

Introducing clr.fund, a trust-minimized quadratic matching fund

clr.fund (pronounced “clear fund”) has been in the works since EthDenver earlier this year, and we were advised to introduce it officially here.

clr.fund is a public goods funding mechanism that:

- is trust-minimized,
- incentivizes funding public goods through quadratically matching funds,
- has a decentralized, transparent process for curating qualified grant recipients, and
- aims to get most of its matching funds from many, small contributions.

Right now, we’re working towards a proof of concept (<https://github.com/clrfund>) and exploring some grant options to see if we can speed up the process.

Why build it?

Quadratic matching offers a clever way to incentivize public goods funding and to distribute the funds, but it’s vulnerable to collusion and sybil attacks. A long-term viable quadratic funding system should address these vulnerabilities.

A public goods funding system also shouldn’t be controlled by a small subset of the community it serves: it should be a public good itself.

A system more transparently controlled by its community will also have an easier time attracting lots of small donations, which is important. Whales probably don’t have strong enough incentives to fund matching pools by themselves long term.

How clr.fund works

It’s a quadratic matching system (see [Vitalik’s post](#)). A continuous series of funding rounds take place, wherein any qualified identity can donate to any number of qualified projects, and a matching pool of funds will supplement those donations, weighting projects that get a high number of donations more. It’s similar to Gitcoin grants, but with the key differences of less trust and more community control.

Contributions to the matching pool

We aim for clr.fund to be a viable recipient of residual fees from layer 2 protocols (e.g. DEXs, DAOs, etc.), block reward funding, and philanthropic donations.

Trust-minimization

We’re aiming for clr.fund to be fairly sybil- and collusion-resistant from the start, using BrightID for sybil-resistance and [MACI](#) for collusion-resistance.

BrightID works by creating a social graph of identities and letting projects using BrightID analyze that graph however they wish. For clr.fund, we’ll likely use BrightID similarly to [Burn Signal](#) in the short term.

A quick and dirty MACI primer: MACI works by letting users register an identity with a coordinator, but letting them invalidate their identity with a new one any time before the end of the round. At the end of the round, the coordinator tallies the votes and creates a zero-knowledge proof, showing that all contributions were counted properly but not revealing who contributed how much to which project. Someone attempting bribery will have difficulty confirming their bribes since they can’t tell what actions the people they bribed took.

Deciding on qualified projects

It isn’t yet clear how important a project curation mechanism is for clr.fund. Right now, MACI imposes a limit on the number of projects (i.e. voting options) that can exist in a single round (16, but 1024 has been tested successfully). If we regularly end up with more projects wanting in to funding rounds than there are slots, we’ll need a curation mechanism.

One suggestion is to use something like Burn Signal, which is sybil-resistant and uses quadratic voting but isn’t collusion-resistant. It’s also tied to the Ethereum community through the ETH token. The top X proposals on Burn Signal when the round starts might qualify. Burned ETH goes to the matching pool. If someone bribes a project past the curation mechanism, they’ll have trouble continuing the bribe during the actual funding round, which is protected by MACI.

This does leave an attack where every project in the round was bribed in, but for that, you might implement a “kill switch” in the round: MACI contributors can vote to kill the round instead of contributing to it. If a quorum is reached (50+%?), the round is killed, and the funds are saved for the next round. The briber will have wasted their money (actually contributing it to the pool!).

Our current plan & long-term possibilities

In the short term, we’re building towards a proof of concept where we can run a small clr.fund round as a test. We’ll hand select the projects and use very little funds. You can see our roadmap [here](#). We’re seeking a few grants to help us build this faster

After the POC, the next step will be an MVP round with a larger matching pool. This would likely involve using the curation mechanism for the first time. The goal here would be to show the system works and to hopefully raise some funds for clr.fund during the round, showing the path to self-sustainability for the project.

Overall, we see clr.fund as a candidate for long-term protocol-level public goods funding for Ethereum, as well as for other protocols / communities.

One long-term possibility involves multiple matching pools processed in each round—anyone could create a pool and be responsible for curating qualified projects in whatever way they wished. Contributions to a project could then be matched by multiple pools (e.g. an ETH 2.0 pool and a DeFi pool, or a decentralized governance pool and a legal compliance pool).

How to get involved

We are currently seeking soft-commitments for contributions to the matching pool for the first few rounds.

If you are interested in this experiment and funding Ethereum public goods, please reach out:

Telegram: <https://t.me/clrfund>

Github: <https://github.com/clrfund>

Blog: <https://blog.clr.fund/>

Twitter: <https://twitter.com/clrfund>