

Comparing Layer-1 and Layer-2 Blockchain Protocols in Reputation-Based Networks

07-400 Milestone 3 Report

March 4, 2022

1 Major Changes

One thing that has changed about the project is that all the members of my lab group that are working on implementing ledger prototypes on various blockchain technologies will be integrating their prototypes into a Node SDK, or a client interface for the BoLT application's middleware. This means that I might not be able to integrate my ledger prototype into the backend as quickly as expected. Currently, we are working on defining an API that encompasses all the properties needed to connect with, send requests, and get responses from the four prototype ledgers.

2 What I Have Accomplished Since My Last Meeting

I have tested the changes to my ledger prototype from my last milestone meeting, namely implementing interest rate calculations for minting and transferring and refactoring the creation of program-derived addresses, by writing client-side code. I have also started writing API documentation for my prototype, which we have used to start planning the architecture of the Node SDK.

3 Meeting Milestones

My milestone for this week was to develop a testing framework for my on-chain program, but I did not reach that milestone since my research group's focus has shifted, as outlined in the first section. Instead, I have been working with others to determine how to integrate our ledger prototypes into a common client interface for the application's middleware.

4 Surprises

One surprise that I ran into when I was implementing interest rate calculations was my inability to use Rust's standard libraries to get the system time. Because of the distributed nature of the ledger, programs are locked out from these functionalities. Currently, I am passing timestamps from the middleware to the ledger, but that raises security concerns, such as the fabrication of timestamps. In particular, this setup is not amenable for a common client interface, so this is one important change that I will have to make to my ledger prototype.

5 Looking Ahead

In the next two weeks, I hope to make progress towards the common client interface, starting by revising the parts of my current prototype that do not fit with our current API, such as the timestamp parameter mentioned in the prior section. There are also other modifications that I will need to make. For instance, in order to move constants used in the client interface into my smart contract, I will need to implement cross-program invocation into for account creation into my ledger function that creates a bolt specification.

In order to meet the API specification, I will also need to implement query functions, such as retrieving all bolts held by a user, bolt specifications owned by a user, and information about a specific bolt specification. I do not expect to accomplish this before the next milestone meeting, but I aim to meet all the API specifications by the milestone meeting that comes after the next one.

6 Revisions to Future Milestones

I am revising my next milestones as follows:

- **Milestone 4** (3/15): Revise the ledger prototype to meet the API specifications for existing functions.
- **Milestone 5** (3/29): Add query functions to the ledger prototype to meet the API specifications.
- **Milestone 6** (4/12): Deploy ledger prototypes from testnets to mainnets. Connect all ledger prototypes to the BoLT system backend for testing.
- **Milestone 7** (4/26): Collect and compare measurements for metrics such as throughput, cost, and latency by running experiments against all ledger prototypes on the BoLT network.

7 Resources Needed

No additional resources are needed at this time.