The Moloch DAO

Beating the Tragedy of the Commons using Decentralized Autonomous Organizations

Ameen Soleimani, Arjun Bhuptani, James Young,
Layne Haber, Rahul Sethuram

DRAFT v1.0

Prepared by Arjun Bhuptani

1. Purpose

Who and What is Moloch?

The name "Moloch" refers to the Canaanite God of child sacrifice [1]. In ancient times, Carthaginians believed that sacrificing a child to Moloch in times of war would increase the odds of their tribe's victory. Members that didn't participate were seen as risking the survival of the entire group. This practice continued for years, despite toll it it took, because of the power of the group's incentive structure.

We first came across Moloch in "Meditations on Moloch" by Scott Alexander [2], which discusses the eponymous character from Allen Ginsberg's poem, *Howl* [3]. In *Meditations,* Alexander points out that while Ginsberg is typically interpreted as discussing the pitfalls of capitalism, the poet is likely instead referring human beings' tendency to consistently choose suboptimal solutions to group coordination problems. Alexander gives the following example of a dictatorless dystopia:

"Imagine a country with two rules: first, every person must spend eight hours a day giving themselves strong electric shocks. Second, if anyone fails to follow a rule (including this one), or speaks out against it, or fails to enforce it, all citizens must unite to kill that person. Suppose these rules were well-enough established by tradition that everyone expected them to be enforced."

In such a society, it becomes rational for all individuals to continue to shock themselves, because if they don't, they are killed and anyone who agreed with them would also be killed (and so on). From our God's Eye view, however, we know that if *everyone*, or at least the majority, of the population coordinated to stop shocking themselves all at once, it would result in a better society for everyone. Alexander goes on to give examples of several real-world situations where this occurs, such as overfishing, arms races, and congress.

In this paper, we take Moloch to mean what Alexander represents in his post: the category of problems associated with collective action, where individual incentives are misaligned with globally optimal outcomes. These are problems that have existed throughout human history under different guises but yet have remained largely unsolved. We see Moloch as a "final boss" of humanity, something which humans have unsuccessfully struggled to overcome for millennia, and which will have to be beaten if we want to ensure the long term survival of our species.

In the context of the Moloch DAO, the name "Moloch" is an ironic joke about the possible futility of attempting to solve coordination problems. It is also a meme in and of itself -- one which we hope spreads awareness of the core problem and the recognition of real life Moloch problems.

Problem Overview

We define public infrastructure (or the blockchain "commons") as technology where the total benefit generated by the technology to the community is greater than the individualized benefit to any particular entity. This conversely means that the cost for any one entity to invest in developing infrastructure is disproportionate to the benefit accrued by it. Naturally, this presents an incentivization problem, often leading to the infrastructure not being developed at all.

In the Ethereum industry, we see this problem most prominently in the development of Eth 2.0. As the recent The State of ETH 2.0 report (commissioned by the founders of this DAO) found, progress towards Eth 2.0 has been slow and conducted mostly by a relatively small group of researchers and implementers with limited funding [4]. Despite the fact that *all* Ethereum projects stand to benefit from the upgrade, only a few major stakeholders are directly involved in funding and development. .

Unfortunately, this is rational economic behavior: building infrastructure for Ethereum means incurring a large personal cost but benefit is split between the entire Ethereum ecosystem [6]. What is the incentive to bear that cost (and, for projects, possibly give up market position) while that is the case?

These types of problems, which we define as Moloch problems, require realigning incentives to solve. By using a DAO structure to pool funds and vote on fund allocation, stakeholders share more effectively coordinate to share costs, resulting in greater overall ecosystem benefit. . Our hypothesis is that this is enough to provide the necessary "activation energy" to trigger widespread coordination between groups which otherwise may have competing interests. The Moloch DAO is an experiment that tests that hypothesis.

Goals

The immediate goal of the Moloch DAO will be to fund and further the development of public infrastructure related to Eth 2.0. This will be done by incentivizing coordination between the various Eth 2.0 projects and major ecosystem stakeholders.

