M^0 Protocol Whitepaper

Author: M^0 Foundation

Abstract: The core M^0 protocol is a coordination layer for permissioned actors to generate *M*. *M* is a fungible token that can be generated by locking Eligible Collateral in a secure off-chain facility. The protocol enforces a common set of rules and safety procedures for the management of *M*.

I. Introduction

The *core* M^0 protocol (which excludes periphery contracts[1] and is hereon referred to simply as the M^0 protocol) is a coordination layer for permissioned actors to generate ***M***. *M* is a crypto asset whose value is designed to be a robust representation of an exogenous collateral basket – a relationship enforced by the financial structure and market incentives of its generators. The purpose of *M* is to become a superior building block for value representation, by combining the convenience of digital money with the risk profile of physical cash. While holders may find this construct appropriate as a vehicle for *cryptodollar* use cases, developers and financial services providers might be interested in it as raw material for the build out of novel products and services – including as collateral for cryptodollars.

When cash was primarily a physical construct, it had several properties that its holders found desirable but have since been lost in the process of digitization. Physical cash is first and foremost self-custodial; it's a bearer instrument which guarantees that it cannot be frozen or seized without due process in the holder's jurisdiction. For example, holders in one nation do not need to be concerned that a far away government can *turn off* the cash in their pocket. This feature allows cash to be credibly neutral, which means that it cannot discriminate against any specific holder. Second, physical cash does not carry additional counterparty risk, it is as good as holding reserves at the issuer's central bank. Finally, physical cash is generally fungible with itself, which is to say that except in extraordinary circumstances where holders are wary to accept a certain serial number, no bill is more or less valuable than another. The downsides of physical cash are that it cannot be transferred electronically, it must be stored in a physically safe location which becomes exponentially more difficult to secure as the quantity held rises, is becoming less broadly accepted as a means of payment, and lacks general digital properties that can allow seamless composability and programmability.

[1] A periphery contract is a smart contract that adds supplement functionality, but exists outside of the core protocol.

Most users have historically believed those properties to be inherited by bank deposits, as if they were merely a digital representation of cash — the fallacy of this false equivalence is becoming evident due to the stress increasingly experienced by the banking system, and is exacerbated by the global nature of cryptodollars. What we call *digital cash* today is typically a commercial bank deposit that can be transferred electronically. It has several beneficial features such as the ability to earn interest, the ability to be transferred over the internet and across large distances, and it offers the peace of mind of digital and custodial security. It is also the most widely accepted form of payment through credit cards, debit cards, ACH, and SWIFT. Unfortunately bank deposits lose many of the desirable features of physical cash. Bank deposits are implicitly custodial and can be seized or frozen without due process in the holder's jurisdiction – they are not credibly neutral. Due to the inherent characteristics of fractional reserve banking, bank deposits hold significant counterparty risk and are only as valuable as the specific bank's balance sheet permits. For this reason they are also not fungible. A deposit in one bank cannot be treated as equal to a deposit in another, and thus introduces exorbitant clearing times between payments, as well as compounding complexity throughout the system.

M is credibly neutral by design, it is by default self-custodial and fungible. Each *M* is the same as every other *M* and there is no ability for the protocol to discriminate against any specific holder(s). *M* is stored and tracked on blockchains, and thus can be stored more securely at scale than physical cash. *M*'s current instantiation is intended to be generated using short term US T-bills, representing the lowest level of counterparty risk excepting physical cash and bank reserves within the US dollar system. The T-bills used to generate *M* must be held exclusively throughout a network of orphaned, bankruptcy-remote entities, which are customized to interact with the M^0 protocol while meeting the formalities of the existing legal system. *M* can be sent anywhere in the world instantaneously using the blockchain rails on which it exists. Interest flowing to the T-bill collateral can be partly collected by the protocol and democratized across permissioned issuers and distributors.

We refer to *M* as raw material for value representation, and not necessarily a cryptodollar in its own right, because the system relies on permissioned issuers (known as *Minters* in the protocol) for generation and distribution. These Minters should be compliant with all applicable regulations and may decide to distribute their own product, for example by wrapping the *M* token in a cryptodollar contract in a way that best meets their requirements. In this capacity, *M* becomes a monetary building block on top of which novel products can be built.

In summary, the M^0 protocol introduces a superior coordination mechanism that democratizes access to the generation and management of programmable, digital cash instruments. It is an infrastructure layer not for the simplistic tokenization of real world bank deposits, but a much more sophisticated way to provide access to the liquidity on high-quality collateral. M^0 intends to redesign the monetary vertical stack, rather than build an additional layer on top of what has ultimately become byzantine infrastructure.

II. Protocol

The M^0 protocol is a set of immutable smart contracts implemented for the Ethereum Virtual Machine. It receives all external inputs from a governance mechanism called the M^0 Two Token Governor, *TTG*, (see III. Governance). The M^0 protocol is a coordination tool for actors permissioned by governance, namely *Minters*, *Validators*, and *Earners*.

II.I Operation

Aside from users accessing the *M* token, which is permissionless, only actors permissioned through the TTG mechanism have access to the M^0 Protocol. The primary actors in the protocol are called Minters and Validators (see III.II Governance Controlled Protocol Actors). Once permissioned by the TTG, Minters and Validators are able to access certain methods in the smart contracts which facilitate the creation, maintenance, and destruction of *M*. *M* is a standard ERC20 token. The core operating condition of the M^0 protocol is to ensure that all *M* in existence is backed by an equal or greater amount of value ***Eligible Collateral*** that is held in an ***Eligible Custody Solution*** (see IV.II Off-Chain Actors and Components). In this context, the protocol acts as the enforcer of a set of rules which controls the generation of *M*, the validation of collateral custody and value, and the assessment of fees when appropriate.

II.I.I Generation of *M*

In order to generate *M*, Minters must have a sufficient off-chain balance of Eligible Collateral which is represented on-chain by a frequently updated and validated number, known as the on-chain ***Collateral Value***. Minters call the **Update Collateral** method to put this number on-chain. They must pass the amount, the list of signing Validators, a list of timestamps associated with the Validator signatures, and valid signature data (from Validators). Whenever timestamps and signature data are passed to a method, the contracts will take the minimum timestamp and the minimum threshold of signatures as defined by the TTG (see III.III Governance Controlled TTG Parameters). Optionally, they can pass a hash of arbitrary metadata and any open Retrieval IDs (see II.I.IV Retrieving Free Collateral) into the method as an argument. The Metadata Hash can be used to retrieve the actual off-chain metadata, which can serve to add context to the update, while Retrieval IDs allow Minters to remove outstanding balance subtractions (see II.I.IV Retrieving Free Collateral). Signature validation may either use standard *ecrecover*[2], which allows for the Minter to obtain the signature off-chain, or EIP-1271 on-chain contract signatures[3]. The collateral balance is an attestation to the value of the Eligible Collateral held in an Eligible Custody Solution (see IV.II Off-Chain Actors and Components for further details).

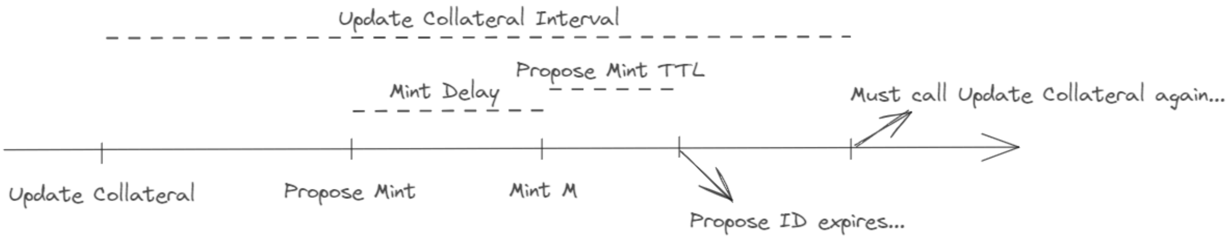
In order to post the value of their Eligible Collateral on-chain, the Minter will need to provide the signature data (from Validators) in the transaction. The presence of the Validator's signature is to reinforce that the value of the Eligible Collateral is correct and reflects the most up-to-date snapshot of the off-chain balances. Minters must update their Eligible Collateral number on-chain and with a valid Validator signature *at least* once every ***Update Collateral Interval*** (see II.III Governance Controlled Protocol Parameters). If a Minter fails to call Update Collateral within Update Collateral Interval of the previous time they called it, their on-chain Collateral Value is assumed to be 0. If the Minter cannot provide valid signature data (from Validators), they cannot successfully call the method. Each time this method is called it will accrue the ***Minter Rate*** (see II.I.II Protocol Fees) on the Minter's current balance of Owed M. If any rules are being violated at the time of the method being called, it will also charge ***Penalty Rate*** on the Minter's balance which is in violation (see II.I.II Protocol Fees).

Example: Eligible Collateral for M has been deemed to be 0-90 day T-bills. Minter 1 has $10,000,000 of T-bills sitting in an Eligible Custody Solution. Minter 1 calls Update Collateral and passes 10,000,000, and a valid Validator signature as arguments. The on-chain Collateral Value of the Minter is now 10,000,000. The next day, $1,000,000 of the T-bills mature and convert to bank deposits, which are not considered Eligible Collateral. The Minter calls Update Collateral and passes 9,000,000, and a valid Validator signature as arguments. The on-chain collateral balance of the Minter is updated to 9,000,000.

Once a Minter has updated their on-chain collateral they are able to generate *M*. They do so by calling the **Propose Mint** method and passing in the amount of *M* they'd like to generate and the address which they would like to generate the *M* to. Once this method is called, it will first call **Get Present Amount** on the Minter's current balance of Owed M. It will then check to ensure that the on-chain Collateral Value multiplied by the ***Mint Ratio*** (see II.III Governance Controlled Protocol Parameters), including the amount they are currently attempting to generate and/or ***Retrieve*** (see II.I.IV Retrieving Free Collateral), is greater than the amount of total Owed M from the Minter. If these checks are passed the method will output a ***Mint ID*** which corresponds to the Propose Mint. A Minter can only have one outstanding Mint ID at any given time. If after the ***Mint Delay*** (see II.III Governance Controlled Protocol Parameters) the Mint ID has not been canceled by the Validator (see II.I.III Cancel and Freeze), the Minter may call the **Mint M** method and pass the Mint ID as an argument to execute the Propose Mint. The Mint Delay was

[2] https://soliditydeveloper.com/ecrecover
[3] https://eips.ethereum.org/EIPS/eip-1271

introduced to avoid atomic Update Collateral calls and Mint calls, and to provide the network of Validators with sufficient opportunity to intervene in the minting process if something is seemingly wrong (see II.I.III Cancel and Freeze). The Minter must call Mint M before the ***Propose Mint Time To Live*** has expired (see II.III Governance Controlled Protocol Parameters). Minters can destroy Owed M at any time by calling the **Burn** method and passing in their Minter Address and the amount of *M* they'd like to burn as arguments. Any address can repay *M* owed by a Minter by calling the Burn method and passing in the amount and Minter address as arguments.



