

Heralding the The First Age

A discussion on the vital importance as well as the risks of token listing on Behodler with a focus on the cryptoeconomics of governance.

by Justin Goro

29 April, 2021

Introduction

Expanding the list of available tokens on Behodler is an obvious way to broaden the appeal of the AMM. In addition to boosting user appeal, every bonding curve added to Behodler adds a level of protection against a class of Scarcity (SCX) whale attacks known as *carpet bombing*. As such, token listing should be given top priority in feature rollout. However, listing without taking certain precautions can lead to a new class of bot attacks I'll label *early bird strikes*.

Since the bonding curves on Behodler follow logarithmic growth, the initial Scarcity created from the first token additions is disproportionately higher than successive tokens. For instance, suppose we consider the initial listing of Dai for swapping on Behodler. The first Dai added (worth approximately one dollar) produces 366 SCX. The second Dai added creates 18 and the third Dai creates 10. It should be apparent that after the first unit, the curve flattens off significantly. If a trader were allowed to scoop up the 366 SCX created from the first unit, this would effect a massively unfair redistribution of wealth that is not justified by early adopter incentives. In the case of Dai, it would lead to 366 SCX acquired for the price of one dollar. It can be argued that any level of SCX above 18 constitutes a whale. Having acquired kraken status through minimal cost, this trader might be tempted to withdraw as much liquidity as possible, making the wealth transfer from Behodler (and hence all remaining SCX holders) real and tangible. The earlier you arrive to a new, empty bonding curve, the more you can withdraw for a given investment. This is the *early bird attack*. It was initially thought that preseeding the first few tokens was sufficient to thwart an early bird attack but recent analysis, both empirical and theoretical has shown that the bonding curves have to be preseeded all the way up to equilibrium levels. Preseeding with anything less opens up the early bird attack vector. However once the equilibrium levels have been reached, the door to early bird attacks is permanently closed. This calls for a new round of innovation on token listing uniquely suited to benefit Behodler.

When Behodler was in the design phase, I added a whitelisting feature to allow specified addresses to trade tokens that were otherwise not officially listed for trading. This whitelisting feature, controlled by governance structures through the DAO, was created to add a certain flexibility to Behodler by allowing future governance to add features through intermediary contracts as innovation demands.

By utilizing this whitelisting feature, we can empower a token listing dapp to control the terms by which as yet unlisted tokens are added to Behodler in a way that both guarantees security and maximizes benefit to the protocol.

This article explains all of the aforementioned concepts in greater detail while providing a proposed token listing solution. This solution will be entirely run by the community through MorgothDAO and will give EYE holders a chance to have a direct say in the calibration of key variables, heralding the dawn of The First Age of a decentralized Behodler powered by MorgothDAO.

Automining

Equilibrium

We define the total value bonded (TVB) of a token trading on Behodler as the dollar value of the full quantity of a token deposited in Behodler. At the time of writing, for instance, the TVB of Dai was approximately \$12000, meaning that Behodler holds 12000 DAI. At market equilibrium, the TVB of every bonding curve on Behodler is the same. That is, if the TVB of the Ether curve is \$10000 then the TVB of MKR is also \$10000. To understand why, one must understand the unique nature of Scarcity. It can be shown that 18 SCX can redeem just under half of the liquidity in any of the Behodler bonding curves. This is regardless of how much value is locked in them. If the TVB of a particular curve is \$100 million, then 18 SCX can redeem \$50 million dollars. We can then prove by contradiction and appeal to the Law of One Price that the TVB of all bonding curves must be equal in value at equilibrium. We can refer to this relationship of equality in value across all curves as **The Law of Uniform Tentacle** because each bonding curve (represented by the tentacles in the logo of Behodler) must be the same length.

Proving of the Law of Uniform Tentacle through intuition

Suppose that the TVB of the ETH curve is \$10 million and that the TVB of the DAI curve is \$4 million. If we redeem 18 SCX of ETH, we'll receive \$5 million. Assuming we can swap this Eth for Dai without slippage somewhere, we can deposit 4 million Dai into Behodler, minting 18 SCX. The net result is that we've made 1 million Dai while keeping our original 18 SCX. In terms of TVB, the ETH curve has fallen to \$5 million in value and the Dai curve has risen to \$8 million. While the relative value positions have swapped, the new difference of \$4 million is lower than the original \$6 million. This process can keep being repeated until the TVB gap closes. At this point, 18 SCX will redeem the same dollar value on both curves and arbitrage trade between them will become impossible. To close the gap, one would use increasingly smaller values of SCX. The choice of 18 is merely illustrative.

Automining defined

Since the liquidity on Behodler is represented by the token Scarcity, minting more SCX on existing bonding curves represents an increase in overall liquidity. Therefore minting SCX is the equivalent of the average TVB increasing.

