

# Stake DAO Votemarket

## *A safe and efficient way to optimize liquidity incentivisation*

Since the launch of the Stake DAO Liquid Lockers, one of the main challenges we have encountered while trying to maximize the profits of sdTKNs was how to extract the highest value possible out of the underlying voting rights. This was particularly true for the CRV locker, for which there are many different bribing solutions, each one of them having their strengths and weaknesses. After a few months of operation and experience, we were not fully satisfied with the existing solutions. Most were off-chain, with huge centralisation and human error risk factors, which often materialized (merkle tree mistakes, distribution delays, etc.). Some involved long risk exposure to the bribed token price, creating some negative game theory pushing users to dump aggressively, others were lacking visibility, etc. We therefore decided to start thinking about our own solution.

## A quick history of the vote incentives market

Bribe.crv.finance was the pioneer in this space, but it is difficult to use and not adapted for retail users as voting on-chain is expensive and you need to claim every week. Furthermore, on bribe.crv.finance, it is impossible to bribe both veCRV and vICVX at the same time, since there would be some bribe overlap, making those bribes very inefficient.

Votium solved this problem by introducing an off-chain solution, and was quickly followed by many other products, each of them trying to bring additional value to users (Pitch, Hidden Hand, Quest...) However, new issues came with those evolutions, deriving notably from their off-chain nature.

Although we are getting closer to perfection with these innovations, we still believe there is value in having a fully on-chain solution that would solve as many of the issues mentioned below as possible.

## Problem identification

To design the product we need for our bribes, we first need to identify what we like and what we lack in the existing solutions. This includes what we like and lack as a voter (indeed, Stake DAO, via the lockers, has to vote and claim bribes for several protocols across different platforms), but also draws from our experiences as a briber, since we tested different protocols when issuing bribes for our sdCRV/CRV and sdBAL/BAL-LP pools.

The key pain points we have suffered from **as a voter** were the following:

1. The distribution process is **not always on-chain**, leading to some human error risks (that materializes several times) and trust issues. It also leads to a lack of composability since it is

impossible to have a fully on-chain and automated system, which would enable us to fully decentralize the lockers.

2. For some platforms, the **distribution arrives two weeks after the vote**, leading to a longer full process and mainly to a longer price exposure to the bribe token, which can impact positively or negatively our bribes in dollar terms.
3. Sometimes bribes need to be claimed every week otherwise they are lost.
4. Difficult accounting of bribes for off-chain solutions since claims for different gauges are bundled into one.

The key pain points we have suffered from **as a briber** were the following:

1. **Uncertainty of bribe efficiency**: we sometimes ended up paying a higher price than the amount of emissions that were directed to our pools following. It would therefore have been more efficient to incentivize the pool directly rather than bribing.
2. **Unjustified fees** taken out of an unsuccessful bribe campaign (when your bribe is not fully distributed on Quest, you still pay a 4% fee on the non-distributed part).
3. **Off-chain** systems leading to a lack of composability and automation.
4. **Potential overlap** when voting and bribing at the same time: we were paying fees on bribes going to ourselves when we were bribing sdCRV/CRV or sdBAL/BAL and also voting with the DAO multisig's sdCRV and sdBAL.
5. **Weekly management needed**: except on Quest, you need to allocate a significant amount of time every week to manage and design your bribe program.

## Overview of today's bribe market

These problems were identified by various bribing platforms, with various attempts to provide solutions, but always with trade offs.

Now we will go through the exercise of identifying those trade offs to make our decision and design the bribing product we need. For this, we will start with a quick overview of the existing solutions present in the market:

