Summary

Most of what happens in the transaction lifecycle team is around managing what happens to a transaction before it gets included on-chain. With that in mind most of our work requires a reasonable understanding of ethereum and the ability to figure out things that most people aren't concerned with. Consequently the space you will find yourself in is one that can not be easily googled and may have no obvious pre-existing solutions for. However, there are many existing non-ethereum technologies that can be leveraged. They may not be suitable as base layer decentralized solutions, but can be bought to market without the need for protocol changes to ethereum.

Tech Stack

Most of our tech stack is written in golang, your solution must be written primarily in go. It should have an appropriate level of testing with automated examples using a commonly used web3 integration library (e.g ethersjs).

Challenge

We want to improve the experience users have of web3 by enabling smarter transactions cancellable transactions. A common thing for users to do is to submit a transaction with a low gas fee. The transaction will sit in the public mempool until such time that the base fee drops enough to allow inclusion. If the user wishes to cancel that transaction they need to submit another transaction with a higher gas fee that will get mined first. This will cost the user some eth.

You need to build a transaction proxy that will only submit transactions to the public mempool when there is a high degree of confidence that the transaction will be included in the next block. If the user submits a 'cancel transaction' it should never be submitted to the public mempool. You should write a json rpc server endpoint that will accept standard raw transactions on eth_sendRawTransaction and store them for later submission (in memory will do) to the public mempool. When the base fee is low enough the transaction should be forwarded/submitted.

Bonus points

Setup metamask to use your RPC server. What problems and challenges are there with this approach? How can they be overcome?

References

https://docs.etherscan.io/tutorials/signing-raw-transactions
https://ethereum.org/en/developers/docs/apis/json-rpc/#eth_sendrawtransaction
https://info.etherscan.com/how-to-cancel-ethereum-pending-transactions/
https://www.jsonrpc.org/specification