Although this scheme is meant to be used primarily in

[MetaCoin

](https://ethresear.ch/t/announcing-metacoin-the-governance-minimized-decentralized-stablecoin/6897) (pegged-coin

system) and

[Reflexer

](https://medium.com/@stefan__ionescu/stability-without-pegs-8c6a1cbc7fbd) (reflex-bond

system), it can be applied to any DeFi protocol.

General Mechanism

In order to have a resilient price feed and at the same time minimize governance's power over oracles, I propose the following on-chain oracle network medianizer:

1. A contract keeps track of whitelisted oracle networks it can call in order to request collateral prices for the pegged-coin/reflex-bond system. The contract is funded by part of the stability fees the system accrues. Each oracle net accepts specific tokens as payment so our contract also keeps track of the minimum amount and the type of tokens needed for each request

2. In order to push a new price feed in the system, a majority of the oracles need to be called beforehand. When calling an oracle, the contract first swaps some COIN (stability fees) with one of the oracle's accepted tokens. After an oracle is called, the contract tags the call as 'valid' or 'invalid'. If a call is invalid, the specific faulty oracle cannot be called again until all the other ones are called and the contract checks if there's a valid majority. A valid oracle call must not revert and it must retrieve a price that has been posted on-chain sometime in the last m

seconds. "Retrieve" can mean different things depending on each oracle type:

- For pull based oracles (we can get a result from them right away), our contract needs to pay a fee and directly fetch the price

- For push based oracles, our contract pays the fee, calls the oracle and needs to wait a specific period of time n

before calling the oracle again in order to get the requested price

1. For pull based oracles (we can get a result from them right away), our contract needs to pay a fee and directly fetch the price

2. For push based oracles, our contract pays the fee, calls the oracle and needs to wait a specific period of time n

before calling the oracle again in order to get the requested price

1. Each oracle result is saved in an array. After every whitelisted oracle is called and if the array has enough valid data points as to form a majority (e.g our contract received valid data from 3/5 oracles), the results are sorted and the contract picks the median. If there's no majority, the array is cleared and the contract needs to wait p

seconds before starting the entire process all over again

Bounded Governance

Ideally, governance would cede control over the medianizer after they whitelist all valid oracle networks and set their parameters. In practice though, oracle networks can be upgraded and projects that depend on them must adapt. Thus, governance can update the medianizer's parameters (but not

change the medianizer address the system depends on) only if they abide by the following rules:

1. When the medianizer is deployed, governance must specify a modification window and a delay window

2. Governance cannot change the modification and delay windows

3. The modification window is a period of time during which governance will be able to modify only one

oracle (they have to specify which one). The delay window forces governance to wait between two modification requests

1. Governance cannot add more oracles. They can only change the address of the oracle they want to modify and also its parameters (e.g payment token)

Apart from oracles, governance will also have bounded power over updating the code that swaps stability fees with oracle specific tokens. There will be multiple DEXs or DEX aggregators integrated with the medianizer and governance will be able to update one DEX integration at a time using the same modification and delay windows.

Examples of DEXs and aggregators used by the medianizer are Uniswap V2, 1split and Kyber Network.

Oracle Backup

Governance may add a backup oracle option that can start to push prices in the system if the medianizer cannot find a majority of valid oracle networks several times in a row.

The backup option must be set when the medianizer is deployed as it cannot be changed afterwards.

Oracle Failure Mitigation

In case of a bug in the medianizer code or wide scale collusion in most whitelisted oracle networks, there are two ways the system can be shut down:

1. A contract that detects whether the medianizer stopped pushing updates for x

days can automatically settle the system; the contract can also settle if the oracle backup has been replacing the oracle networks for too long

1. Governance can burn a certain amount of tokens to trigger settlement