Constitutional Proposal

Abstract / Motivation

Currently, the DAO receives transaction fees on Nova at an address accessible via the core governance system. This means that any time the DAO wishes to transfer these funds, it must pass a constitution proposal. We propose instead a system in which these funds are "automatically" (permissionlessly) sent from Nova to the Arbitrum DAO Treasury on Arbitrum One. This gives the DAO faster, easier accessibility to these funds, and should make bookkeeping simpler.

Details

Currently, the portion of the Nova transaction feesthat the DAO collects are sent via thefund distributor contracts to 0xf7951D92B0C345144506576eC13Ecf5103aC905a, the address alias of the core governance l1 timelock. This allows the DAO to spend the funds via a round-trip core proposal initiated from the core governor on Arbitrum One.

A system in which these funds were sent to the DAO Treasury without the DAO needing to take action is preferable for several reasons:

- Spending from the DAO treasury requires a lower quorum (3% vs. 5%)
- Spending from the DAO treasury is faster (doesn't impose the additional ~ 2 weeks of delay
- Keeping all fees in a single location makes bookkeeping simpler and easier to reason about.

We propose an update in which all Nova fund distributors that distribute to the timelock alias are updated to instead distribute to a system of "fee routers", via which permissionless transactions will allow the funds to be routed to the Arbitrum Treasury.

The lifecycle works as follows:

1. A distributeRewards is called on a RewardDistributor contract, sending funds to a ChildToParentRouter contract.
2. Either upon receiving funds or via a call to routeFunds, the ChildToParentRouter creates an L2-to-L1 message which sends the contract's full Ether balance.
3. The L2-to-L1 message is executed, transferring the Ether to a ParentToChildRouter contract on L1.
4. routeFunds is called on ParentToChildRouter, creating a retryable ticket which transfers its full Ether balance to the DAO Treasury on Arbitrum One.

Implementation

- Fee Router contracts:
- Github link: fund-distribution-contracts/src/FeeRouter at 61f4f60384e2ecba8250287dfb2778ce30bd82b0 · OffchainLabs/fund-distribution-contracts · GitHub
- Deployments:
- ParentToChildRewardRouter | Address 0x3EB1F638C115F7dE148FAE18b8217e6c200c8657 | Etherscan
- ChildToParentRewardRouter | Address 0x47a85c0a118127f3968a6a1a61e2a326517540d4 | Arbiscan
- ParentToChildRewardRouter | Address 0x3EB1F638C115F7dE148FAE18b8217e6c200c8657 | Etherscan
- ChildToParentRewardRouter | Address 0x47a85c0a118127f3968a6a1a61e2a326517540d4 | Arbiscan
- Audit Link: fund-distribution-contracts/audits at cb9ef7b5a5730fb9eb4bfdd2f9bc1f716a6e9253 · OffchainLabs/fund-distribution-contracts · GitHub
- Github link: fund-distribution-contracts/src/FeeRouter at 61f4f60384e2ecba8250287dfb2778ce30bd82b0 · OffchainLabs/fund-distribution-contracts · GitHub
- Deployments:
- ParentToChildRewardRouter | Address 0x3EB1F638C115F7dE148FAE18b8217e6c200c8657 | Etherscan
- ChildToParentRewardRouter | Address 0x47a85c0a118127f3968a6a1a61e2a326517540d4 | Arbiscan

- ParentToChildRewardRouter | Address 0x3EB1F638C115F7dE148FAE18b8217e6c200c8657 | Etherscan
- ChildToParentRewardRouter | Address 0x47a85c0a118127f3968a6a1a61e2a326517540d4 | Arbiscan
- Audit Link: fund-distribution-contracts/audits at cb9ef7b5a5730fb9eb4bfdd2f9bc1f716a6e9253 · OffchainLabs/fund-distribution-contracts · GitHub
- Upgrade Action contract:
- ArbOS30 AIP: add AIPNovaFeeRoutingAction by DZGoldman · Pull Request #230 · ArbitrumFoundation/governance · GitHub
- ArbOS30 AIP: add AIPNovaFeeRoutingAction by DZGoldman · Pull Request #230 · ArbitrumFoundation/governance · GitHub

Additional Action

Sometime after this proposal action, an additional proposal can be submitted to transfer the funds remaining in the timelock's address alias to the fee router.

Timeline

NOTE:

If this vote passes on Snapshot, the tentative plan is to upgrade Arbitrum Sepolia to arbOS 30 the week of June 10th.

If Arbitrum Sepolia upgrades to arbOS 30, there will be an ecosystem-wide plan to ensure non-breaking compatibility of infrastructure and applications with Stylus, which will take several weeks or months. The ultimate goal is that the experience for EVM developers remains as unchanged as possible if arbOS reaches mainnet. Once enough confidence is reached, there will be an onchain vote to upgrade Arbitrum One and Nova to arbOS 30. The Arbitrum community will be informed of any significant updates as they arise.