, this means that cryptocurrencies are subject to high price volatility. Volatility can be a desirable feature for a financial investment that tries to capture capital gains. But it is a major impediment to cryptocurrencies becoming more widely accepted as a medium of exchange". [34] Therefore we conclude that it would be better for us to continue down this path with this initiative.

### 2.6.3 Volatility

The purpose of stablecoins is to minimize the volatility of the price, and to keep things relatively "stable". Though every stablecoin utilizes different methods to do so, algorithmic stablecoins focus on utilizing rebases to maintain stability. So what exactly is volatility in cryptocurrency?

"In a 24-day trade, it is defined as the dispersion of the price of an asset as shown from its starting price. An investment is considered volatile if its prices move aggressively up or down daily, as can be seen in the cryptocurrency market." [35] Put simply, it is the standard-deviation of daily returns.

When comparing the 4 algorithmic stablecoins, a volatility timespan was predetermined - 90 days, as some coins had been active for less than a year. Therefore the numbers shown below portrays the asset's annualized volatility calculated over the past 90 days of daily returns.

| Name | Annualized Volatility (90D) |
|---|---|
| **Empty Set Dollar (ESD)** | 2.25 |
| **Basis Cash (BAC)** | 2.00 |
| **FRAX (FRAX)** | 0.338 |
| **Ampleforth (AMPL)** | 1.75 |

Figure 7: Annualized Volatility of Algorithmic Stablecoins over the past 90 Days of Daily Returns

The ideal goal of a stablecoin is to have the least amount of annualized volatility as possible - with the numbers above, it would be whatever is closest to 0. The larger the number, the higher the volatility. Observing the results, it can be observed that FRAX has the least volatility [36], while ESD [37] has the highest. Ampleforth and Basis Cash seem to be somewhere in between, with a volatility coefficient of 1.76 [38] and 2.00 [39] respectively. With our endeavors, we would like to see Ampleforth reach an even lower coefficient to witness true algorithmic stability.

## 2.7 Related Works Summary

With the performance limitations of blockchains currently hosting the Ampleforth protocol properly investigated, many details came to light in helping future developers understand the pros and cons of utilizing each platform. Following the previously introduced Blockchain Trilemma, we were able to compare metrics involving Decentralization, Scalability, and Security, as seen in Figure 8. With our ultimate goal of assessing the advantages and disadvantages of building the Ampleforth Protocol on the Internet Computer, we have come to the conclusion that this

platform is extremely well built for scalability, however, that its decentralization and security capabilities are too early in its stages to receive a proper assessment. This same issue was also observed when investigating the NEAR Protocol, as data on Deposit Times and Transactions Fees could not be found. When comparing platforms with all available information, however, it seems that TRON had the best outcome for all 3 categories. Therefore it would be ideal for us to have the Internet Computer have similar or improved metrics as the TRON blockchain platform.

As for the understanding of what constitutes a good algorithmic stablecoin, we were able to fully compare them based on three metrics: token-based governance, elastic supply, and volatility. The summary of our findings can be viewed in Figure 20. We were able to find that all of them were governed by a DAO, and had quite similar methods when it came to making changes to the protocol. All except one had an elastic supply, which also happened to be the one with the lowest volatility. Ampleforth did not have the highest or lowest volatility coefficient, therefore providing us with space to further improve our project.

| Name | Network Permission | Consensus Protocol | Block Time | Transactions Per Second | Deposit Times | Transaction Fee | Interoperability | Scalable |
|---|---|---|---|---|---|---|---|---|
| Ethereum | Permission-less | Proof-of-Work | 13.4s | 25 TPS | 5 mins | $13.05 | No | No |
| Polkadot (Acala) | Permission-less | Nominated Proof-of-Stake | 5s | 1,000 TPS | 2 mins | The length fee is the product of a constant per-byte fee and the size of the transaction in bytes. | Yes | Yes |
| TRON | Permission-less | Delegated Proof-of-Stake | 2.5s | 2,000 TPS | 1 min | $0 | Yes | Yes |
| NEAR | Permission-less | Thresholded Proof-of-Stake | 1.5s | 100k TPS | ? | ? | Yes | Yes |
| The Internet Computer | Permission-less | Threshold Relay (Bonded Validator Model, PoW + PoS) | 0.18s | 500-1000 TPS | ? | ? | Yes | Yes |

Figure 8: Metrics Comparing All of Ampleforth's Blockchain Platforms

| Name | Token-based Governance | Elastic Supply | Volatility (90D) |
|---|---|---|---|
| Empty Set Dollar (ESD) | Yes | Yes | 2.25 |
| Basis Cash (BAC) | Yes | Yes | 2.75 |
| FRAX (FRAX) | Yes | No | 0.344 |
| Ampleforth (AMPL) | Yes | Yes | 1.76 |

Figure 9: Algorithmic Stablecoin Comparisons

# 3 Solution

We designed an Ampleforth implementation on the Internet Computer to discover whether or not we could improve specific attributes such as decentralization, scalability, and security. While developing this prototype we encountered several challenges that required us to adjust our implementation process. One significant challenge was the pre-deployment stage of Dfinity and the Internet Computer at the time of conducting this research. This meant that certain canisters were built with varying sdk (Software Developer Kit) versions. Our prototype design consists of three integral canisters:

- ERC-20 - dfx (Software Developer Kit) 0.6.0
- Supply Policy - dfx 0.6.0
- Market Oracle - dfx 0.6.25

Whilst Dfinity does already have an ERC-20 in Motoko they, unfortunately, do not yet have a built-in oracle canister. The developer team did however refer us to another Github project currently in development that was using a GO-based plugin to feed external data to an Internet Computer canister. In order to get a working prototype for the market oracle, we decided to separate our implementation into two separate prototypes as the ERC-20 and the market oracle were built using different versions of dfx. We were unable to circumnavigate this issue given the relatively short time to conduct the research. In separating the prototype into two we get a full working implementation for each component and are able to test each as necessary.

## 3.1 Architecture

Prototype 1 - ERC-20 and Supply Policy

This first prototype focuses on parsing the data after it has been pulled from an oracle. Since the market oracle is not connected to this prototype we drip feed the daily oracle price to the Supply Policy manually using a bash script. We use the oracle rate from the first 60 days of Ampleforth's price history sourced from a website called Coin Tools [40] that tracks the daily rebase history of Ampleforth on Ethereum.

The software development kit deploys canisters to the Sodium network, a local Internet Computer instance running on your machine. As we have just mentioned briefly the Supply Policy canister is drip-fed the oracle rate (token price). The Supply Policy canister will then determine:

- If the oracle rate is within the price threshold:
  - No rebase to perform.
- If the oracle rate is outside the price threshold:
  - Rebase to perform.
  - Calculate the percentage adjustment (Supply Delta) for the total supply of AMPLs.
  - Return percentage adjustment (Supply Delta).

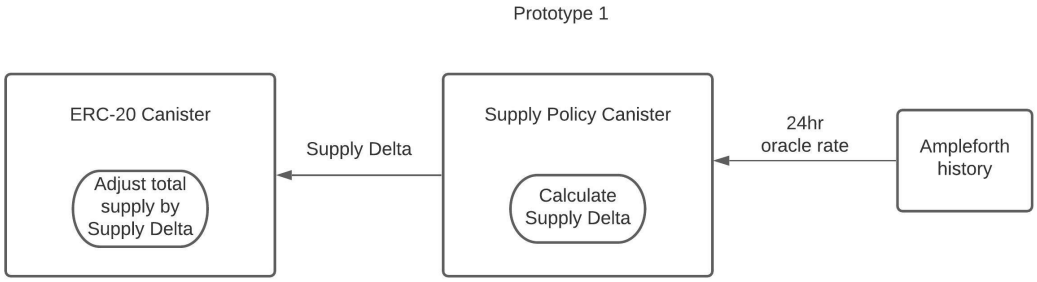


Figure 10: Prototype 1

Once that percentage adjustment is returned by the Supply Policy it is given to the ERC-20. Our ERC-20 is an extension of Enzo Haussecker's motoko implementation [41]. Our version of the canister has an additional rebase function which adjusts the total supply of AMPLs on the network.

During the course of developing our prototype, we ran into several complications. For one, since the ERC-20 is built using SDK version 0.6.0 we were unable to utilize floats that are available in more recent versions. This meant that, in order for our prototype to work, all our values had to be multiplied by a magnitude of 100,000 in order to perform the calculations. Once calculated the values are reverted back to their original order of magnitude. This of course increases the line count of these two canisters as we handle the additional conversion.

