Gitcoin Grants Round 9: The Next Phase of Growth

2021 Apr 02

See all posts

Special thanks to the Gitcoin team for feedback and diagrams.

Special note: Any criticism in these review posts of actions taken by people or organizations, especially using terms like "collusion", "bribe" and "cabal", is only in the spirit of analysis and mechanism design, and should not be taken as (especially moral) criticism of the people and organizations themselves. You're all well-intentioned and wonderful people and I love you.

Gitcoin Grants Round 9 has just finished, and as usual the round has been a success. Along with 500,000 in matching funds, $1.38 million was donated by over 12,000 contributors to 812 different projects, making this the largest round so far. Not only old projects, but also new ones, received a large amount of funding, proving the mechanism's ability to avoid entrenchment and adapt to changing circumstances. The new East Asia-specific category in the latest two rounds has also been a success, helping to catapult multiple East Asian Ethereum projects to the forefront.

CLR MATCHING ROUND 9

East Asia Grants

GITCOIN

		NUMBER OF CONTRIBUTIONS	TOTAL CONTRIBUTED	CLR MATCHING
1	DAOSquare	1,525	$8,185	$19,005
2	CREAM	792	$5,195	$5,520
3	DAppChaser	730	$3,217	$4,161
4	Gas Now	545	$4,494	$3,032
5	EthGrounds InariDAO	624	$2,470	$2,774
6	ETHPlanet	599	$2,901	$2,644
7	reality.eth	622	$ 11,414	$2,581
8	Curve Swaps	654	$2,182	$2,459
9	Ethereum.cn	543	$3,030	$2,290
10	BaoBoard	457	$2,512	$1,767

CLR MATCHING ROUND 9

Infra Grants

GITCOIN

		NUMBER OF CONTRIBUTIONS	TOTAL CONTRIBUTED	CLR MATCHING
1	Ethereum Swarm	3,439	$31,889	$38,865
2	Gitcoin Grants	2,350	$27,328	$20,553
3	Umbra	2,235	$12,044	$11,101
4	Circles UBI	1,864	$11,352	$9,393
5	DAppNode	1,803	$15,191	$8,785
6	Prysm	1,463	$22,801	$7,321
7	Wallet Connect	1,451	$ 24,448	$6,869
8	beaconcha.in	1,337	$12,272	$5,802
9	ethers.js	1,269	$15,914	$5,650
10	Nethermind	1,464	$14,096	$5,316

However, with growing scale, round 9 has also brought out unique and unprecedented challenges. The most important among them is collusion and fraud: in round 9, over 15% of contributions were detected as being probably fraudulent. This was, of course, inevitable and expected from the start; I have actually been surprised at how long it has taken for people to start to make serious attempts to exploit the mechanism. The Gitcoin team has responded in force, and has published a blog post detailing their strategies for detecting and responding to adversarial behavior along with a general governance overview. **However, it is my opinion that to successfully limit adversarial behavior in the long run more serious reforms, with serious sacrifices, are going to be required**.

Many new, and bigger, funders

Gitcoin continues to be successful in attracting many matching funders this round. BadgerDAO, a project that describes itself as a "DAO dedicated to building products and infrastructure to bring Bitcoin to DeFi", has donated $300,000 to the matching pool - the largest single donation so far.

