The Meta-DAO Manifesto

Brahma <0xBrahma@tutamail.com>

September 12, 2022

Abstract

In 2020, Robert Leshner introduced the Compound Finance governance system. Since then, it has gained widespread adoption. The problem is that it only works for very simple projects. To organize larger ones, you need a Meta-DAO: a DAO that is itself broken into smaller DAOs.

In this paper, we introduce a Meta-DAO implementation. This implementation is built upon a new primitive called conditional tokens. DAOs can use conditional tokens to estimate how an improvement proposal passing would impact a token's market capitalization. In our implementation, DAOs in the Meta-DAO execute improvement proposals when they would increase the Meta-DAO's market capitalization. They reject proposals when they would decrease it. The end-result is a system where decisions are made by a combination of markets and code.

1 Introduction

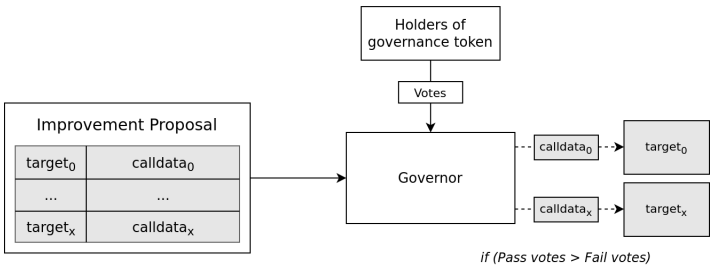

Figure 1: The Compound Finance governance system

Much of decentralized finance today is managed by the Compound Finance governance system or another system that was inspired by it. The centerpiece of the system is the *governor*, the smart contract that manages decisions. Accounts get the governor to take actions by introducing improvement proposals. Improvement proposals contain a list of targets, smart contracts to call, and

calldatas, low-level machine instructions to those smart contracts.[2] For example, a proposal to pay a security auditor 200k in USDC would contain USDC as the target and a 200k transfer instruction as the calldata. After a proposal is raised, holders of the COMP token can use their tokens to vote *Pass* or *Fail*. If a proposal garners enough *Pass* votes, the governor executes it. It does this through calling each target with its corresponding calldata.

This system works well for simple DAOs. When there's only one product that the DAO manages, a small number of contributors to the DAO, and there exist a few token-holders with large stakes (so-called whales), it works.

But not all DAOs are like this. Some DAOs, like MakerDAO, manage complex products with many integrations. Others, like Gnosis, Abracadabra, and Sushi, market multiple products under a single brand. It's hard for every decision these DAOs make to get pushed through as an improvement proposal. In addition, many DAOs have very dispersed token-ownership, so that the incentive for each holder to do research on proposals is small.[3]

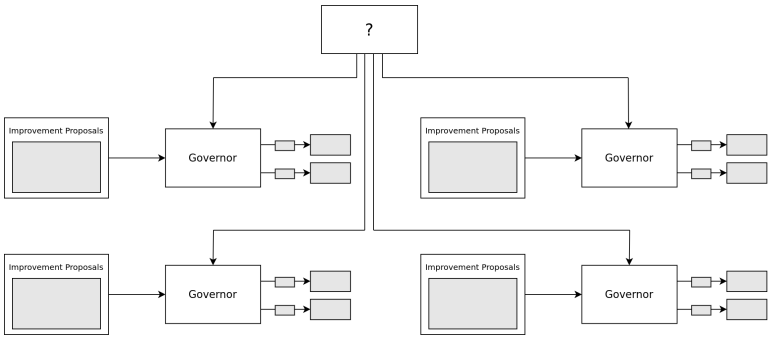

Figure 2: A Meta-DAO

The natural solution to this problem is to build a Meta-DAO, a single DAO that is itself broken up into multiple DAOs. Each DAO would focus on a smaller piece of the puzzle, with some coordination mechanism to coordinate their activities. This would prevent any individual DAO from being overloaded with decisions.

Indeed, many communities have raised the need to reorganize themselves like so.[4, 5, 6] But so far, the mechanism for decision-making in these DAOs is unclear. In this paper, we propose an implementation driven by code and markets.

2 Conditional tokens

At the core of our system is a primitive called conditional tokens. Conditional tokens give their owners the right to claim underlying tokens in certain cases.

Users mint and burn conditional tokens, also called *cndTokens*, by interacting with the conditional vault smart contract.