	yBribe	Votium	Hidden Hand	Pitch	Quest
<b>Brief description of the solution</b>	On-chain contract that makes claimable the balance of the bribe to people who were voting on Curve on last Snapshot date (every Thursday at 12am)	Started with vICVX bribing and expanded to veCRV bribing, but kept the off-chain approach of creating a biweekly merkle airdrop to reward users who voted for a bribed gauge during the past two weeks. Automatically blacklists Convex from merkle to avoid vote overlap.	Fork of Votium with no blacklist, but available on many different protocols, including Tokemak which has no veTOKEN	Off-chain solution which allows Protocols to set a maximum price per vote.	Off-chain contract which allows for fixed bribe price and multi-week programs by using merkle airdrops. It also allows blacklisting addresses in an off-chain manner
<b>Strengths and weaknesses from a user (voter) point of view</b>					
<b>Strength</b>	<ul style="list-style-type: none"> <li>- Fully on-chain</li> <li>- Bribes distributed immediately after the vote</li> </ul>	<ul style="list-style-type: none"> <li>- No need to claim bribes every week</li> <li>- Bundle claims of different gauges with same token</li> </ul>	<ul style="list-style-type: none"> <li>- No need to claim bribes every week</li> <li>- Bundle claims of bribes from different gauges/protocols</li> <li>- Bribes distributed immediately after the vote</li> </ul>	<ul style="list-style-type: none"> <li>- No need to claim bribes every week</li> <li>- Bundle claims of different gauges</li> <li>- Bribes distributed immediately after the vote</li> </ul>	<ul style="list-style-type: none"> <li>- Multi-week visibility on bribe sustainability</li> <li>- Bundle claims of different gauges</li> <li>- No need to claim every week</li> <li>- Relatively good visibility on bribe level</li> <li>- Bribes distributed immediately after the vote</li> </ul>
<b>Weaknesses</b>	<ul style="list-style-type: none"> <li>- User needs to claim every week</li> <li>- Lack of visibility on bribes sustainability</li> <li>- Unfriendly UI</li> </ul>	<ul style="list-style-type: none"> <li>- Off chain</li> <li>- Bribe distributed 2 weeks after the vote, leading to longer exposure to bribe token price</li> <li>- 2% fee paid by voters</li> </ul>	<ul style="list-style-type: none"> <li>- Off chain</li> <li>- Lack of visibility on bribes sustainability</li> </ul>	<ul style="list-style-type: none"> <li>- Off chain</li> <li>- Lack of visibility on bribes sustainability</li> <li>- Lack of visibility on bribes level</li> </ul>	<ul style="list-style-type: none"> <li>- Off chain</li> <li>- Lack of clarity on eligibility to Quest</li> <li>- Designed for long term commitment with a First-come-first-served mechanism</li> </ul>
<b>Strengths and weaknesses from a protocole (briber) point of view</b>					
<b>Strength</b>	<ul style="list-style-type: none"> <li>- Fully on-chain</li> <li>- Unclaimed bribed are rolled over automatically</li> <li>- Any token can be accepted</li> </ul>	<ul style="list-style-type: none"> <li>- Possibility to bribe both veCRV and vICVX without bribe overlap</li> </ul>	<ul style="list-style-type: none"> <li>- Many protocols accepted</li> </ul>	<ul style="list-style-type: none"> <li>- Possibility to set up a maximum price per bribe</li> <li>- Fees paid only on the claimed part</li> </ul>	<ul style="list-style-type: none"> <li>- Possibility to create multi-week programs</li> <li>- Total visibility on bribe level</li> </ul>
<b>Weaknesses</b>	<ul style="list-style-type: none"> <li>- Lack of visibility on bribes efficiency</li> <li>- Not possible to blacklist addresses to avoid bribe overlap or self-bribing</li> <li>- 1% fee</li> </ul>	<ul style="list-style-type: none"> <li>- Off chain</li> <li>- Bribe distributed 2 weeks after the vote, leading to big dumping events every other week</li> <li>- 2% fee</li> </ul>	<ul style="list-style-type: none"> <li>- Off chain</li> <li>- Lack of visibility on bribes efficiency</li> <li>- 4% fee</li> <li>- Not possible to blacklist addresses to avoid bribe overlap or self-bribing</li> </ul>	<ul style="list-style-type: none"> <li>- Off chain</li> <li>- 4% fee</li> <li>- Not possible to blacklist addresses to avoid bribe overlap or self-bribing</li> </ul>	<ul style="list-style-type: none"> <li>- Off chain</li> <li>- Fee paid on total amount deposited regardless of the bribe efficiency</li> <li>- 4% fee</li> <li>- If the bribe market goes down, the briber will overpay its bribes. If it goes up, the bribe won't get filled.</li> </ul>

If we sum up the information available in this table, we can identify a few points which are key to the user experience of a bribing platform.

For voters, the key points to consider are:

- On-chain v/s off-chain.
- Visibility of bribe level and sustainability: “how much will I get in total if I vote for this gauge?” It takes into account the bribe level itself and whether this bribe will last for two weeks, or maybe only one.
- Fees.
- Management flexibility: possibility for users to claim whenever they want, to have a minimum of gas heavy transactions, to vote easily, etc.
- Price exposure: this relates to how long the voter is exposed to the price of the token used for the bribe. The shorter the exposure, the easier it is to select the highest bribe.