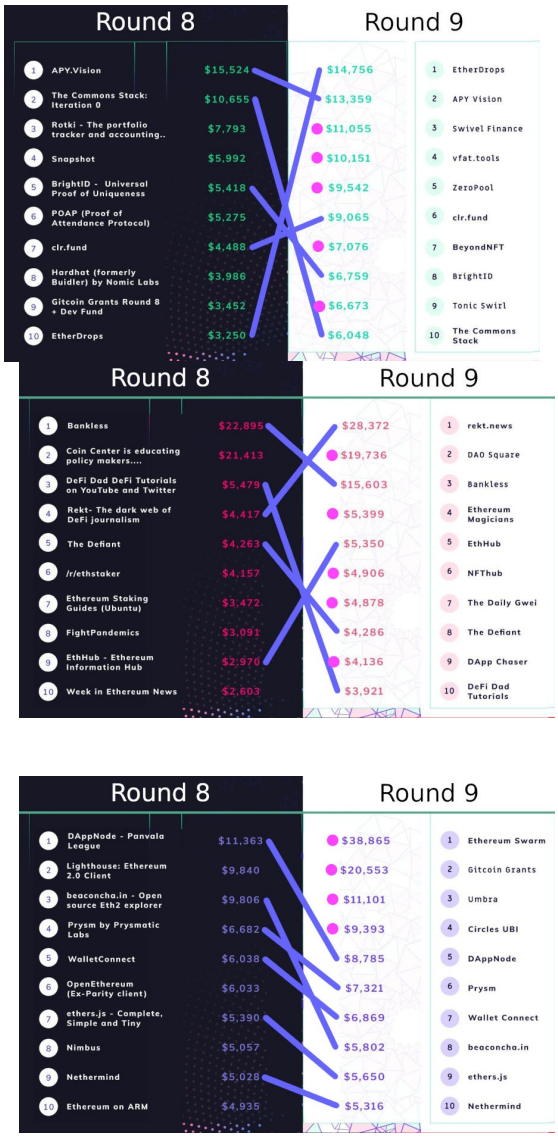
Other new funders include Uniswap, Stakefish, Maskbook, FireEyes, Polygon, SushiSwap and TheGraph. As Gitcoin Grants continues to establish itself as a successful home for Ethereum public goods funding, it is also continuing to attract legitimacy as a focal point for donations from projects

wishing to support the ecosystem. This is a sign of success, and hopefully it will continue and grow further. The next goal should be to get not just one-time contributions to the matching pool, but long-term commitments to repeated contributions (or even newly launched tokens donating a percentage of their holdings to the matching pool)!

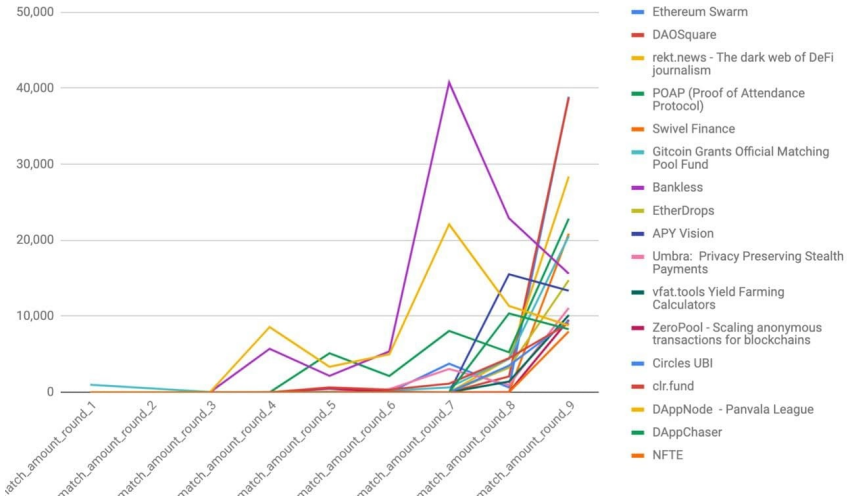
Churn continues to be healthy

One long-time concern with Gitcoin Grants is the balance between stability and entrenchment: if each project's match award changes too much from round to round, then it's hard for teams to rely on Gitcoin Grants for funding, and if the match awards change too little, it's hard for new projects to get included.

We can measure this! To start off, let's compare the top-10 projects in this round to the top-10 projects in the previous round.



In all cases, about half of the top-10 carries over from the previous round and about half is new (the flipside, of course is that half the top-10 drops out). The charts are a slight understatement: the Gitcoin Grants dev fund and POAP appear to have dropped out but actually merely changed categories, so something like 40% churn may be a more accurate number.

