ANASTASIA LABS

Closout Report Video Script

Project Number	1000013
Project manager	Jonathan Rodriguez
Date Started	2023-10-08
Date Completed	2024-03-31

Contents

“The Trifecta of Data Structures: Merkle Trees, Tries, and Linked Lists for Cutting-Edge Contracts”

(Estimated Time assumes 120 - 150 words per minute)

Introduction

(Estimated Time: 45 seconds)

Hello, Cardano community! I’m Mladen Lamesevic from Anastasia Labs.

I’m excited to present the closeout report for our project, “The Trifecta of Data Structures: Merkle Trees, Tries, and Linked Lists for Cutting-Edge Contracts.”

In this presentation, I’ll cover why we embarked on this project, how we executed it, and the outcomes we achieved. Let’s dive in!

Project Context and Importance

(Estimated Time: 1 minute 30 seconds)

First, a bit of context.

We entered the Dev Tools, API, or Library Challenge because the Cardano ecosystem faces scalability challenges due to constraints like the single UTXO model and the 16kb transaction size limit.

Additionally, there is a noticeable absence of shared design patterns and limited practical examples and educational resources for scaling solutions in the Cardano community.

To address these challenges, we focused on implementing robust on-chain data structures —Merkle Trees, Tries, and Linked Lists— in two of Cardano’s scripting languages, Aiken and Plutarch.

You might be asking yourself why we chose to go with Aiken and Plutarch as opposed to the other Cardano languages, well…

Aiken, is particularly known for its expressive syntax, strong typing system and is rapidly gaining community adoption whereas Plutarch is known for its ability to manipulate outcomes at a lower level (Close to UPLC), which enables more control over the execution of smart contracts, leading to more efficient and compact scripts.

These structures are essential for enhancing the scalability and efficiency of DApps on Cardano and provide efficient solutions within the blockchain’s constraints.

Project Objectives

(Estimated Time: 30 seconds)

Our primary objectives were:

- To design and develop efficient data structures for the Cardano blockchain.
- To provide detailed, easy-to-follow documentation to help developers implement these structures.
- To ensure these data structures are thoroughly tested and production-ready.

Execution and Milestones

(Estimated Time: 1 minute 30 seconds)

We divided our project into three phases: design and development, testing, and documentation.

Phase 1: Design & Development

We successfully designed and developed the following data structures in both Plutarch and Aiken, each with its unique purpose and benefits:

- **Merkle Tree:** The Merkle Tree is valuable in proving the presence of arbitrary data within the tree structure. By carrying only the root hash in the script, an efficient and space-saving proof can be generated, ensuring data integrity and validity.
- **Trie (Stick Breaking Set):** Developed with comprehensive functionality for Aiken and Plutarch, the Trie is particularly useful in facilitating mutable data storage in scripts by leveraging the sharing of common prefixes. This approach optimizes storage efficiency and enables more extensive data manipulation within the constrained on-chain environment.
- **Linked List:** Created to enhance scalability and throughput in smart contracts, the Linked List leverages the EUTXO model to significantly enhance scalability and throughput. By linking multiple UTXOs together through a series of minting policies and validators, it improves the user experience interacting with smart contracts concurrently.

We approached the challenge by doing research on data structures and developing those that could have the biggest impact. With this outlined approach of the execution of these technical solutions, we received a total funding of ₳238,374. To date, we have received ₳105,000, with ₳133,374 remaining to be distributed upon the completion of all milestones.

Each design pattern was tagged with a release version, enabling the community to track progress and updates.

Phase 2: Thorough Testing

(Estimated Time: 30 seconds)

For the successful execution of these technical solutions, we received a total funding of ₳238,374. To date, we have received ₳105,000, with ₳133,374 remaining to be distributed upon the completion of all milestones. We conducted rigorous testing for all three data structures to ensure their robustness and efficiency.

Each data structure was subjected to unit tests, with results integrated into a CI/CD pipeline. We ensured that all design patterns passed the unit testing phase successfully.

Phase 3: Comprehensive Documentation

(Estimated Time: 1 minute 30 seconds)

We focused on creating comprehensive, user-friendly, and visually engaging documentation for each data structure. Our aim was to ensure that developers of varying skill levels could understand, implement, and leverage these data structures effectively.

- **Merkle Tree:** The Merkle Tree documentation is a treasure trove of information, providing a deep dive into the concepts, implementation steps, and practical examples. It includes interactive diagrams that visually explain the structure and operation of the Merkle Tree, along with code snippets and sample usage scenarios. This comprehensive guide empowers developers to understand and apply the Merkle Tree in their projects effectively.
- **Trie:** The Trie documentation serves as a comprehensive guide, offering step-by-step instructions that demystify this complex data structure. It features detailed diagrams illustrating the Trie's structure and functionality, and provides hands-on tutorials with sample code. This documentation is designed to help developers understand its practical implementation and unlock the full potential of Tries in their applications.
- **Linked List:** The Linked List documentation goes beyond the basics, offering detailed tutorials and scenarios that showcase its application in smart contracts. It presents clear diagrams to explain the Linked List structure, and provides code examples and use cases. This documentation demonstrates how the Linked List can be used to enhance scalability and throughput in smart contracts, providing developers with the tools they need to create more efficient and powerful applications.