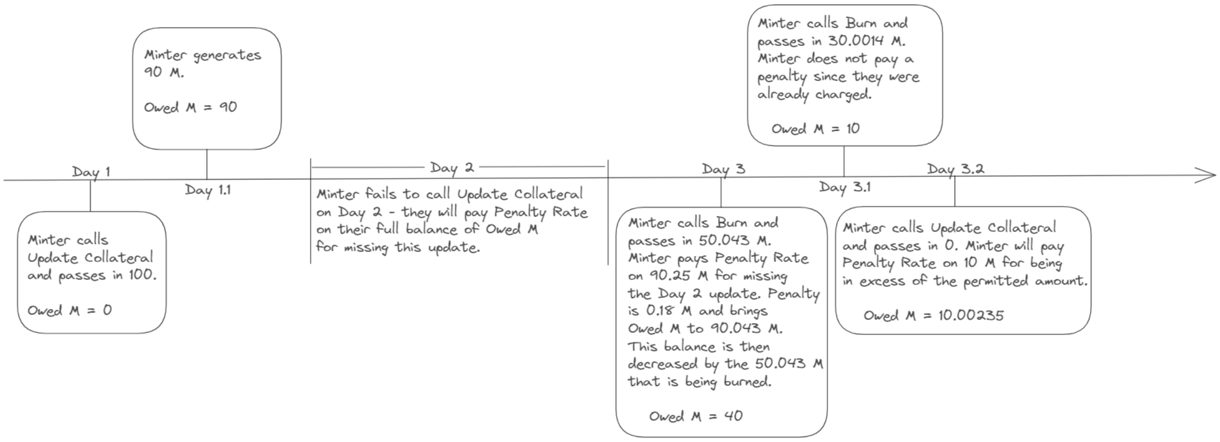
The *M* generation process and its limiting factors

II.I.II Protocol Fees

There are two fees assessed on Minters in the M^0 protocol. The first is called Minter Rate, a governance controlled parameter, which is levied continuously on the Minter's balance of Owed M. This fee compounds on a continuous basis. The beneficiaries of Minter Rate are the ***Earn Mechanism*** (see II.I.V The Earn Mechanism) and the ***ZERO*** holders (see III. Governance). The second is Penalty Rate, another governance controlled parameter, which is levied on balances that are in violation of protocol rules and has the same beneficiaries as the Minter Rate. One of the primary invariants of the protocol is that the balance of a Minter's Owed M should not exceed the on-chain Collateral Value (sans open Retrieval IDs) multiplied by the Mint Ratio. Penalty Rate is imposed upon any balance in excess of this amount. If a Minter has not called Update Collateral within Update Collateral Interval, they will incur Penalty Rate on their entire balance of Owed M for each Update Collateral Interval that they miss – i.e. when Update Collateral is not called in the TTG-specified time, the system interprets its value to be zero. Unlike Minter Rate, Penalty Rate is not continuously levied on the Minter's balance of Owed M, but is charged discretely as a one-time percentage fee on their delinquent balance at the moment the balance is checked, and then added to the Minter's Owed M. Collecting the Minter Rate is contained in the Get Present Amount method, which is exclusively embedded in all other Minter methods and cannot be called independently. It is called in most methods, including Burn. The Minter Rate is mechanically affected by updating a global index value which is applied to all Owed M in the Minter's balance, as is Penalty Rate. Penalty Rate is contained in the **Impose Penalty** method, which is exclusively embedded in the Update Collateral,Burn, and **Deactivate Minter** methods. When Impose Penalty is called in conjunction with Update Collateral, it checks for both missed Update Collateral Interval periods and the Minter's balance of Owed M relative to their on-chain Collateral Value discounted by the Mint Ratio. When it is called in conjunction with Burn, it only checks for missed Update Collateral Interval periods. This is done to ensure that the Minter is not penalized on the same errant balance more than once. Impose Penalty will also account for whether it has already been called in the current Update Collateral Interval period and will not charge a Minter twice for the same missed period.

Example: A Minter calls Update Collateral; the Get Present Amount method is called along with the Impose Penalty method. The Minter's balance of Owed M prior to the method call is 8,000,000 M. First, Get Present Amount is used to fetch the latest index and apply the Minter Rate to the Minter's balance of Owed M, accounting for continuous compounding. Assume that this increases the Minter's Owed M to 8,000,010 M. If the Minter's on-chain Collateral Value is 8,000,000 and the Mint Ratio is 90%, then the maximum amount of Owed M Minter 1 should have is 7,200,000 M– but the actual on-chain number is 8,000,010 M. Therefore Minter 1 will incur Penalty Rate on (8,000,010 M - 7,200,000 M)=800,010 M. If Penalty Rate is 0.01%, then the Minter's Owed M is incremented to 8,000,010 M + 800,010 M 1.0001 = 800,090.001 M.*

The following is a diagram which demonstrates a hypothetical sequence where a Minter incurs Penalty Rate charges. The example below describes this hypothetical sequence.



Minter calls Update Collateral and passes in 100.

Owed M = 0

Minter generates 90 M.

Owed M = 90

Minter fails to call Update Collateral on Day 2 - they will pay Penalty Rate on their full balance of Owed M for missing this update.

Minter calls Burn and passes in 50.043 M. Minter pays Penalty Rate on 90.25 M for missing the Day 2 update. Penalty is 0.18 M and brings Owed M to 90.043 M. This balance is then decreased by the 50.043 M that is being burned.

Owed M = 40

Minter calls Burn and passes in 30.0014 M. Minter does not pay a penalty since they were already charged.

Owed M = 10

Minter calls Update Collateral and passes in 0. Minter will pay Penalty Rate on 10 M for being in excess of the permitted amount.

Owed M = 10.00235

Hypothetical sequence where a Minter incurs Penalty Rate charges.

Example (some balances are rounded):

Assume that the Mint Ratio is 90%, Update Collateral Interval is 24 hours, Minter Rate is 5% APY (and therefore ~0.00058% per hour), and Penalty Minter Rate is 0.02%.

<u>On Day 1</u>, a Minter calls Update Collateral and passes in 100 as the value. Its Owed M is 0.

Later that day (Day 1.1) the Minter generates 90 M.

On Day 2 the Minter fails to call Update Collateral.

*On Day 3 the Minter calls Burn and passes in 50.043 M. Assume that less than 48 hours have passed since the Mint call on Day 1. First, Get Present Amount is called and applies the latest index to the Minter's balance of 90 M. This increases the Minter's Owed M to (90 * (1 + (0.0000058 * 48))) = 90.025 M. Next, Impose Penalty is called and is applied to the Minter's updated balance. Since the Minter missed one Update Collateral Interval period (on Day 2), they are penalized one time. The Minter's Owed M is increased to (90.025 * (1 + (0.0002 * 1)) = 90.043 M. The amount passed into the Burn method (50.043) is now subtracted from this new balance and reduces the Minter's Owed M to (90.043 - 50.043) = 40 M. Recall that burn only checks for missed Update Collateral Interval periods and does not check the Minter's current on-chain Collateral Value and therefore does not penalize any currently errant balance. The Minter then calls Burn again on Day 3 and passes in 30.0014 M. Assume 6 hours have passed since the previous Burn call. First, Get Present Amount is called and applies the latest index to the Minter's balance of 40 M. This increases the Minter's Owed M to (40 * (1 + (0.0000058 * 6))) = 40.0014 M. Impose Penalty is then run but does not charge the Minter an additional Penalty Rate because it has already paid for all missed Update Collateral Interval periods. The amount passed into the Burn method (30.0014) is now subtracted from this new balance and reduces the Minter's Owed M to (40.0014 - 30.0014) = 10 M. Finally, also on Day 3, the Minter calls Update Collateral and passes in 0 (most likely because their Eligible Collateral has matured and is sitting in ineligible bank deposits). Assume another 6 hours have passed. Once again, Get Present Amount is run to apply the latest index to the Minter's balance of M. This increases the Minter's Owed M by (10 * (1 + (0.0000058 * 6))) = 10.00035 M. Next Impose Penalty is called. There are no charges for missed periods because these were already paid when the Minter called Burn. A check for an errant balance is now run and produces 10.00035 M since the Minter's entire balance is in excess of their maximum permitted Owed M due to the on-chain Collateral Value being 0. The Minter's Owed M is increased to (10.00035 * 1.0002) = 10.00235 M.*

II.I.III Cancel and Freeze

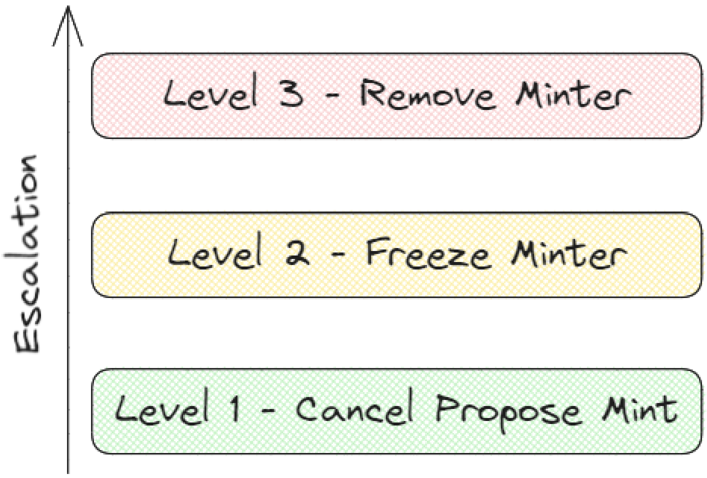
There are two methods which can be used to stop an errant generation of *M* or to stop an errant Minter in the case of an emergency.

The first method, **Cancel**, can be called by any Validator on any Mint ID associated with the generation of *M*. The calling actor must pass the Mint ID as an argument to the method. Calling this method will cancel the specified Mint ID and cancel the proposal. The Cancel method can be called at any time until Mint is called. Minters do not have access to the Cancel method because submitting a new Propose Mint will automatically cancel and replace any that are currently outstanding.

The second method, **Freeze**, can be called by any Validator on any Minter by passing the Minter address as the argument of the method. Calling Freeze will disable Propose Mint and Mint for **Minter Freeze Time** (see II.III Governance Controlled Protocol Parameters). It can be called multiple times on the same Minter to reset the Minter Freeze Time window. If Freeze is called during an already existing Minter Freeze Time, the Minter Freeze Time window will restart from the beginning.

Example: A Validator calls freeze on a Minter and the Minter Freeze Time window is 6 hours. After 5 hours, the Validator calls Freeze again on the same Minter. The Minter is now frozen for an additional 6 hours. This means that the Minter will be frozen for 11 hours in total, unless a Validator calls Freeze again before the end of the second 6 hour period.

These methods can be thought of as being a part of a series of escalations the system (first through Validators and then through the TTG) can levy against an errant Minter. The final escalation being removal of the Minter. Removal of a Minter would similarly be done via a TTG proposal.



The three levels of escalation against an errant Minter

II.I.IV Retrieving Free Collateral

Any Minter with an excess of off-chain value (both Eligible Collateral and other forms of value that may reside in the Eligible Custody Solution) relative to their Owed M, can remove this value from the Eligible Custody Solution. They can do so by calling the **Propose Retrieval** method and passing the amount they wish to retrieve from custody. Like most methods this will first call Get Present Amount. After that it will check that the on-chain Collateral Value, after subtracting the amount the Minter is trying to retrieve and any other open Retrieval IDs, is sufficient to support the Minter's remaining balance of Owed M. Unlike Propose Mint, which is limited to one Mint ID at a time, there is not a limit to the number of outstanding Retrieval IDs. If this check passes, it will sideline this balance to be deducted from future calculations where it is relevant. Finally it outputs a ***Retrieval ID***. The subtraction from the on-chain Collateral Value when relevant will remain until the Retrieval ID is closed.

In order to close the Retrieval ID and eliminate the subtraction on the Minter's on-chain Collateral Value, the Minter must pass the Retrieval ID into an Update Collateral. In signing off on this transaction, the Validator is attesting that the Retrieval ID has been fully processed off-chain, or will not be processed at all, and that the new on-chain Collateral Value is correct.