Suppose we invent a token that requires SCX in order to be minted. Let's call this token Dearth. The way it works is you send in \$10 worth of SCX to a contract which then mints up 1 unit of Dearth. The dollar price is gleaned by inspecting the DAI bonding curve on Behodler (recall Behodler's original purpose was to be an onchain oracle).

We then list Dearth as a tradeable token on Behodler. Suppose the average TVB on Behodler at the time is \$1000. The Law of Uniform Tentacle requires we list Dearth prepopulated with \$1000 of tokens in order to not destabilize prices and liquidity levels on Behodler. Suppose in this example that there is no Dearth in existence. Listing Dearth means we have to mint up \$1000 which requires we lock \$1000 of SCX in the Dearth minting contract. If the SCX required to mint the new Dearth has to be minted afresh then it means that listing Dearth will *drag* new liquidity into Behodler. Since the relationship between Dearth and SCX is variable, there will be a continual back and forth flows between SCX and Dearth in order to maintain the Uniform Tentacle relationship as prices drift in dollar terms. This will lead to regular SCX burning. Each time we list a token like Dearth, we induce second order minting of Scarcity and burning. This is why we refer to this process as automining because Behodler is essentially mining for its own liquidity through allowing the trade of SCX based tokens.

Automining LP tokens to tilt the SCX price

While listing a playful wrapper like Dearth is an interesting exercise in demonstrating automining, there are superior choices of automining tokens to be listed. In particular Uniswap pairs that contain SCX as one side of the pair. In addition to the benefits of Dearth, they facilitate trade of SCX which itself leads to burning. Suppose we list the SCX/ETH pair and suppose the average TVB on Behodler is \$1000. The Law of Uniform Tentacle implies that the TVB of SCX/ETH on Behodler has to be initialized at \$1000. This means that \$1000 worth of LP has to be minted and deposited into Behodler which ensures that the SCX/ETH pool on Uniswap is \$1000 at a minimum. As liquidity on Behodler rises and hence the average TVB, the value of liquidity locked in the SCX/ETH pool will rise.

As the liquidity on the SCX/ETH pool rises, trade volume and average trade size will increase. This will lead to larger quantities of SCX being burnt on a regular basis. In Uniswap V2 and Sushiswap, when one token burns in a trade, the price of that token rises. Because of the [principle of price tilting](#), this creates an incentive to mint SCX on Behodler and sell on Uniswap. Therefore simply listing an LP on Behodler leads to price tilting, one of the goals of the Liquid Queue. The more SCX pairs we list on Behodler, the more regular and pronounced this effect will be.

Market differentiation

The final reason listing LPs is desirable is gas cost. If we want to realize the internal value of an LP, the gas of unwinding on Uniswap is fairly high and significantly higher than performing a swap on Behodler. Therefore, listed LPs offer holders of LP tokens a cheaper exit alternative to that offered on Uniswap. We'll also be able to provide a very low gas alternative for holders of Uniswap LP to migrate their LP to Sushi in order to farm Sushi tokens. Currently the migration cost on Sushi approaches an Eth in value depending on gas fees. On Behodler this could be done for just over \$30 at a gas price of 130 Gwei at current Eth prices.

Carpet Bombing revisited

In the [article exploring attack vectors on Behodler](#), one of the vectors listed involved hitting all the bonding curves in the span of 1 transaction, preventing a natural market correction by keeping prices all relatively unchanged. It was shown that given current block gas limits, this attack is only really possible when the number of tokens listed on Behodler is less than about 100. Depending on gas saving strategies and token implementations, this number may vary in practice but not too significantly. Therefore, one defense against such an attack is to list more tokens. The risk of carpet bombing necessitates that before any large scale liquidity mining operations can commence, Behodler should have at least 150 tokens listed so that newly empowered whales are prevented from striking down the overall price levels. In particular, because of the principles of automining, the best tokens to list are Uniswap V2 and Sushiswap LPs, and later, Uniswap V3 *tokenized positions* (an upcoming Behodler innovation) as this would have the secondary effect of drawing Scarcity back into the other AMMs, undoing the effects of the carpet bombing (and increasing burning which itself has second order effects).

Recap

Before continuing it's worth drawing all of the above concepts into one coherent summary. Firstly, Behodler needs to list more tokens to resist carpet bombing attacks. However, the Law of Uniform Tentacle requires we list every new token prepopulated with the average TVB on Behodler in order to avoid early bird strikes. In addition, releasing the initial burst of SCX into the wild risks creating obese whales who acquired their SCX at almost no cost, divorcing their incentives from the health of Behodler. Therefore token listing needs to be carefully handled before any large scale mining operation can occur. Fortunately it

was shown that listing SCX wrappers such as Uniswap LP tokens with SCX as one half of the pair takes advantage of the Law Of Uniform Tentacle by acting to draw in additional liquidity into Behodler in a process known as automining.

One problem with listing a hundred tokens at once is cost. If the average TVB on Behodler is \$1000 and we list 100 new tokens then each token must be prepopulated with \$1000 bringing the total liquidity cost to \$100,000.