For bribers, the key features to consider are:

- On-chain v/s off-chain.
- Visibility of bribe efficiency: this relates to what happens to the funds invested in the bribe if not enough votes go to the bribe gauge.
- Fees.
- Management flexibility: how easy it is to maintain a bribe program when decisions are made by governance and/or transactions are done by a multisig, or directly by a contract.
- Possibility to avoid bribe overlap: this relates to the possibility to blacklist addresses that do not need to be bribed (e.g. addresses bribed in another way, or the briber’s address itself if they vote).

## Technical solution

For the needs of Stake DAO’s Liquid Lockers, an on-chain solution is seen as the best long term solution. This comes with a few challenges, namely the gas optimisation and trust-less aspect.

Then, when we look at the other aspects we want to prioritize, we believe that visibility is the most important criteria for both the voter and the briber. We also believe that the possibility to blacklist addresses is going to be key to adapt to everyone’s situation, as the protocols who bribe also generally have their own voting power.

Now, as mentioned above, there is no perfect solution: you always have trade offs. The reason why most protocols went for an off-chain solution is that so far no one has found a gas efficient solution to allow users to enable bundled claims of several weeks and bribes in a single transaction.

We came to the same conclusion: to minimize gas costs, it is more efficient to have a very simple bribe contract that will unfortunately require users to claim every week. This is particularly true for bribes which will require a blacklist.

Stake DAO Votemarket will work as follows.

## 1-Bribe Creation

In a trust-less fashion, bribe managers create multi-week bribe programs (minimum two weeks), by **depositing** a certain amount of tokens (any ERC20) into the contract, **setting the program length** (number of weeks), and **setting a maximum price** per bribe. This will enable the bribe manager to design long term programs which will follow the bribe market with limited recurring maintenance, and ensure no inefficient bribes are being distributed.

A good way to avoid inefficient bribing is by setting a maximum price per bribe equivalent to the amount of incentives that will be directed by this bribe.

Bribers will also have the possibility, but not the obligation, to blacklist one or several addresses, so that they can maximize the efficiency of the program, while still voting themselves or bribing wrappers (vICVX or vLAURA for example).

Finally, bribers have the possibility to make their bribe upgradeable, if they plan to keep depositing incentives in it and increasing the bribe duration. This can notably be useful in case of on-chain integration of the bribe inside a broader architecture.

## 2- Voting and claiming the bribes

Voters will simply vote via the underlying platform's gauge vote.

The bribe contract will check the user voting power for a given gauge (using the "bias" variable), and distribute the weekly claimable amount proportionately to each user's vote, taking into account blacklisted addresses and maximum price per bribe.

This can be summed up by the following equations:

Claimable bribe for one period:

$$B = \frac{\text{Total bribe} - \text{Total claimed}}{\text{Remaining weeks}}$$

User voting power:

$$Vu = \text{user\_slope} * (\text{end\_lock} - \text{currentPeriod})$$

User claimable bribe:

$$Bu = \text{Min} \left( \frac{B \times Vu}{(V - Vb)}, Vu \times M \right)$$

Where:

*Bu*: Bribe claimable by user

*B*: Total claimable bribe for the period

*Vu*: User voting power

*V*: Total votes for the gauge (fetched from the gauge controller, using the bias returned by the *points\_weight* function)

*Vb*: Sum of voting powers of all blacklisted addresses

*M*: Maximum bribe price set by the contract

Voters need to claim every week. If they don't, the unclaimed amount will be added back to the bribe program and distributed over the remaining periods.

However, to limit the amount of transactions users will need to call to claim their bribe, a *claimAll* function has been implemented which will enable the user to claim all different bribes at once.

The 2% fee taken by the platform is charged during the claiming process, to make sure that bribers don't pay fees on bribes that will never reach the voter.

### 3- Update bribe

To avoid needing to recreate a new bribe at the end of the defined period, or to adjust the bribe to changing market conditions, the bribe manager can choose to make his bribe upgradeable. In this case, while the bribe is not finished, it can be upgraded with a one week delay. Any upgrade will be effective only from the next period. To make sure users are aware of the reduced visibility ahead of the upgrade, a message is displayed mentioning bribe conditions are changing during the next period, and how they are changing.

