Milestone Report: Off-chain Components for Decentralized Farming (Milestone 3)

Documentation & Testing Report

April 5, 2025

Contents

1 Introduction

This milestone report outlines the completion of **Milestone 3**: "Off-chain Components" for our Cardano Catalyst project focused on decentralized farming solutions. The deliverables for this milestone include:

- Development of comprehensive off-chain infrastructure for managing transactions and interactions with staking contracts, top-pools contracts, and governance-based farms.
- Implementation of a robust backend system comprising blockchain querying, transaction parsing, database management, and a user-friendly API.
- Documentation of the implemented off-chain tools, their usage, and API endpoints.
- Presentation of successful integration between off-chain infrastructure and on-chain components through testnet operations.

This document provides detailed documentation of the developed off-chain tools and summarizes the testing performed to demonstrate functional integration with on-chain smart contracts.

2 Off-chain Components Overview

Our off-chain infrastructure comprises two key components:

a) **Transaction Building Tools:** Python scripts for creating, managing, and submitting transactions related to staking, top-pools contracts, and governance-based farms. These tools align closely with our on-chain components, specifically:

- Staking Contracts
- Top-Pools Contracts
- Governance-Based Farms

b) **Blockchain Querying and API System:** A backend consisting of a chain querier and transaction parser, a structured database, and an API server. This system scans the blockchain, parses relevant transaction data, organizes it efficiently in a database, and exposes it via an accessible API for frontend integration.

The following sections document these components in greater detail, providing guidance on their usage and functionality.

3 Off-chain Tools Documentation

Below is the complete list of implemented off-chain Python scripts, executable via:

```
python3 -m muesliswap_onchain_staking.offchain.<component_name>
```

- **batch_stake_orders**: Processes multiple staking orders simultaneously to enhance transaction efficiency.
- **batch_unstake_orders**: Efficiently handles multiple unstaking requests in a single transaction.
- **cancel_stake_order**: Allows users to cancel previously submitted staking orders before they're processed.
- **create_farm**: Initializes and deploys new farming contracts on-chain.
- **mint_free**: Utility script to mint test tokens on the testnet environment for testing purposes.
- **place_stake_order**: Places a staking order on-chain, recording the staker's intent to lock tokens in exchange for rewards.
- **place_unstake_order**: Records a user's request to unstake previously staked tokens.

- **unstake**: Executes an unstaking transaction, returning the tokens to the user's wallet.
- **update_emission_rate**: Updates the emission rate of rewards in a specific farming contract.
- **top_pools.init**: Initializes the top-pools contract, setting up initial parameters and rankings.
- **top_pools.update_params**: Updates the governance parameters within the top-pools contracts.
- **top_pools.update_ranking**: Refreshes pool rankings based on liquidity, ensuring accurate reward allocation.

These tools enable efficient and reliable interactions with the on-chain smart contracts.

4 API Backend System

The API backend comprises three main components:

4.1 Chain Querier and Transaction Parser

The chain querier continuously scans the Cardano blockchain via Ogmios, identifying transactions relevant to our decentralized farming solution. Parsed transactions extract essential data, which is then stored into the database.

Key features include:

- Handling blockchain rollbacks.
- Robust parsing and error handling to ensure data consistency.
- Integration with Ogmios for reliable real-time blockchain querying.

4.2 Database Structure

The backend utilizes an SQLite database with the following essential data models:

- **Blocks & Transactions**: Stores block and transaction metadata.
- **Transaction Outputs**: Records detailed information about transaction outputs, datums, and involved tokens.
- **Farm States**: Tracks farming parameters, rewards, emission rates, and staking details.
- **Staking Positions**: Records user staking positions, associated cumulative rewards, and staking durations.

The database ensures efficient and structured access to all relevant blockchain data.

4.3 API Server and Endpoints

The API server, built using FastAPI, exposes a straightforward HTTP interface for accessing farming and staking data:

- **GET /api/v1/farms**: Provides comprehensive data about all active farms, including reward tokens, emission rates, and total amounts staked.
- **GET /api/v1/staking/positions**: Retrieves staking positions for a given user wallet, detailing staking amounts, durations, and earned rewards.
- **GET /api/v1/health**: Checks system health and synchronization status with the Cardano blockchain.

This backend API significantly simplifies frontend integration and improves user experience.