The remainder of this article proposes a solution which meets all of these criteria. The solution takes the form of a fairly simple and modular dapp which will be entirely calibrated in real time by holders of the EYE token in order for the community to fine tune and direct the token listing process according to popularity. The governance driven dapp will be the first live demonstration of MorgothDAO minions in action. The minion in command of the Dapp will be Ancalagon. To be clear the primary purpose of the token listing dapp is to mine for liquidity on a per token basis so that the token can be listed on Behodler for swapping.

The spirit of dapps on Behodler

In any given cryptocurrency community, [there exist individuals of varying preferences](#). Some are passive hodlers, others are active participants who play the varying games and incentives carefully. Some are enthusiastic voices, others are silent critics. Some understand the long term value and others are aped in for maximum APY (wen moon). There are principled ideologues who watch for signs of centralization and mathematical geniuses who simply enjoy the dynamic complexity. It's important to make sure that anything built on Behodler caters to the interests of all the subgroups and doesn't just satisfy one constraint. The best way to do that is to ensure that the dapp leaks benefits into the protocol. As an example of this in action, Behodler's initial pitch was to be a dapp that ensures regular burning of WeiDai. So in addition to giving traders an alternative to Uniswap, it gives holders of WeiDai a guarantee against price inflation while underpinning it with the guarantee of dollar stability. In this way, the ethos of the [Thrifitcoin](#) is guaranteed by Behodler's existence. This is aside from the actual main function of Behodler which is to facilitate swapping and liquidity growth. So the spillover doesn't subtract or interfere with the main purpose. Similarly, the purpose of the Liquid Queue is to not just increase liquidity but to be fun and a little addictive.

This is the spirit in which all dapps integrated with Behodler should strive for. To be very specific of a token listing dapp, this is an enumeration of the goals:

1. Incentives should primarily draw in liquidity.
2. The incentives should be sustainable. Relying on the remaining EYE liquidity mining fund is not good enough.
3. SCX should be carefully and slowly released so as to not put Behodler at risk of carpet bombing and so as not to dilute the value for existing holders.
4. The economic risk of using the dapp should be low.
5. The dapp should be community governed in a manner that utilizes Game Theory to lead to the most efficient outcome and in a way that creates a sink for EYE. This will give EYE its first utility. Rather than thinking of EYE as a governance token, it's more accurate to think of it as a utility token that governs.
6. The dapp must be fun to use.

A word on implementation

Unlike previous dapps released in the WeiDai Ecosystem, this one will be deployed to the Kovan testnet for user feedback and familiarity. By the time it launches on mainnet, all UX surprises, confusions and glitches should be ironed out entirely. One of the lessons gleaned from the Liquid Queue Beta is that it was a bit too open for a beta that uses real money. We should never again skip the traditional testnet approach to feature launching. Take it as a given that the following dapp will be available on Kovan both before and during the running of the mainnet version so that the community can throw around experimental fake money to get a handle on how things work.

Description of a community token listing Dapp: Morta

The Dapp is as yet unnamed, another task the community can collectively decide on. The contract which gatekeeps tokens on Behodler is called Lachesis, named after one of the fates (Morai) of Ancient Greek mythology who was responsible for measuring the life of mortals. We'll continue with this nomenclature as a placeholder and refer to this token prelisting dapp as Morta, another of the sisters of Fate. The goal of Morta will be to take potential tokens for Behodler and get their TVB up to comply with the Law of Uniform Tentacle. At this point, a governance action can permanently list the token on Behodler for trade.

Morta will present a list of all tokens that have the potential to list on Behodler. Think of this as the final waiting room for potential tokens. These tokens will be comprised of utility and governance tokens such as Aave, Sushi and vBTC as well as liquidity token pairs for Uniswap, Sushiswap and later Uniswap3 tokenized positions.

Whitelisting

Behodler has a whitelisting feature that allows specific addresses to be given the ability to trade tokens that are otherwise forbidden on Behodler. If we utilize this feature to give Morta full control over the minting of SCX from new tokens, we can set the "terms and conditions" of how these tokens are added to Behodler and what happens to the SCX. We now have the foundation to put a safe token mining dapp in place.

Iterative Description

To assist the reader in following along on the journey to the final dapp, it will be described iteratively, each iteration improving on the shortfalls of the last. The theory motivating the dapp will be described afterwards in the appendix so that the reader arrives armed with intuitions on their side. Just to be clear, the iterations do not reflect versions of the dapp in reality. They are just an explanatory literary device.

First iteration: Delayed SCX purchase

Suppose we wish to list Sushi on Behodler. We want to allow holders of Sushi to supply Sushi in return for receiving Scarcity. However, we don't want to give it all away, lest we get an early bird strike or a slew of carpet bombing. So we create a contract for staking SCX. A user supplies Sushi to a whitelisted contract which deposits the Sushi in Behodler, generating SCX. After some lock period, the user is able to retrieve their SCX. To reduce the danger of releasing all the SCX at once, we divide it into 10 even chunks. released over 3 months.