We pick Eth 2.0 as our initial goal because we believe it to be the most obvious, most pressing instance of a Moloch problem which affects everyone in this ecosystem.

The secondary goal of Moloch will be to research collective incentivization and iterate towards a generalized version of Moloch that is applicable to an even wider range of suboptimization problems regardless of scale or scope.

2. Game Theory

Philosophy

The ideas behind the Moloch DAO's mechanism design come largely from historical discussions of the core problem. While reviewing these discussions is outside of the scope of the paper, we recommend reading:

- Tragedy of the Commons - Garrett Hardin [5]
- The Prisoner's Dilemma [6]
- Logic of Collective Action - Mancur Olson [7]
- Governing the Commons - Elinor Ostrom [8]

Aligning Incentives

Finding a solution to this problem requires restructuring incentives so that costs and benefits are split equally amongst all participants. The Moloch DAO does this by pooling user funds and locking them up in a contract called the Guild Bank. Contributors to the Guild Bank are given voting rights over how those funds should be spent, proportional to their contribution relative to the total pool.

$$votes\ (\%\ of\ total) = \frac{contribution}{total\ value\ of\ Guild\ Bank}$$

This core mechanism is simple and we do not expect to drive new behavior. However, two additional mechanisms exist which make the DAO much more interesting:

Members of the DAO are able to liquidate their votes to retrieve a proportional share of the funds from the guild. This gives a strong financial incentive to members to try to maximize the value of their votes either by increasing their proportional ownership of the pool OR by increasing the value of the pool overall. Note that liquidating their votes means sacrificing decision-making capabilities over the future of the pool.

Self vs. Group Interest

If, for instance, members need to build shared infrastructure which would result in collective benefit (represented by an increase in the value of the resource), then the value calculus for whether to build the infrastructure is now based on the probability that the group's total cost would be lower than the group's total benefit. In other words, for a rational investor, it makes sense to invest in a proposal when that proposal is likely to increase the value of the pool by more than the cost that is split amongst all members.

The simplest way of representing this cost is to inflate the supply of the index token so that all members are diluted proportionally. As an analogy, imagine if 50% the companies represented by the S&P500 need some open source accounting infrastructure. This infrastructure would cost any individual company 5% of their stock price to build and would only improve each company's stock value by 1%. Building the infrastructure would not be economically rational for any company individually, because the cost would far outweigh any benefit.

However, if the investors in the S&P500 inflated the supply of the index by just 0.01% (i.e. 5% divided by 500), diluting themselves down by that amount, they could pay that to an independent organization to build and open source the technology and have each company integrate it. This would yield a stock price benefit of 1% across 50% of the companies the index (0.5% benefit to the S&P500 overall) meaning that the investors would have gained 0.49% value on the index.

Note here that we are not actually inflating the value of the underlying asset (ether, in this first version), just of the index (voting shares in the Moloch DAO) which represents the asset. This is important because the investors, i.e. the primary beneficiaries of the the index, are the ones who pay the cost of coordination. This holds true in the Moloch DAO as well.

Restricting Access

We assume that members of the DAO will be at least somewhat aligned on overarching purpose - whether it be to fund Eth2.0 development or some other infrastructural goal. Second, we assume that members will be willing to sacrifice short term benefit in order to pursue the purpose of the DAO. For ETH 2.0 we assume that rational long-term ETH holders and projects building on the platform will see the value in contributing to its ongoing development, lest they see the value of their holdings drop and their platform be technologically surpassed.

In a truly permissionless paradigm, these would not be acceptable assumptions to make. We instead choose to restrict access to joining the guild by having existing members vote on new entrants in a similar fashion to funding proposals.

This solves both of the above problems. For the first, existing members are naturally incentivized to only admit new entrants who are aligned with the guild's interests in order to share costs with them. For the second assumption, we expect that when considering a new applicant, existing members will consider previous interactions with that applicant which will drive applicants towards longer-term benefit strategies. We cover this more in depth in "Ragequitting" below.