Furthermore, since wallet canisters do not yet exist publicly for developers we followed Enzo Haussecker's example and instead created each wallet as a hidden local directory encrypted with a key for access.

Prototype 2 - Market Oracle

The Market Oracle was also tested on Dfinity's Sodium network and runs using dfx version 0.6.25 (dfx is Dfinity's software development kit). The foundations for our market oracle rely on the oracle framework created by Hypotenuse Labs [42]. Using a GO-based plugin they were able to input data sourced from external weather APIs into an Internet Computer canister. To suit our needs we were able to adjust the code in the following ways:

- Queries Coingecko HTTP endpoints for price of AMPL
- Query for other token metrics such as:
  - Market capitalization
  - 24 hour volume change
  - 24 hour price change
  - Last updated timestamp

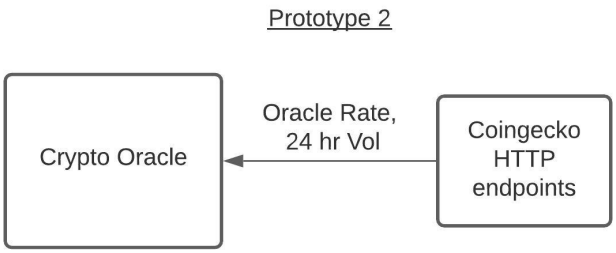


Figure 11: Prototype 2

## 3.2 Formulas

In the following section, we will cover the three most important formulas utilized by our prototypes and what they achieve. The first formula that we will introduce is run by the Supply Policy when the oracle rate is outside the target price threshold.

Supply Percentage Adjustment

$$SupplyDelta = (((Oracle\ Rate - Price\ Target) / Price\ Target) * 100) / reactionLag \quad (1)$$

Again the *Oracle Rate* is given to the supply policy by a bash script. The script collects these oracle rates (token prices) from the Ampleforth price history. The *Price Target* is the CPI adjusted 2019 US dollar. This value equals $1.005 and is constant for all but one of the rebases in the first 60 days. Lastly we have the *reaction lag*, first introduced in the Ampleforth whitepaper, its purpose is described as 'to dampen the supply change' [5]. At launch, this value was set to 30 meaning that the supply adjustment would be applied as if it was over 30 days.

Total Supply Adjustment

This formula is fairly self explanatory. It changes the total supply of tokens on the network by the necessary percentage.

$$totalSupplyInt = totalSupplyInt * SupplyDelta \quad (2)$$

Split Ratio

The last formula that we will introduce is the conversion rate between a user's internal balance and their external balance. This is referred to in the Ampleforth whitepaper the *'Split Ratio'* [5]. This allows for rebases to be performed more efficiently as now we can avoid creating a transaction for each wallet.

$$hiddenSupply / totalSupplyInt \quad (3)$$

## 3.3 Findings

Next, we will present our results for the ERC-20 and Supply Policy tests conducted on the Internet Computer's Sodium network as well as tests of the official Ampleforth implementation on a Ganache Ethereum network (personal Ethereum blockchain).

- Rebase Tests on Internet Computer Sodium Blockchain
- Rebase Tests on Ganache Ethereum blockchain
- Oracle Tests
- Token Transfer Tests

The goal of our prototype was to make canister speeds as fast as possible. The faster the ERC-20 can execute, the faster those changes can be propagated across the network, hence improving protocol scalability. Ideally, we can also increase the speed of the market oracle, however, we will see that ultimately its speed has little effect on scalability.

Rebase Tests on Internet Computer Sodium Blockchain - Prototype 1

- Rebase: positive rate no change CPI
  - Time to execute ERC-20 rebase
  - Time to execute SupplyPolicy rebase
  - Time to execute one whole rebase

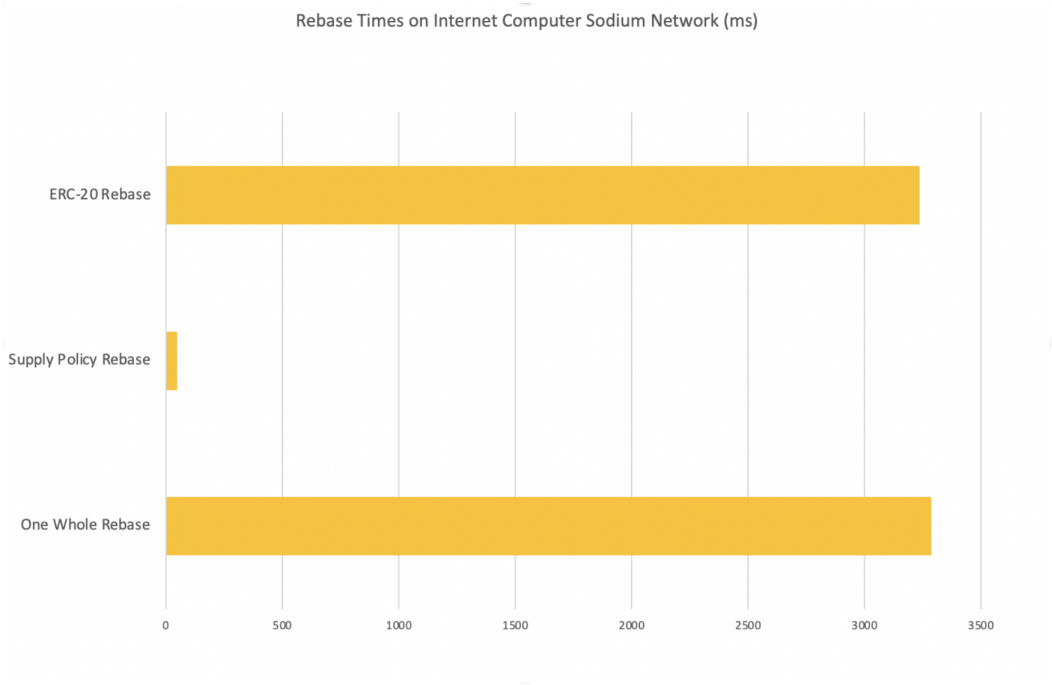


Figure 12: Internet Computer Rebase Times

We can see from the figure above that one whole rebase on our prototype takes around 3 seconds to complete. As we will see in the next section these times are slower than the Ampleforth implementation on Ganache. We discuss the significance of our rebase times in the following section once we have presented the findings for the same tests using the Ampleforth Ganache implementation.

Rebase Tests on Ganache Ethereum blockchain

The Ampleforth implementation that we are using for these tests is sourced from Ampleforth's public Github repository [42].

- Rebase: positive rate no change CPI
  - Total rebase speed
    - Get data from market oracle
    - Get data from cpi oracle
    - Emit Rebase
    - Update last Rebase Timestamp
    - Update Ampleforth global supply
    - Emit Rebase
    - Time to increment epoch

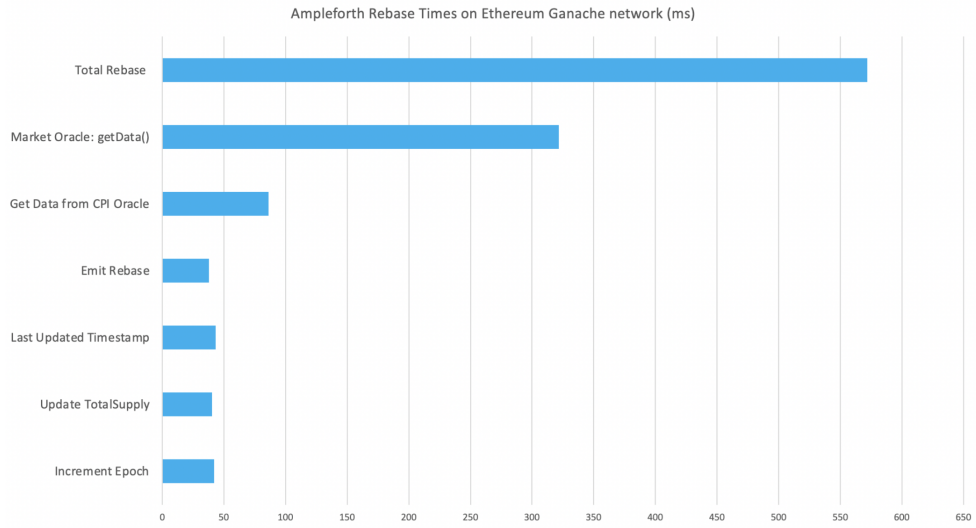


Figure 13: Ampleforth Rebase Times on Ganache Ethereum Instance

You will notice that since this is essentially the official Ampleforth implementation its rebase functionality is far greater than what we were able to build in just four months. However, with one rebase only taking around 570 ms, it is several seconds faster than our prototype on the Internet Computer test-net (test network).