Criticism of The First Iteration

While this first iteration satisfies the condition of delaying the dissemination of SCX, locking up user funds for long periods is an unpleasant UX experience. Dividing into multiple locks may improve the long wait or just frustrate the user by forcing them to endure the gas cost of many withdrawals.

Digression on concepts

The Law Of Uniform Tentacle tells us that at equilibrium, the value of locked liquidity (TVB) in all bonding curves is equal. We'll refer to this number as the prevailing TVB. Our goal is to list a token on Behodler but we can only do so when we've crowdfunded a large enough pot of the token so that the token bonding curve can be preseeded with enough liquidity to match the prevailing TVB. When we've raised enough tokens, we'll say we've crossed the critical threshold for listing to take place.

Second iteration: SCX farming

A more user friendly experience would be to create a farming dapp similar to so many of the token farms on Binance Smart Chain (BSC) or to Sushiswap's Onsen farms. Here we wish to allow the user to stake their input token, to then generate SCX from the staking and to release the SCX at a fixed ongoing rate. To achieve this, when a liquidity provider deposits their token, the contract deposits into Behodler to generate SCX which is then put into a reward pot. Users are then rewarded in linear proportion to their staked tokens. This does mean that the SCX slowly drains out which means the staking period, by definition is finite. Liquidity providers can unstake and are not locked in. When a user leaves, the SCX required to redeem their tokens is calculated and withdrawn from the reward pot. A standard `removeLiquidity` call to Behodler is made and the resulting tokens are forwarded to the departing user. A discerning user may notice that there may not be enough SCX to redeem tokens for an unstaking user because it was given away as staking rewards. Bear in mind that the first unit of a token provided produces 366 SCX. If we're at the point where 200 units of a token have been added, then the SCX cost of redeeming tokens to give to a departing user will be very low. The excessive SCX produced from the early units provides some wiggle room here. But if users farm long enough, eventually there will be an SCX shortfall that will not be possible to meet.

The hope is that enough farmers join the pool to reach the critical threshold required to comply with the Law of Uniform Tentacle before the staking rewards run out. At this point, the contract locks all withdrawals, and the token becomes tradeable on Behodler.

Criticism of The Second Iteration

Since SCX is scarce, the rewards could run out before a token has raised enough liquidity. More importantly, if all of the SCX produced is used to pay out farmers, then we still run the risk of future carpet bombs.

Third iteration: Fractional SCX farming

The initial kick of SCX produced provides enough breathing room to guarantee a high APY to future providers without having to hand out all of the SCX. To illustrate this concept, consider listing the Sushi token again. Suppose 3 people provide 1 Sushi each. The first person to farm will produce 366 SCX which will

be added to the pot. The second person will produce 18 and the third 10. In total, for 3 Sushi, the reward pot has 394 SCX. The price of Sushi at the time of publication was \$11. the price of SCX was \$334. This means that for \$33 of Sushi, we have \$135,536 dollars of rewards. This is clearly a bit excessive. We could

guarantee providers triple their wealth positions (\$66) and burn the rest of the SCX for the benefit of the community.

Digression on Picking the perfect level of SCX to give away

By now the reader may be coming to an intuitive understanding that there will be a lower bound of staking rewards that are equal in value to the input tokens provided and an upper bound which equals the entire SCX produced. Neither extreme is desirable but there is an optimal sweet spot which is just high enough to entice the staker but not so high that it undermines the value of SCX. This concept is formalized into a mathematical relationship in the appendix.

Criticism of The Third Iteration

While we can now reduce the risk of carpet bombing through burning the vast majority of SCX produced, we've now accelerated the reckoning at which the dapp can no longer pay rewards to stakers.

Fourth iteration: Protocol token farming for eventual SCX swap

Many farming dapps, inspired by Sushi's vampire drain, reward users with a token which is minted every block at a fixed rate. While this seems to suggest unlimited inflation, there are usually sinks elsewhere in the system to absorb the newly minted tokens. If this is done correctly, the net tokens minted could be zero. To provide an example of how this is possible, consider Ethereum once EIP1559 has been implemented. In this EIP, gas fees are burnt while miners are rewarded a fixed issuance of Eth per block. If the Eth issued is 0.5 per block but the fees amount to 0.6 Eth per block then the net growth of Eth is negative (deflationary), even though on the surface it appears inflationary.

For all the major inflationary tokens such as Sushi, Dogecoin and Eth, consider that the token issuance is linear and fixed unlike fiat currencies which compound their inflation with percentage growths. Sushi uses allocation points to distribute the new rewards across all the mining pools. For a token that issues 1 token per block, the first year on Ethereum will see approximately 2102400 tokens issued. The following year will see another 2102400, implying an inflation rate of 100%. By year 3, the inflation rate has halved to 50% (2102400/4204800). By year 7, the inflation rate has dropped to 14%. Linear issuance tokens are therefore characterized by falling inflation rates, even though the token issuance is unlimited.