Ragequitting

We recognize that in any voting-based system, there are a large number of edge cases created by possible collusion or unavailability of stakeholders. To handle these, we built a catch-all mechanism that allows participants to exit with their funds if they did not agree with the result of a vote. This is done by allowing members to "Ragequit" the guild within a "grace period" after voting on a proposal completes but before those members' ownership is affected by that proposal.

Let's say for instance that 99% of the guild colludes and submits a proposal to issue 99% more Shares to themselves and dilute the remaining 1% down to effectively zero. In this case, the 1% would Ragequit the guild during the grace period, negating the majority voting bloc's attack.

Note that the remaining members after the grace period ends bear the cost of the proposal. In the case of a contentious vote where a large minority exits (e.g. a 51% attack), this means that cost is greatly increased for those who stay. To enforce this, we restrict Ragequitting to only members who voted "No" in a given proposal if the proposal passes. This *forces* "Yes" voters to bear the cost of a malicious proposal.

We expect this to create an interesting additional meta-incentive for mutual cooperation, since guild members would be strongly disincentivized to submit proposals that might cause a large proportion of other members to ragequit.

Readers may note that an attack vector exists where participants can repeatedly Ragequit to either avoid dilution or grief the guild. This would be an effective method to avoid paying the cost of a single proposal. However, if the member wanted to avoid paying they would have to Ragequit the entirety of their shares, completely removing them from the guild. They would then have to reapply as a member, and we expect that existing members would be hesitant to readmit the defecting applicant. From existing game theoretic research, we are confident that this will be a strong incentive for members to not abuse the ragequit mechanism. [9]

Hypercompetition and Forked Evolution

The Ragequit mechanism can push Moloch to fragment into sub-entities when contentious disagreements are reached. We also upgrade Moloch by having all members ragequit and start over.

This is a feature, not a flaw. We want Moloch to be forked, upgraded and iterated on rapidly, both with new onchain and offchain mechanisms. A major driver for this will be guild size - members are incentivized towards increasing the pool value since it would mean a lower per-person cost per proposal, but are also incentivized to keep guild small enough to retain voting power and keep guild philosophy aligned on a specific goal.

Once this balance is exceeded, we believe that there are strong incentives for new applicants to fork the core guild and build their own. Since the code is open source, and since they could easily add rewards for early participants in their own guild, we hope that this will eventually create a hypercompetitive market for Moloch DAOs [10].

Free Riders

Any solution to the Tragedy of the Commons would be remiss without a discussion of the free rider problem. Simply put, how do we ensure that other entities do not also share in the benefit of an ecosystem when the cost is paid only by a single or small group of projects?

Solving this problem is not possible when the Guild Bank is collateralized only with wETH as all ETH holders would always share in some of the benefit of Ethereum's upgrade and the increased price/utility of ETH. However, we do still believe that there are enough stakeholders who care enough about coordinating around Eth 2.0 that we will be able to validate the technology and core game theory.

When the Guild Bank contains assets besides just ETH, this will become a much more complex calculus. We don't know for sure yet what effect this will have on the guild, but we're highly interested in collecting data and contributing to our more general theoretical understanding when it is live.

3. Minimum Viable DAO

Learning from Historical DAOs and Blockchain Development

In building the Moloch DAO, our first concern was to ensure the security of deposited funds at all times. We looked to past major security breaches on Ethereum, such as the 2016 DAO hack and both Parity multisig hacks, to put together a set of design principles for secure development.

First and foremost, we set a goal of putting only the absolute minimum set of functionality on-chain. In the vast majority of edge cases, social coordination mechanisms (offline coordination) can be used as roundabout solutions to problems. By minimizing on-chain coordination, we reduce the potential attack surface that we need to account for.

Second, we decided to follow iterative development methodology by choosing to only increase complexity upon validation of core game theory and technology assumptions. For instance, in the DAO MVP, only wETH is allowed as a tribute. While doing it this way does not let us validate the entirety of the game theory proposed above, building the MVP this way allows the Moloch DAO to be immediately useful to the ecosystem.