In this section, we will be summarizing and comparing the significance of our rebase tests. As well as commenting on the applicableness of such direct comparisons between prototypes on each respective network. The first metric that we will be comparing is the speed for one whole rebase to take place on each respective test network. As we can see the official Ampleforth implementation running on Ganache is far quicker than our prototype on the Internet Computer.

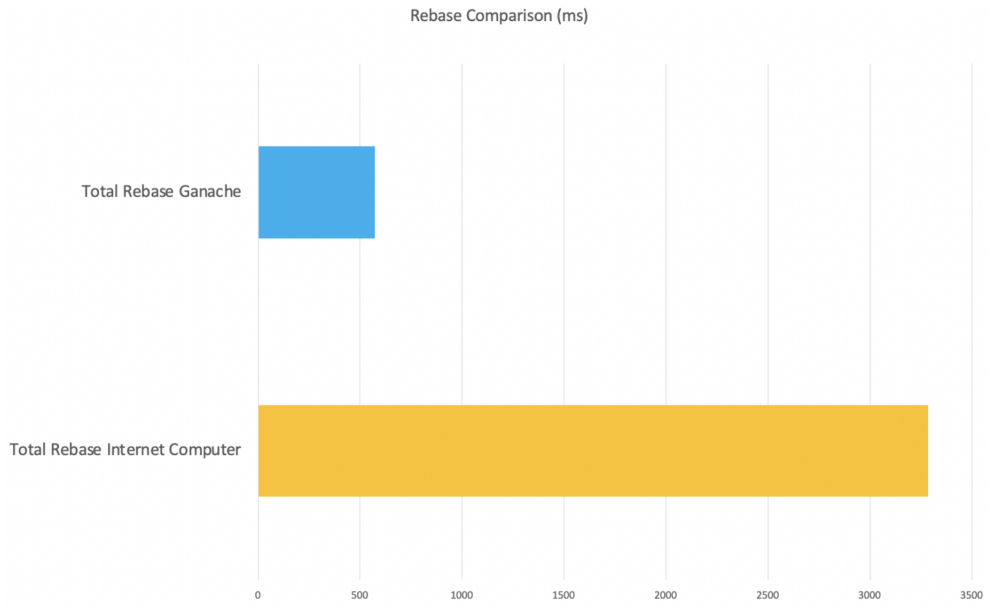


Figure 14: Total Rebase Time Comparison

The speed of the rebase is important as it directly affects the protocol's ability to scale. When a rebase is called, it is essential that the change in the state of the ERC-20 is replicated across the network in an efficient manner. The faster the ERC-20 canister executes the faster the network can come to a consensus. We can see that on the local networks the official Ampleforth implementation is able to perform this rebase at a faster rate.

Again, when analyzing this comparison we must take into account that these tests are performed on local test networks. Neither blockchain instance, Ganache or Sodium, takes into account transactional delay. When we are able to test on the Internet Computer's main-net we predict that our application will perform and scale far more efficiently once fully deployed as the Internet Computer is able to finalize transactions at a rate more than five times that currently on Ethereum.

Oracle Tests

In this section we compare the time taken to retrieve oracle data on the Internet Computer's Sodium network against the time taken to retrieve oracle data running on a Ganache Ethereum network. Both of these times are likely to increase when deployed to their respective main nets as Sodium and Ganache tests do not account for the transactional delay. Tests of the Ampleforth Market Oracle are conducted using Ampleforth's open-source Github repository [7].

- Ampleforth's Market Oracle on Ganache network instance.
- Prototype 2 - Coingecko Market Oracle on Sodium network instance.

Again we can see that Ampleforth's ganache instance performs very well against our prototype. However, if we were to test each implementation again on their respective main nets we predict that our prototype would stay relatively constant or might even improve. On the other hand, the market oracle on Ethereum would almost certainly increase once the transactional delay of the Ethereum main-net is accounted for.

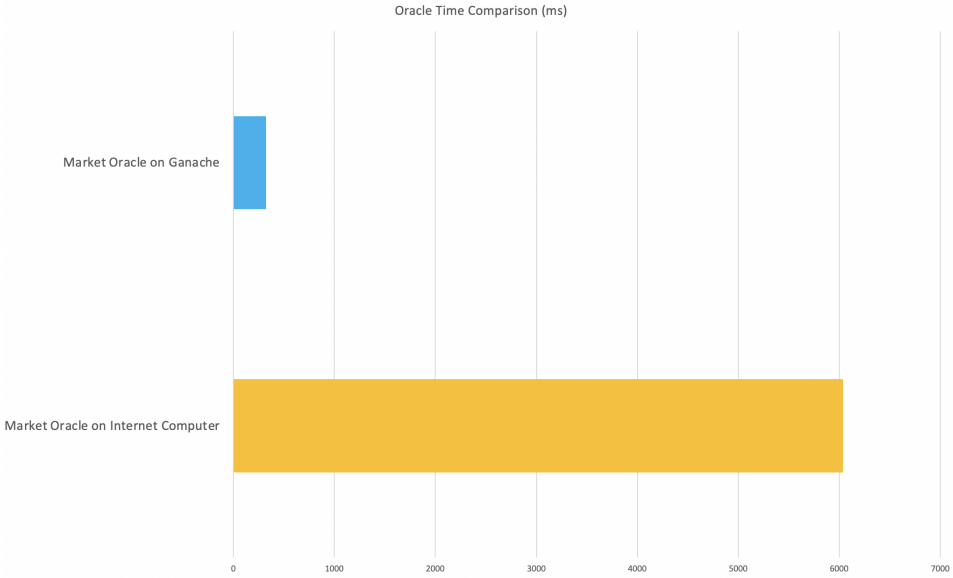


Figure 15: Oracle Query Time Comparison

Unlike the rebase speed of the ERC-20, the oracle speed has little effect on the protocol's ability to scale. This is due to the fact that the rebase is only called once a day at 2 am UTC. Whether the ERC-20 gets that data in 250 ms or 6000 ms makes little difference to the average user. However, we may see that the speed of our prototype oracle becomes more significant as we extend its functionality to query multiple sources. With multiple providers, an oracle must be able to distinguish which ones are valid in a timely manner. If this process takes too long oracles may run out of time to query sources. Hence we can see that the speed of an oracle (with multiple providers) is also tied to its security. Oracles must be able to verify sources and pass the information on before the rebase window closes else they run the risk of failure. This is very important as the Ampleforth protocol is entirely dependent on a functioning oracle to perform daily rebases. Ultimately Ampleforth is only as secure as its oracle and therefore such a system must be airtight. Later on, we will discuss how oracle security can be improved through the manifest and accountability contracts.

Token Transfer Tests

Next, we will consider how long it takes for each respective ERC-20 to transfer tokens from wallet A to wallet B. Again, it is important to remember that these times do not account for the transactional delay expected on deployed blockchains.

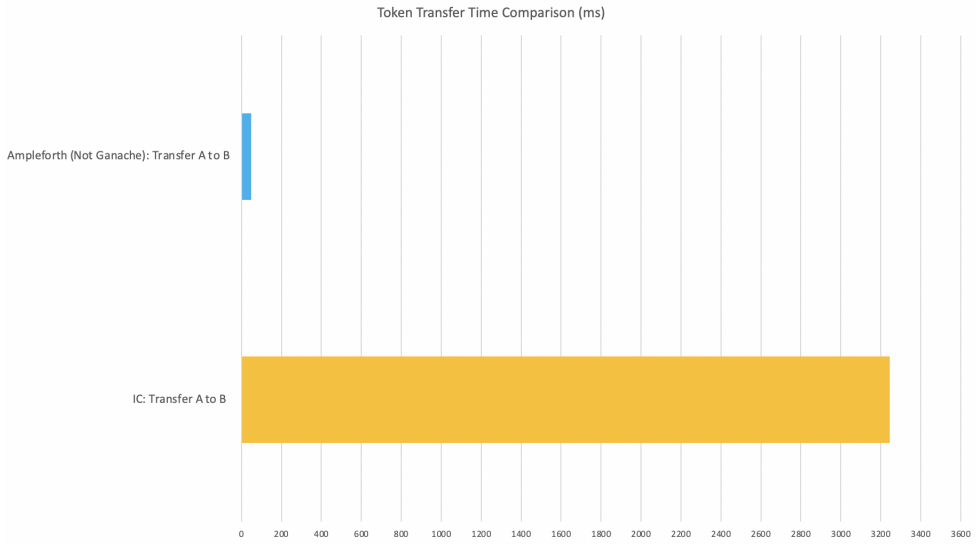


Figure 16: Token Transfer on Ethereum vs Internet Computer

As you can see in the figure above we were not able to test the token transfer speed on a Ganache instance. The Ampleforth transfer speed indicated above is calculated by a unit test of Ampleforth's ERC-20 transfer function. However, speeds on a personal Ganache instance are unlikely to be significantly higher as neither test considers congestion on the blockchain.