Taking advantage of these dynamics, our dapp now issues a new token a fixed rate. For simplicity, let's assume we're only farming 1 input token. In keeping with the popular food nomenclature for reward tokens, let's call the reward token Flan (as in the dessert). And to distinguish it from the input token being mined, we'll assume that Aave is the token we're listing.

A holder of Aave will stake their Aave and receive 0.001 flan per second per Aave staked. For instance, if someone stakes 34 Aave for 5 minutes, they'll farm

```
(34*5*60*0.001 =) 10.2 flan in total.
```

Since there is no constraint on the flan minted, holders of Aave can come and go, receiving staking rewards. At some point, it is hoped that enough Aave is staked to trigger the listing of Aave on Behodler. Until there is enough Aave, it simply sits in the staking contract. Once the threshold is reached, the entire portion is converted to Scarcity via the addLiquidity function. The circulating flan then immediately gets a fixed exchange rate. Holders of flan can send in their flan and receive the Scarcity they're owed. Here we can also

burn some of the Scarcity, without risking being undercapitalized. The more SCX we burn, the lower the final flan exchange rate. But since the initial SCX is much more than the market price implies, we have the ability to inflate the flan supply quite significantly without undercutting user rewards.

Criticisms of the Fourth Iteration

While the circulation of flan ensures that rewards flow to users in a UX friendly, stable manner, flan essentially has no value until the liquidity is ported. This makes it hard to price on secondary markets like Uniswap or Behodler. It would be preferred if users could immediately offload their tokens as they receive them. Imagine receiving flan and knowing its precise dollar value already and not having to wait for a promised SCX.

Fifth iteration: Multiple staking contract

Here the setup is identical to the fourth iteration but repeated many times in parallel. That is, there are multiple staking contracts for each potential token. Some will reach their critical threshold faster than others, depending on user preference. Those that finish first provide a price for all outstanding flan, facilitating price discovery. As tokens begin to list, more and more SCX will come online to be used as a backing for the circulating flan.

By having many token listing contracts coming and going, flan begins to find a support base of SCX that allows secondary markets to function efficiently, providing holders of flan to sell and for those interested in acquiring flan the opportunity to do so without resorting to staking liquidity. This extra base of HODL buyers provides a demand base to increase the incentive to farm for liquidity because the secondary market price of flan rises.

Criticisms of the Fifth Iteration

There is no guarantee that secondary markets will have sufficient liquidity to allow for flan trading. One workaround is to list flan itself as a token to list on Behodler so that at the very least liquidity for flan increases on Behodler. However, this precludes other exchanges which cuts off the ecosystem from bot and speculative purchasing power found on Uniswap and Sushiswap.

Sixth iteration: Protocol token farming backed by open market operations (OMO).

As in the previous iteration, flan is issued by multiple contracts. Before launching Morta, we create a flan/SCX pool on Sushiswap. Since no flan exists yet, when the first staking contract deploys, the LP can be seeded with say 100 SCX and mint 1000 flan. This is then used to initiate the flan/SCX pool with an arbitrary starting point.

Here the staking mechanics are the same as before. However, when a staking contract reaches the critical threshold to satisfy the Law of Uniform Tentacle, things change:

1. As before SCX is generated from addLiquidity and a portion is burnt.
2. Half of the remaining SCX is used to buy flan on another AMM such as Sushiswap.
3. The contract now holds SCX and flan equal in value. The SCX and flan are then wrapped as LP tokens and locked away permanently.

In this way, holders of flan can't actually redeem their flan for anything but the market operations of the staking contract have ensured that the flan price is boosted and that the liquidity of flan is deepened at the end of every token listing round. A spontaneous market for flan is created and supported by every listing event.

The appendix contains a mathematical dive into showing how flan optimizes a number of variables relating to Scarcity burning and value retention.

Digression on End of Staking implications

If the reaching of the prevailing TVB results in all staked tokens being locked and then transferred, it may appear that the very last person to stake receives absolutely no reward for doing so. For this, reason it's important to explore an end of staking reward. One approach is to just allow the flow of flan to staked users to continue for another day. Another is to give the final user who triggers the end of staking a disproportionate reward as an incentive to wind things up.

Criticisms of the Sixth iteration

While a deep flan/SCX pool is great for both tokens, the front end routers used by Uniswap and Sushiswap still favour Eth pools disproportionately. We've found a way to increase the depth of Eth based liquidity pools for both EYE and SCX: simply list the ETH/EYE and ETH/SCX pairs on Behodler and as the AMM grows, automining will take care of the rest. But we have no such analog for flan.

Seventh iteration: Protocol token farming, backed by OMO with and automining underpinning

Here the mechanics are exactly the same as the sixth iteration but our choice of tokens to be farmed includes a number of flan Uniswap and Sushiswap pairs, starting with flan/ETH. This bootstrapping of flan enables the token listing dapp to accelerate its addition of other tokens by issuing a high value token as a staking reward. That is to say, if the price of flan rise and the issuance per block is fixed, then there's more money to go around.

