

Reduce Transaction Payload

Imagine we have a method M to our smart contract. M takes 6 parameters as inputs. Due to some contract restriction, their types are fixed as below:

- Param1: uint256
- Param2: uint256
- Param3: uint256
- Param4: address
- Param5: uint256
- Param6: uint256

Upon close examination, we realized that the range of the parameters are as follows:

- Param1: greater than 0 but less than 1, only needs first 4 decimals
- Param2: greater than 0 but less than 1, only needs first 4 decimals
- Param3: greater than 1 but less than 2, only needs first 4 decimals
- Param4: none zero address
- Param5: greater than 0 but less than 1, only needs first 4 decimals
- Param6: greater than 1 but less than 2, only needs first 4 decimals

Due to contract logic, except Param4, the rest of the parameters are scaled to 1e18 for decimals processing. At the same time, if users are not using what parameters to use for Param1 - Param6, the contract will provide them with default values. In this case, we realized 90% of the users will use default values instead.

For this task, please design a solution that reduces the transaction data sent from Web3 client to this method M as much as possible. You can assume M is just a dummy function, it does not do anything, what matters is the transaction data reduction.

For the language, you should use solidity as the smart contract language. For the client, you can use Typescript/Rust. If it is possible, please provide a full docker dev environment.

To submit the exercise, you can upload to a public git repo or private git repo and add our devs to the project.