On local test networks, it is again faster to transfer tokens on the official Ampleforth instance. However, this does not take into account congestion nor the high gas prices on Ethereum for such a simple transaction. Once deployed on the Internet Computer main-net we expect our transfer function to scale far more efficiently due to the Internet Computer's superior block rate and lower transaction fees.

## 3.4 Working Prototype Rebase (First 60 Days)

To demonstrate the correctness of our solution we simulated the first 60 rebases of the Ampleforth rebase history. Oracle rate data was collected from Coin Tools [40] and then fed into Supply Policy which then returns the supply adjustment. That generated value is then given to the ERC-20 rebase function, which adjusts the total supply accordingly.

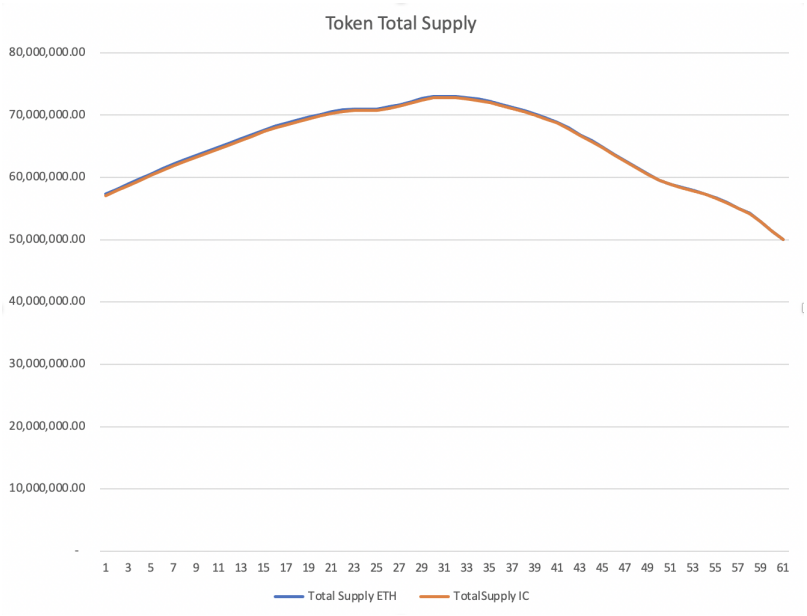


Figure 17: Total Supply After First 60 Rebases

The graph above demonstrates that we are able to simulate almost the exact same supply history with a prototype on the Internet Computer. Again, the oracle rate and price target data are retrieved from the Ampleforth rebase history and are used to adjust the total supply over the first 60 days.

# 4 Results and Discussion

## 4.1 Main Challenges and Issues

As we have already briefly mentioned there have been quite a few challenges in implementing our prototypes. Our research was conducted in the four months leading up to the launch of the Internet Computer. This meant that resources and infrastructure were limited, as to be expected of any pre-deployment blockchain. For example, there was no built-in oracle and hence no way to obtain external real-world data for a canister. This resulted in us working off another project also currently in development on the Internet Computer. Many of the resources that we utilized were only experimental versions, for example, the ERC-20 developed by Enzo Haussecker. This meant that oftentimes canisters were written in different versions and hence unable to work together in tandem.

Another challenge is that we were unable to test our prototypes of the Internet Computer's main-net which finally launched on May 7th. This launch date was unfortunately too late for us to test and incorporate the results into the paper, however, these tests are essential, and we will likely conduct them in the near future. . Unlike the Ganache and Sodium networks, tests on the Internet Computer main-net will take into account transactional delay. Furthermore, it is in this realm that the Internet Computer is likely to surpass Ethereum for scalability as its block rate is simply far superior.

## 4.2 Blockchain Scalability Comparison

Before diving into the results of our findings within the prototype, it is vital to understand the differences between the two blockchain platforms as they have significant impacts on performance. Specifically in this section, we will focus on measurements from outside resources concerning each

blockchain's scalability. The definition of scalability here will be identical to the one utilized in the related works section – it is how much a blockchain can expand its size efficiently without bound. Therefore, figures such as block rate and transactions per second will be identified for the audience to gain a deeper understanding of each platform's environment for our prototype.

Unfortunately for this paper, there were only a few variables of each blockchain platform that could be directly compared with accurate and concise information. This was either due to a lack of information as the Internet Computer is still new or because of the differences in architecture and functionality between the two blockchains. Therefore, we will have a separate section for the Ethereum blockchain platform to provide analytical insight into certain speeds and data outputs that we have discovered. This "Ethereum Figures" section will provide data on metrics such as difficulty (T), hash rate (TH/s), and uncle rate (%). Furthermore, the differences in architecture and inability for comparison will be elaborated upon within each section below.

### 4.2.1 Block Rate

To begin, we would like to first compare the block rate of each blockchain platform. The block rate defines the time it takes for a single block to be mined. For Ethereum, "the average block time of the network is evaluated after n number of blocks, and if it is greater than the expected block time, then the difficulty level of the proof of work algorithm will be reduced, and if it is less than the expected block time then the difficulty level will be increased" [43]. This evaluation differs for the Internet Computer, as block finalization occurs at different speeds in each subnet. Therefore, the block rate is measured as the sum of block rates of all subnets available on the main network.

Through the Dfinity Explorer dashboard, we were able to acquire the average block rate of the Internet Computer. After the official launch of the Genesis main net, the average block rate was 5.5 bps. Ethereum's average block rate was also acquired in a similar manner, but through Etherscan - a block explorer and analytics platform for Ethereum. Through this method, Ethereum's average block rate was found to be 0.076 bps. These values inform us that while the speed of our prototype was found to be slower on the Internet Computer test net, both test nets do not account for the transactional delay expected on a deployed blockchain. Therefore, if we were to deploy our prototype on the Internet Computer main net, it would most likely scale far better as the block rate is significantly greater than Ethereum.

The Internet Computer's Block Rate: 5.5 bps [44]

Ethereum's Block Rate: 0.076 bps [45]

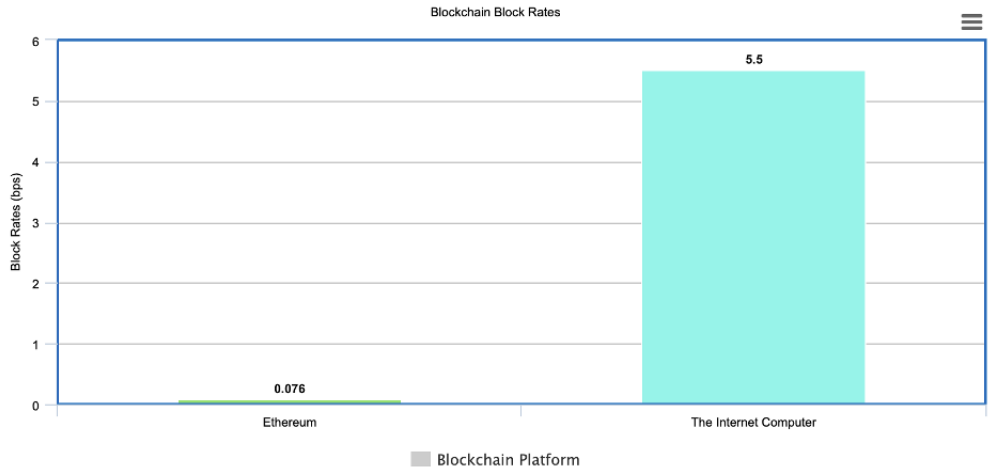


Figure 18: Blockchain Block Rate Comparisons

### 4.2.2 Transactions per Second

Next, we would like to compare the transactions per second (TPS) of each blockchain platform. The definition of transactions per second here will be identical to the one utilized in the related works section – it is best described as the number of transactions that are processed in one second throughout a blockchain. A transaction usually consists of "cryptographically signed instructions from accounts... [and] will initiate a transaction to update the state of the... network" [46]. Traditional Proof-Of-Work consensus algorithms such as Ethereum often perform very poorly in this field [15]. At only 19.2 TPS according to Etherscan, this is a significant weakness for the blockchain platform.