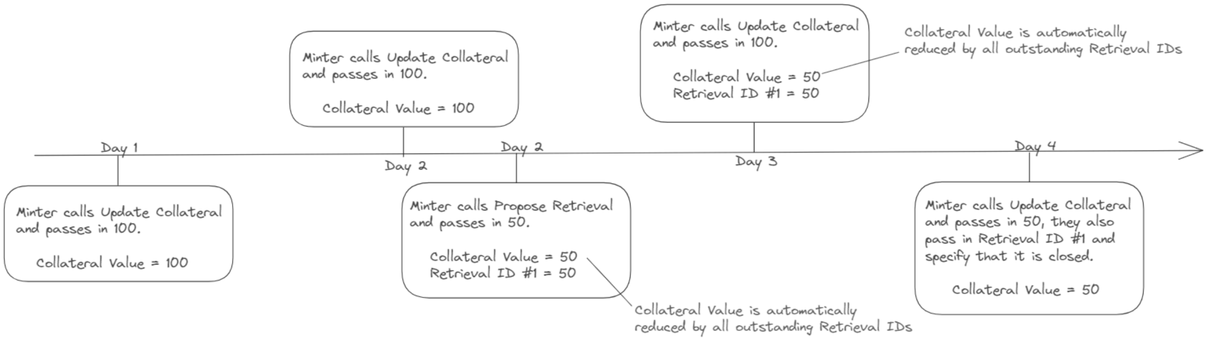

Day 1
Minter calls Update Collateral and passes in 100.
Collateral Value = 100
Minter calls Update Collateral and passes in 100.
Collateral Value = 100
Day 2
Day 2
Minter calls Propose Retrieval and passes in 50.
Collateral Value = 50
Retrieval ID #1 = 50
Collateral Value is automatically reduced by all outstanding Retrieval IDs
Minter calls Update Collateral and passes in 100.
Collateral Value = 50
Retrieval ID #1 = 50
Collateral Value is automatically reduced by all outstanding Retrieval IDs
Day 3
Day 4
Minter calls Update Collateral and passes in 50, they also pass in Retrieval ID #1 and specify that it is closed.
Collateral Value = 50

Diagram demonstrating the effect of Retrieve on a Minter's on-chain Collateral Value.

II.I.V The Earn Mechanism

The ***Earn Mechanism*** is a mechanism in the protocol which allows Approved ***Earners*** (see II.II Governance Controlled Protocol Actors) to earn the ***Earner Rate*** (see II.III Governance Controlled Protocol Parameters). The Earner Rate, while input as an explicit value from the TTG, is bound in the smart contracts to be the lower of its input value or the maximum it can be without expending more *M* than is accruing from Minter Rate. To elaborate, the *utilization* of the Earn Mechanism can be considered to be the total amount of Owed M that is currently paying the Minter Rate (hereon referred to as *active M*) divided by the total amount of *M* in the Earn Mechanism. If a Minter is depermissioned by the TTG, their *M* will be deducted from the active M and thus lower utilization. The Earner Rate is then the minimum of the value input by the TTG, or the Minter Rate multiplied by utilization, which represents the lowest rate that would be safe to offer before more M is being paid to Earners than is being collected from Minters.

Once approved by the TTG, an Approved Earner can call the **Start Earning** method. This will check if the address is on the Approved Earners list and if so the address will begin to earn the Earner Rate on a continuously compounding basis. If an address is removed from the Earner's list, **Stop Earning** can be called with the address in question passed as an argument to the method, which will cease the accrual of the Earner Rate on the address' balance.

II.I.VI Removing a Permissioned Actor

Permissioned actors can be removed from the system by passing a proposal in the TTG mechanism removing them from a specific list (e.g. the Minter list). Once an actor is removed they are no longer able to call the methods in the contracts, preventing them from engaging in further activity with the protocol. Once a Minter is removed from the Minter list, they cease to pay the Minter Rate on their Owed M. The value of current Owed *M*, and potentially penalties for missed intervals, is stored for repayment by the Minter or anyone interested in their off-chain collateral retrieval. This is to ensure that *M* does not get paid to the Earn Mechanism and to the *ZERO* holders if a Minter is no longer actively in the system. The Burn method always remains available to anyone even if a Minter is no longer permissioned. This ensures that off-chain actors can facilitate the winddown of the Minter's operations and destroy the Owed M in order to retrieve the value from the Eligible Custody Solution. Once a Minter is removed from the Minter list, any actor in the system can call the **Deactivate Minter** method to cease the accrual of Minter Rate and the imposition of further Penalty Rate charges.

II.I.VII Example Interactions and Flows

The following is an example of how Permissioned Actors are intended to interact with the protocol, and how *M* is intended to flow through the protocol.

Step 1: Minters, Validators, and Earners propose their addresses to the M^0 TTG.

Step 2: The TTG accepts or rejects the proposals.

Step 3: Minters that were approved by the TTG are now on the Minter List. These Minters call for their first on-chain Collateral Value update.

Step 4: This Minter has contracted with at least one Validator off-chain. The Validator(s) have full access to the records and statements of the Eligible Custody Solution. The Validator(s) will check to ensure that the proposed on-chain Collateral Value is less than the dollar value of the Eligible Collateral in the Eligible Custody Solution. Once they have confirmed this to be true, they will provide the Minter with their signature and a timestamp from when they performed the balance check.

Step 5: Once the Minter has obtained a valid signature and timestamp from the Validator(s) they will call Update Collateral to push the balance on-chain.

Step 6: Now that the Minter has a positive on-chain Collateral Value, they are able to generate *M*. They will call the Propose Mint method and specify the amount they wish to generate. As long as this amount is within the bounds of the current on-chain Collateral Value (excluding any open Retrieval IDs) multiplied by the Mint Ratio[4], the method will output a Mint ID. This Mint ID will be inactionable for some Mint Delay, and then actionable for Propose Mint Time To Live.

Step 7: The Mint ID must be outstanding for Mint Delay. This is to ensure the superset of Validators have the opportunity to scrutinize the Propose Mint and call the Cancel and/or Freeze methods if something is amiss. Once Mint Delay has passed, the Minter can call Mint M and generate the *M*.

Step 8: Now that the *M* is generated, the Minter begins to pay Minter Rate on their Owed M. They will be subject to Penalty Rate charges on this balance if they do not keep their on-chain Collateral Value up to date or allow their on-chain Collateral Value to decrease below the permitted level. If the latter circumstance occurs, it is likely because the assets comprising the off-chain collateral have matured and are sitting in bank deposits.

Step 9: Assume that this Minter now wishes to repay some of the *M* they have generated and to retrieve a portion of the collateral from the Eligible Custody Solution. The Minter will first call Burn and specify the amount of *M* they'd like to repay. This will reduce their Owed M by this amount. Assuming that now there is a positive spread between the permitted amount of *M* that the Minter can generate and their Owed M, they can call the Retrieve method. The Retrieve method will first check if their on-chain Collateral Value, after the retrieval, is in compliance with the protocol's rules. If it is, the method will output a Retrieval ID and reduce the Minter's on-chain Collateral Value by the amount specified.

Step 10: The Minter can now go to the operator of the Eligible Custody Solution and show them the Retrieval ID and request that they allow them to redeem the corresponding value.

Step 11: Once the transaction has completed and cleared the Minter will call Update Collateral and input the new Collateral Value and the Retrieval ID in order to remove the subtraction to

[4] Note that if the on-chain Collateral Value was not updated within Update Collateral Interval it will be 0.

on-chain Collateral Value associated with the Retrieval ID. The Minter once again requires the Validator signature data and timestamp for the transaction to succeed.

II.II Governance Controlled Protocol Actors

Minters - A list of addresses (the Minter list, where the addresses are known as Minter address) maintained by the TTG mechanism which are able to access the minting functionality. The minting functionality allows for addresses on the Minter list to update on-chain Collateral Value associated with their Minter address, Propose Mint *M*, Mint *M*, Burn *M*, and to Retrieve off-chain collateral.

Validators - A list of addresses (the Validator list, where the addresses are known as Validator IDs) maintained by the TTG mechanism which act as a security layer for protocol. Validators are required to provide signatures for the Update Collateral method. They also have the ability to call the Cancel method on any Mint ID, and the Freeze method on any Minter address.

Earners - A list of addresses (the Earner list, where the addresses are known as Earner IDs) maintained by the TTG mechanism. These addresses are able to control whether they earn the Earner Rate.

II.III Governance Controlled Protocol Parameters

Note: These values will be set after launch through the TTG mechanism. It is not possible to deploy the protocol with preset parameters.

Minter Rate - The annualized percentage charged continuously to Minters on their Owed M. It is alterable with a Standard Proposal.

Logic: This annualized percentage should (generally) be less than the average rate on the Eligible Collateral being earned by Minters. This spread, adjusted for the Mint Ratio, is the profit margin of the Minter.

Penalty Rate - The percentage charged on Owed M that is in excess of the amount a Minter is permitted to have generated. It is assessed any time Impose Penalty is called, which is embedded in both Update Collateral and Burn. It is alterable with a Standard Proposal. This is a fixed percentage and not an annualized rate.

Logic: This percentage should be sufficiently high to deter Minter offenses, but not so high as to overly punish Minters that are victims of circumstance.

*Example: Minter 1 has 1,000,000 Owed M but has not updated their on-chain Collateral Value within Update Collateral Interval, and hence the on-chain Collateral Value is 0. Whenever they call Update Collateral or Burn, and Impose Penalty is consequently called, they will pay Penalty Rate on 1,000,000 - (0 * Mint Ratio). So they will pay Penalty Rate on their full debt. 1,000,000 * .01 = 10,000 M in Penalty Rate charges. This assumes that only one Update Collateral Interval period was missed.*

Earner Rate - The annualized percentage paid to *M* in the Earn Mechanism. If the cumulative *M* paid out via the Earn Mechanism is going to be greater than the amount of *M* being generated by the Minter Rate, the Earner Rate is automatically discounted to whichever percentage will reduce this mismatch to 0. ZERO holders receive all remaining M that is not paid out to the Earn Mechanism for their participation in protocol governance. It is alterable with a Standard Proposal.

Logic: This annualized percentage should be consistent with the yield demanded by institutional holders of M. It is mechanically prevented from exceeding the cumulative level of *M* generated by the Minter Rate. It should not be set so low that it results in insufficient demand for *M* and thus an inefficient market for Minters.

Mint Ratio - This percentage is the fraction of a Minter's on-chain Collateral Value that they can generate in M. It effectively controls the leverage of a Minter and the over-collateralization of *M*. It is alterable with a Standard Proposal.

Logic: This percentage controls the leverage of Minters and the over-collateralization of *M*. It should be set high enough to encourage attractive Minter economics, but not so high that it compromises the stability of *M*.

Mint Delay- This amount of time is the period between when a Minter has called Propose Mint and when they can first call Mint *M*. It serves as a protective measure to ensure all actors have sufficient time to audit each Mint. It is alterable with a Standard Proposal.

Logic: This amount of time should be long enough to ensure proper auditability and to afford Validators a chance to call Cancel or Freeze if necessary. It should not be so long that it introduces unnecessary friction into the Minting process and reduces the efficiency of Minters and the stability of *M*.

Propose Mint Time To Live - This is the amount of time after the Mint Delay that a Proposed Mint has to be called before it expires. It serves as a protective measure to ensure that Minters cannot call Propose Mint and then execute the Mint at a much later date. It is alterable with a Standard Proposal.

Logic: This amount of time should be long enough to give Minters a chance to react to the Mint Delay lapsing and execute their Mint, without being so long that it compromises the integrity of the previous Validator checks.

Update Collateral Interval - This amount of time is the period between which Update Collateral must be called by a Minter. If they do not call Update Collateral within this amount of time after their previous call, their on-chain Collateral Value is assumed to be 0 and they will incur Penalty Rate on the next update. It is alterable with a Standard Proposal.

Logic: This amount of time should be long enough to ensure that Validators can reliably check the status of the off-chain structures and balances, but not so long that it compromises the integrity of those checks.

Update Collateral Threshold - This number of signatures is the minimum number of Validator signatures required to execute Update Collateral. If a Minter cannot provide this number of signatures, they cannot successfully call Update Collateral. It is alterable with a Standard Proposal.

Logic: This number of signatures should ensure that the Update Collateral process is as secure as possible given the number of Validators in the network. It should not be set so high that Minters cannot reliably call Update Collateral.

Minter Freeze Time - This amount of time is the duration for which a Minter will not be able to call Propose Mint or Mint after having the Freeze method called by a Validator on their address. It is alterable with a Standard Proposal.

Logic: This amount of time should be sufficient for the Minter to remedy an issue, but not so long that it materially disrupts its normal course of business.

