

Cardano-node-emulator: 2026 Update and Maintenance proposal

I. Introduction

Context

[cardano-node-emulator](#) (CNE) is a tool designed to simulate the behaviours of a Cardano node. It is principally used for the validation and testing of smart contracts. CNE has spawned from the old mono-repo [plutus-apps](#) which was put into maintenance mode three years ago, and is now archived since December 2024. This repository was unmaintainable due to its monolithic structure and the loss of relevance of some of its components. However, among those components, a few remained relevant at the time of CNE's creation, or simply appeared in its dependency tree, and thus have been placed within CNE. For example, CNE contains *plutus-script-utils* which provides the notion of typed validators, a notion that used to be almost universally used when defining plutus scripts, and that has slowly fallen into irrelevance, even more so since Conway, where scripts can now effectively be used for several purposes..

CNE is currently used by various companies for testing purposes, and specifically by Modus Create which makes use of the tool in their own open source testing tool, [cooked-validators](#). As it was no longer maintained, Modus Create has taken maintainership of CNE, with two goals in mind:

- Update CNE to the latest version of its Cardano dependencies
- Update content that is no longer relevant for developers or testers of smart contracts

While this first effort and the associated maintenance that occurred during the past year was necessary, this was just the beginning of the road for CNE. CNE will need to be updated to the new upcoming Dijkstra era, and more work needs to be done to keep its components relevant. Additionally, some investment into reaching out to the community should also be performed.

Proposal

Modus proposes to continue its effort in being the principal maintainer of cardano-node-emulator for the upcoming year. This effort will consist of 4 main axes:

- Prepare the release of Dijkstra, study the impact of the changes on CNE, and implement the necessary updates to be ready for the release.
- Keep maintaining the tool, integrating pool requests from the community, engaging with community members on Discord and fixing possible bugs and issues.

- Engage with the community, in a way that makes it visible that CNE can once again be relevant for developers of smart contracts, as it used to be in the past.
- Update heavily outdated parts of CNE. Each individual project will be subject to a relevance study, and updated if applicable.

The motivation behind these 4 elements will be detailed in the next section.

Motivation

The past year has been brighter for CNE. Its core component has been fully updated to the latest version of Cardano packages living in Conway, *plutus-script-utils* has been fully revamped to replace typed validators with multi-purpose scripts, and some community engagement has been fostered through a dedicated Discord channel, and a community meeting every other month. It is, however, just the beginning.

CNE still suffers from various shortcomings, and requires a significant amount of work, with respect to the following 4 categories, corresponding to the 4 aspects of our proposal.

The upcoming **Arrival of Dijkstra** with the new version of Plutus is being prepared throughout the Cardano ecosystem, and CNE should follow this path. Being up to date is key to being relevant, and a necessity if CNE can start being used and loved again.

Issues, pull requests, and bugs should be addressed as they appear. Should the engagement from the community increase, then contributions and reports will increase as well, and time needs to be allocated to that purpose. As in all open source projects, fostering the link between the developers and the external contributors of a tool is of the most importance.

Speaking about **the community**, there is some effort to be put into having the Cardano community be aware of the resurgence of CNE. Most people have either never heard of CNE, are unaware that it now contains some of the tools they used to use, or have completely forgotten about it, and for good reasons ! It is time to market that CNE is being maintained, and still has a lot to offer to developers or testers of smart contracts. An example, which we intend to replicate to other areas in the tool, is the creation of an abstract for multi-purpose scripts, replacing the old typed validators, which can be efficiently used to write Plutus smart contract, but is yet to be made visible as such.

Finally, some **old code** needs revamping. CNE is born from a subset of projects that used to live within plutus-apps. These projects were at various stages of development, some of them being quite mature, while others remained in early drafts. This is the case of plutus-ledger and cardano-node-socket-emulator, which both need some serious refactoring. Overall, the whole codebase should be scrutinised in an attempt at modernizing it.

Scope

This proposal pertains to the code and ancillary documents hosted in the following repository: <https://github.com/IntersectMBO/cardano-node-emulator>.

Staffing

We propose one maintainer, staffed 20% throughout the year.

Mathieu Montin is responsible for the smart contract auditing activity within Modus create, and was the leader of the former high assurance software team within Tweag. He has significant experience with Plutus and Cardano. He is the main developer and maintainer of [cooked-validators](#), which itself uses CNE to assist in auditing Plutus contracts.