5 Testing Report

We conducted comprehensive testing to validate the successful integration and functionality of our off-chain infrastructure with the previously developed on-chain components. All transactions were constructed using the off-chain scripts described above and submitted to the Cardano Pre-production Testnet. Below we detail the individual test scenarios, including transaction descriptions and links for verification.

5.1 Test Scenarios and Results

1. **Farm Creation**
   - **Description:** Deployment and initialization of a new farming contract.
   - **Result:** Successfully created farm with correct initialization parameters encoded in the datum.
   - **Transaction:** e06940f81e0b9a788c3334913875c167bb8b2f1508f278cae970259cd7c33067
2. **Emission Rate Update**
   - **Description:** Adjusted the emission rate of rewards within an existing farm contract.
   - **Result:** Verified the successful update of emission parameters in the contract datum.
   - **Transaction:** 758202b3582552a811a1055be1445e97f6520cc55e858b470b4a231b8d292e7a
3. **Place Stake Order**
   - **Description:** Submitted an individual staking order, recording the user's intent to participate in farming.
   - **Result:** Successfully recorded the stake order with accurate staking parameters.
   - **Transaction:** 0dfee56c3bc9774f1dd17fffffc0b6fdf2d38e28354edfb633756a07a41e8d44
4. **Batch Stake Orders**
   - **Description:** Batched processing of multiple staking orders within a single transaction to improve efficiency.
   - **Result:** Successfully processed and confirmed multiple staking orders simultaneously.
   - **Transaction:** a110b110b803ee05301b961878834ed525afcf60f22767f3df1f74bce1ba09b7
5. **Top Pools Contract Initialization**
   - **Description:** Initialized the top-pools contract, specifying the initial ranked list of pools and associated reward amounts.
   - **Result:** Successfully deployed initial top pools UTxO and verified datum correctness.
   - **Transaction:** 51f670e93809c66511bfb825651b63ccac94456b4daa48abbe1a9f048f2e3cfb
6. **Top Pools Parameter Update**
   - **Description:** Updated rewards for a specific pool in the ranking list, ensuring no unauthorized changes were introduced.
   - **Result:** Successfully validated the update operation and contract constraints.
   - **Transaction:** 635a1ebd46b660b88389c4a0fdf55628ff28474952201365f49518ef4fbadf80
7. **Top Pools Ranking Update**
   - **Description:** Updated pool rankings by swapping positions within the top-pools list.
   - **Result:** Successfully executed ranking swap; verified datum reflects correct list order.
   - **Transaction:** 07a26abd71d2ce51688b889f307442b2973fb2b78760f0cd96db57204155303c
8. **Governance-Based Tally Creation**
   - **Description:** Initiated a governance tally proposing an update to a farm's reward parameters.

- **Result:** Tally successfully created and recorded on-chain, ready for community voting.
- **Transaction:** c20cb35ac518ff580874a1e870fe2635699f1f2217018fe149a0a3132a29b3fd

9. **Governance Stake and Voting**
    - **Description:** Created a governance stake and subsequently cast a vote within the tally.
    - **Result:** Verified successful stake creation and vote inclusion in the governance tally.
    - **Stake Transaction:** a321eb9246044d437a7b199de7e519a02b876fec785db5869975cc58f4292bd0
    - **Vote Transaction:** c6341d632b068ac044769e26e2e35ed9124502ad202c3eb88b9cd9f07529527d

5.2 Observations

- All off-chain tools successfully constructed valid transactions.
- Interactions between off-chain scripts and on-chain contracts behaved as intended.
- Datums and transaction outputs were verified through public blockchain explorers.

6 Conclusion

The successful completion of **Milestone 3** demonstrates a fully functional off-chain infrastructure integrated seamlessly with our decentralized farming smart contracts. This milestone significantly advances our project, enabling effective transaction management, robust backend querying, and a user-friendly API for frontend integration.

Our next steps include:

- Deploying an alpha version for collecting user feedback.
- Integrating necessary adjustments based on internal and community input.
- Officially launching the complete farming solution on MuesliSwap.
- Preparing final deployment documentation, a public closeout report, and video materials for community transparency and engagement.

These upcoming steps ensure our decentralized farming system aligns closely with community expectations, leading to a robust and sustainable farming experience on Cardano.