While the Internet Computer's official TPS figures have yet to be released to the public, other groups have released estimated numbers. For example, according to the DEXON Foundation, the blockchain TPS ranges from 500 – 1000 TPS. They state, "Dfinity contains a randomness beacon which generates new randomness by a VRF (verifiable random function) with information from a newly confirmed block. They use this randomness to select a leader and electors for a round. By hypergeometric distribution, Dfinity only samples hundreds of nodes to notary a block instead of using all nodes, and this is correct with high probability" [47]. Even at the lowest estimate of 500 TPS, this is significantly greater than the current results for Ethereum, suggesting that the Internet Computer is very likely to scale much faster. Similarly to the block rate, these values signal that the deployment of our prototype onto the Internet Computer main net would most likely result in execution speeds that are much greater than Ethereum.

Ethereum's TPS: 19.2 tps [48]

The Internet Computer's TPS: 500 - 1000 tps [47]

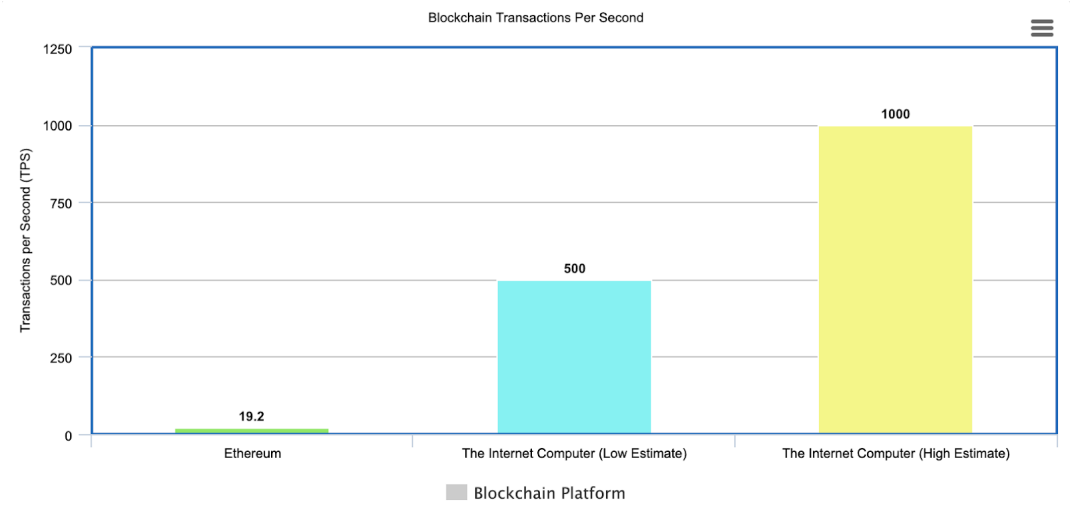


Figure 19: Blockchain Transactions per Second

### 4.2.3 Ethereum Figures

Now we would like to present data that is specifically relevant to the Ethereum blockchain, such as difficulty (T), hash rate (TH/s), and uncle rate (%). While this will not provide a direct comparison against the Internet Computer, we believe the data points could help in understanding the internal architectural speeds of the consensus algorithm.

To begin, the first data point is the Ethereum network difficulty. This key value is "the difficulty of a problem that miners must solve to find a block. It shows how many times on average miners should calculate a hash function to find a cryptocurrency block" [49]. This difficulty fluctuates depending on the number of miners, but in the past year, it seems to have grown almost exponentially, as it currently stands at 7,615.430 T.

The second data point is the hash rate of the Ethereum network. This key-value represents the processing power of the blockchain platform, and "is a calculated numerical value that specifies an estimate of how many hashes are being generated by Ethereum miners trying to solve the current Ethereum block or any given block. The calculation uses the current mining difficulty and the average Ethereum block time between mined blocks versus the defined block time as variables..." [50]. Represented as TeraHashes per second (TH/s), the current hash rate of Ethereum stands at 596.1013131 TH/s.

The last and final data point is the uncle rate of the Ethereum network. This value represents the rate at which uncle blocks are appearing on the blockchain. "Uncle blocks are created in Ethereum blockchains when two blocks are mined and submitted to the ledger at roughly the same time. Only one can enter the ledger as a block, and the other does not" [51]. The appearance of an uncle indicates the gas necessary for the transaction – if the uncle rate is low, the gas limit must be increased, and if the uncle rate is high, the gas limit must be decreased. Represented as a percentage, the current uncle rate of Ethereum stands at around 4.59%.

The reason all three data points do not exist for the Internet Computer is because the two consensus algorithms differ. Ethereum's network difficulty, hash rate, and uncle rate are only applicable to its Proof-of-Work consensus protocol, which does not fully apply to the Internet Computer. Therefore, no such figures are produced by Dfinity's platform.

Difficulty (T): 7,615.430 T [52]

Hash Rate (TH/s): 596.1013131 TH/s [53]

Uncle Rate (%): 4.59% [54]

## 4.3 Application Scalability Comparison

### 4.3.1 Lines of Code (LOC)

Now we would like to compare the lines of code of each architectural component within the original Ampleforth stablecoin on Ethereum, against our prototype on the Internet Computer. To briefly introduce, Lines of Code (LOC) refers to "a software metric used to measure the size of a computer program by counting the number of lines in the text of the program's source code" (source). While there are many different metrics to base this calculation on, this project will be basing its numbers on Logical Lines of Code (LLOC). This means that an empty line or comment will not be included in the final count. To measure each source code most accurately, we then utilized a functionality on Visual Studio Code called VSCodeCounter. This provided us with a consistent method that produces accurate results.

Furthermore, to make this comparison as fair as possible, we only measured the LOC of the three main architectural components of the Ampleforth stablecoin – the ERC-20, Supply Policy, and the Market Oracle. While the original stablecoin contained many other pieces of code, we wanted to make sure it was truly comparable with our prototype which was still in its early stages. Therefore, it is important to note, these figures could greatly shift in the future with several updates. One other factor that could also impact these figures is if the ERC-20 canister we are utilizing is updated to be compatible with floats. Currently, we are only able to process integers, therefore there are lines of code dedicated to tackling this issue. Therefore, this number would be greatly reduced once resolved. A fully functional version of our Ampleforth prototype on the main net of the Internet Computer would look quite different from our current result as we are lacking the full implementation, so we are hoping to take further measurements in the future.

ERC-20 Original: 192

ERC-20 IC Canister: 198

Supply Policy Original: 108

Supply Policy IC Canister: 19

Market Oracle Original: 144

Market Oracle IC Canister: 64

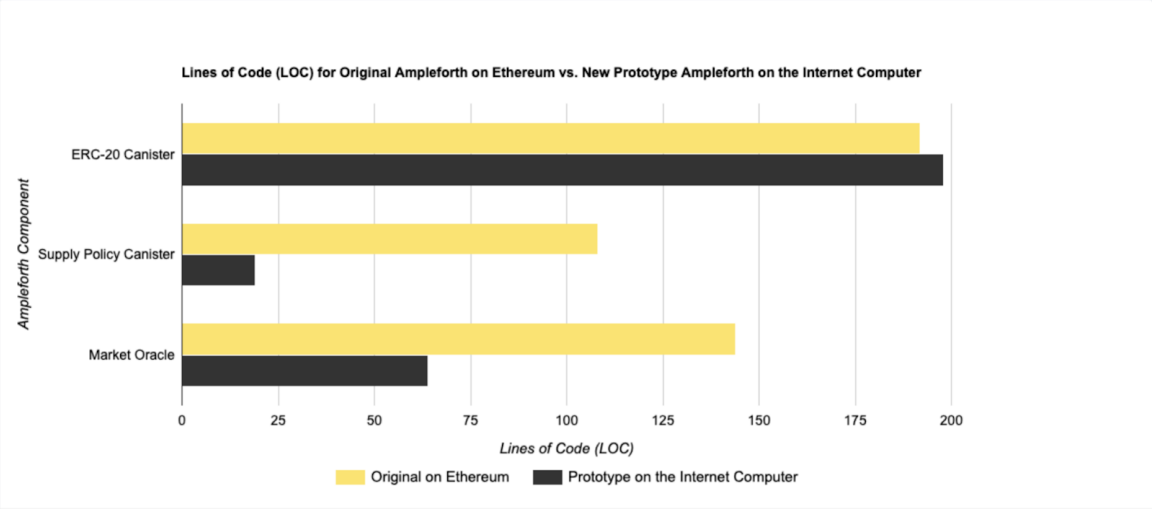


Figure 20: Lines of Code (LOC) Comparison

## 4.4 Future Improvements

Our current prototype is not the complete implementation of the currently existing Ampleforth stablecoin on Ethereum – therefore our primary goal is certainly to achieve this milestone on the Internet Computer. However, we have also started brainstorming ways to improve this implementation once complete. We would like to utilize this section to introduce three of these ideas that could potentially take our project further.