Lastly, we observed that there are complexity tradeoffs to building in upgradeability in contracts. We experimented with a library-based architecture, but found that the additional friction of use and increased attack surface were not valuable enough to focus on over validating the core use case. As such, we approached upgrading from the simplest standpoint: participants can coordinate offchain when an upgrade needs to happen by deploying a new DAO smart contract and exit Moloch if they choose to participate in the upgrade.

Moloch Shares

The Moloch DAO uses Shares to satisfy game theoretic conditions. Shares are requested on a per-proposal basis and are expected to be issued proportional to the applicant's tribute amount as compared to the total pooled funds. Shares confer the right to vote on proposals for how the resources in the Guild Bank should be allocated. They can also be *irreversably* redeemed via Ragequit for a portion of the Guild Bank's held funds at any time proportional to ownership.

$$redeemable\ funds\ = \frac{Your\ Shares}{Total\ outstanding\ Shares} \times total\ value\ of\ Guild\ Bank$$

For example, if the Guild Bank holds 100 wETH with Shares distributed among 10 members and each member has 10 Shares, then the value of each share is 1 wETH and each member owns 10% of the GB. If a single member leaves and liquidates their 10 Shares, the Guild Bank total value drops to 90 wETH. However, each member still holds 10 Shares and so still owns 10 wETH. Their relative ownership of the GB (and thus their voting power) changes from 10% → 11.11%.

Shares are *not* transferable. This means they are incompatible with the ERC20 standard. We do this to ensure that votes cannot be bought and sold on the open market, and so that if members are bribed to submit proposal or votes it can still be more easily attributed to them.

Joining Moloch

Joining the guild requires submitting a membership proposal along with a tribute in wETH. A membership proposal requests a certain number of Shares in return for the tribute. Existing members of the guild vote on the proposal, including whether to grant the requested Shares. If the membership proposal is accepted, the tribute tokens are deposited into the Guild Bank and new Shares are minted and issued to the applicant. If the membership proposal is rejected, the tribute tokens are returned.

In order to reduce spam, membership proposals may only be submitted by existing guild members. This means that applicants must convince an existing member to champion their proposal when they apply. Additionally, membership proposals require a 10 ETH deposit, 9.9 ETH of which is returned after the proposal is processed, regardless of outcome. The remaining 0.1 ETH is reserved to incentivize processing the proposal once it is ready.

Aborting a Membership Proposal

A vulnerability discussed in this construction is the possibility of a member submitting a proposal with an applicant's tribute and maliciously requesting fewer Shares than what the applicant was expecting. If the proposal passed, this would mean that the guild purchased the applicant's wETH at a *steep* discount.

To counteract this, new applicants are able to abort their membership proposal during an Abort Period that starts immediately when the proposal is submitted. . Aborting during this time will prevent submitting any future votes on the proposal and will cause the vote to fail regardless of the votes already cast. Aborting will automatically return all tribute to the applicant just as if their proposal had been rejected. The 10 wETH proposal deposit, however, will still only be returned to the member who submitted the proposal once it is processed. Since it was their fault for submitting the proposal with the incorrect number of shares requested, they should not stand to benefit (by having their deposit returned early) from the applicant aborting the proposal.

Note that, by default, the Moloch DAO will have an Abort Period that lasts for 1 day into the Voting Period. This stops potential griefing vectors where applicants maliciously create proposals that they intend to abort in order to waste members' time. Members can vote with confidence on proposals that have been in the Voting Period for at least 1 day knowing they can no longer be aborted.

Submitting Grant Proposals

Submitting a grant funding proposal utilizes the same underlying contract mechanisms as those used by membership proposals. As above, the proposal may only be submitted by a member of the guild. In this case, the proposal would contain some work to be delivered in exchange for a requested number of Shares. For recordkeeping, the proposal details can be hashed and stored on IPFS.

Members then vote on the proposal. If accepted, the Guild Bank mints new Shares and issues them to the grant funding applicant.

