ANASTASIA LABS

SMART HANDLES

Project Closeout Report

SMART HANDLES

On-chain framework for using DApps without going through their official frontends

- Customizable with arbitrary logic
- Best suited for dedicated tasks, such as swapping ADA to MIN via Minswap
- Discoverable addresses by storing ADA Handles in instances
- All-round suite for easy product deployment
- Accompanied by a wallet PR as guidance for wallets to follow

WHAT WILL THE EXPERIENCE LOOK LIKE?

1. User sends some ADA to e.g. `$ada-to-min` instance, reserving 1 ADA for the routing agent
2. Routing agent handles the route and collects its fee
3. Minswap batchers send original user proper amount of MIN tokens

OBJECTIVES
&
CHALLENGES

- Create an abstract contract with flexibility
- Accompany the contract with an off-chain SDK
- Complementary CLI generator package

- Implement simple and advanced example instances
- Submit sample transactions on preprod testnet
- Open a PR to an open source wallet

5 PHASES OF DEVELOPMENT

1. CONTRACT & UX DESIGN

- Guaranteed integrity for all instances
- Focus on convenience for both users and router agents

2. ON-CHAIN DEVELOPMENT

- Minimal performance cost by employing Plutarch for the wrapper contract

3. OFF-CHAIN SDK

- Consistent and robust interface between all endpoints

4. CLI AGENT

- Convenient experience for users and router agents via a CLI application generator

5. WALLET INTEGRATION

- PR in an open source wallet (GameChanger) to provide a basic guideline for wallets

OVERVIEW

ON-CHAIN STIPULATIONS

1. Incentive for decentralized agents
2. Guaranteed route destination
3. Support for "single" and "batch" targets
4. Full ownership of funds for users in case of failed routes
5. Flexibility through advanced datums

OFF-CHAIN ENDPOINTS

All endpoints comprise of both single and batch target,
while also supporting simple and advanced cases

1. REQUEST

- Specify assets to be locked at the instance for the simple case
- Specify a multitude of values for the advanced case, such as:
    - Possible owner
    - Router fee
    - Reclaim router fee
    - etc.

2. ROUTE

- Output datum maker functions for simple and advanced cases
- Required mint configurations for the advanced case
- Additional callback for tweaking the transaction for both

3. RECLAIM

- No required values for the simple case
- For advanced reclaims:
  - Output datum maker function
  - Configurations for potential required mint
  - Additional callback for tweaking the transaction

CLI AGENT

Generated applications will offer 3 endpoints

`monitor`

- Primary endpoint for router agents
- Periodic polling an instance's address
- Routing upcoming requests and accruing the fees
- Optional flag for advanced reclaims

`submit-simple`

- Lovelace count to be locked
- Along with any assets

`submit-advanced`

- Lovelace count to be locked
- Along with any assets
- Additional values corresponding directly to the off-chain SDK

KEY ACHIEVEMENTS

- Development of a Wrapper Contract
- Development of a Complete Off-Chain Suite
- Provision of a Sample Reference for Instance Developers

KEY LEARNINGS

- Ups and Downs of Adding Features to Frameworks
- Importance of Emulated Tests
- Importance of Testnet Transactions

NEXT STEP

Support for Datum-less UTxOs

REPOSITORIES

- On-Chain Contract
  - https://github.com/Anastasia-Labs/smart-handles
- Off-Chain SDK
  - https://github.com/Anastasia-Labs/smart-handles-offchain
- CLI Agent
  - https://github.com/Anastasia-Labs/smart-handles-agent