The first improvement focuses on the security of Ampleforth and is done through the addition of two new functions within the oracle – the Manifest function and the Accountability function. To avoid the lack of transparency and potential misbehavior, the Manifest function is an oracle design that "requires oracles to explicitly declare their manifest... Such a manifest would contain oracle metadata (like oracle contact information), deployed data sources, intended frequency of oracle updates, and precise description of the price derivation" [55]. This would provide all parties with a more secure guarantee of the reliability of the information they are receiving from the oracle itself. To hold oracles accountable in the blockchain ecosystem, the Accountability functions aim to punish misbehaving oracles. "The platform is initiated with an accountability contract with which oracles deposit some specific amount of crypto assets. Whenever an oracle misbehaves, violating its manifest, it can be punished by the accountability contract which verifies the reported misbehavior and executes all (or the part of) the deposit" [55]. This would introduce incentives to oracles, keeping them in check.

The second improvement is focused on the decentralization of Ampleforth, specifically on the market oracle. The main idea is to increase the number of verified data providers for the market oracle to query. Currently, the original Ampleforth's market oracle is "powered by nine different incoming price feeds from three different aggregators" [56]. For example, one of their sources, Chainlink, provides "daily AMPL/USD volume-weighted average price (VWAP) from... different aggregators: BraveNewCoin, Kaiko, CryptoCompare, AnyBlockAnalytics, and CryptoAPIs" [56]. This makes it easier for the stablecoin to aggregate and compare needed information for accuracy. For our current prototype, however, we are either only pulling our information from CoinGecko, or utilizing the Ampleforth rebase history for our testing. Therefore, we believe it is a necessary improvement to increase our number of verified data providers in the future.

The third and final improvement is a suggestion to the previous improvement, where we attempt to connect to Chainlink ourselves to gain access to the same decentralized market providers as Ampleforth. As stated on their website, "Chainlink's decentralized oracle network provides reliable, tamper-proof inputs and outputs for complex smart contracts on any blockchain" [57].

Currently, Chainlink does not allow for integration with the Internet Computer. As Chainlink expands to other blockchains there is potential to increase our protocol's decentralization by querying a decentralized oracle. Improving the decentralization of our oracle is a pivotal step for both accuracy and security. By having multiple verified data providers it is easier to track anomalies in the data. Furthermore, the risk of having no market data is low since broken oracles can easily be replaced with another. Ensuring the security of an oracle is essential. Ampleforth is completely reliant on oracle data and would immediately encounter serious issues if there was no data provider for a rebase.

# 5 Conclusion

With the performance limitations of blockchains currently hosting the Ampleforth protocol clearly established, it has become quite clear that there is a need for a stable cryptocurrency on a more tamperproof, efficient, and fault-tolerant platform. Therefore, for our project, we designed and developed an Ampleforth implementation on the Internet Computer to discover whether it would improve specific attributes such as decentralization, scalability, and security. This prototype had three functional components – the market oracle, supply policy, and ERC-20 token canister – which we were able to utilize and test for research results. In addition to this prototype, we also:

1. Created local wallets with encrypted keys.
2. Allowed for the exchange of tokens between wallets on the network.
3. Encoded the protocol's rebase mechanism into the token canister so that a rebase occurs across the network.
4. Collected data for the rebase from certified market oracles.

To measure the impact of building an implementation of Ampleforth on the Internet Computer, we conducted several different tests between the two platforms. The first test measured the execution times for Token and SupplyPolicy canisters on the Internet Computer versus Ethereum and produced the results in Figure 12 and Figure 13. The direct comparison of these results can be seen in Figure 14. The second test then compared the time taken to retrieve oracle data on the Internet Computer (sodium network) against the time taken by Ampleforth's market oracle running on Ganache (personal Ethereum blockchain). The results can be seen in Figure 15. Finally, the third test considered how long it takes for each respective ERC-20 to transfer tokens from Wallet A to Wallet B. The results can be viewed in Figure 16. When comparing these results between the two blockchain platforms, it becomes clear that there is a dramatic difference in Ampleforth execution speeds. However, as we have mentioned already only the speed of the ERC-20 drastically increases scalability. ERC-20 speeds are likely to change when the prototypes are tested on the Internet Computer main-net as sodium tests do not account for the transactional delay. If this was accounted for, prototypes deployed to the Internet Computer main-net are likely to perform and scale far more efficiently than on Ethereum. The Internet Computer main net simply has a far superior block rate and hence can deal with transactional delay much more efficiently.

While we were able to accomplish our main objectives, there were several issues that prevented a straightforward implementation. The first was different build versions between our ERC-20 and market oracle. This prevented us from achieving our goal of a fully automatic implementation of Ampleforth, though we were able to come up with an alternative architecture to overcome this issue temporarily. The second was how we were unable to run prototype tests on the main net due to the extended launch date of Genesis. Due to Dfinity's blockchain being so brand new, the timeline of our project, unfortunately, fell a bit short before the launch date. Therefore, our current prototype has been built upon the Sodium net instead. As we are provided more time

after the initial phase of this project, we are planning to resolve all these issues for more accurate implementation.

In addition to fixing existing issues, we hope to also implement additional features for the stablecoin to have better security and decentralization. The first implementation is to add manifest and accountability functions. This will help users avoid the lack of transparency in the data they receive from oracles, and also hold them accountable. The second implementation is to increase the number of verified data providers for the Market Oracle to query. Currently, our prototype only has one verified data provider – CoinGecko – and we believe this can be improved. To have the same, if not more data providers than the official Ampleforth stablecoin would be ideal. Finally, to build upon the previous implementation, we would like to connect our oracle to Chainlink so that market providers are truly decentralized.