III. Governance

The M^0 protocol uses a new on-chain governance mechanism called a TTG to manage its various inputs. TTG stands for *Two Token Govern*or and is a mechanism by which holders of the voting tokens are penalized for failing to vote. There are two utility tokens used in the M^0 TTG: ***POWER*** and *ZERO*. *POWER* is used to vote on active proposals and can be considered the primary management token of the mechanism. *POWER* holders will earn *ZERO* in exchange for their direct participation in governance. If a *POWER* holder delegates their balance to an address that is not also the holder of the tokens, it is this address which receives the *ZERO* rewards. *ZERO* holders are comparatively (to *POWER* holders) passive in the voting process and only vote on important changes. *ZERO* holders at any time may **Reset** (see: III.II.II.III ZERO Threshold Proposals) the *POWER* token supply to themselves. The goal of the TTG mechanism is to ensure credible neutrality of governance. In any system there are two extremes that must be avoided: capture and fraud. In one case, the system is captured by actors whose primary interest is not in efficient protocol operation and it ceases to function in a way where all users are treated the same. In the other, the protocol ceases to function for anyone except the fraudulent actor. It is this dichotomy that is at the heart of the two token design. *POWER* holders are treated as a managerial class that is able to earn compensation through continued benevolent participation. This continued benevolence is judged by the *ZERO* holders who can always strip the *POWER* holders of their management rights, and thus their ability to earn future ownership in the protocol. If the composition and decisions of *POWER* holders trend towards either extreme, it is in the interest of *ZERO* holders to call Reset in order to restore balance.

III.I Inputs

The M^0 TTG (hereafter TTG) is responsible for the following inputs to the M^0 protocol and to itself (see Section III.III Governance Controlled TTG Parameters, Section II.II Governance Controlled Protocol Actors, and Section II.III Governance Controlled Protocol Parameters for further context on the actors and parameters listed below):

Governance Controlled TTG Parameters

- Proposal Fee
- *POWER* Threshold
- *ZERO* Threshold
- CASH Toggle

Governance Controlled Protocol Actors

- A list of approved Minters
- A list of approved Validators
- A list of Approved Earners

Governance Controlled Protocol Parameters

- Minter Rate

- Penalty Rate
- Earner Rate
- Mint Ratio
- Mint Delay
- Propose Mint Time To Live
- Update Collateral Interval
- Number of Signatures
- Minter Freeze Time

III.II Operation

The TTG is used to vote on proposals seeking to amend the actors and variables. New lists and variables may be added arbitrarily over time, however the core protocol is immutable and these additions will not directly impact its operations. The implementation which controls the M^0 protocol is deployed exclusively on the Ethereum Mainnet.

III.II.I Epochs

The mechanism practically operates in 30-day epochs, meaning in the standard operating procedure proposals are only passed on a 30-day cycle. In practice, this broader epoch is split into two epochs of 15 days. These numbers were chosen to align with an average calendar month. The first epoch (the **Transfer Epoch**) is a non-voting period where transfers and delegation are enabled. The second 15-day epoch (the ***Voting Epoch***) is where voting takes place on Standard Proposals and transfers and delegation are disabled. These restrictions on on-chain activity only apply to *POWER* tokens, there are no restrictions placed on the *ZERO* token. The conceptual 30-day epoch is split into these two smaller epochs to ensure correct accounting for voting and inflation. This means that proposals are collected in one 15-day epoch (whether it be the Voting Epoch or the Transfer Epoch), are voted on in the following 15-day Voting Epoch, and should be executed in the following 15-day Transfer Epoch. Proposals that passed but were not executed eventually expire. Therefore a proposal may spend as short as 15 days plus two blocks, or as long as 60 days, from submission to execution. During the 15-day Transfer Epoch, holders may transfer their balances, reassign delegations, and purchase *POWER* that is being auctioned.

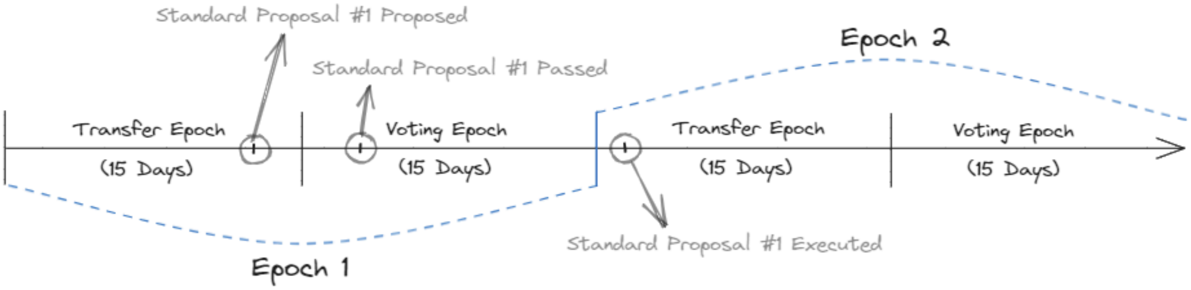


Diagram showing the Transfer and Voting Epochs

III.II.II Proposals

The TTG has three types of proposals: (1) ***Standard Proposals***, (2) ***POWER Threshold Proposals***, and (3) ***ZERO Threshold Proposals***. The use of threshold in the terminology represents a *yes threshold*, meaning that the threshold percentage of yes votes must be reached in order for the proposal to pass. A proposal that requires a threshold never explicitly fails (although it will eventually expire), but cannot be executed without reaching the requisite number of yes votes. For example, if a proposal requires a 10% yes threshold, it will pass as soon as 10% of the relevant token supply has voted yes. If this proposal never reaches 10% of the relevant token supply voting yes, it will expire without passing. These proposal types are described below in further detail.

III.II.II.I Standard Proposals

Standard Proposals are voteable only by *POWER* holders and require a simple majority of participating tokens to pass. Standard proposals do not require a threshold percentage of yes votes to pass. If there are only 100 *POWER* tokens voting in a Standard Proposal and 51 vote yes while 49 vote no, it will pass. If 50 vote yes and 50 vote no, it will fail as it requires the yes balance to exceed the no. Standard Proposals are the only proposal type which are “mandatory” for POWER holder participation. Lack of participation will result in POWER holders being diluted in terms of their overall voting power in the system and will cause them to forfeit any ZERO rewards for which they were otherwise eligible. A successful Standard Proposal will have its Proposal Fee available to be returned to the proposer.

Example: During a Voting Epoch a proposal to change Minter Rate is made. Only 10% of the POWER tokens choose to participate. When voting on the proposal, 60% of the participating POWER tokens vote yes. This proposal is passed because the yes votes are greater than the no votes. The total number of POWER holders participating is only relevant in votes which require a POWER Threshold.

III.II.II.II POWER Threshold Proposals

A *POWER* Threshold Proposal can be used to submit anything which would otherwise be a Standard Proposal, except it requires a *POWER* Threshold and is immediately votable and subsequently immediately executable rather than only being votable and executable in the future epochs. For this reason, *POWER* holders are likely to use this proposal type in the case of an urgent or emergency situation. If a *POWER* Threshold has not been reached before the proposal expires, the proposal cannot be executed. *POWER* Threshold Proposals expire at the end of the next epoch.

*Example: During the Transfer Period a POWER Threshold Proposal is made to change Minter Rate, which requires a POWER Threshold due to it being an POWER Threshold Proposal. The POWER Threshold is set to 75%. The proposal becomes immediately votable and a full 100% of the POWER supply chooses to participate – 80% vote affirmatively. This proposal has passed and is instantly executable upon the POWER Threshold being met. This is because it achieved yes votes from greater than 75% of the POWER supply (100% * 80% = 80% yes). Note that the proposal very well could have gone on to accumulate more yes votes, but was likely executed immediately or soon after meeting the POWER Threshold.*

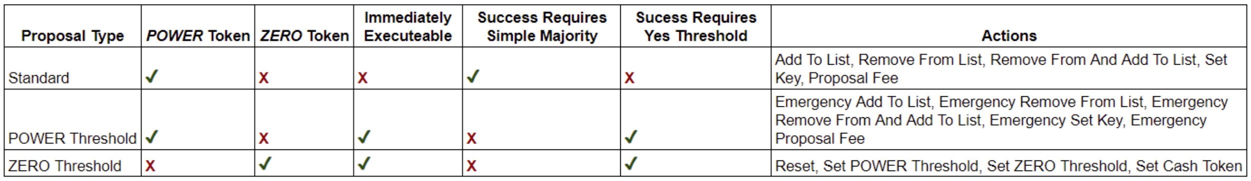
III.II.II.III ZERO Threshold Proposal

A *ZERO* Threshold Proposal is used for Reset, to toggle CASH between *WETH* and *M*, and to set the *POWER* and *ZERO* Thresholds themselves. The Reset method is a special feature reserved for the *ZERO* token holders which allows a yes threshold of *ZERO* holders to change the current governor of the system to a new version with a new *POWER* token that is claimable pro rata to *ZERO* holders or to existing *POWER* holders. They do this by creating a proposal to call the Reset method. Mechanically this is affected by replacing the current governor (*POWER* token address) of the system with a new instance, where the starting balance of the *POWER* tokens are proportional to each *ZERO* holder's balance in the epoch before the Reset. It is immediately executable upon achieving a yes threshold. It is intended for *ZERO* holders to use this feature should they find something irreparably wrong with the composition and/or voting patterns of the current *POWER* holders and wish to take on voting responsibility themselves. It is anticipated that *ZERO* holders will only Reset the governor to the existing *POWER* holders if the token is nearing an overflow, which will not happen for 150+ years. There is technically no limit to how many times Reset can be called, but it is not anticipated to be frequently used (if it is ever used in the first place). If a Reset is executed in the middle of a Voting Epoch, all active and/or unexecuted proposals are effectively canceled because they are using the obsolete governor.

Example: The ZERO Threshold is set to 60%. POWER holders seem likely to pass a proposal to add a perceived malicious actor to the Minter List. A user submits a proposal to Reset. This proposal is immediately votable – it achieves 70% of the total ZERO supply voting yes. This proposal is passed and is immediately executable. All currently votable proposals, including the proposal to add the perceived malicious Minter, are effectively canceled.

III.II.II.V Proposal Matrix

The diagram below details the three types of proposals in the TTG and highlights which actions are associated with each type.



Proposal Type	*POWER* Token	*ZERO* Token	Immediately Executeable	Success Requires Simple Majority	Sucess Requires Yes Threshold	Actions
Standard	✓	x	x	✓	x	Add To List, Remove From List, Remove From And Add To List, Set Key, Proposal Fee
POWER Threshold	✓	x	✓	x	✓	Emergency Add To List, Emergency Remove From List, Emergency Remove From And Add To List, Emergency Set Key, Emergency Proposal Fee
ZERO Threshold	x	✓	✓	x	✓	Reset, Set POWER Threshold, Set ZERO Threshold, Set Cash Token

Diagram of Proposal Types with associated token responsibility, mechanics, and actions.

III.II.III Checkpoints and Voting

A checkpoint of balances is taken at the start of epochs and the balances contained in these checkpoints are used for voting throughout the epoch. During a Transfer Epoch, only the balance a user possessed at the checkpoint will be counted towards voting on *POWER* Threshold Proposals, which will not include standard inflationary proposals by definition. In order to vote the *POWER* owner's delegate address (hereon referred to as the *POWER* holder) calls the **Cast Votes** method on the array of proposals they wish to vote on and specifies yes/no for each proposal. There is no abstain option in the TTG.

III.II.IV Proposing

Anyone with an ethereum address and ***WETH*** or *M* may submit a proposal. The TTG is to be deployed with *WETH* as its internal currency (known as *CASH* in the mechanism), and therefore any Standard Proposal submission must pay a Proposal Fee in *WETH,* or at a later date *M* depending on the current CASH toggle setting, in addition to gas fees (see III.III Governance Controlled TTG Parameters). A proposal that passes makes the Proposal Fee available to be returned to the proposer upon execution.

There are two primary structures of proposals that can be managed through the TTG: (1) configuring a registrar used by the protocol, i.e. adding and removing addresses from arbitrary lists/sets and setting arbitrary variables; (2) setting governance parameters. The M^0 protocol looks to the registrar to use certain variables and sets of addresses in its processes.