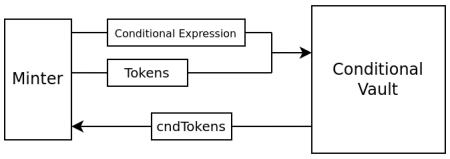

Figure 3: Minting cndTokens

To mint cndTokens, a user must deposit some tokens and specify a conditional expression. Conditional expressions are programmable expressions that can be evaluated by the conditional vault. For example, a user could pass the following expression:

```
block(150).transaction_count > 1000
```
[1]

When block 150 is mined, it will contain either more than 1,000 transactions or 1,000 or fewer transactions. In the former case, the expression will evalute to true; in the latter, it will evaluate to false. Before block 150 happens, the conditional vault can't evalute the expression.

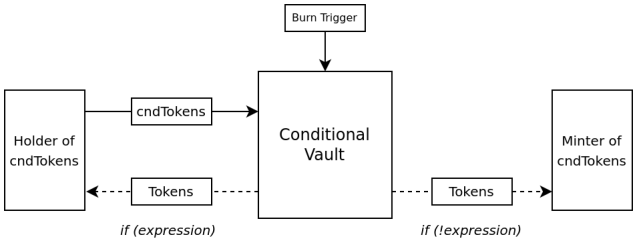

Figure 4: Burning cndTokens

Once the conditional vault can evaluate an expression, anyone can trigger the burning of cndTokens tied to that expression. If that expression evalutes to true, the conditional vault sends the deposited tokens to whoever held those cndTokens. If it evalutes to false, the vault sends the tokens to whoever minted the cndTokens.

[1] This, like the other conditional expressions, is pseudocode. It's helpful to think of the conditional vault as an abstract machine that evaluates arbitrary conditional expressions, but an implementation of the conditional vault doesn't need to allow such generality. Instead, it can allow the user to select and configure between a few pre-defined alternatives. For example, one could build a conditional vault that only accepts conditional expressions like this one, where the function signature to mint cndTokens looks like *mint(block_number, required_transaction_count)*.

3 Conditional markets

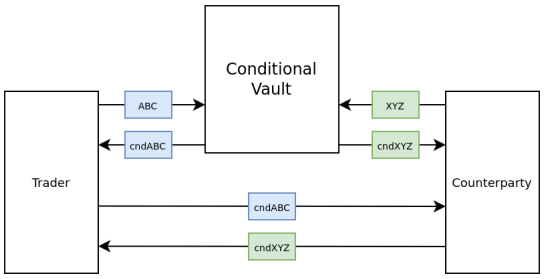

Figure 5: A conditional swap

With conditional tokens, traders can execute conditional swaps, swaps that get finalized only if something happens. These are performed through the following steps:

1. A trader deposits the token they want to sell into the conditional vault, and supplies a conditional expression that only evalutes to true if the desired event happens. If they want to swap ETH into SOL, but only if Solana can achieve 20,000 transactions per second of usage by block 500,000, they would mint cndETH using the conditional expression "block(500_000).transaction_count $>$ 20_000."[2]

2. Someone else wishing to take the other side of the swap converts their tokens into cndTokens, using the same conditional expression. In this example, they would convert their SOL into cndSOL.

3. Both sides swap their cndTokens. Here, the trader would receive cndSOL and their trading partner would receive cndETH.

If the condition evalutes to true, both sides can redeem their conditional tokens for normal tokens. Otherwise, they will get back what they originally put in. In our example, the trader will only receive SOL if block 500,000 contains more than 20,000 transactions in it. Otherwise, they will receive their original ETH.

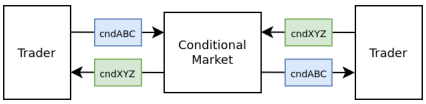

Figure 6: A conditional market

[2] Assume that manipulating this number is prohibitively expensive.

Instead of trading directly with each other, most traders will likely prefer to trade through markets. Conditional markets allow this. As long as traders swap cndTokens for other cndTokens that share the same conditional expression, conditional swaps work the same.

In allowing traders to do conditional swaps, conditional markets open the door for price discovery around alternate realities. If there was a cndETH to cndSOL pair like the one in our example, this pair would not have the same exchange rate as a normal ETH to SOL pair. Because those conditional swaps are guaranteed to be reverted unless Solana can get a large amount of activity, this type of cndSOL is likely to trade at a premium relative to normal SOL.

In general, cndTokens trade at different prices than their normal counterparts, with their price dependendant on how much those normal tokens would be worth if the event encoded in the conditional expression came true.

4 Estimating impact of proposals

Once conditional expressions can be made dependent on a given improvement proposal passing or failing, conditional markets can be used to estimate the impact of a proposal on a token's market capitalization.