Eighth iteration: Protocol token farming, backed by OMO, with automining, orchestrated by EYE governance

"...and out of the pits of Angband there issued the winged dragons, that had not before been seen" - The Silmarillion, J.R.R. Tolkien

MorgothDAO has been in control since the launch of Behodler 2 but has mostly been lying in wait. Now the time has come to turn EYE toward its true purpose of governance and to release the first of Morgoth's minions. Before outlining how this will work, it's worth explaining a bit of the taxonomy of MorgothDAO.

A DAO of DAOs

All of the governance functions of Behodler are enumerated into powers. For instance, there's a "ADD_TOKEN_TO_BEHODLER" power and a "CONFIGURE_Scarcity" power. Every power has one domain. A domain is a contract. For instance, CONFIGURE_Scarcity has the Scarcity contract as its domain.

ADD_TOKEN_TO_BEHODLER has Lachesis as its contract. A domain can be associated with more than one power but a power can only refer to one domain.

While the list of powers and domains is small to start with, there exists a mechanism on Morgoth to add more powers and more domains. In this way, Morgoth's authority can grow arbitrarily large. As you may have guessed, the ability to create new powers is itself a power.

So who gets these powers? The wielders of powers in Morgoth are minions. These minions include Sauron, Glaurung, Saruman, Smaug, Gothmog etc. Each minion can have one or more powers or none at all. Each minion is mapped to a controlling Ethereum address. These addresses can be human controlled or be contracts themselves. In a sense, each minion can be controlled by its own DAO, making Morgoth a DAO of DAOs with infinite flexibility. In this way, Morgoth is not opinionated on the style or ideology of governance. Some minions could be simple democracies, some might be liquid democracies, some might be authoritarian oligarchies, some might make use of innovative governance techniques such as holacracy or quadratic voting. Morgoth cares not how each minion organizes itself.

Pouring power

This taxonomy allows us to divide up all the governance powers of the entire Behodler ecosystem among the minions. So who's doing the dividing? Well Melkor of course. When MorgothDAO launched, there was only one minion, Melkor. And he had all the powers in himself. And as Morgoth grows, Melkor will increasingly pour out powers into minions and in this way, as the minions become empowered, Melkor's direct role diminishes. Instead of leaving Behodler fully open and vulnerable to takeover from nefarious interests, we'll gradually dial down the power of Melkor as the other minions mature and take over, ensuring that decentralization is handled in the most responsible manner while also occurring as swiftly as possible. There's a function "castMelkorIntoTheVoid" which is the final step in completely disempowering Melkor.

Ancalagon

The dapp Morta will become a domain on Morgoth. Then all the governance functions will be enumerated as powers and all of those powers will be poured into the minion, Ancalagon. We can then put a mini DAO in place which controls Ancalagon. The mini DAO will act according to the will of EYE holders.

Eighth iteration revisited

The eighth iteration of Morta will be the same as the seventh but this time the community of EYE holders will have direct control over the calibration of every token being farmed. This will include how much flan to issue per second and how much SCX to burn on listing (within reasonable bounds). Community level fine tuning in this manner will allow those with their ears to the ground to jump in and help steer the ship in a way that most maximizes the value of EYE and SCX. There are precise metrics that will help inform these decisions (discussed in the appendix) so that those governing token listing dapps have a sense of whether their decisions are harmful or helpful. Front end indicators and health statistics as well as complementary analyses of the kinds we saw popup during the LQ beta will also assist the community in fine tuning and discovering optimal variable. In the event of an emergency, EYE holders can also pause or cancel farming on a token. By requiring EYE to be staked in order for governance actions to be performed, we're providing a demand base for EYE. The precise nature of the governance is beyond the scope of this document and will be fleshed out in due course. However, staking EYE and possibly EYE/ETH and EYE/SCX LP will be at the heart of the process.

Ninth iteration: Eighth iteration but with a Bonfire.

The ninth and tenth iterations are to demonstrate the future utility of flan in addition to token listing and illustrates the benefit of pursuing the creation of flan.

Now that flan exists, we have an economic tool to benefit the ecosystem further. Provided we don't inflate beyond the demand base created by new token listings, we can direct some of its power to supporting both the value of Scarcity and EYE. This will both strengthen the health of Behodler, add value to governance staking and provide a demonstration of how powerful a tool flan will become going forward. This is how it will work:

1. If flan mints at a fixed emission per block, we can allocate portions of that minting toward a new small dapp called Bonfire without increasing the inflation rate in exactly the same way that Sushi's audited Masterchef contract does with *allocPoints*.
2. Bonfire receives a fixed rate of flan per block (or per second). Flan therefore accumulates in Bonfire
3. Using governance levers, we set a price in EYE. Suppose the price is 1000 EYE.
4. Anyone can send in 1000 EYE and receive the accumulated flan.
5. The EYE sent in is burnt, creating an ongoing deflationary pressure on EYE.