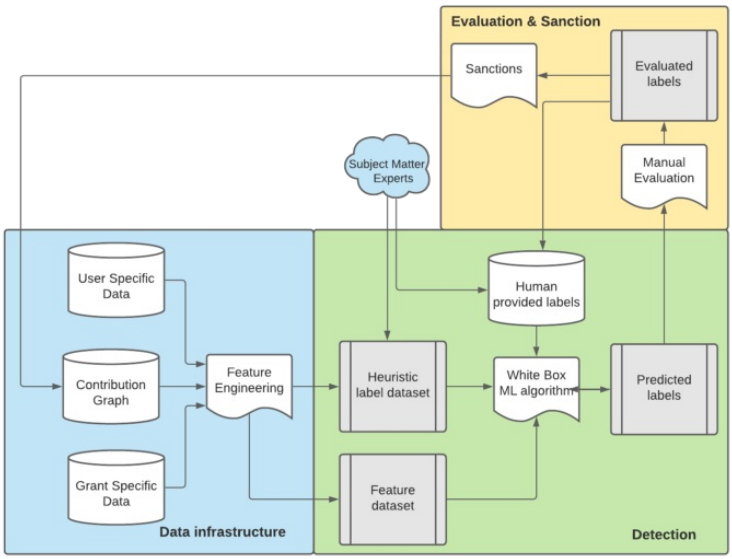


If you check the results from round 8 against round 7, you also get about 50% churn, and comparing round 7 to round 6 gives similar values. Hence, it is looking like the degree of churn is stable. To me, it seems like roughly 40-50% churn is a healthy level, balancing long-time projects' need for stability with the need to avoid new projects getting locked out, but this is of course only my subjective judgement.

Adversarial behavior

The challenging new phenomenon this round was the sheer scale of the adversarial behavior that was attempted. In this round, there were two major issues. First, there were large clusters of contributors discovered that were probably a few individual or small closely coordinated groups with many accounts trying to cheat the mechanism. This was discovered by proprietary analysis algorithms used by the Gitcoin team.

For this round, the Gitcoin team, in consultation with the community, decided to eat the cost of the fraud. Each project received the maximum of the match award it would receive if fraudulent transactions were accepted and the match award it would receive if they were not; the difference, about $33,000 in total, was paid out of Gitcoin's treasury. For future rounds, however, the team aims to be significantly stricter about security.



A diagram from the Gitcoin team's post describin their process for finding and dealing with adversarial behavior.

In the short term, simply ignoring fraud and accepting its costs has so far worked okay. In the long term, however, fraud must be dealt with, and this raises a challenging political concern. The algorithms that the Gitcoin team used to detect the adversarial behavior are proprietary and closed-source, and *they have to be closed-source* because otherwise the attackers could adapt and get around them. Hence, the output of the quadratic funding round is not just decided by a clear mathematical formula of the inputs. Rather, if fraudulent transactions were to be removed, it would also be fudged by what risks becoming a closed group twiddling with the outputs according to their arbitrary subjective judgements.

It is worth stressing that this is not Gitcoin's fault. Rather, what is happening is that Gitcoin has gotten big enough that it has finally bumped into the exact same problem that every social media site, no matter how well-meaning its team, has been bumping into for the past twenty years. Reddit, despite its well-meaning and open-source-oriented team, employs many secretive tricks to detect and clamp down on vote manipulation, as does every other social media site.

This is because *making algorithms that prevent undesired manipulation, but continue to do so despite the attackers themselves knowing what these algorithms are, is really hard*. In fact, **the entire science of mechanism design is a half-century-long effort to try to solve this problem**. Sometimes, there are successes. But often, they keep running into the same challenge: collusion. It turns out that it's not that hard to make mechanisms that give the outcomes you want if all of the participants are acting independently, but once you admit the possibility of one individual controlling many accounts, the problem quickly becomes much harder (or even intractable).

But the fact that we can't achieve perfection doesn't mean that we can't try to come closer, and benefit from coming closer. Good mechanisms and opaque centralized intervention are substitutes: the better the mechanism, the closer to a good result the mechanism gets all by itself, and the more the secretive moderation cabal can go on vacation (an outcome that the actually-quite-friendly-and-cuddly and decentralization-loving Gitcoin moderation cabal very much wants!). In the short term, the Gitcoin team is also proactively taking a third approach: making fraud detection and response accountable by inviting third-party analysis and community oversight.

Picture courtesy of the Gitcoin team's excellent blog post.