In order to propose a change to a list, a user submits a Standard Proposal or a *POWER* Threshold Proposal calling the **Add To List** or **Remove From List** methods along with the address they wish to add or remove. There is also a method called **Remove From And Add To List** which facilitates swapping an address on a list. In order to add a new list to the TTG a proposer will create a proposal which uses Add To List and will specify a new list, which is created simultaneously to the proposal being executed. Since the M^0 core protocol is immutable, any list added after deployment can only be used to manage periphery smart contracts and cannot impact core operations.

Example: Alice wishes to add her company's address to the list of approved Minters in the M^0 protocol. Alice calls Add To List, specifying the Minters list along with the address she would like to gain permission to mint M in the protocol.

In order to propose a change to a configuration contract proposers call the **Set Key** method[5]. In order to propose a configuration change at the registrar, proposers create a proposal for the governor to call the registrar's Set Key method. The update either results in the first setting or overwriting of a value for a given key (i.e. variable name).

Example: Bob wishes to change Minter Rate from 3% to 4%. Bob calls Set Key and specifies the configuration contract he'd like to change along with the new value he would like it to contain.

Once a proposal passes, and assuming the requisite amount of time has passed, anyone can call the **Execute** method to execute the action on chain. They must pass the proposal arguments into the Execute method.

III.II.V Inflation Mechanics

In each epoch the supply of *POWER* is inflated by 10% and the supply of *ZERO* is inflated by up to 5,000,000 tokens. This inflation is claimed *pro rata* by participating *POWER* holders, specifically by the delegate address. Any *POWER* that remains undistributed (or that could not be claimed because the holder did not fully participate in that epoch) is auctioned off to the highest bidder in a pay-as-bid Dutch auction (hereon "Dutch auction"). When there are tokens to auction, the auction starts at the beginning of the Transfer Epoch and ends at the finish of the

[5] In certain user interface implementations, this method may be referred to as "Set Protocol Configuration"

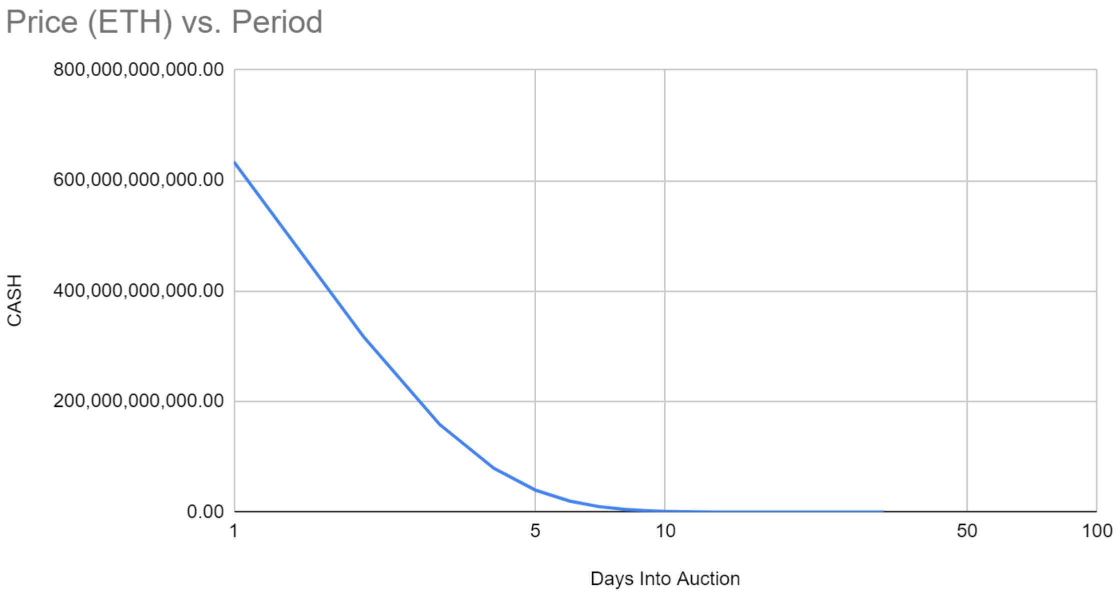
Transfer Epoch. Therefore if a user purchases *POWER* during an auction they will always be able to use those tokens to vote in the subsequent Voting Epoch. Each participating *POWER* holder during the Voting Epoch will also receive their *pro rata* (based on their percentage of total voting power) share of the 5,000,000 *ZERO* tokens.

Once a Standard Proposal has been submitted it can be voted on in the following Voting Epoch, unless it is a *POWER* Threshold Proposal or a *ZERO* Threshold Proposal, which can be voted on at any time. When a Standard Proposal becomes available for voting, it is mandatory for *POWER* holders to vote on it or else the owner of the *POWER* tokens will lose relative voting weight in the system. If a *POWER* owner's delegate fails to vote on *any* proposal in an epoch, they will forfeit any *POWER* or *ZERO* inflation they would have otherwise been able to claim. *POWER* holders must vote on all Standard Proposals in the Voting Epoch; *POWER* Threshold Proposals and *ZERO* Threshold Proposals do not factor into *POWER* inflation dynamics. There is no inflation if an epoch only has *POWER* Threshold Proposals and/or *ZERO* Threshold Proposals, or no proposals at all.

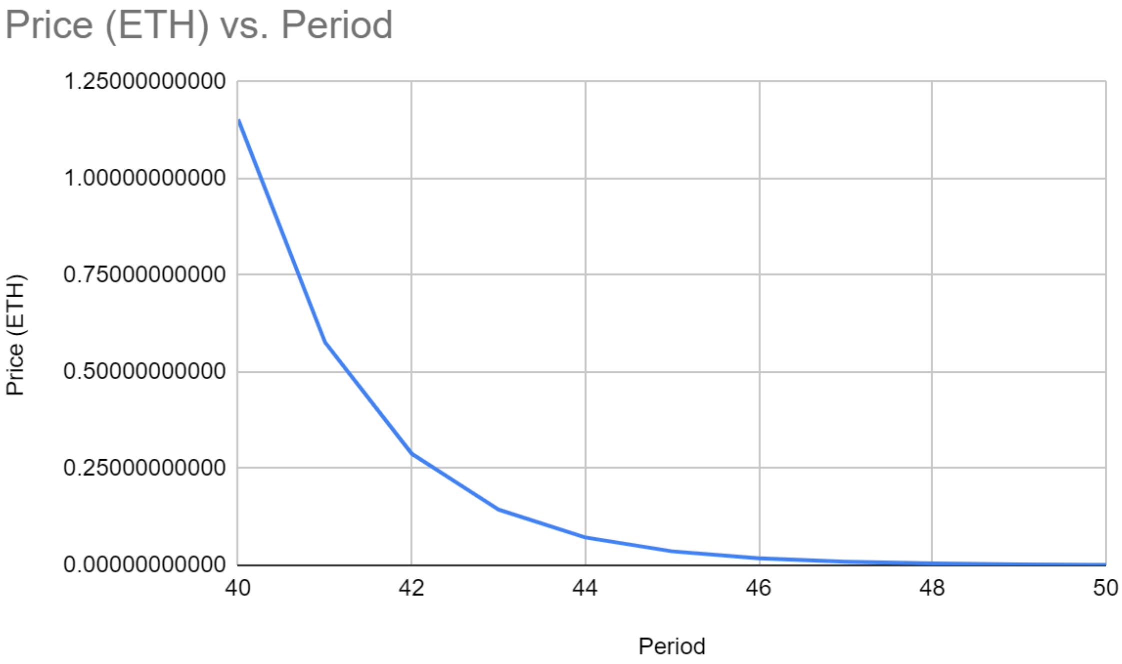
Example: *Alice is a POWER holder with 1,000 POWER tokens, the total POWER supply is 10,000; hence Alice controls 10% of the POWER voting weight. In epoch 1 she participates fully by voting on all Standard Proposals. Simultaneous to participating, Alice claims an additional 100 POWER tokens and 500,000 ZERO tokens, i.e. 10% of a POWER inflation of 1,000 tokens and 10% of a ZERO inflation of 5,000,000 tokens. In epoch 2 Alice fails to vote on a Standard Proposal and is therefore not able to claim any POWER or ZERO tokens. The 110 POWER tokens (i.e. 10% of the new additional POWER supply of 1,100 tokens) that should have gone to Alice are auctioned in the Dutch auction. Bob buys these 110 tokens for 1 WETH (the internal currency of the TTG) and can now participate in the following voting epoch. The ZERO tokens that should have gone to Alice are simply never minted.*

III.II.VI Dutch Auction

Once the Transfer Epoch has begun the Dutch auction will begin simultaneously if any *POWER* holder failed to participate. The price per basis point (0.01%) of *POWER* token, calculated as a percentage of the total *POWER* supply, in the Dutch auction will start at 2^{99} wei (the smallest unit of *WETH*) and decrement the exponent approximately every 3.6 hours. During the period between exponent decreases the price linearly declines. This means that after the first 3.6 hours of the auction the price will be 2^{98} wei and linearly decrease to 2^{97} wei over the following 3.6 hours. In the implementation, bitwise shifting is used to achieve this effect. That is to say that at the midpoint between two exponents, the value is halfway between them. In order to purchase *POWER* in the Dutch auction, purchasers must call the **Buy** method. The diagrams below illustrate the intended price curve of the Dutch auction in ETH over the 15 day period. The first demonstrates the entire price curve, while the second shows a smaller slice of the curve to demonstrate its semi-linearity.



The price of 1 basis point of POWER in ETH at each 3.6-hour period (shown in days).



A "slice" of the price curve between periods 40 and 50.

III.II.VII Delegation

Both *POWER* and *ZERO* owners may delegate their voting power to an arbitrary Ethereum address during the Transfer Epoch. Delegated *POWER* will retain its inflation in the *owner* address, while *ZERO* rewards will be claimable by the *delegate* address. Owning *ZERO* does not earn the owner or the owner's delegate inflation or rewards aside from *M* generated by the protocol fees, and thus delegation does not have any impact on the owner outside of transferring voting power. *ZERO* does earn fees from the Proposal Fee on failed Standard Proposals and from the Minter Rate. Delegation snapshots are taken at the beginning of the epoch and close at the end of the epoch, and the values in the snapshots are subject to change until the epoch closes. Both *POWER* and *ZERO* follow the ERC20 standard and holders must call the **Delegate** method and provide the address they wish to delegate to. For holders that do not actively Delegate, the default delegation is set to the address which owns the tokens. Users do not need to alter their delegation in each epoch unless they wish to change delegates.

III.II.VIII *ZERO* Claiming of Residual Value

In exchange for ZERO holders' participation in protocol governance, they will receive the remainder of the protocol fees. The anticipated accumulation of tokens to the *ZERO* holders are Proposal Fee payments from proposal submission, the payments from *POWER* token auctions, and a portion of Minter Rate and Penalty Rate charges to Minters (see II.I.II Protocol Fees). Proposal Fee and auction payments are collected in *WETH* or *M*, depending on the status of the CASH toggle, and Minter Rate is collected in *M*. At any time a *ZERO* holder may call the **Claim** method in order to claim this accumulated value. The amount of claimable tokens are pro rata to each account's *ZERO* balance on the close of each epoch they are trying to claim for. They pass the array of epochs (or a starting epoch and ending epoch) they are seeking to claim for as arguments to the method.

III.III Governance Controlled TTG Parameters

CASH - The internal currency of the TTG. It is used to pay Proposal Fee and to purchase *POWER* in the Dutch auction. It can be toggled between *WETH* and *M*.

Logic: This token must be permissionless and well distributed in order to prevent takeover of the TTG. It should also have sufficient value to its holders in order to deter spam and to increase the efficiency of the Dutch auction.

Proposal Fee - The amount paid in CASH to submit any proposal. It is alterable with a Standard Proposal.

Logic: This amount should be sufficiently high to deter spam, but not so high as to deter legitimate proposals.

POWER **Threshold** - The number of yes votes as a percentage of the total *POWER* supply required to pass proposals which require a *POWER* Threshold.