In this way we have an ongoing perpetual incentive to burn EYE. To recap, Bonfire is a simple contract that is topped up with flan which it uses to pay users in return for them sending in EYE which is burnt.

Tenth and final iteration: Burn SCX in Bonfire

Everything is the same as in the ninth iteration but this time, 10% of the EYE sent in to Bonfire is set aside in a second Bonfire contract. Here, the EYE accumulates and receives a price in SCX. Again we set an arbitrary price using governance. Let's say 1 SCX. Anyone with SCX can send in their SCX to receive the EYE. The SCX is burnt in its entirety.

The appendix contains a section explaining the purpose and dynamics of the Bonfire addition.

Conclusion

Listing new tokens is essential to both secure Behodler from carpet bombing whales, to enable future liquidity mining events and to broaden its appeal.

However, listing a token without preseeding it with a level of liquidity equal to the prevailing TVB dilutes Scarcity and opens up Behodler to a high speed liquidity withdrawal known as the early bird attack.

Since, by definition, a higher value of SCX is created from the preseeding than is justified by the prevailing market price, it was suggested that some of this surplus SCX be used to incentivize users to bring their liquidity to Behodler. Various options were explored but it was found that the slower this SCX is released, the safer it is to the ecosystem and so we gravitated toward a farming dapp where SCX is the reward token. However, since SCX is limited in supply, this design necessitates a limited duration staking round which may mean some tokens never raise the liquidity required to be listed. It was then shown that a fixed rate per block minted token would allow stakers to earn rewards in perpetuity. Furthermore, if the SCX generated from the listing event is used to deepen the liquidity of the reward token through open market operations then users can earn the token, knowing they can always sell it.

Finally it was shown that the reward token cryptoeconomically guarantees that SCX and EYE will always have a strong demand base.

Appendix

Mathematical analysis of Flan

Residual Scarcity

For every token listed on Behodler, query the balanceOf function on Behodler's address. Then run that number through the transformations that would generate Scarcity on Behodler. Sum up all the Scarcity. In a world of no burning, this number should be identical to Scarcity's total supply. Yet if you then query Scarcity's total supply, you'll notice that the true total supply is lower than the calculated version. This gap is known as residual Scarcity and represents the growing sink.

Flan Growth constraints

Suppose we wish to list a new token, T. The prevailing TVB implies that at the current market price we have to list t units of T. When we deposit t units, we generate x units of Scarcity. The current market price for 1 unit of T as shown on prevailing exchanges (both centralized and decentralized) is p . For purposes of comparison, let's assume that p is the dollar price. We define the total market value of all t tokens on Behodler as

$$m = pt \quad [1]$$

Next we wish to take the Scarcity produced by depositing t units into Behodler and calculate the dollar value so as to compare to m . Let the dollar price of Scarcity as shown on Uniswap or as shown by setting the redeemable Scarcity on the Dai Behodler curve at 1 unit as c . Then the total market value, s , is

$$s = cx \quad [2]$$

We've defined t as the equilibrium level of t required for the T bonding curve to comply with the Law of Uniform Tentacle. Let q be an arbitrary number of units of T below equilibrium such that $q < t$. Let y be the SCX produced from adding q . Then we know that

$$cy > pq \quad [3]$$

This follows from the logarithmic nature of Scarcity production where the first derivative of $x = f(t)$, $dx/dt \propto 1/t$ (there's a constant scaling factor thrown in there that has no bearing on final price comparisons).

This means that right up until t ,

$$cx > pt \quad [4]$$

In other words the market value of the total Scarcity produced exceeds the market value of the tokens added. If this Scarcity were circulated to the general public, it would lead to an excessive claim on liquidity so that when redeemed would leave Behodler with less liquidity than before the SCX was minted. This is an unsustainable position and redistributes wealth from existing SCX holders to those who get in on the bonding curve early.

There must be some quantity of SCX, a , such that

$$ca = pt, \quad a < x \quad [5]$$

If we take a and release it to contributors of T in linear proportion of their contributions then liquidity providers will get the exact value of what they put in. This is the lower bound of rewards since it provides no net incentive to contribute liquidity. On the other extreme, if we distribute x to contributors in proportion of their contributions then each contributor will receive rewards far in excess of their contributions which can create all sorts of nasty secondary incentives (eg. pump and dump). The excess Scarcity, $e = x - a$ is the range that we have to work with and within it will be the optimal level of Scarcity to release such that holders of t have an incentive to provide Scarcity but not so high that we undermine the value of Scarcity. Let this optimum be given by k . Our staking contract must then burn $x - k$ Scarcity to guarantee the health of SCX.