Inviting community oversight is an excellent step in preserving the mechanism's legitimacy, and in paving the way for an eventual decentralization of the Gitcoin grants institution. However, it's not a 100% solution: as we've seen with technocratic organizations inside national governments, it's actually quite easy for them to retain a large amount of power despite formal democratic oversight and control. **The long-term solution is shoring up Gitcoin's *passive* security, so that active security of this type becomes less necessary.**

One important form of passive security is making some form of unique-human verification no longer optional, but instead mandatory. Gitcoin already adds the option to use phone number verification, BrightID and several other techniques to "improve an account's trust score" and get greater matching. But what Gitcoin will likely be forced to do is make it so that some verification is required *to get any matching at all*. This will be a reduction in convenience, but the effects can be mitigated by

the Gitcoin team's work on enabling more diverse and decentralized verification options, and the long-term benefit in enabling security without heavy reliance on centralized moderation, and hence getting longer-lasting legitimacy, is very much worth it.

Retroactive airdrops

A second major issue this round had to do with Maskbook. In February, Maskbook announced a token and the token distribution included a retroactive airdrop to anyone who had donated to Maskbook in previous rounds.

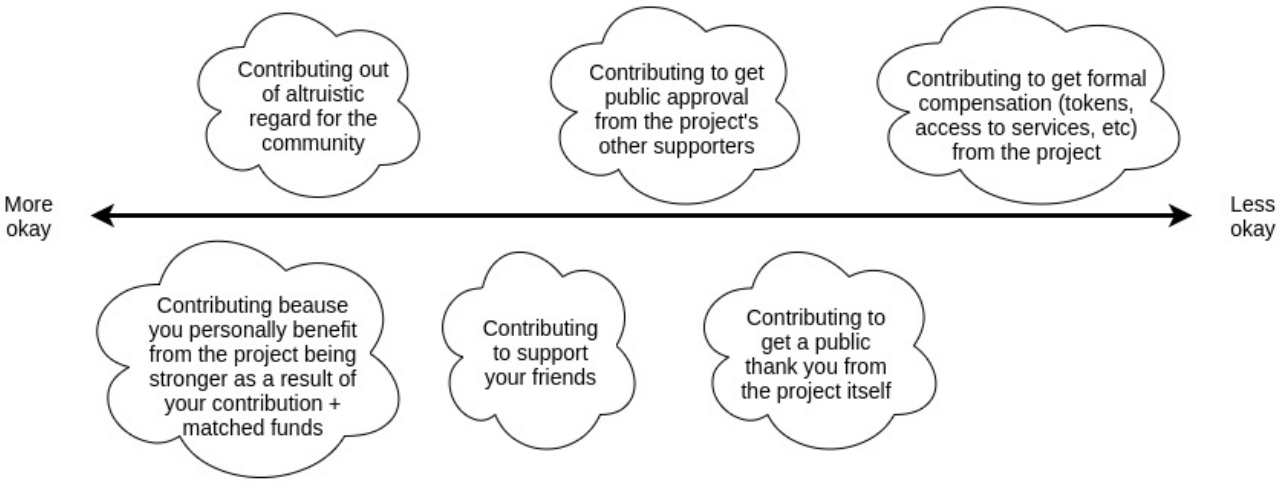
	Level	Regular Users	Power User
Mask Insider	NFT	Interacted but no longer hold	All Holders
	Red Packet	>=1, <=5 interaction	> 5 interactions
	ITO	>=1 interaction	
Ecosystem Participants	Voters	>=2, <=10 votes	>10 votes
	Gitcoin	Donated to >= 2 rounds (8 in total)	Mask donor before 1/1/2021
	ENS		Linked with Twitter

The table from Maskbook's announcement post showing who is eligible for the airdrops.

The controversy was that Maskbook was continuing to maintain a Gitcoin grant this round, despite now being wealthy and having set a precedent that donors to their grant might be rewarded in the future. The latter issue was particularly problematic as it *could* be construed as a form of obfuscated vote buying. **Fortunately, the situation was defused quickly; it turned out that the Maskbook team had simply forgotten to consider shutting down the grant after they released their token, and they agreed to shut it down. They are now even part of the funders' league, helping to provide matching funds for future rounds!**

Another project attempted what some construed as a "wink wink nudge nudge" strategy of obfuscated vote buying: they hinted in chat rooms that they have a Gitcoin grant and they are going to have a token. No explicit promise to reward contributors was made, but there's a case that the people reading those messages could have interpreted it as such.

In both cases, what we are seeing is that *collusion is a spectrum, not a binary*. In fact, there's a pretty wide part of the spectrum that even completely well-meaning and legitimate projects and their contributors could easily engage in.



