ANASTASIA LABS

Closeout Report

Project: The Trifecta of Data Structures
URL: Catalyst Proposal
Number: 1000013
Start Date: 2023-10-08
End Date: 2024-03-31

Contents

List of KPIs

Challenge KPIs

1. **Enhancing Scalability of Cardano:** Given the inherent constraints of the blockchain, such as the ledger rules and transaction size limitations, we challenged ourselves to build advanced data structures (Merkle trees, Tries, and Linked Lists). These structures enable larger and more complex smart contract applications, which are significant to enhancing the scalability of DApps on Cardano.
2. **Addressing Insufficient onchain data structures hinders scalability of Cardano:** Recognising the absence of shared design patterns and limited availability of practical examples for scaling solutions in the Cardano community. We implemented advanced data structures (Merkle trees, Tries, and Linked Lists in Aiken and Plutarch) and provided comprehensive, well structured and easy to understand documentation and tutorials, thereby enriching the educational resources available to developers whilst providing efficient and scalable data structures for Cardano smart contracts as well as serving as a valuable resource for developers looking to understand and implement these data structures in their own projects.
3. **Ensuring Code Quality and Production-Ready Resources:** The project upheld high standards of code quality, adherence to best practices and readiness for production in the implementation of these structures through thorough code reviews and unit tests, to ensure reliability and efficiency for the developer community.

Project KPIs

List of project KPIs and how the project addressed them:

1. **Provide generic and production-ready implementations:** The team developed robust, optimized, and well-tested implementations of Merkle trees, Tries, and Linked Lists in both Aiken and Plutarch, providing functional and efficient data structures for real-world smart contract applications.
2. **Ensure robustness through thorough Testing:** The project ensured the reliability of the data structures through comprehensive code review and unit testing, validating their correctness and efficiency.
3. **Make the project fully open-source:** All developed code, documentation, tutorials and examples of validator scripts to validate and showcase these library implementa-

tions have been made publicly available under an MIT license and can be found in the Main Github Repo.

Key achievements

- **Implementation of Advanced Data Structures:** We successfully developed and implemented advanced data structures (Merkle trees, Tries, and Linked Lists) in Aiken and Plutarch. These structures enhance the scalability of DApps on Cardano by allowing for more efficient use of the limited 16kb transaction size and the single UTXO model.
- **Extensive documentation and tutorials:** We are proud to have enriched the Cardano community with educational resources by providing comprehensive documentation and tutorials to help developers understand and utilize these data structures in their projects.

Key learnings:

We grew a lot in experience implementing Merkle Trees, Linked Lists and Tries in Plutarch and Aiken in various ways;

- **Efficient Data Integrity with Merkle Trees:** We gained a deep understanding of constructing and verifying Merkle proofs, which enhanced our ability to implement secure and efficient data validation processes.
- **Flexibility and Efficiency with Linked Lists:** Leveraging Plutarch's functional programming paradigm and Aiken's robust type system to create Linked Lists that are both efficient and easy to maintain, highlighted the importance of designing for efficient insertion and deletion operations.
- **Managing Mutable Data with Tries:** learned to optimize storage by sharing common prefixes and implementing fast update and lookup operations.
- **Leveraging Plutarch and Aiken's Strengths:** We got to appreciate Plutarch's composability and functional programming features, which facilitated the construction of intricate data structures. Aiken's expressive syntax and strong typing system contributed to writing clear and maintainable code, ensuring that smart contracts are both robust and efficient.

Next steps

Continuous Improvement: Based on the feedback and insights from the community, we plan to continue refining and optimizing our implementations of Merkle trees, Tries, and Linked Lists in both Aiken and Plutarch.

Community Engagement and Collaboration: We intend to monitor various metrics on GitHub as indicators of community interest and adoption such as tracking the number of stars, forks, pull requests and contributions in the repository. Feedbacks, discussions, and questions on the project's GitHub repository will also be monitored to assess the level of community engagement with the project.

Final thoughts/comments:

We are incredibly proud to add open source libraries to the Cardano communities. Our implementations of advanced data structures in Aiken and Plutarch not only enhance the scalability of DApps on Cardano but also serve as a valuable resource for the developer community. We look forward to seeing how these tools are used and adapted in future Cardano projects.

Resources

Links to other relevant project sources or documents:

Project	Plutarch	Aiken
Main Github Repo	Merkle trees	Merkle trees
Catalyst Proposal	Linked Lists	Linked Lists
	Tries	Tries

Closeout Video

link11_placeholder