Logic: This percentage should be low enough to ensure that in an emergency situation, enough POWER holders can be collected to pass a proposal. It should be high enough that a malicious proposal cannot be passed instantly.

Example: POWER Threshold = 80%. Therefore if there are 10,000 total POWER in existence, 8,000 POWER will need to vote affirmatively for a proposal to pass that requires a POWER Threshold.

ZERO **Threshold** - The number of yes votes as a percentage of the total *ZERO* supply required to pass proposals which require a *ZERO* Threshold.

Logic: This percentage should be low enough that it is possible to call Reset if necessary. It should be high enough to ensure that Reset is not called without a very high level of consensus.

Example: ZERO Threshold = 60%. Therefore if there are 1,000,000,000 total ZERO in existence, 600,000,000 ZERO will need to vote affirmatively for a proposal to pass that requires a POWER Threshold.

III.IV Immutable TTG Parameters

Epoch Duration - The combined length of the Voting Epoch and the Transfer Epoch. *Set to 30 days*.

Logic: This amount of time should be short enough to permit for timely management of the protocol, but long enough for all *POWER* holders to both socialize and physically vote on Standard Proposals.

Voting Epoch Duration - The length of the Voting Epoch. *Set to 15 days*.

Logic: This amount of time should be long enough to permit *POWER* holders to physically exercise their vote.

Transfer Epoch Duration - The length of the Transfer Epoch. *Set to 15 days*.

Logic: This amount of time should be long enough to contain the Dutch auction and for any *POWER* holders that may wish to perform transfers or re-delegations to do so.

Auction Duration - The length of the Dutch auction for unclaimed *POWER* inflation. *Set to 15 days* and overlaps perfectly with the Transfer Epoch.

Logic: This amount of time should be long enough to cross all conceivable prices while still promoting efficient price discovery. Note that this length matches the Transfer Epoch, so that tokens acquired in the Transfer Epoch will be included in the checkpoint for the following Voting Epoch.

Dutch Auction Exponent - The exponent with a base of 2 that determines the starting auction price. *Set to 99*.

Logic: This number should produce a sufficiently high starting price per *POWER* token such that the market price of *POWER* in Cash is never above this price. It should not be so high as to cause the auction to exceed the Transfer Epoch before reaching 0 given the Dutch Auction Period Time setting.

Dutch Auction Periods - The number of equal periods that must fit into the Transfer Epoch. *Set to 100*.

Logic: This number of periods defines how often the Dutch auction will decrease the Dutch Auction Exponent. At the current setting the Dutch auction will decrement the Dutch Auction Exponent approximately every 3.6 hours.

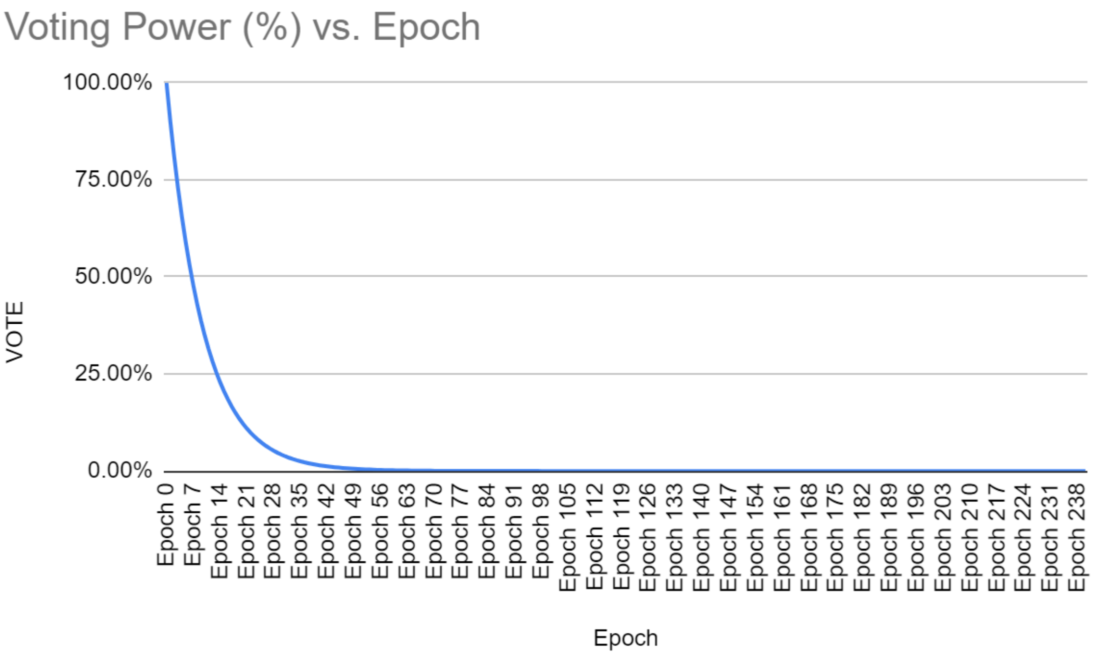
III.V Immutable *POWER* Parameters

***POWER* Initial Supply** - The initial supply of *POWER* tokens before any inflation. *Set to 10,000*. Decimals are 0.

Logic: The initial supply of *POWER* should be sufficient to distribute to the initial holders in the network, but not so high as to cause a premature overflow error. The lack of decimals (and thus lack of subdivision of tokens) was also chosen for this reason. See the diagram under *POWER* Inflator for further analysis. Another factor to consider in this initial supply is the “dust level,” meaning the level at which in a Reset a *ZERO* holder would not receive any *POWER* tokens. For example, since new *POWER* (post-Reset) is based on existing *ZERO* balances, anyone who owns less than 1/ *POWER* Initial Supply of a *ZERO* token will get 0 *POWER* after a Reset.

***POWER* Inflator** - The percentage inflation of the *POWER* supply per active epoch. This occurs only in epochs with a votable proposal, hence why they are referred to as active. If a *POWER* holder fails to fully participate in an epoch with at least one votable, their balance of *POWER* tokens will not decrease but their percentage of overall voting power will decrease by 1 / (1 + *POWER* Inflator). *Set to 10%*.

Logic: This percentage must sufficiently encourage *POWER* holder participation without occasional lapses in participation destabilizing the system. At 10% inflation, a *POWER* holder can expect to lose ~45% of their voting power if not participating for 6 epochs (note that epochs in our implementation correspond roughly to calendar months), ~70% of their voting power if not participating for 12 epochs, and ~90% of their voting power if not participating for 24 epochs. To a lesser extent this number should also take into account a realistic number of epochs with votable proposals to ensure that the operation of the protocol is not impacted by an overflow error. This would require either a Reset or a Hard Fork. The following diagrams demonstrate the intended impact of the *POWER* Inflator on an inactive participant, and a comparison of *POWER* Inflator and decimal setting as they impact overflow times assuming a 30-day total epoch time.



Voting power as a percentage of starting voting power (y-axis) over missed epochs with a votable proposal (x-axis), using a 10% POWER Inflator

	decimals	**decimals**	**decimals**
inflator	**18**	**6**	**0**
10%	110.91 years	135.083 years	147.16 years
20%	58.00 years	70.58 years	76.91 years
30%	40.33 years	49.08 years	53.5 years

Years to overflow assuming 12 epochs per year with votable proposals. Comparison of POWER Inflator (left column) and decimal choice for POWER token (top row). Assumes a starting POWER supply of 10,000.

III.VI Immutable *ZERO* Parameters

***ZERO* Initial Supply** - The initial supply of the *ZERO* token before any rewards. *Set to 1,000,000,000*. Decimals are 6.

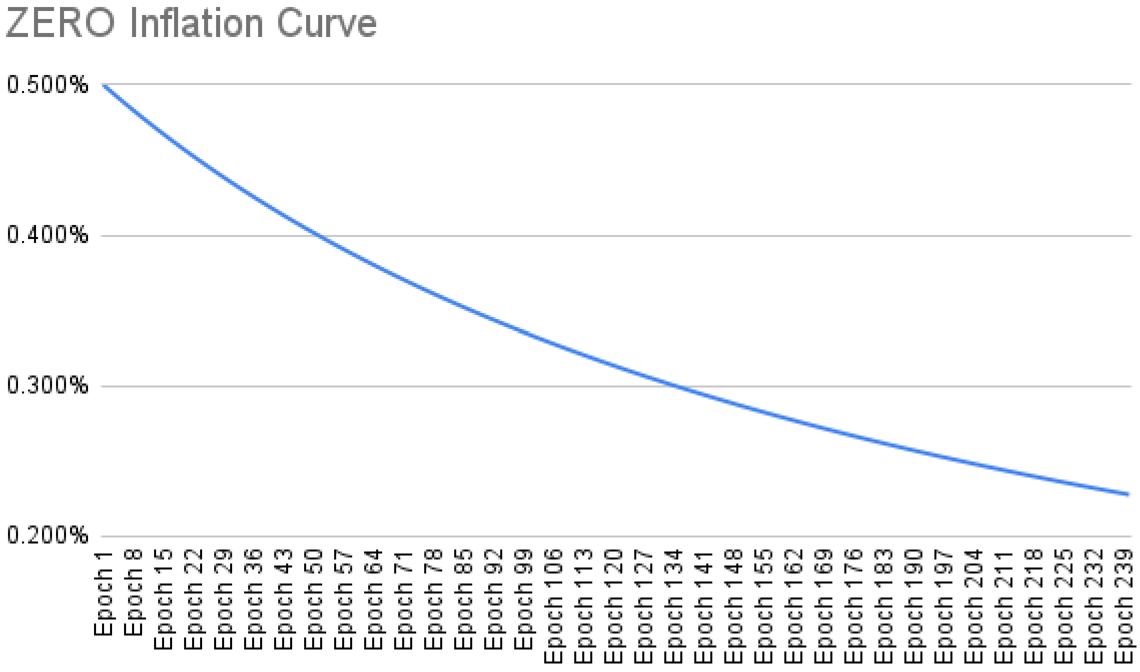
Logic: The initial supply of *ZERO* should be large enough to promote a high level of decentralization (as it pertains to the Reset method), and small enough to not break common integrations.

***ZERO* Reward** - The maximum amount of *ZERO* given to *POWER* holders in a Voting Epoch. Each *POWER* holder is given their pro rata share of the reward if they fully participate and the tokens are claimed simultaneously to voting. Tokens which go unclaimed are never minted. No tokens are distributed in an epoch with no active proposals. *Set to 5,000,000*.

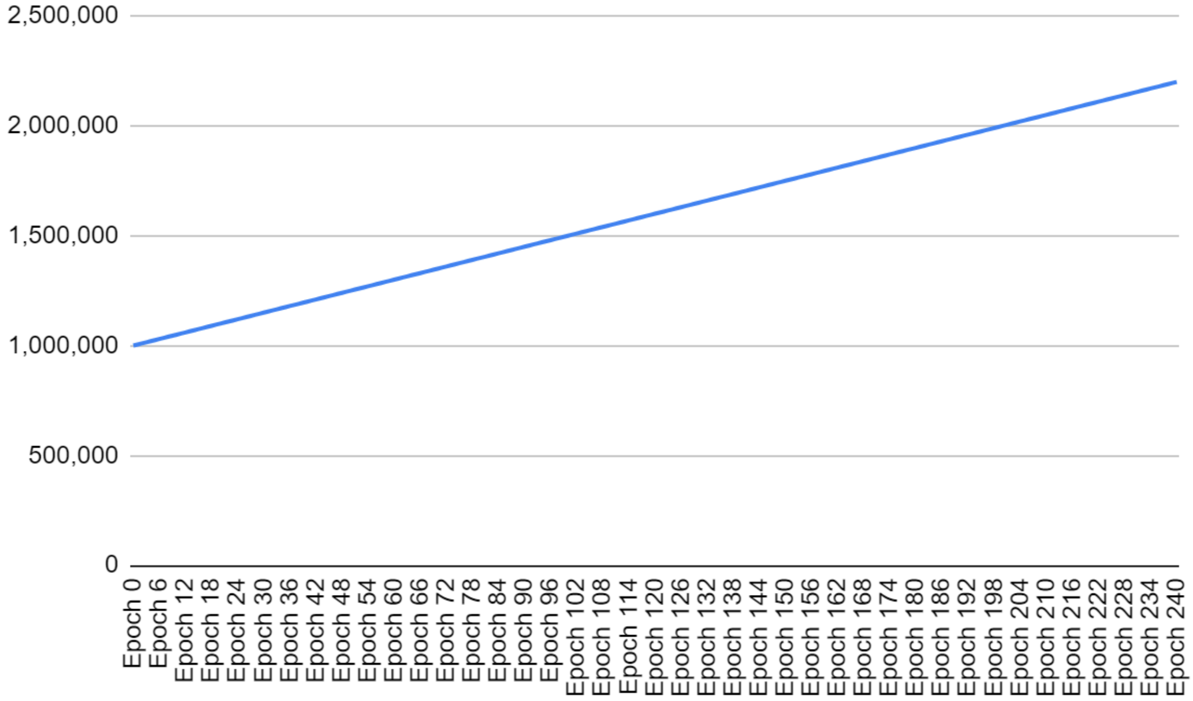
Logic: The reward amount should provide enough incentive to Standard Proposal voters to consistently vote while not destabilizing the protocol through unpredictable distribution. An additional consideration is the maximum level of decentralization that can be achieved given average gas fees – i.e. the reward must at least cover the average gas fee of participants and thus the average gas fee puts a practical boundary on the percentage of *POWER* one must own to justify participation.

