GPv2VaultRelayer

Architecture

TheGPv2VaultRelayer contract is an important component used to protect user funds from malicious solvers. As previously mentioned, theGPv2Settlement contract allows using arbitrary on-chain liquidity through interactions (such as performing a swap onBalancer V2 , or performing a Paraswap trade). IfVault andERC-20 allowances were made directly to aGPv2Settlement contract, a malicious solver could drain user funds through the interaction mechanism. However, since these allowances are made to theGPv2VaultRelayer contract and interactions to the contract are strictly forbidden, malicious solvers have no direct access to user funds. TheGPv2Settlement contract uses theGPv2VaultRelayer to withdraw user tokens only as part of the trade, which contains strong guarantees that the user's signed order parameters are respected.

TheGPv2VaultRelayer has access to user balances through 3 mechanisms:

1. FallbackERC-20 Allowances
2. Balancer External Balances
3. Balancer Internal Balances

Guarantees and Invariants

- TheGPv2VaultRelayer
- is only able to transferERC-20
- tokens to theGPv2Settlement
- contract

FallbackERC-20

Allowances

The third and final method of approving tokens for CoW protocol is to use directERC-20 allowances to theGPv2VaultRelayer . This works like most other trading protocols, where for each token you want to sell, an allowance must first be approved for theGPv2VaultRelayer contract.

Orders with thesellTokenBalance flag set toerc20 will withdraw using this process. ThebuyTokenBalance flag can also be set toerc20 in order to receive trade proceeds directly inERC-20 amounts.

Balancer External Balances

The first mechanism that theGPv2VaultRelayer contract can use to withdraw userERC-20 tokens is throughVault external balances. This works by having anERC-20 allowance for the Balancer Vault, and a relayer approval for theGPv2VaultRelayer contract.

This allowance and approval combination allows theGPv2VaultRelayer contract to transferERC-20 tokens through theVault . Roughly speaking, the process works in the following way:

1. GPv2VaultRelayer
2. request to the Balancer Vault anERC-20
3. transfer from the user account to theGPv2Settlement
4. contract
5. The Balancer Vault verifies that theGPv2VaultRelayer
6. contract is:
7. a. Authorized by Balancer governance to act as a relayer
8. b. The user has set an approval for that specific relayer
9. The Balancer Vault issues anERC-20
10. transfer from the user account to theGPv2Settlement
11. contract using the Vault's existingERC-20
12. allowance

This system for withdrawing user funds has several advantages such as:

- It can reuse existingVault
- ERC-20
- allowances and doesn't require new ones specific to the CoW Protocol.
- Upgrades to the CoW Protocol contract would only require a single relayer approval for all tokens instead of individualERC-20
- approvals for each token being traded.
- TheGPv2VaultRelayer
- approval can be revoked by a single transaction to theVault

- instead of multiple transactions to eachERC-20
- token for which the user wants to remove the approval.

Orders with thesellTokenBalance flag set toexternal will withdraw using this process.

Balancer Internal Balances

The second mechanism is to use balances internal to theVault . The Balancer V2 vault can accrueERC-20 token balances and keep track of them internally in order to allow extremely gas-efficient transfers and swaps. The CoW Protocol contracts can make use of this in order to decrease the gas cost of settling a user order on-chain. In order for this to work, the user must approve theGPv2VaultRelayer contract and have internalVault balances available.

Internal balances can be withdrawn from theVault at any time for theirERC-20 equivalent amounts.

Orders with thesellTokenBalance flag set tointernal will withdraw using this process. ThebuyTokenBalance flag can also be set tointernal in order to receive trade proceeds in internal balances instead ofER20 token balances.

Data Types and Storage

Nil

Functions

For the Protocol

transferFromAccounts

This function is used for transferringERC-20 tokens from users to theGPv2Settlement contract in the course of settling a batch auction.

function

transferFromAccounts (GPv2Transfer . Data [ ]

calldata transfers)

external onlyCreator { vault . transferFromAccounts (transfers , msg . sender) ; }

batchSwapWithFee

This function is used in the course of settling a single trade on-chain. It is called by theGPv2Settlement contract and is used to perform a batch swap on the Balancer V2 vault. The function is defined as:

function

batchSwapWithFee (IVault . SwapKind kind , IVault . BatchSwapStep [ ]

calldata swaps , IERC20 [ ]

memory tokens , IVault . FundManagement memory funds , int256 [ ]

memory limits , uint256 deadline , GPv2Transfer . Data calldata feeTransfer)

external onlyCreator returns

(int256 [ ]

memory tokenDeltas)

Indexing

Nil

Off-chain

Nil