II. Detailed update plan

The CNE repository contains various projects, each of which does not require the same level of update work. Here is a brief overview of the specific needs for each of them. Note that not all of the below will likely be doable in a year, but we will pick among those ideas, and make priorities according to the needs of the community.

freer-extra is a small library that provides extra tools around freer monads. This effect library is used by CNE at a few places where effects are relevant. The main question to ask is why this extension was developed in the first place, and if it could be replaced by existing similar libraries, as plenty exist. Overall, we will pinpoint the locations in CNE where this library is used to assess whether it should be replaced, kept, or removed altogether. If these utilities are relevant, maybe they could be contributed to the actual freer library instead of living in CNE.

cardano-node-socket-emulator is an interface library to allow using the emulator instead of the real node in a similar way, using a socket. It misses a lot of features and is overall in a stand-by state. We will not make a priority to update this project in particular, but we see no reason to remove it from the repository. Thus, we will keep updating it through the various Cardano dependencies updates.

plutus-script-utils is the project that provides utilities for writing scripts. It used to be the host for typed validators, and in the last year has been updated to provide consistent utilities for multi-purpose scripts. There are two things we will address: whether it makes sense to keep the old utilities available in their own package, as we currently do, and whether it should stand alone in its own repository, to be given more visibility. We will investigate the use of template haskell there to facilitate the building of scripts from their separate purpose logics.

plutus-ledger contains legacy code mostly, with helpers to make a bridge between the Plutus and Cardano apis. This is the project that needs the most attention in terms of updates. The codebase is not clean, not consistent, and very much outdated. However, the idea behind the library is sound and should be kept alive. We will investigate what needs to be kept, and what needs to go, and implement the changes accordingly.

cardano-node-emulator is the main component of the repository, which consists of the emulator itself. In the past year, some new features have been added to manually tamper with the internal emulated ledger state that is maintained within the emulator. This is to be continued and updated, as the actual Ledger internal state has been revamped and will thus change when entering Dijkstra. We will also make a clearer separation between function interfacing the emulator with the ledger, and monadic operations living in the emulator itself. On a higher level, we will investigate what features such an emulator should exhibit, against other similar tools such as cooked-validators.

III. Regular maintenance

Maintenance responsibilities

After the initial phase of update, we propose to take over principal maintenance of the above CNE repository. In particular, this would extend to the following activities:

1. Maintaining the backlog and roadmap

- a. Develop and publicise a medium-term roadmap for the development of the cardano-node emulator, based upon the results of planning, community discussion and tracked issues.
- b. Based upon this roadmap, we will maintain a short-term backlog of work items.

2. Reviewing and Merging PRs

- a. Monitor the active PR lists and shepherd them to completion.
- b. Assist external contributors in making contributions.

3. Stakeholder and Community Engagement

- a. Identify and maintain the list of key stakeholders for the project
- b. Create channels for contributors to engage with developing the backlog and priorities for the project.
- c. Identify work items particularly suitable for work on by newcomers and external contributors.
- d. Regularly inform stakeholders about the state of the project.

4. Establishing clear project governance

- a. Establish project governance structures according to the Intersect governance framework

(<https://intersect.gitbook.io/open-source-committee/policies/governance>)

- b. Establish project documentation as per the Intersect documentation framework (<https://intersect.gitbook.io/open-source-committee/policies/documentation>)

5. Perform regular maintenance

- a. Work on the backlog items as identified in (1)
- b. Keep the project up-to-date with dependencies - in particular, with the current iteration and era of cardano-node.
- c. Maintain a CHANGELOG of all major changes to the codebase.

Communication Channels

As part of the maintainer role, the team will engage and communicate with stakeholders and the community in order to guide priorities and work. This communication will be held on the dedicated slack channel and other monthly meetings.

Contract Basis

Time

We propose a fixed term contract of one year aligned with the Intersect budget process, on an amortised basis of 4 engineer days per month. We believe that this time would be sufficient to carry out the regular maintenance responsibilities required for the two mentioned repositories.

We expect that the amount of time needed will vary from month to month and depending on upstream development. We therefore might expect to spend more time as new eras are implemented and correspondingly less during the intra-era period.

Price

We propose an all-inclusive price of \$97,875.00 USD to cover ongoing support for 12 months. This pricing is based on our long-term partnership and past engagements with Intersect, ensuring we provide the best possible support.

Breakdown:

- Initial Phase: 20 engineer days dedicated to project setup and initial development.
- Ongoing Support: Continuous development, updates, and troubleshooting over the next 12 months.

This comprehensive pricing reflects our commitment to delivering exceptional value and sustained collaboration.