ANASTASIA LABS

Proof of Achievement - Milestone 5

CML Report

Project Number 1100024
Project Manager Jonathan Rodriguez

ANASTASIA LABS

Contents

Project Name: Lucid Evolution: Redefining Off-Chain Transactions in Cardano
URL: Catalyst Proposal

Evidence Definition

Provided documentation detaling Lucid Evolution aligns seamlessly with the latest advancements in Cardano's CML is available at https://github.com/Anastasia-Labs/lucid-evolution

CML Integration and Lucid Evolution

Context

Lucid Evolution's relationship with Cardano Multiplatform Library (CML) began with addressing fundamental challenges in transaction processing and memory management. Our initial contributions focused on resolving memory leaks and optimizing WASM builds for JavaScript interfaces, as evidenced in our early work with CML's memory management systems documented here.

Current State of Integration

Today, Lucid Evolution maintains a sophisticated fork of CML (anastasia-labs/cardano-multiplatform-lib v6.0.2-2) that extends beyond the current mainline CML version (6.0.1). This advancement isn't about version numbers – it represents our commitment to pushing the boundaries of what's possible with Cardano's off-chain operations.

Our fork incorporates features and optimizations that anticipate future network requirements, as CML repository naturally takes a longer time to make new releases.

Technical Contributions

Our team has been active in implementing updates for the upcoming hardfork. A example is our work on set tag serialization, where we've developed a solution that address both current and future protocol requirements. The implementation of the 258 tag serialization requirement showcases our approach:

```
impl<T: Serialize> Serialize for NonemptySet<T> {
    fn serialize<'se, W: Write>(
        &self,
        serializer: &'se mut Serializer<W>,
        force_canonical: bool,
    ) -> cbor_event::Result<&'se mut Serializer<W>> {
        if let Some(tag_encoding) = &self.tag_encoding {
            serializer.write_tag_sz(258,*tag_encoding)?;
        }
        serializer.write_array_sz(
            self.len_encoding
                .to_len_sz(self.elems.len() as u64, force_canonical),
        )?;
        // Implementation continues...
    }
}
```

We are committed to maintaining backward compatibility while preparing for future protocol changes.

Memory Management and Performance Optimization

Building on our earlier contributions to memory management in CML, we've continued to refine and enhance these systems. Our implementation includes UTXO management that enables efficient chaining of transactions within a single block. These improvements have been important in minimizing memory leaks and enhancing stability, particularly in WASM

Cross-Network Compatibility and Future-Proofing

Our implementation ensures seamless operation across all Cardano network environments. This includes current mainnet protocols, preview/testnet features, and upcoming hardfork requirements. Diagnostic notation of our serialization shows this compatibility:

```
{
    0: 258_1([
        [
h'f9aa3fccb7fe539e471188ccc9ee65514c5961c070b06ca185962484a4813bee',
h'8e782b6a21fbe1361c6220bea7d1327fd8c3d6c1f0d34361193f7a98a5609e6dca
08e10c3dc291b746c26018721c786ecc3753c5318458d7da4cc61ef0f97f05'
        ]
    ])
}
```

Active Development

Our team maintains engagement with the CML development community, as evidenced by our recent contributions to discussions about serialization requirements and protocol updates.

References to our work and further evidence

Repositories

- Cardano Multiplatform Library (CML) Main Repository: github.com/dcSpark/cardano-multiplatform-lib
- Lucid Evolution Repository: github.com/Anastasia-Labs/lucid-evolution
- Our Custom CML Fork: github.com/Anastasia-Labs/cardano-multiplatform-lib

CML / Lucid Evolution Interactions

- Set Tag Serialization Discussion: Issue 364
- Implementation PR: PR 365
- Commit Reference: b1bedaff

Memory Management Improvements

- Memory Leak Resolution: CML PR 182
- Transaction Chaining Implementation: PR 141