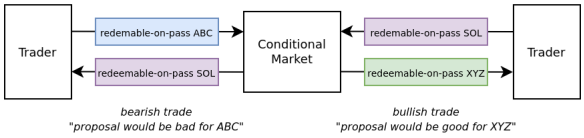

Figure 7: A conditional-on-proposal market

These conditional expressions have the following form, where x is an improvement proposal number and y is either *Pass* or *Fail*:

```
require(proposal(x).state != Pending) && proposal(x).state == y
```

We call cndTokens where y is *Pass* redeemable-on-pass tokens and cndTokens where y is *Fail* redeemable-on-fail tokens.

Importantly, any token can be converted into a redeemable-on-pass or redeemable-on-fail token that is dependent on an improvement proposal, irrespective of where the improvement proposal was raised. Figure 7 could depict trading around an ABC proposal, an XYZ proposal, or a proposal to a different DAO.

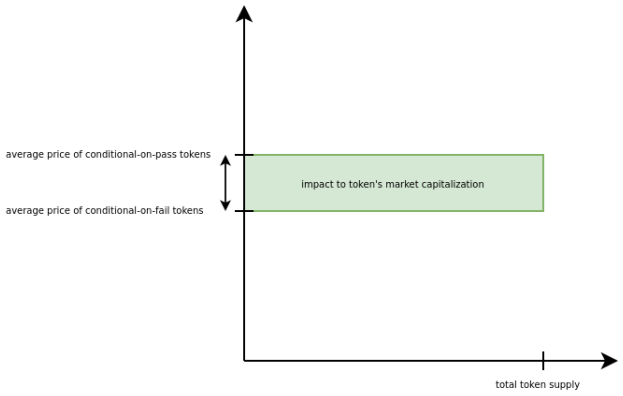

Figure 8: Calculating impact to a token's market capitalization

In an efficient market[3], investors will buy and sell conditional-on-pass and conditional-on-fail tokens until their prices reflect how they expect that proposal to impact the token's market capitalization.[7, 8, 9] For example, if a proposal to DAO ABC would burn $5M of the DAO's stablecoins for no reason, we would expect the market capitalization of ABC based on the price of its conditional-on-pass tokens to be $5M less than the market capitalization of ABC based on the price of its conditional-on-fail tokens.

We can also work in the reverse order, backing into the market's expectation of a proposal's impact by looking at the prices of its conditional-on-pass and conditional-on-fail tokens. All that's required is getting the difference in prices between the two conditional tokens and multiplying this by the token's total supply.

[3] Although the market for conditional tokens is unlikely to be a perfect one, the less efficient it is the stronger the incentive for users to step in and correct any mispricings.

5 Making decisions

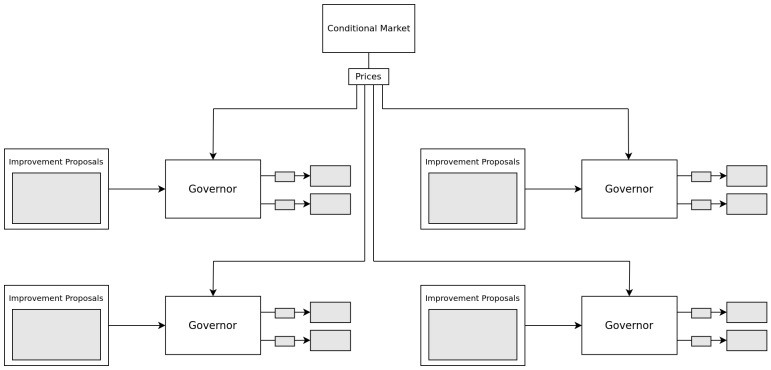

Figure 9: How decisions are made

We call the DAOs which make up Meta-DAO *members,* or *member DAOs.* To make decisions on improvement proposals, members perform the following algorithm:

1. After an improvement proposal is raised, wait 10 days, during which the market can bet on its impacts by buying and selling this proposal's redeemable-on-pass and redeemable-on-fail tokens. This doesn't just concern this member's tokens; users can convert any member's tokens into cndTokens tied to this improvement proposal.

2. For each member DAO that had cndTokens tied to this improvement proposal, calculate the impact that the proposal would have on the member's market capitalization. [4]

3. Add together all of the impacts calculated from the prior step. When the sum is positive, execute the proposal.

There is no special weight applied to any of the impacts. In other words, members put the interests of all members, including their own, on equal footing.

[4] To prevent manipulation of this with proposals that burn the holdings of random token-holders, members don't have the ability to burn any member tokens other than the tokens stored in the member itself.

