Rigil Testnet

TheSUAVE Rigil Testnet is live and public:

- Block Explorer
- Faucet
- EthStats
- Technical Docs
- chainId: 16813125
- Rigil Kettle Address:0x03493869959c866713c33669ca118e774a30a0e5
- Localhost Kettle Address:0xb5feafbdd752ad52afb7e1bd2e40432a485bbb7f

We have RPC nodes you can connect to:

RPC URL

https://rpc.rigil.suave.flashbots.net Connect to Rigil

RPC Key Differences

In order to keep some data in transactions confidential, SUAVE JSON-RPC extends the usual Ethereum JSOPN-RPC methods. Some methods in theeth_ namespace are overloaded to support confidential compute requests.

1. eth_sendRawTransaction

Creates a new message call transaction or a contract creation for any signedConfidentialComputeRequest .

1. eth_call

Executes a new message call immediately without creating a transaction on the block chain. It follows the same format as the defaulteth_call with two extra parameters:

- IsConfidential
- : Set to true to execute as a confidential request and access theMEVM
- methods.
- ExecutionAddress
- :address
-
    - (optional) The execution address that performs the execution.
- eth_kettleAddress

Returns the list of available addresses in the Kettle to execute the confidential compute request.

Testing the RPC

The easiest way to test your connection to an RPC endpoint is via a simple curl command.

Remote curl request

curl

-X POST \ -H

"Content-Type: application/json"

\ --data

'{"jsonrpc":"2.0","method":"eth_kettleAddress","params":[],"id":1}'

\ https://rpc.rigil.suave.flashbots.net

Local curl request

curl

-X POST \ -H

"Content-Type: application/json"

\ --data

'{"jsonrpc":"2.0","method":"eth_kettleAddress","params":[],"id":1}'

\ http://localhost:8545

Expected Response

If your connection is working properly you should get a response such as:

{ "jsonrpc" : "2.0" , "result" : "0x30870" , "id" : 1 } Note that the only difference between these two is the URL at the end of the curl request.

SUAVE Transactions

The example above follows theexact same API interface as the original go-ethereum client. However, if we grab a random transaction hash from theRigil Explorer , we can see the core difference with the SUAVE Rigil RPC: a new SUAVE transaction type.

Remote curl request

curl

-X POST \ -H

"Content-Type: application/json"

\ --data

'{ "jsonrpc": "2.0", "method": "eth_getTransactionByHash", "params": ["0x294b510e4fd257dec3d27b051f157489446c38828ff5f6b8d8c194797c6ddaab"], "id": 1 }'

\ https://rpc.rigil.suave.flashbots.net

Response

{ "jsonrpc" :

"2.0" , "result" :

{ "blockHash" :

"0x165444c40f3f963c91c0b8f30d40d2946c4264cb889e8254cf6668319d662fa9" , "blockNumber" :

"0xb0d61" , "chainId" :

"0x1008c45" , "confidentialComputeResult" :

"0xee2cc364000000000000000000000000000000000000000000000000000000000000000006039d118e4d92d41988043f41a0ae2412d5891812600c63b0dfba4cd7954e94b796e22ce51ae18ed57501f9df3a
, "from" :

"0xdacc37d00411381083d36864b9bfb06bc31a7aec" , "gas" :

"0x989680" , "gasPrice" :

"0x3b9aca00" , "hash" :

"0x294b510e4fd257dec3d27b051f157489446c38828ff5f6b8d8c194797c6ddaab" , "input" :

"0xee2cc364000000000000000000000000000000000000000000000000000000000000000006039d118e4d92d41988043f41a0ae2412d5891812600c63b0dfba4cd7954e94b796e22ce51ae18ed57501f9df3a
, "nonce" :

"0x5" , "r" :

"0xc9d97182f9cb70986e4096be9fb299a120f714aaf15897e1cd5c08a7281ac86f" , "requestRecord" :

{ ... } , "s" :

"0x765917eb8554a00d43162f9b9fff8646e8a7ce1756855d84f7627ec9571689d1" , "to" :

"0xa60f1b5cb70c0523a086bbcbe132c8679085ea0e" , "transactionIndex" :

"0x0" , "type" :

"0x50" , "v" :

"0x1" , "value" :

"0x0" } , "id" :

1 } This response has a couple fields that aren't in your traditional Ethereum transaction type, namely:

- confidentialComputeResult
- executionNode
- requestRecord

To dive deeper into these differences checkout the SUAVE chain specs . Edit this page Previous Tools Next SUAVE Standard Library