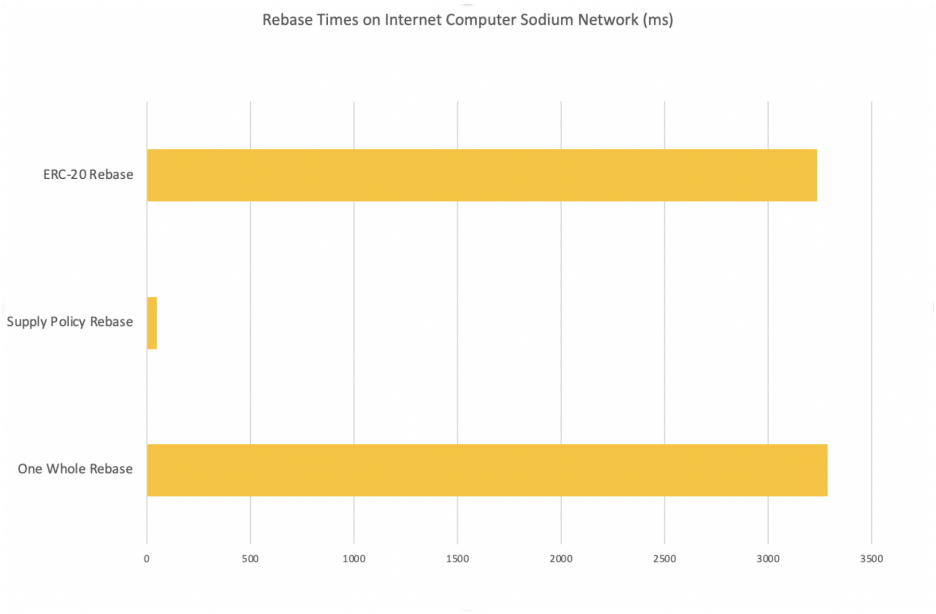


Figure 12: The Internet Computer Rebase Times

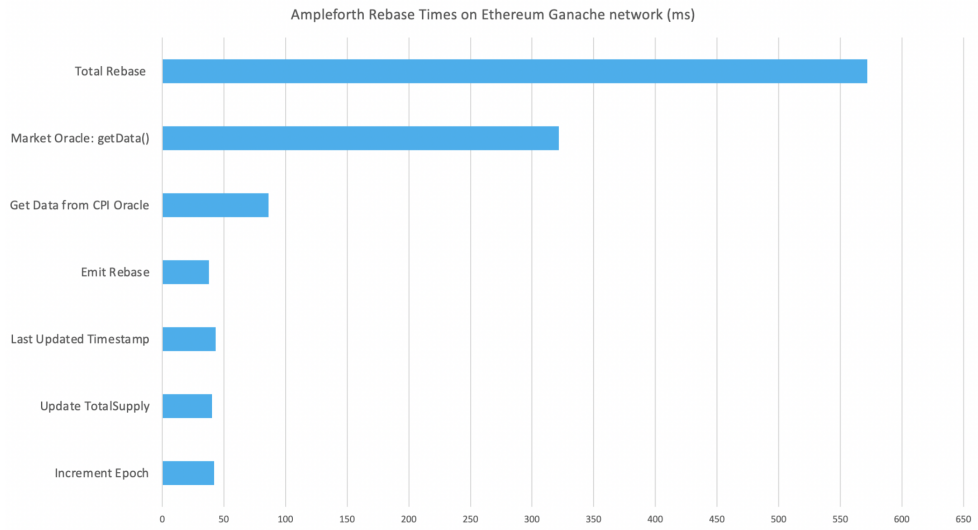


Figure 13: Ethereum Execution Times

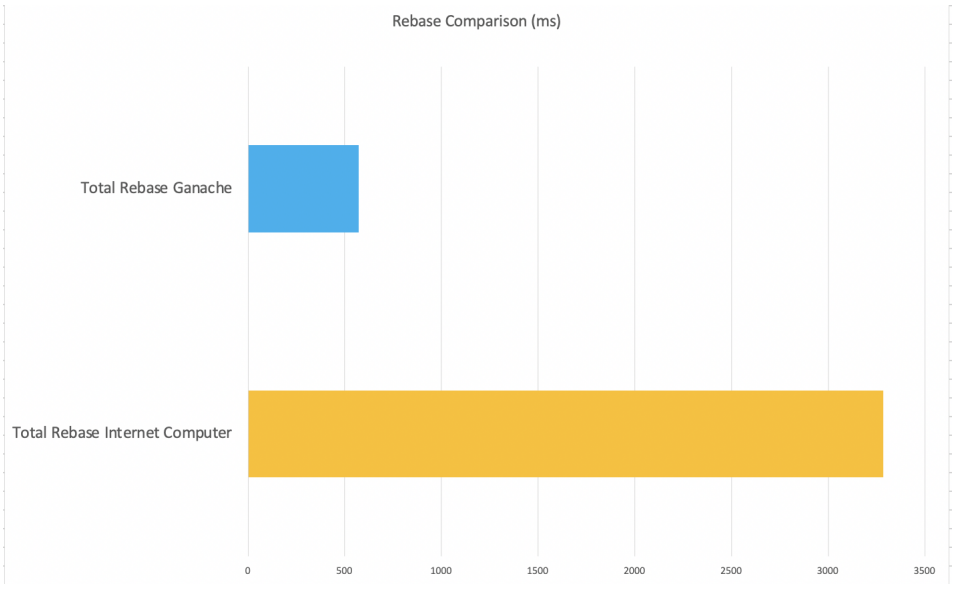


Figure 14: Rebase Times Comparison

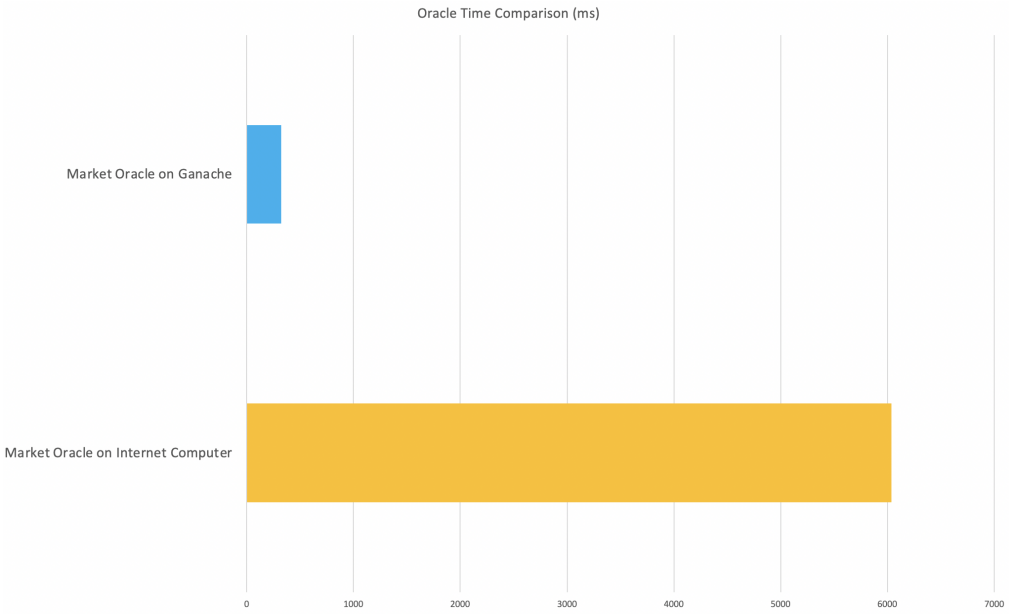


Figure 15: Oracle Speed Ganache vs Sodium

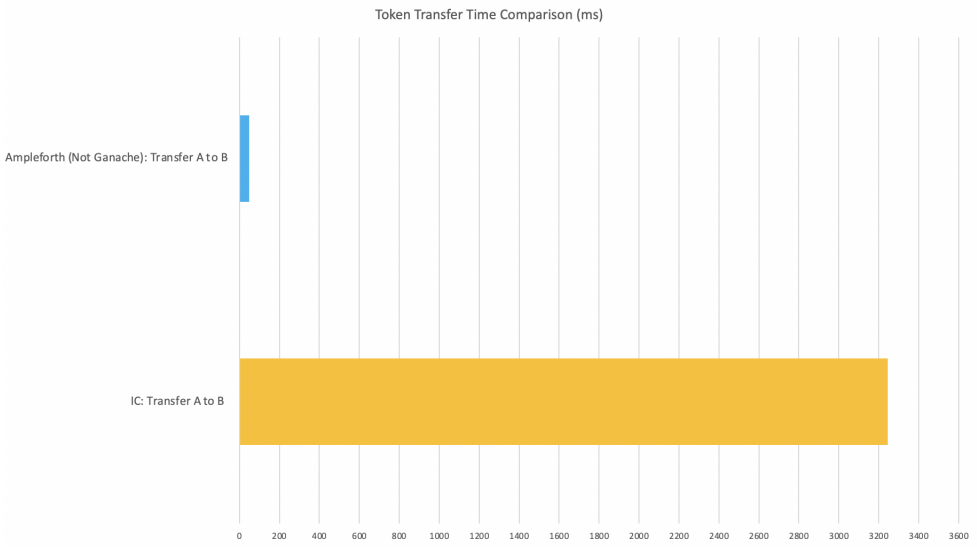


Figure 16: Token Transfer on Ethereum vs Internet Computer

## 6 Acknowledgments

First and foremost, we would like to express our sincere gratitude to our advisors/mentors, Profs. Luyao Zhang, Olivier-Gilles Marin, and Yulin Liu for their continuous support during our study and research, and for their patience, wisdom, and immense knowledge. Their guidance was what truly helped us succeed with this initiative.

We would also like to thank the team at the DFINITY Foundation - specifically Nicolas Zoghb, Dylan Miller, Enzo Haussecker, Igor Lilic, Elizabeth Yang, and Michael Hunte for all the necessary support and in-depth information that was provided to us during this study. Without their help, we would not have achieved as much as we have, and we are extremely grateful.