Above, we defined the discrepancy between the true Scarcity supply and the implied supply as residual Scarcity. This is the same as the total Scarcity burnt. Let the growth in residual Scarcity per second be given by r . This number will obviously vary from block to block but over a range, we just average it out. Suppose we list a token for staking and raise t units of T . the whole process takes 1 hour. We want to release k units to the public. We know $k - a$ is a surplus we're using to induce suppliers with an incentive to provide. As long as the following relationship holds,

$$k - a \leq 3600r \quad (3600 \text{ seconds passed in the hour of crowdfunding}) \quad [6]$$

then net Scarcity produced will not be positive. That is, we'll have no net Scarcity inflation. One of the major implications from the equation 6 is that if we can increase r , we can increase k , meaning that we have more leg room in which to reward stakers without destabilizing Scarcity.

Scarcity dynamics

With respect to assisting the task of token listing, the purpose of automining is to increase r in equation 6. By increasing the levels of Scarcity locked in Uniswap and Sushiswap, we increase both the potential trade size and arbitrage opportunities for traders. Let's generalize equation 6 further by representing the crowdfunding duration with the letter z for zeit, the German word for time, because we're running out of letters:

$$k - a \leq zr \quad [7]$$

We now have two variables at our disposal for optimization, r and z . By increasing z , we can increase $R(z)$, total Scarcity burnt during the crowdfunding period which allows us a greater value for k .

In the initial iterations of Morta, the stake duration was precisely to maximize z . However, this comes at the cost of UX and at an opportunity cost. If users are not rewarded immediately, they have to consider if their staked tokens would be better suited to other yield opportunities in the DeFi ecosystem. The staking rewards are then discounted by the best alternative opportunities.

Flan solves this by offering an immediately monetized reward for staking while still maximizing z by delaying the release of Scarcity until the very last possible moment.

The versions of Morta that utilize market operations to support the price of flan indirectly through pooling and purchasing operations on Uniswap and Sushiswap assists in providing SCX for trading which itself raises the value of r .

In this way, flan maximizes both r and z simultaneously. It should also be noted that when Scarcity is added to AMMs, less of it breaks free into the wild than if given directly to stakers. This gives us greater flexibility to increase k . Suppose that for a given k -a of SCX sent to Uniswap, some proportion is permanently locked away because of Uniswap pricing dynamics. Let this proportion be b for *bhāga*, the Punjabi word for fraction. Then we have,

$$(1-b)(k-a) \leq zr, \quad 0 < b < 1 \quad [8]$$

From equation 8, we can see that using AMMs to lock up SCX/Flan provides yet more room in which to increase k , the reward alpha for stakers.

It should also be clear that prior listing events leave a permanent liquidity mark on the external AMMs. This implies that each listing event has a permanent effect on raising r . In other words, each successive token listing event is able to "stand on the shoulders" of the prior history of listing events by providing increasingly high levels of k , as represented by a flan emission rate. In other words, if the flan price rises because of increasing SCX being pooled with flan, the rate of token emission in flan need not fall. This may eventually catapult Behodler as the go to place for initial token offerings which requires governance decisions, which draws in EYE.

The purpose and dynamics of Bonfire

Equation 8 is for one token listing contract. However, Morta will have many tokens running simultaneously so that the market may decide on which tokens should be most urgently added to Behodler. If we sum k -a across all the staking contracts, we'll arrive at the total value of SCX available for distribution through the issuance of flan. Let this value be represented by V . Suppose we have 100 tokens listed and the average k -a is 2 SCX. In total, V is then 200 SCX. This implies that over the total lifetime of the token listing events, $R \geq 200$. Now suppose we wish to allocate some flan issuing power to Bonfire without raising V beyond 200 SCX so that the effect on Scarcity is neutral. We lower the average k -a issuance on each contract to 1.9 SCX. This frees up $(100 \times 0.1) = 10$ SCX for issuance by Bonfire. We calculate the max flan issuance by using the flan/SCX price and find that the price of flan in terms of SCX is 1 SCX buys 240 flan. This means we have 2400 flan up for minting over this period of staking. 1000 EYE is worth 1200 flan. 2 people each send in 1000 EYE, each receiving 1200 flan. 1600 EYE is burnt, the remainder 400 is sent into the SCX Bonfire contract. From the above exchange rates we see that 1 SCX is about 20 EYE. This means the EYE in the SCX bonfire contract is worth 20 SCX. Someone sends in 20 SCX for burning.

This last action raises the value of R . From equation 8, we can see that this raises z_r which means that k can be raised.

In other words, although Bonfire draws away flan that could otherwise be used for rewards, it indirectly raises the amount of possible flan available for rewards.

The final purpose of Bonfire is to inspire imagination in the community for other creative uses of flan .

Recall that as staking contracts are added to Morta, their available k -a will increase. While offering this entire increase to stakers is possible, another possibility it siphon off some of that growth for use in other areas in Behodler in which to add value.

Technical challenges

Quantifying the best ranges in all the values up for calibration by the community will be quite tricky. This is why a listing Morta on Kovan first will allow users to experiment with governance decisions and witness the results of the experiments first hand before we apply the lessons of the testnet to a final mainnet implementation.