Inter IIT Tech Meet 11.0

Consensys Metamask Snaps

Problem Statement

The wallet experience can be enhanced and customized with Snap, a software that runs in a different environment. To modify MetaMask utilizing Snaps, we are developing for new and intriguing use cases.

For instance, utilizing internal APIs, MetaMask may easily add new APIs, support various blockchain protocols, or change the functionality of already existing features.

Goals and Strategy

Transaction approval and storing data features.

Schedule actions with cron jobs

Populate MetaMask's pre-transaction window with custom transaction insights

Control non-EVM accounts and assets in MetaMask

Notify users in MetaMask

Display a custom confirmation screen in MetaMask

SOLUTION FORMULATION

Simplify

It delivers a solution to another side of using a decentralized network, that is, high gas and transaction fees by reducing the number of transactions needed

this is done by optimizing the Max-Flow when a number of shared transactions are being done in a relatively bounded group of people.

Smart Contract Automation

It allows users to do recurring payments or execution of a smart contract and even scheduling one time transfers.

Transaction Insights

The API is utilized to provide additional data about the safety of the ongoing transaction before the transaction is carried out.

SOLUTION FORMULATION

Non-EVM based asset management

The Snap has the capability to store and manage private keys of not only EVM based keys but also other networks like solana, tron, bitcoin, etc.

Contact Book

The Snap can also be used to store data and so contacts can be stored while Simplify functionalities allows the contact book to synchronize data over devices.

Problems Faced

Ideation

The Problem Statement was quite new most people in team had experience with smart contracts but required some more learning to develop on this platform

Many Ideas that were thought at the first go have already been implemented in a Dapp or Snap like Password Manager, Loans using blockchain, gas fee, portfolio/staking, non-EVM wallet for bitcoin.

While other ideas either could exist as separate Dapp without snap or couldn’t be implemented with Snap.

Problems Faced

Technical

Features like Custom UI were released in between our solution of the problem statement while onCronjobHandler was not available in @metamask/snap-types library but rather it was present in @metamask/snaps-types.

Web3 is quite new and Snaps even newer. The core team knew about most specifics while other members needed to catchup to some of the concepts.

And members with knowhow in this field were the only ones having knowhow in some other problem statements too.

Not able to use certain libraries because of the structure of the snap. But this led us to think in new ways to implement the same.

Further Scopes

- Smart Contract automation can also set the amount of times the transaction needs to be done
- Smart Contract includes a way to pay for variable bills according to the usage.
- Not only automation on server side instead of client side but also interactive and easy deployment of the smart contract.
- Removing some rules made during coding of the algorithm can make it even more optimized

Further Scopes

- Adding charts to predict gas fees into transaction insights and the ability to automate contracts on the bases of it.
- Coins like bitcoin and filecoin weren't added as many implementations of them already exist, they can be added and swapping tokens can be implemented.
- History for non-EVM based networks can also be stored in database

DEVELOPER DOCS REFERRED AND FRAMEWORKS USED

1. https://github.com/MetaMask/snaps-monorepo/discussions/675
2. https://docs.metamask.io/guide/snaps.html?utm_source=ethlisbon&utm_medium=event&utm_campaign=2022_Oct_ethlisbon-hackathon-page_awareness_event
3. https://developers.tron.network/docs/tronweb-1
4. https://docs.solana.com/developing/clients/javascript-api
5. https://medium.com/@mithunmk93/algorithm-behind-splitwises-debt-simplification-feature-8ac485e97688

Thank You!!

Open for Questions