## References


[1] S. Coelho-Prabhu, "A beginner's guide to decentralized finance (defi)," *Coinbase*, 2020.

[2] "Frequently asked questions," *DFINITY*, 2021.

[3] "The internet computer," *DFINITY*, 2021.

[4] B. Simon, "Stability, elasticity, and reflexivity: A deep dive into algorithmic stablecoins," *Deribit Insights*, 2020.

[5] M. R.-C. Evan Kuo, Brandon Iles, "Ampleforth: A new synthetic commodity," *Ampleforth*, July 12, 2019.

[6] CertiK, "The blockchain trilemma: Decentralized, scalable, and secure?" Oct 2019. [Online]. Available: https://medium.com/certik/the-blockchain-trilemma-decentralized-scalable-and-secure-e9d8c41a87b3

[7] A. Kadiyala, "Nuances between permissionless and permissioned blockchains," *Medium*, Feb 2018. [Online]. Available: https://medium.com/@akadiyala/nuances-between-permissionless-and-permissioned-blockchains-f5b566f5d483

[8] E. Kuo, B. Iles, and M. Rincon-Cruz, "Ampleforth: A new synthetic commodity," May 2019.

[9] M. Scherer, "Performance and scalability of blockchain networks and smart contracts," 2017. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1111497/FULLTEXT01.pdf

[10] S. M. Hosseini Bamakan, A. Motavali, and A. Babaei, "A survey of blockchain consensus algorithms performance evaluation criteria," *Expert Systems with Applications*, vol. 154, p. 113385, 04 2020.

[11] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[12] P. Wackerow, "Proof-of-stake (pos)," *ethereum.org*, 2020. [Online]. Available: https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/

[13] W. Mahmood and A. Wahab, "Survey of consensus protocols," *papers.ssrn.com*, Oct 2018. [Online]. Available: https://ssrn.com/abstract=3556482

[14] S. Bano, A. Sonnino, M. Al-Bassam, S. Azouvi, P. McCorry, S. Meiklejohn, and G. Danezis, "Sok: Consensus in the age of blockchains," in *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, ser. AFT '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 183–198. [Online]. Available: https://doi.org/10.1145/3318041.3355458

[15] *Polkadot Network*. [Online]. Available: https://polkadot.network/launch-npos/

[16] I. Polosukhin, "Thresholded proof of stake," *Medium*, Apr 2020. [Online]. Available: https://medium.com/nearprotocol/thresholded-proof-of-stake-67b74e616a92#:~:text=NEAR%20uses%20an%20election%20mechanism

[17] *NEAR Protocol*, Oct 2019. [Online]. Available: https://near.org/papers/the-official-near-white-paper/#how-near-works

[18] D. Fucci, S. Romano, M. Baldassarre, D. Caivano, G. Scanniello, B. Thuran, and N. Juristo, "A longitudinal cohort study on the retainment of test-driven development," vol. Rev.1, May 2018. [Online]. Available: https://arxiv.org/pdf/1807.02971.pdf

[19] S. Richards, "Gas and fees," *ethereum.org*, Jan 2021. [Online]. Available: https://ethereum.org/en/developers/docs/gas/

[20] Etherscan, "Ethereum average gas price," *ycharts.com*, 2021. [Online]. Available: https://ycharts.com/indicators/ethereum_average_gas_price#:~:text=Ethereum%20Average%20Gas%20Price%20is

[21] BtcTurk, "What is tron (trx)?" *Medium*, Oct 2020. [Online]. Available: https://btcturk.medium.com/what-is-tron-trx-4f87f845288f#:~:text=On%20the%20TRON%20network%2C%20transaction

[22] *undefined*. [Online]. Available: https://wiki.polkadot.network/docs/en/learn-transaction-fees

[23] R. Belchior, A. Vasconcelos, S. Guerreiro, and M. Correia, "A survey on blockchain interoperability: Past, present, and future trends," 2020.

[24] P. Wegner, "Interoperability," *ACM Comput. Surv.*, vol. 28, no. 1, p. 285–287, Mar. 1996. [Online]. Available: https://doi.org/10.1145/234313.234424

[25] D. Williams, "Announcing internet computer "mainnet" and a 20-year roadmap," *Medium*, Feb 2021. [Online]. Available: https://medium.com/dfinity/announcing-internet-computer-mainnet-and-a-20-year-roadmap-790e56cbe04a

[26] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Generation Computer Systems*, vol. 107, Aug 2017. [Online]. Available: https://arxiv.org/pdf/1802.06993.pdf

[27] *Smith + Crown*, 2021. [Online]. Available: https://smithandcrown.com/glossary/governance-token/#:~:text=Governance%20tokens%20confer%20holders%20the

[28] [Online]. Available: https://docs.emptyset.finance/governance/protocol

[29] [Online]. Available: https://basiscash.fyi/governance/#:~:text=Those%20proposals%20deemed%20worthy%20to

[30] 2020. [Online]. Available: https://docs.frax.finance/smart-contracts/governance

[31] 2021. [Online]. Available: https://www.ampleforth.org/governance/

[32] Jul 2019. [Online]. Available: https://smithandcrown.com/research/distributed-governance-beyond-token-based-voting/

[33] Mar 2021. [Online]. Available: https://academy.binance.com/en/articles/elastic-supply-tokens-explained

[34] N. Cachanosky, "Elastic cryptocurrency supplies: A step in the right direction – aier," Sep 2018. [Online]. Available: https://www.aier.org/article/elastic-cryptocurrency-supplies-a-step-in-the-right-direction/

[35] S. Gupta, "Cryptocurrency volatility - a friend or a foe. volatility in trading markets." Mar 2020. [Online]. Available: https://aithority.com/guest-authors/cryptocurrency-volatility-a-friend-or-a-foe/

[36] [Online]. Available: https://messari.io/asset/frax/metrics

[37] [Online]. Available: https://messari.io/asset/empty-set-dollar/metrics

[38] [Online]. Available: https://messari.io/asset/ampleforth/metrics

[39] [Online]. Available: https://messari.io/asset/basis-cash/metrics

[40] "Ampleforth rebase history." [Online]. Available: https://www.coin-tools.com/ampl/ampl-rebase-history/

[41] Enzoh, "enzoh/motoko-token." [Online]. Available: https://github.com/enzoh/motoko-token

[42] "Ampleforth." [Online]. Available: https://github.com/ampleforth

[43] P. Siriwardena, "The mystery behind block time," Jul 2018. [Online]. Available: https://medium.facilelogin.com/the-mystery-behind-block-time-63351e35603a

[44] [Online]. Available: https://www.dfinityexplorer.org/#/

[45] [Online]. Available: https://etherscan.io/chart/blocktime

[46] K. Ziechmann, "Transactions," Mar 2021. [Online]. Available: https://ethereum.org/en/developers/docs/transactions/#

[47] "Dexon comparison to other blockchain." [Online]. Available: https://dexon-foundation.github.io/wiki/Blockchain-Comparison

[48] [Online]. Available: https://etherscan.io/

[49] "Ethereum eth network difficulty chart - 2miners." [Online]. Available: https://2miners.com/eth-network-difficulty

[50] "Ethereum hashrate chart." [Online]. Available: https://www.coinwarz.com/mining/ethereum/hashrate-chart

[51] W. Kenton, "What is an uncle block?" Jan 2021. [Online]. Available: https://www.investopedia.com/terms/u/uncle-block-cryptocurrency

[52] [Online]. Available: https://etherscan.io/chart/difficulty

[53] [Online]. Available: https://etherscan.io/chart/hashrate

[54] [Online]. Available: https://ycharts.com/indicators/ethereum_uncle_rate#:~:text=Ethereum%20Uncle%20Rate%20is%20at,long%20term%20average%20of%209.27%25.

[55] B. Liu and P. Szalachowski, "A first look into defi oracles," *CoRR*, vol. abs/2005.04377, 2020. [Online]. Available: https://arxiv.org/abs/2005.04377

[56] E. Kuo, "The ampleforth chainlink oracle integration is going live," Mar 2020. [Online]. Available: https://medium.com/ampleforth/the-ampleforth-chainlink-oracle-integration-is-going-live-16053ccdebd5

[57] "Blockchain oracles for hybrid smart contracts: Chainlink." [Online]. Available: https://chain.link/