Example: Alice has 10% of the POWER supply delegated to her. She fully participates in a Voting Epoch and is rewarded with 10% of the reward, in this case 500,000 ZERO tokens.

See the diagrams below for further illustration of the potential impact of the reward on the *ZERO* supply over time.



Maximum *ZERO* inflation per epoch over 240 epochs (20 years)



Maximum *ZERO* supply over 240 epochs

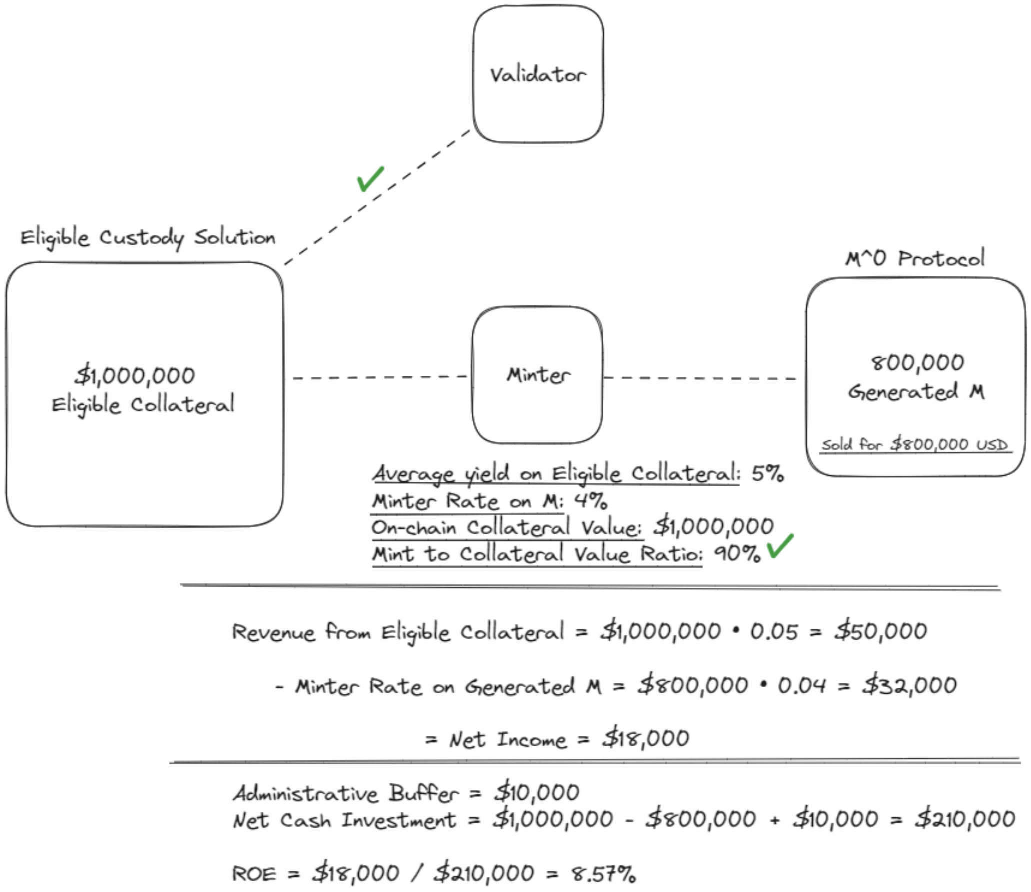
IV. The M^0 Economy

The M^0 protocol is a coordination mechanism. It is not intended to replace existing financial actors, but rather to provide novel and more efficient means by which they can interact. We believe that a universal blockchain-based protocol, where rules and transparency are enforced by code, is superior to the feudal and opaque landscape of value transmission present today.

The M^0 Protocol is intended to coordinate Minters, Validators, and Earners. It is anticipated that off-chain this may correspond to financial services providers (such as cryptodollar issuers), auditors, and institutional holders of *M*.

IV.I Minters

Minters are primarily incentivized to join the protocol because they want to earn the spread between the yield (net of expenses) on their Eligible Collateral and the protocol's Minter Rate. In addition, the ***Mint Ratio*** will determine the attractiveness of Minting relative to the yield spread. The effective ROC (*return on capital*) of a Minter is net yield generated on the Eligible Collateral, divided by the net cash investment, which is the capital invested into Eligible Collateral, minus their Owed M (assuming they were able to sell this *M* at $1) plus the Administrative Buffer. It is anticipated that Minters in the M^0 protocol will correspond to financial services providers (such as cryptodollar issuers) off-chain. The ultimate function of the Minter is the generation and management of the supply of *M*. Considering the initial Eligible Collateral is intended to be short term T-bills, it is assumed that the Minter Rate will need to be less than the US Federal Funds rate. See the diagram below for a visual example of the basic Minter economics.



Average yield on Eligible Collateral: 5%
Minter Rate on M: 4%
On-chain Collateral Value: $1,000,000
Mint to Collateral Value Ratio: 90% ✓

Revenue from Eligible Collateral = $1,000,000 • 0.05 = $50,000

- Minter Rate on Generated M = $800,000 • 0.04 = $32,000

= Net Income = $18,000

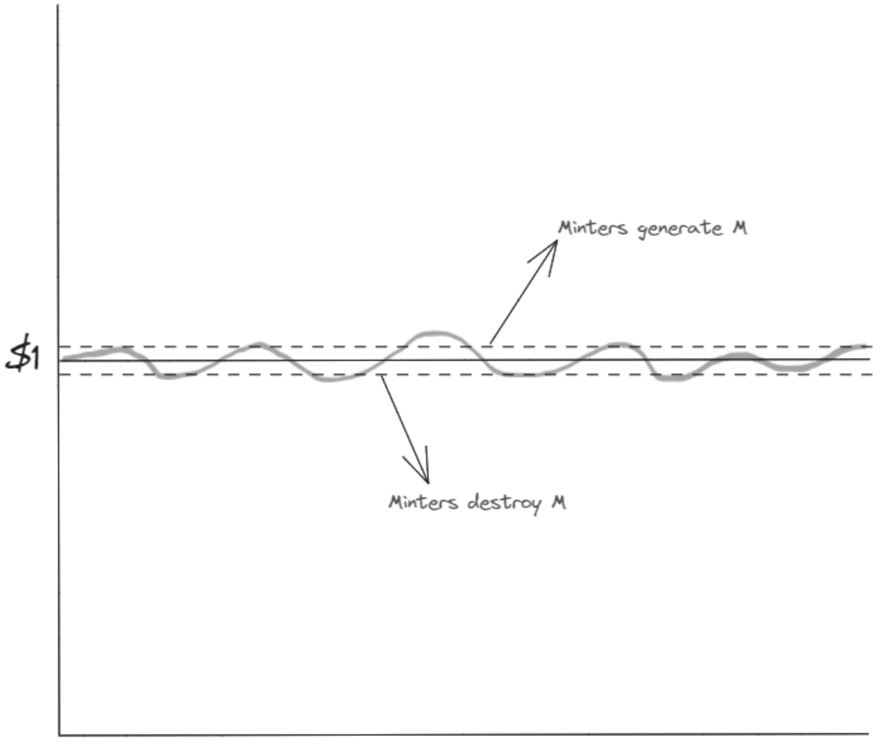
Administrative Buffer = $10,000
Net Cash Investment = $1,000,000 - $800,000 + $10,000 = $210,000

ROE = $18,000 / $210,000 = 8.57%

Diagram demonstrating the basic economics of a Minter

It is also anticipated that Minters will engage in arbitrage. If *M* is trading above $1 on secondary markets, it is logical for Minters to deposit Eligible Collateral in order to generate more *M*. This will boost their net yield. Conversely, if *M* is trading below $1 on secondary markets, it is logical for Minters to repurchase *M* and to use it to Retrieve Eligible Collateral. This will also boost their net yield. It is for this reason that *M* is expected to trade with some volatility around (US)

$1 – the mechanism relies on sometimes inefficient and unpredictable market forces to achieve an average price of $1 over the long term.



A hypothetical price curve for *M* with expected Minter activity

There will be no built in mechanism at the protocol level to ensure the price stability of *M*; rather, that will be achieved via the economic incentives described and visualized above.

IV.II Validators

Economically, all Validators must be incentivized off-chain or use periphery smart contracts. There is no Validator compensation in the core protocol. This decision was made because the Validator landscape is complex and the chance of accurately encapsulating these complex economic arrangements on-chain was nil. For the basic function of providing signatures for Update Collateral, it is expected that Validators and Minters will enter into binding, off-chain legal agreements specifying applicable terms and appropriate compensation. It is anticipated that Validators in the M^0 protocol will eventually correspond to auditors off-chain. The ultimate function of the Validator is to provide as close to real time attestation of the Eligible Collateral being used to generate *M* as possible.

While the protocol considers all Validator addresses to be fungible, there are in fact many specializations that could occur off-chain in the Validator ecosystem. For instance, there may be Validators that specialize in signing off on on-chain Collateral Value updates, while others act as "sentinels" and exclusively exist to call Cancel and Freeze on errant Minters. The level of specialization could even go beyond the initial methods of the protocol as periphery contracts are added by the ecosystem.

IV.III Earners

Earners are simply addresses approved by the TTG to earn the Earner Rate. It is expected that, throughout the cycle, the Earner Rate will remain comparable to the US Federal Funds rate as well in order to entice Earners to continue to hold *M*. The Earner Rate can be used as an additional tool to encourage *M* price stability around $1. If the price of *M* is above $1, the TTG can lower the Earner Rate in order to discourage holding of *M* and to encourage selling of *M* for alternative sources of yield. If the price of *M* is below $1, the TTG can raise the Earner Rate in order to encourage the holding and purchase of *M*. It should be noted that the Earner Rate can be higher than the Minter Rate as long as the amount of *M* being paid out via the Earner Rate is less than the amount of total *M* generated from Minter Rate.

It is anticipated that Earners in the M^0 protocol will correspond to institutional holders of *M* off-chain, and to issuers and distributors that maintain *M* inventory. The ultimate function of Earners is as a source of demand for *M*, making it more likely that Minters can efficiently generate *M*. This is effectively to say that Earners align nicely with the ultimate distributors of *M* to the broader market.

Example: The TTG permissions a large cryptocurrency exchange to the Earners list. This exchange has millions of users and is regulated/licensed appropriately in the jurisdiction(s) it serves. Once permissioned, the exchange can use customer's M to earn the Earner Rate. They can now pass a portion or all of the Earner Rate on to their customers.

