Gasless Poll Voting

Gasless voting in the governance portal is an enhancement to the on-chain polling system offered by MakerDAO. At a high level, the voting portal frontend now offers the ability for a voter to submit their vote preference at no cost. The gasless voting system uses a relayer service to securely submit the vote metadata to a polling contract on the Arbitrum blockchain—an L2 built on top of Ethereum.

These votes are merged with votes from the L1 Ethereum network and the results are presented seamlessly in the UI.

System Overview

Technical Diagram

?

User Flow

The general flow works like this:

1. User makes their vote selections and then submits their vote via the gasless system. The gasless poll voting system is opt-out, meaning that it will be used by default unless the user specifies that they rather use legacy (L1) voting.
2. The user signs a message consisting of their vote selections and some metadata with their wallet, such as Metamask.
3. The governance portal backend does some validation to prevent spam, and ensure the vote is secure.
4. The vote is submitted by our relayer to the polling contract on the Arbitrum network, gas is paid for by the relayer.
5. The governance polling database listens for the event and writes it to the table. When requested by the frontend, the votes from both networks are merged and calculated.
6. The frontend displays the poll results in the UI.
7.

Spam Protection Checks

Gasless Poll Voting is available to all users who pass the following checks, which have been introduced as a means to prevent spam:

1. Address has at least 0.1 MKR of poll voting weight.
2. Address has not used the Gasless Poll Voting system in the last 10 minutes.
3. Address has not voted more than once in any of the polls included in the ballot.
4. Address is not a multisig wallet. (note: Off-chain message signing is not yet supported bySafe
5. multisigs)
6.

Detailed overview

User Vote

On the polling page, users make one or more voting selections, and navigate to the review page. Here, some initial checks are performed, and if the user is eligible they will be presented with the option to submit their votes using the gasless system (or use the legacy system if desired).

?

If the user is ineligible to use the gasless system, or the gasless system is unavailable, they will be given the opportunity to vote with the legacy system, as well as an indication of the reasons why they are not eligible.

?

Signing Comments

After clicking “submit”, the user is asked to sign a message with voting options and some metadata. Using thesignTypedData_v4 method from the EIP-712 spec, the user can verify that information they are signing is correct. The wallet will present a decoded view of the signed data similar to the one below.

?

Explanation of each field:

- voter
- = the address casting the vote (the address that's signing the messafe will always be an EOA)
- pollIds
- = the IDs of the polls that your ballot contains

- optionIds
- = the 'option IDs' that you're voting on for each poll ID
- nonce
- = the nonce is a number that is incremented each time you vote via signature, which prevents replay attacks
- expiry
- = signed messages expire after a few hours, which cancels the vote in case the system fails to deliver the vote transaction to the L2
-

The signed message is run through some additional validity checks to check for spam in our backend API. If it passes these checks, the relayer takes over. We verify that:

- the user hasn't voted in any of the polls yet
- the user has at least 0.1 MKR in voting weight
- the user hasn't submitted another gasless poll vote in the last 10 mins
- the signature on the signed message is valid
- the votes are not expired
- the poll IDs included in the ballot are still active
- the nonce for the account voting is correct
- the relayer provided by the DUX Core Unit has funds (aETH)
-

Defender Relay

The defender relay is an Ethereum-account-as-a-service provided by OpenZeppelin that allows us to send transactions to the Arbitrum network without having to manage private keys, nonces, etc. This wallet is pre-funded by DUX with enough Arbitrum ETH to send transactions on behalf of the gasless voting user.

Our backend API utilizes the defender relay SDK to submit the transaction to the Arbitrum polling contract. We have worked with the techops core unit to setup vital statistics monitoring of the relayer.

?

Arbitrum Polling Contract

- https://github.com/makerdao-dux/polling-contract
- Arbitrum Mainnet Address:0x4f4e551b4920a5417F8d4e7f8f099660dAdadcEC
- Arbitrum Goerli Address:0x4d196378e636D22766d6A9C6C6f4F32AD3ECB050
-

The Arbitrum polling contract is based off the Ethereum Mainnet polling contract with some modifications to handle the vote hash validation, and emit events with the necessary additional metadata. When the relayer address calls thevote function, the contract will emit an event to the chain that will be scraped by the gov polling db.

Gov Polling DB

The gov polling db ETL service runs on both networks scraping blocks, listening for events. When it discovers aVoted event, it writes the parameters from the event to the respective table, keeping track of all such events. There is an API function which merges the votes from both chains, returning the latest valid vote.

The frontend uses graphQL to request the vote events from the API function in the database, then displays the results in the UI, indicating which chain the vote is from, and a link to the transaction.

?

Page last reviewed: 2022-11-09 Next review due: 2023-11-09

Previous Practical Guide to VotingNext On-Chain Governance Last updated1 year ago On this page *System Overview * Technical Diagram * User Flow * Spam Protection Checks * Detailed overview * User Vote * Signing Comments * Defender Relay * Arbitrum Polling Contract * Gov Polling DB

Was this helpful? Edit on GitHub