Voting on Proposals

Proposals are voted on in the order that they are submitted and can be parallelized subject to certain limitations. For the MVP, the voting period of each proposal is 7 days. Up to 5 proposals may be submitted per day and so there are a maximum of 35 proposals being voted on at any given time (each staggered by 4.8 hours).

Votes are won by simple majority with no quorum requirement. Unlike other voting systems where members are immediately locked into the outcome of a vote, because Moloch provides a grace period during which members who voted No can exit, we felt it was still safe not to require quorum.

Members are only able to vote once on proposals. Votes are counted and finalized at the end of the voting period but BEFORE the start of the grace period or the subsequent issuance of new Shares. This means that an application for new Shares issued will not be added to Yes or No votes on active proposals. Additionally, members who Ragequit on one proposal will not be removing their votes from the vote on a different active proposal. Because Ragequitting members who voted No do not have their votes deducted and members who voted Yes can't ragequit, the vote tally for a proposal is final as soon as its Voting Period is complete.

Grace Period

After votes are finalized, the Grace Period begins. As mentioned in "Ragequitting", the Grace Period lasts 7 days and allows members to exit the guild should they strongly disagree with the outcome of a vote. Members can only freely exit if they vote No on a proposal. Members who vote Yes on a proposal that passes will be forced to bear the cost of dilution from that proposal.

At the end of the Grace Period, the proposal is processed by calling the processProposal function. A reward of 0.1 ETH is deducted from the 10 ETH proposal deposit and sent to the member that called the function to incentivize timely processing.

Because proposals are parallelized, there are some edge cases around Ragequitting and the Grace Period. First, Ragequitting a proposal will mean that all active (i.e. being voted on) proposals and all other proposals currently in the Grace Period will not affect the member. This could result in a bad outcome for the member if they were simultaneously issued Shares in another proposal being processed at the same time (thus disincentivizing them from Ragequitting). It could also be the case that a new applicant, after joining and being issued shares, immediately suffers dilution because of a malicious proposal queued directly after theirs.

These types of cases are handled by the staggered queue. In the absolute worst case, a member will always have a maximum of 4.8 hours where all previous proposals have completed being processed and they can still ragequit.

Dilution Bound

As a safety mechanism, we have built a dilution bound to stop a large set of colluding actors from forcing a minority of guild members to experience massive dilution in a single hit by all

Ragequitting at the same time. We specify a Dilution Bound field in the contract (default = 3) which bounds the maximum dilution that a member can suffer.

For instance, if 80% of the voting power of the guild were to Ragequit all at once, the remaining members would suffer a 5x dilution. When the proposal is being processed, the Dilution Bound would be triggered and the proposal would fail, i.e. no new Shares would be issued. Since Ragequitters would have taken their proportional holdings of ether, the total Guild Bank balance would be reduced but the remaining members would have exactly the same ether as before the proposal was processed. If the remaining members then wanted to, they could re-submit the proposal.

Initializing the Guild

Since the process of adding new members to the guild and minting new Voting Shares requires a vote of all guild participants, adding the initial set of participants to the guild will require a different process.

For the MVP, the Moloch DAO contracts will be deployed with one “summoner” member address in the constructor. Then, this member, utilizing their one vote, will manually add a set of initial founders to the guild by issuing Shares for a fixed entry tribute.

References:

[1] Moloch Wikipedia Page
[2] Meditations on Moloch - Scott Alexander
[3] Howl - Allen Ginsberg
[4] Logic of Collective Action - Mancur Olson
[5] The State of Ethereum 2.0 - Kyokan
[6] Funding the Evolution of Blockchains - Fred Ehrsam
[7] The Tragedy of the Commons - Garrett Hardin
[8] The Prisoner's Dilemma
[9] Governing the Commons - Elinor Ostrom
[10] Iterated Prisoner's DIlemma Strategies - Juriˇsi ´c et al.
[11] Accelerating Evolution Through Forking - Fred Ehrsam