6 Funding the commons

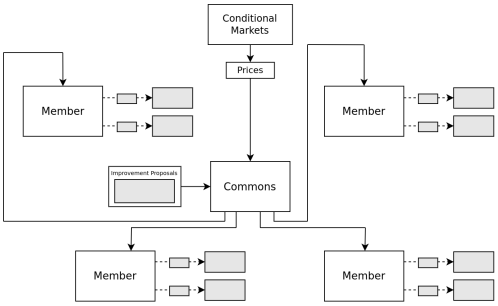

Figure 10: Commons DAO improvement proposals

The last mechnism needed is some way for improvement proposals to atomically multicast to multiple members at once, where either all the members take an action or none of them do. This is needed for funding of common goods proposals, where either the action should go through, and all relevant members should chip in by sending money to a collective pot, or it should fail, and no members should send money to the collective pot.

This is accomplished through a special-purpose DAO called the commons DAO. The commons DAO has no token of its own, and exists only to serve the interests of the members. All members execute any machine instructions that the commons DAO gives them. The commons DAO uses the same decision-making algorithm as the members, taking decisions when they appear to be in the benefit of the group.

The commons DAO also stores the list of active members. For a new member to join a Meta-DAO, an improvement proposal containing a machine instruction to push a new member to the list needs to pass through the commons.

7 Incentive analysis

In this section, we explore types of decisions and their incentives.

7.1 Proposals without externalities

Many member DAO decisions only affect that member. Examples of proposals in this category include:

- Paying out regular contributors at a previously agreed-upon rate
- Tweaking pricing of a dApp
- Allocating capital to do a re-design of a website

In these cases, investors are incentivized to buy or sell conditional tokens of that member but not tokens of other members. If other members' tokens are converted into a conditional variant that is dependent on this proposal passing or failing, the incentive is for those tokens to trade near price parity.

At price parity, a DAO calculates the impact of a proposal on a DAO as 0, and it doesn't affect the decision. As such, decisions without externalities are likely to be determined by the impact that a proposal would have on the DAO where the proposal was raised.

7.2 Proposals with externalities

Other proposals affect multiple members, even if they only involve an action by one. For example, members may share a common brand, and one member may be considering a product that could generate incremental revenue at the cost of overall brand image.

Here, investors are incentivized to sell those negatively-affected members' redemable-on-pass tokens and buy their redeemable-on-fail tokens. This doesn't prevent the proposal from proceeding; the proposal will pass if the losses to those other members are smaller than the gains by the member where the proposal was raised. This can also work in reverse: proposals can pass even if they have a negative impact on the member where they're raised if the positive impact to other members is greater than that negative impact. In other words, decisions are not always win-win, but in an efficient market they are always positive-sum.

7.3 Commons decisions

Proposals raised to the commons DAO create similar incentives to proposals raised to a member that have externalities. These proposals pass or fail based on whether they have a positive-sum impact on the broader group of members. Since the commons DAO has no token, its independent interests, if any, aren't factored into the decision-making process.

8 Conclusion

We have introduced a mechanism for decision-making in Meta-DAOs. Instead of human decision-makers, it relies on open and permissionless markets. Meta-DAO is not a process for humans to follow, but is instead a combination of programs and state, intended to be stored on the Solana blockchain.

References

[1] Robert Leshner, "Compound Governance," https://medium.com/compound-finance/compound-governance-5531f524cf68, February 2020.

[2] Various contributors, "Compound Protocol," https://github.com/compound-finance/compound-protocol/tree/86c0116833596dcae7bec15eac2ad5a14da1806d, March 2020.

[3] Mancur Olson Jr., "The logic of collective action: public goods and the theory of groups," 1965.

[4] @tracheopteryx and @lex_node, "YIP-61: Governance 2.0," https://gov.yearn.finance/t/yip-61-governance-2-0/10460, April 2021.

[5] pet3rpan, "KPI Sub-DAO Structure," https://pet3rpan.mirror.xyz/5nsl1Xnr-BwmNAPoNeMlghPiTFMhxNg4zE7suaR0sXc, January 2022.

[6] Rune Christensen, "The Endgame Plan parts 1&2," https://forum.makerdao.com/t/the-endgame-plan-parts-1-2/15456, May 2022.

[7] Robin Hanson, "Shall we vote on values, but bet on beliefs?" In *Journal of Political Philosophy,* 2013.

[8] Paul A. Samuelson and William D. Hordhaus, "Economics," 1948.

[9] Aswath Damodaran, "Investment valuation: tools and techniques for determining the value of any asset," 1995.