Note that this is a somewhat unusual "moral hierarchy". Normally, the more acceptable motivations would be the altruistic ones, and the less acceptable motivations would be the selfish ones. Here, though, the motivations closest to the left *and* the right are selfish; the altruistic motivation is close to the left, but it's not the *only* motivation close to the left. The key differentiator is something more subtle: **are you contributing because you like the consequences of the project getting funded (inside-the-mechanism), or are you contributing because you like some (outside-the-**

mechanism) consequences of *you personally* funding the project?

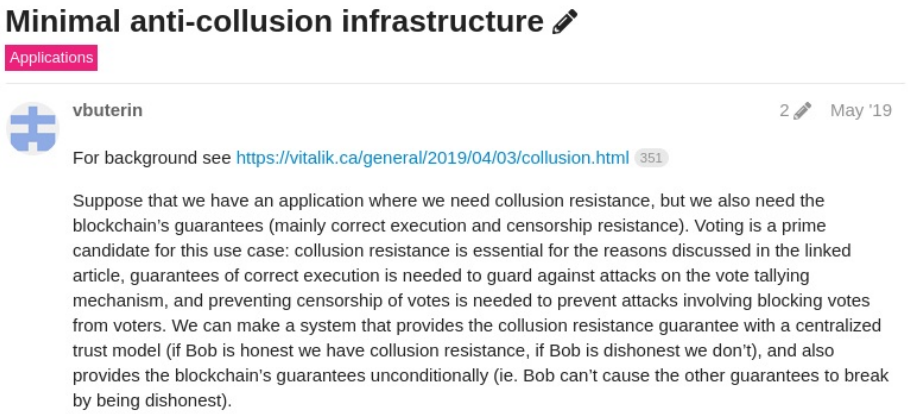
The latter motivation is problematic because it subverts the workings of quadratic funding. Quadratic funding is all about assuming that people contribute because they like the consequences of the project getting funded, recognizing that the amounts that people contribute will be much less than they ideally "should be" due to the tragedy of the commons, and mathematically compensating for that. But if there are large side-incentives for people to contribute, and these side-incentives are attached to that person specifically and so they are not reduced by the tragedy of the commons at all, then the quadratic matching *magnifies* those incentives into a very large distortion.

In both cases (Maskbook, and the other project), we saw something in the middle. The case of the other project is clear: there was an accusation that they made hints at the possibility of formal compensation, though it was not explicitly promised. In the case of Maskbook, it seems as though Maskbook did nothing wrong: the airdrop was retroactive, and so none of the contributions to Maskbook were "tainted" with impute motives. But the problem is more long-term and subtle: **if there's a *long-term pattern* of projects making retroactive airdrops to Gitcoin contributors, then users will feel a pressure to contribute primarily not to projects that they think are public goods, but rather to projects that they think are likely to later have tokens**. This subverts the dream of using Gitcoin quadratic funding to provide alternatives to token issuance as a monetization strategy.

The solution: making bribes (and retroactive airdrops) cryptographically impossible

The simplest approach would be to delist projects whose behavior comes too close to collusion from Gitcoin. In this case, though, this solution cannot work: the problem is not projects doing airdrops *while* soliciting contributions, the problem is projects doing airdrops *after* soliciting contributions. While such a project is still soliciting contributions and hence vulnerable to being delisted, there is no indication that they are planning to do an airdrop. More generally, we can see from the examples above that policing motivations is a tough challenge with many gray areas, and is generally not a good fit for the spirit of mechanism design. But if delisting and policing motivations is not the solution, then what is?

The solution comes in the form of a technology called MACI.