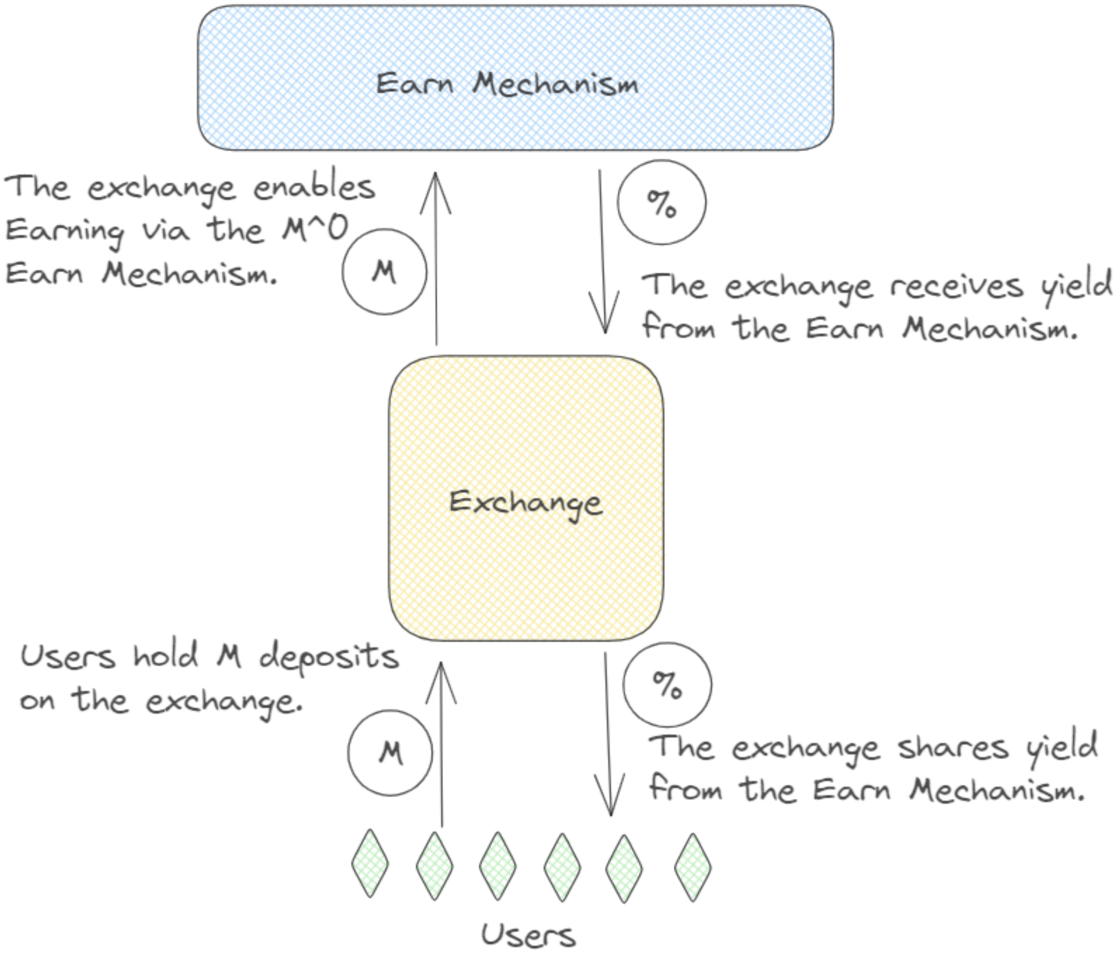

Earn Mechanism
The exchange enables Earning via the M^0 Earn Mechanism.
M
%
The exchange receives yield from the Earn Mechanism.
Exchange
Users hold M deposits on the exchange.
M
%
The exchange shares yield from the Earn Mechanism.
Users

Diagram showing how Earners may distribute *M* yield to end users

V. Off-Chain Ecosystem

The M^0 protocol relies on the presence of several off-chain actors and components, and a feedback process referred to as "guidance" to function properly.

V.I Guidance

It is anticipated that as the M^0 ecosystem grows, an increasing number of stakeholders will have an interest in its continued development and improvement. These stakeholders are referred to as *Think Tanks*. Much as the Basel Committee and its cooperating international regulators have provided (theoretically non-committal) guidance and rules for the banking sector, it is anticipated that similar groups and perhaps new M^0-specific institutions will provide guidance for the M^0 ecosystem and protocol. The protocol is designed in such a way that it can formally adopt guidance through a governance vote and enforce this guidance throughout the system via the Validators. It is intended to function as follows:

Step 1: Think Tanks provide guidance.

Step 2: The TTG adopts or rejects this guidance by voting on a hash of a public document containing it.

Step 3: Permissioned Actors adjust their behavior accordingly.

Step 4: Validators withhold signatures, Cancel Propose IDs, or Freeze Minters who have not adjusted their behavior to be in line with the guidance. If there is still non-compliance, the TTG can remove errant actors from the system.

This combination of steps creates the "guidance feedback loop."

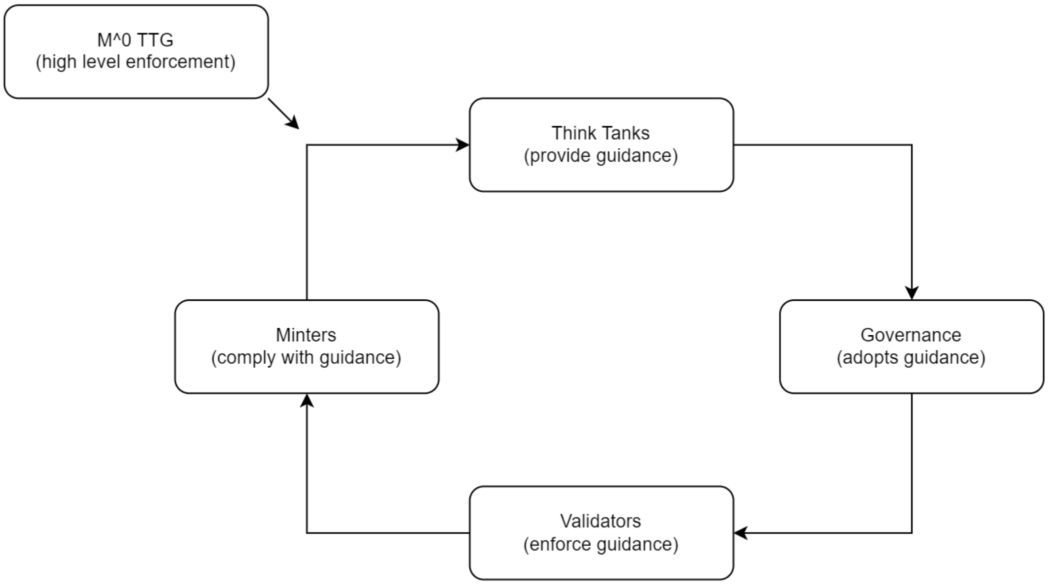


Illustration of the guidance feedback loop

Example: Think Tanks interested in the progress of the M^0 ecosystem and protocol determine that only certain jurisdictions are appropriate to host Eligible Custody Solutions. These Think Tanks put out guidance amending the definition of an Eligible Custody Solution to include a list of appropriate jurisdictions, and stipulate that all Minters should be in compliance with this list within 180 days of the proposal's execution. The document is hashed so that all actors can *validate they have the same and correct version of the guidance, and this hash is submitted to the TTG under a guidance list. The POWER holders will then vote on the proposal. If it passes, the guidance should be considered approved and enforceable. Once the 180 day window is complete (which is also defined in the guidance), Validators should begin to withhold signatures on Update Collateral calls from Minters that attempt to include Eligible Collateral held in an unapproved jurisdiction in their on-chain collateral value. Validators can also call Cancel or Freeze on a Minter that is errant in other ways not capturable in the Update Collateral Method. Validators not enforcing the approved guidance should expect to be removed by the TTG.*

V.II Off-Chain Actors and Components

The following components and their associated guidance represent those contemplated for the launch of the protocol. This list is not exhaustive and will likely change over time. While much of the guidance is currently quantitative, there is nothing prohibiting qualitative guidance that is left to the interpretation of the various actors. The guidance will also include legal templates and agreements that will contain terms required to ensure smooth operation of the off-chain components.

Eligible Collateral - A description of portfolio composition which can be placed in Eligible Custody Solutions and be used to generate an on-chain Collateral Value, which is subsequently used in the generation of *M*.

Guidance at launch: 30-90 day US Government T-bills.

Eligible Custody Solution - A description of entity structures, jurisdictions, contractual agreements, and other details that will suffice for the custody of Eligible Collateral.

Guidance at launch: Orphaned SPV[6] in approved jurisdiction.

Administrative Buffer - An amount of value that a Minter may be compelled to set aside to the Eligible Custody Solution Operator that is not included in the Minter's Eligible Collateral. This value is intended to be used by the Eligible Custody Solution Operator to facilitate the orderly winddown of the facility should the Minter become inactive or incapacitated.

Guidance at launch: $100,000.

[6] An SPV (Special Purpose Vehicle) is a legal entity structured intentionally to serve a specific purpose. In the case of the M^0 ecosystem, that purpose is to securely custody collateral.

Eligible Custody Solution Operators - The professional corporate servicing agents (e.g. Trustees) that manage the Eligible Custody Solution.

Guidance at launch: Any governance-approved Operator that is technically and operationally equipped (and licensed, as applicable) to act as manager for an orphaned SPV structure in the approved jurisdictions as well as capable to read and interpret the relevant on-chain information.

Banks - The banks that hold deposits on behalf of the Eligible Custody Solution.

Guidance at launch: Any bank in the approved jurisdiction(s).

Custodians - The custodians that hold Eligible Collateral (excluding bank deposits) on behalf of the Eligible Custody Solution.

Guidance at launch: Any bank in the approved jurisdiction(s) that are operationally equipped to custody the Eligible Collateral.

Minter Wind Down Procedures - The procedures a Minter is expected to follow should they be removed from the Minter list.

Guidance at launch: Minters will be given a 90 day grace period after being removed from the Minter list to fully wind down their operations (a "cooperative wind down"). This means they will have zero remaining Owed M. Should they fail to do so in this timeframe, the remainder of their off-chain value will be forfeit including the Administrative Buffer. The Administrative Buffer, in addition to remaining off-chain value in excess of Owed M, is intended to be used to finance the various off-chain actors that will be involved in the uncooperative wind down.

The information contained in this document is being provided solely for informational/discussion purposes and the reader should not construe anything contained herein to be a solicitation or an offer of sale of securities. Nor should you construe the contents of this document as legal, tax or financial advice. Any potential participant in the M^0 ecosystem is urged to consult their own advisors for any legal, tax or financial questions.

To the extent this document contains any forecasts, projections, goals, plans, and other forward-looking statements regarding M^0's project, position, results, and/ or other data, you acknowledge that such forward-looking statements are based on our assumptions, estimates, outlook, and other judgments made in light of information available at the time of preparation of such statements and involve both known and unknown risks and uncertainties. Accordingly, any forecasts, plans, goals, and other statements contained herein may not be realized as described, and actual financial results, success/ failure or progress of development, and other projections may differ materially from those presented herein.

Neither the $M token, nor any governance tokens associated with the M^0 project, will be offered to US persons (without a valid exemption). Any token described in this document has not been registered or qualified under any state or national securities law or regulation.

Participation in the purchase, use, or investment in any digital assets, tokens, or cryptocurrencies involves inherent risks, including but not limited to market volatility, regulatory changes, and technological risks. Prospective investors should conduct their own research and seek the advice of a qualified financial advisor or legal counsel before making any decisions.

The team and contributors associated with this project make no representations, warranties or guarantees regarding the accuracy, completeness, or reliability of the information contained in this whitepaper or any linked materials. They disclaim any liability for any direct, indirect, or consequential losses or damages arising from reliance on this information or any errors or omissions in its content.

By accessing, reading, or using this whitepaper, you acknowledge and agree to the terms outlined in this disclaimer. You are solely responsible for evaluating the risks and merits associated with any actions you take related to the contents of this document.