The only thing that can be changed are :

- Bribe duration
- Total bribe
- Maximum price per bribe.

All of them can only be adjusted upward.

### 4- Closing the bribe

At the end of the last period, voters still have one week to claim their reward. After that, the bribe manager can close the bribe program. While doing so, all unclaimed funds still remaining inside the bribe will be sent to the manager, and **no fees** will be charged on those funds.

### Practical example

Bob DAO is looking to incentivise providing liquidity to their liquidity pair BOB/ETH on Curve. They decide to go for a vote incentive rather than direct LP incentive since the bribe market is currently lower than the liquidity market (bribes are cheaper than incentives directed by votes).

Their governance votes for a one-month incentive program of 10,000 BOB tokens with a security to avoid paying bribes more than the incentives directed by the votes acquired.

On Stake DAO Votemarket, they create a bribe with the following parameters:

- Protocol to bribe: Curve
- Reward token: BOB
- Number of weeks: 4
- Max BOB reward per veCRV: 0.005 (equivalent of 0.007 CRV, the amount of liquidity directed by one veCRV vote over one week)
- Blacklisted address: 0xB0B...0 (their multisig address which owns some sdCRV and veCRV which also plans to vote for the BOB/ETH gauge but doesn't want to dilute the bribe for other voters)
- Total BOB rewards: 10,000

Peter and Mary see that there will be a minimum 2,500 BOB reward for the next 4 weeks and decide to vote for the BOB/ETH gauge on Curve. The Bob DAO multisig also votes:

- Peter: 400,000 veCRV
- Mary: 600,000 veCRV
- 0xB0B...0: 1 million veCRV

The total votes for the gauge is 2 million veCRV. The total claimable is  $10,000/4 = 2,500$  BOB. The bribe price is 0.0025 BOB/veCRV. As the multisig is blacklisted, only Peter and Mary's votes are considered. Peter can claim 980 BOB and Mary can claim 1,470 BOB (taking into account the 2% fee paid when rewards are claimed).

Peter claims, but Mary is distracted and forgets to claim. Her pre-fee reward is therefore rolled over and reused over the remaining periods.

Period number two therefore has 3,000 BOB rewards. The bribe price is 0.003 BOB/veCRV. This time, Mary doesn't forget to claim: she receives 1,764 BOB and Peter receives 1,176 BOB.

The vote incentive market goes up and Mary decides to change her vote to vote for another gauge. Now Peter is the only one still eligible to BOB vote incentives.

Since everything was claimed in period 2, the BOB reward for period 3 is still 3,000 BOB. The bribe price before cap is 0.0075 BOB/veCRV. This time, not enough votes were eligible, so the maximum price was reached. The total reward will therefore be  $0.005 \times 400,000 = 2,000$  BOB. Peter will therefore be able to claim 1,960 BOB and 40 BOB will go to Stake DAO treasury.

The total claimed rewards since the beginning of the program are  $1,000 + 3,000 + 2,000 = 6,000$  BOB. Therefore, there are still 4,000 BOB rewards to be distributed for the last period. Again, Peter is the only one eligible, and the bribe price reaches its ceiling. Pierre can again claim 1,960 BOB, and 2,000 will remain unclaimed.

After the end of the last claiming period, Bob DAO will close the bribe, receive their 2,000 unused BOB back, and be able to issue a new bribe program. They will have achieved a total of 2.8m votes over 4 weeks, directing a total of 19,600 CRV incentives to the BOB/ETH gauge, with just 8,000 BOB spent. With a BOB/CRV ratio of 1.4, this means they have achieved a 1.75x better outcome than what they would have achieved by directly incentivising the BOB/ETH pool with BOB.

## Conclusion

The solution proposed by Stake DAO Votemarket aims to maximize visibility for users and bribe managers, while providing the security that only on-chain architectures can offer, and keeping a decent management flexibility on both the user and briber fronts.

The objective of this tool is to develop a new standardized approach to liquidity incentivisation, where projects can incentivize their liquidity in a more efficient manner. The on-chain approach allows a scalability that off-chain based solutions needing recurring maintenance will not be able to follow without a growing risk.

Using bribes as the main tool for liquidity incentivisation will lead to reward token uniformisation and higher fluidity of the liquidity market. The current version of Stake DAO Votemarket is on Mainnet, but in the future, cross chain bribing solutions will be a main catalyst of new chain liquidity bootstrapping.