

Micro Oracles: A Simple Commitment-Augmented Funding Mechanism

Ajesiroo
ajesiroo@protonmail.com

January 12, 2024
Draft

Abstract

Standard on-chain funding mechanisms are often prone to misaligned incentives, both in the form of grants, where mismanaging of resources is common, and in the form of post factum rewards, where backers occasionally renege on prior assurances. Multi-node oracles are sometimes used to mitigate these drawbacks by requiring confirmation of an off-chain event prior to disbursement, but they are typically difficult to implement as they require significant considerations around economic viability and security. We propose a simple oracle system that can be reduced to two parties. A contribution is held in a smart contract and a portion of the amount is reserved, representing a minimum commitment level. Upon the actualisation of an off-chain event, the backer can either transfer the full contribution to the recipient or revert the majority of the contribution and burn the reserved portion in the process. The minimum commitment level not only serves as a signalling device for the recipient, but disincentivises the backer from making allocations arbitrarily.

1 Introduction

Conventional on-chain contribution schemes often carry the same fundamental trust assumptions observed in the context of off-chain equivalents. This most typically takes the form of contributors trusting the recipient, as we see in applications such as Gitcoin Grants and Giveth, but in less common cases the dynamic is reversed and in the setting of post factum reward schemes, the recipient must trust that contributors will fulfil pre-defined assurances upon the realisation of a deliverable [1][2]. In both scenarios there is significant potential for misaligned incentives; up-front funding often allows recipients to mismanage resources, sometimes to the point of outright fraud, and post factum rewards can enable contributors to renege on prior commitments even after some criteria is satisfied. Recent on-chain approaches that try to mitigate some of these issues typically centre on oracles consisting of many nodes, sometimes in tandem with cryptoeconomic incentives to encourage participants to attest to an event honestly [3][4][5]. The major downside with these mechanisms, however, is that they consist of many “moving parts” and their inherent complexity not only makes them difficult to implement, but require significant considerations both around their economic viability and the multitude of potential attack vectors.

A complementary approach that does not attenuate counterparty trust assumptions to the same extent as the aforementioned oracles, but nevertheless incentivises the various parties to follow through on reasonable expectations, involves programatically reserving a portion of a contribution in such a way that it can only be released to the recipient along with the remainder of the contribution, or burned and effectively making it inaccessible by any party. This encourages the recipient to develop some pre-defined deliverable, as they are able to observe a degree of commitment from the counterparty, while simultaneously allowing the backer to revert the majority of the funding if expectations were not met. The major advantage of this approach is simplicity: rather than requiring many participants to ensure the viability of the oracle, the mechanism can be reduced to a single contributor and recipient, which in turn alleviates some of the by-products stemming from more complex mechanisms.

2 Micro Oracles

A commitment level, typically several percentage of the overall contribution, is held in the core smart contract comprising the system, and the backer can release this amount together with the primary component of the contribution at any point in the future. The backer can also opt to revert the majority of the contribution to the originating address, but in doing so the reserved portion is burned, thus the latter represents a *minimum commitment* on the part of the backer. This has a signalling effect that may be instrumental in the recipient's decision to develop their project further, while at the same time providing some incentive for the contributor to remain nominally consistent with their indicated intentions.

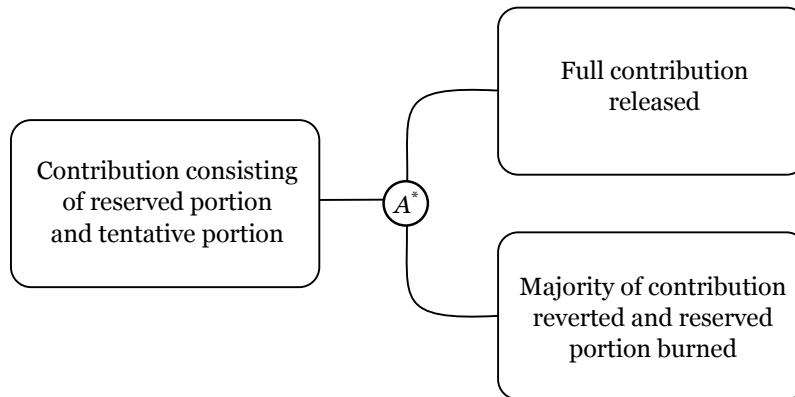


Figure 1

Where A^* is the allocation function.

2.1 Commitment Levels

A variety of different formulas can be used to determine the portion of the contribution that represents the minimum commitment level. Perhaps most logically, it can be derived as a fixed percentage of the overall amount, generally between 4%-15%, and this attribute of scaling in proportion with contribution allows for a greater absolute commitment in the context of larger contributions where more is at stake for both parties. Alternatively, a regressive or capped approach can be used, where the relative commitment level as a percentage decreases as the overall contribution rises, enabling more manageable minimum commitments despite larger contribution sizes. A flat rate could also be considered an option, although at the expense of the flexibility that the aforementioned approaches provide.

2.2 Privacy

For certain applications, there may be an expectation of privacy for both the contributor and the recipient. The current best practice to enable pseudonymity between the various participants is through the use of zero-knowledge succinct non-interactive arguments of knowledge (zk-SNARKs). An intermediate step before interacting with the main system would be required, where private addresses are assigned to regular externally owned accounts. A less encompassing approach to private addresses would be to obfuscate only the individual transactions, and while being more straightforward to implement, it does not provide the same level of privacy as the former.

2.3 Other Considerations

One of the main drawbacks with the mechanism is that it could lead to misuse under certain circumstances by the contributor. If the recipient is not fully cognizant of the implications of the non-committed portion of the contribution, and the system is used as a primary rather than auxiliary source of funding, a malicious participant could attempt to use it as a device to incentivise the development of a deliverable with the intention of reverting the majority of the contribution from the outset. This would still come at a cost to the bad actor, as the portion that represents the commitment level will necessarily be burned, but it is nevertheless an important consideration. For this reason, it is crucial that the inherent properties of the mechanism, including its limitations compared to oracles with stronger mitigation of trust assumptions, is described clearly as part of the user experience. As touched upon previously, the system may be best suited as a complementary source of funding, particularly where there is already an impetus to develop the project, or as an approach for smaller milestones of an overarching project.

Another facet that should be explored is the implementation of minimum hold periods, which in practical terms, amounts to first checking if `block.number` exceeds a pre-defined threshold prior to calling the transfer function. The utility of such a check may be marginal, however, as the backer will still be able to ultimately release or revert the main component of the contribution.

References

- [1] *Gitcoin Grants*. Retrieved Jan 12, 2024 from <https://grants.gitcoin.co>
- [2] *Giveth*. Retrieved Jan 12, 2024 from <https://giveth.io>
- [3] Ajesiroo. 2021. *Ternary Funding and Joint Tokens: A Trustless Approach to Public Goods Funding*. Retrieved Jan 12, 2024 from <https://ajesiroo.github.io/tf.pdf>
- [4] Clément Lesaege, Federico Ast, and William George. 2019. *Kleros Short Paper*. Retrieved Jan 12, 2024 from <https://kleros.io/whitepaper.pdf>
- [5] 2020. *UMA Data Verification Mechanism: Adding Economic Guarantees to Blockchain Oracles*. Retrieved Jan 12, 2024 from <https://github.com/UMAProtocol/whitepaper/blob/master/UMA-DVM-oracle-whitepaper.pdf>