Proposal: Accounting for votes ARB that locked in various projects

(for discussion)

Constitutional

Abstract

A large part of ARB tokens (more than 13,500,000) are locked in the various protocols, but users cannot participate in governance with these tokens.

cp0x

believes that it is necessary to take these tokens into account in the governance system if there is a technical possibility for this in the smart contracts of projects (for example, there is a function that allows you to understand the exact number of arb tokens for a certain account).

Motivation

Users who use ARB in various projects are the most active, but cannot be a part of the governance, since their tokens due to technical implementation are not on their account. Such users also often have interest in participating in the DAO governance

Rationale

This way we can increase:

- the number of votes

, since at the moment users participating with their tokens in various protocols and they are not able to participate in voting

- quality of votes

, as we will involve more experienced and active users in the voting process

Specifications

Quite a large part of ARB tokens (more than 13,500,000) are locked in the following protocols:

- Uniswap

. more than 500,000 ARB locked ([ink](ink))

- SILO

. more than 5,000,000 ARB locked ([ink](ink))

- GMX

. More than 4,000,000 ARB locked ([ink](ink))

- Camelot

. More than 3,000,000 ARB locked ([ink](ink))

- Sushi

. More than 1,000,000 ARB locked ([ink](ink))

etc. It's just a few examples (volumes are constantly changing)

Steps to Implement

1. Select protocols to be taken into account in the calculation

2. Analysis of the possibility of accounting for user tokens for the selected protocol

3. For each protocol, write code that will take into account not only the presence of ARB tokens in the account of

everyone who delegated to the Delegate, but also the presence of ARB tokens in the above-mentioned projects

4. After appropriate audit, apply the code so that the current getVote() function takes this proposal into account

Timeline

Estimated schedule:

1. Select protocols. 1 month from the end of voting on Snapshot

2. Analysis by smart contract developers: 2 days for every protocol (up to 10 protocols are expected)

3. Audit code: depends on results of analysis

4. Apply the code

Total duration is about 16 weeks (4 month)

Overall Cost

Can be organized a grant for this work, with a cost of $100/hour

If it is impossible for employees to participate, you can hire programmers, 3 people for 4,000 ARB per month.

Audit - 10,000 ARB, it is possible to use the grant among individuals, since many companies do not check security, but issue certificates without greatly improving security.

Total amount: 3p x 4.000 ARB x 3 month + 10.000 = 46.000 ARB