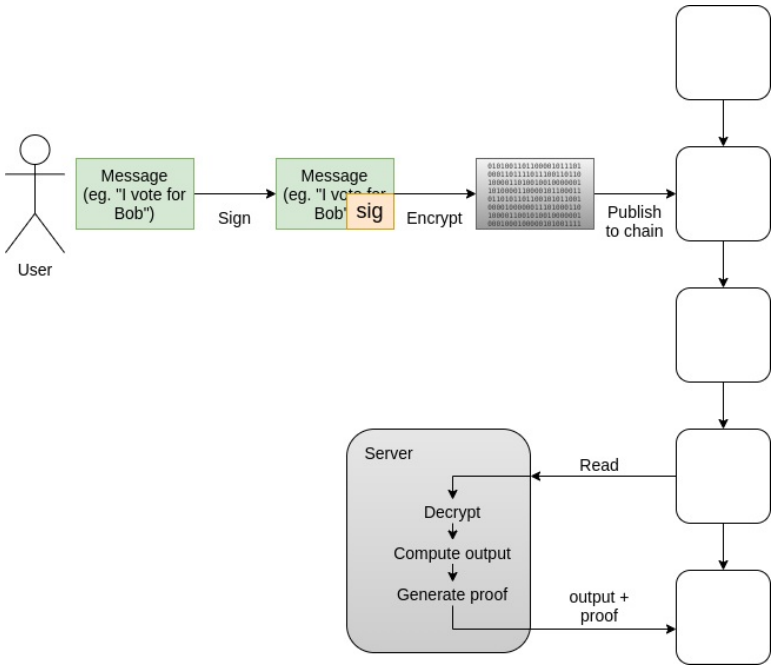
MACI is a toolkit that allows you to run *collusion-resistant applications*, which simultaneously guarantee several key properties:

- **Correctness**: invalid messages do not get processed, and the result that the mechanism outputs actually is the result of processing all valid messages and correctly computing the result.
- **Censorship resistance**: if someone participates, the mechanism cannot cheat and pretend they did not participate by selectively ignoring their messages.
- **Privacy**: no one else can see how each individual participated.
- **Collusion resistance**: a participant cannot prove to others how they participated, *even if they wanted to prove this*.

Collusion resistance is the key property: it makes bribes (or retroactive airdrops) impossible, because

users would have no way to prove that they actually contributed to someone's grant or voted for someone or performed whatever other action. This is a realization of the secret ballot concept which makes vote buying impractical today, but with cryptography.

The technical description of how this works is not that difficult. Users participate by signing a message with their private key, encrypting the signed message to a *public key* published by a central server, and publishing the encrypted signed message to the blockchain. The server downloads the messages from the blockchain, decrypts them, processes them, and outputs the result along with a ZK-SNARK to ensure that they did the computation correctly.



Users cannot prove how they participated, because they have the ability to send a "key change" message to trick anyone trying to audit them: they can first send a key change message to change their key from A to B, and then send a "fake message" signed with A. The server would reject the message, but no one else would have any way of knowing that the key change message had ever been sent. There is a trust requirement on the server, though only for privacy and coercion resistance; the server cannot publish an incorrect result either by computing incorrectly or by censoring messages. In the long term, multi-party computation can be used to decentralize the server somewhat, strengthening the privacy and coercion resistance guarantees.

There is already a quadratic funding system using MACI: clr.fund. It works, though at the moment proof generation is still quite expensive; ongoing work on the project will hopefully decrease these costs soon.

Practical concerns

Note that adopting MACI does come with necessary sacrifices. In particular, there would no longer be the ability to see who contributed to what, weakening Gitcoin's "social" aspects. However, the social aspects could be redesigned and changed by taking insights from elections: elections, despite their secret ballot, frequently give out "I voted" stickers. They are not "secure" (in that a non-voter can easily get one), but they still serve the social function. One could go further while still preserving the secret ballot property: **one could make a quadratic funding setup where MACI outputs the value of *how much* each participant contributed, but not who they contributed to**. This would make it impossible for specific projects to pay people to contribute to them, but would still leave lots of space for users to express their pride in contributing. Projects could airdrop to *all* Gitcoin contributors without discriminating by project, and announce that they're doing this together with a link to their Gitcoin profile. However, users would still be able to contribute to someone else and collect the airdrop; hence, this would arguably be within bounds of fair play.

However, this is still a longer-term concern; MACI is likely not ready to be integrated for round 10. For the next few rounds, focusing on stepping up unique-human verification is still the best priority.

Some ongoing reliance on centralized moderation will be required, though hopefully this can be simultaneously reduced and made more accountable to the community. The Gitcoin team has already been taking excellent steps in this direction. And if the Gitcoin team does successfully play their role as pioneers in being the first to brave and overcome these challenges, then we will end up with a secure and scalable quadratic funding system that is ready for much broader mainstream applications!