SendRawTransactionConditional explainer

A sequencer rpc,eth_sendRawTransactionConditional , allowing callers to conditionally include a transaction based on a set of provided options.

This feature is meant to unblock use cases that require atomic inclusion, otherwise possible on Ethereum through specialized block builders like Flashbots. Some examples:

- 4337 Bundlers utilizing a shared mempool of UserOperations. Reverted transactions due to conflicting UserOperations would make it too costly for bundlers to operate on L2, forcing the use of private mempools.

Specification

This endpoint is an extension ofeth_sendRawTransaction , with an extra parameter of "options" with the following structure:

{ "knownAccounts": [optional] A map of accounts with expected storage. The key is account address If the value is hex string, it is the known storage root hash of that account. If the value is an object, then each member is in the format of "slot": "value", which are explicit slot values within that account storage. "blockNumberMin": [optional] minimal block number for inclusion "blockNumberMax": [optional] maximum block number for inclusion "timestampMin": [optional] minimum block timestamp for inclusion "timestampMax": [optional] maximum block timestamp for inclusion } The "cost" of a given conditional isapproximately determined by the number of storage lookups that would be incurred by validating this conditional. There's a predefined max of1000 that is allowed to prevent DoS.

Since the sequencer is not compensated for the additional state checks, otherwise through the GAS of the transaction, a configured rate limit is applied to this cost.

To also disincentive the use of this endpoint for MEV in comparison toeth_sendRawTransaction , the conditional is checked against the parent of the latest block. This conditional is checked once at the RPC layer prior to mempool submission. If rejected against chain state, the RPC will return an error with the following spec

- error code-32003
- (transaction rejected) with a reason string as to the specific failed check
- error code-32005
- (conditional cost) with a reason string if the conditional cost exceeded the maximum OR the cost was rate limited due to network conditions.

Successful submission doesNOT guarantee inclusion! The caller must observe the chain for a receipt with some timeout to determine non-inclusion. The conditional is also re-checked in the block building phase to handle intra-block conflicts.

⚠ Conditional transactions are tied to the block builder they are submitted to. This means that these transactionsare not gossiped between configured peers!!!

If you are running an active/passive setup with replicas that gossip txs to an active sequencer, this endpoint should be fronted by a proxy that can broadcast the request to all replicas.

How to enable

This feature can be enabled with the addition of a flag to op-geth.

- --rollup.sequencertxconditionalenabled
- (default: false) a boolean flag which enables the rpc.
- --rollup.sequencertxconditionalcostratelimit
- (default: 5000) an integer flag that sets the rate limit for cost observable per second.

⚠ It is not advised to publicly expose this sequencer endpoint due to DoS concerns. This supplemental proxy,op-txproxy , should be used to apply additional constraints on this endpoint prior to passing through to the sequencer.

Features Security