All documentation is well-organized, well-structured, and accessible to developers of varying skill levels and can be found in our Main Github Repo.

Achievements and Outcomes

(Estimated Time: 1 minute)

We are thrilled to share the remarkable outcomes of our project, which we are hopeful will significantly contribute to the Cardano ecosystem:

- **Implementation of Advanced Data Structures:** Our team has pioneered the development and implementation of 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.
- **Quality Assurance:** Through rigorous code reviews and comprehensive unit tests, we have ensured that our implementations are not only reliable but also efficient, setting a benchmark for future developments in the community.

Key Learnings and Challenges

(Estimated Time: 1 minute)

Throughout the project, we encountered several challenges and gained valuable insights:

- **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:** We learned to optimize storage by sharing common prefixes and implementing fast update and lookup operations.
- **Further Experience with Plutarch and Aiken:** We got to appreciate Plutarch's ability to provide more efficient and precise control over data structures, facilitating the construction of intricate data structures and optimizing the performance of smart contracts on the Cardano blockchain.

Aiken's expressive syntax and strong typing system also contributed to writing more clear and maintainable.

Demonstration of Project Outputs

(Estimated Time: 2 minutes)

Let me demonstrate the project outputs and show you how they work:

Merkle Tree: We developed the Merkle Tree structure in both Aiken and Plutarch. The Merkle Tree is used to prove the presence of arbitrary data within the tree structure. For instance, by carrying only the root hash in the script, an efficient and space-saving proof can be generated. This ensures data integrity and validity, which is crucial for secure and efficient data validation processes. You can find the detailed implementation and usage examples in our GitHub repository.

Trie (Stick Breaking Set): The Trie data structure we implemented helps in managing mutable data by sharing common prefixes, thus optimizing storage. This structure is particularly useful for fast update and lookup operations, essential for complex DApp functionalities. We have included comprehensive tutorials and interactive diagrams in our documentation to help developers understand and utilize this structure effectively.

Linked List: We created a Linked List that leverages the EUTXO model to significantly enhance scalability and throughput in smart contracts. By linking multiple UTXOs together through a series of minting policies and validators, it improves the user experience by allowing concurrent interactions with smart contracts. Our documentation provides detailed tutorials, scenarios, and code examples demonstrating the practical applications of Linked Lists in smart contracts.

Future Prospects and Community Impact

(Estimated Time: 1 minute)

These data structures hold immense promise for Cardano's ecosystem.

As Cardano continues to grow and evolve, developers will seek efficient solutions to overcome scalability challenges. Our data structures provide precisely that—scalability within the blockchain's constraints.

Imagine DApps that can handle more users, transactions, and complex smart contracts seamlessly. We anticipate early adopters will soon recognize their value, paving the way for broader adoption.

- **Sparking Interest and Collaboration:** We've already generated interest and discussions within the community. By inviting developers to collaborate, participate in workshops, webinars, and hackathons we foster a vibrant ecosystem and accelerate adoption.
- **Continuous Improvement:** Feedback from the community ensures ongoing enhancements, making our data structures even stronger and more reliable.
- **Comprehensive Documentation:** We are committed to improving our documentation with tutorials, real-world examples, and troubleshooting guides, empowering developers of all skill levels.
- **Long-Term Commitment:** We will keep our libraries up-to-date with the ever-evolving Cardano ecosystem, ensuring our data structures remain relevant. By maintaining the open-source approach and transparent development process, we believe will inspire confidence and trust within the community.

Conclusion:

(Estimated Time: 30 seconds)

In conclusion, our project has successfully bridged a critical gap in the Cardano ecosystem by providing essential data structures and comprehensive documentation. We believe these tools will significantly enhance the scalability and functionality of Cardano-based applications.

Thank you for your support and interest in our work. For more information, visit our GitHub repository at https://github.com/Anastasia-Labs/data-structures.

If you want to know more about Anastasia Labs or contact us, you can visit

- Our website at https://anastasialabs.com/
- Follow us on twitter at https://x.com/AnastasiaLabs
- Join our discord community: https://discord.com/invite/8TYSgwthVy

See you next time!

Goodbye!

(Total Estimated Time: 8 minutes)

Verification Against Criteria:

1. Which challenge did you enter and why?

   Met: Described in "Project Context and Importance" with reasons for choosing the challenge.

2. What was the approach you submitted in your proposal application which was eventually funded?

   Partially Met: Discussed the focus on implementing robust on-chain data structures but lacks specific mention of the approach submitted in the proposal.

3. Please explain any particular technical solutions you proposed and the amount of funding you received.

   Met: Explained technical solutions (Merkle Trees, Tries, Linked Lists) as well as the amount of funding received.

4. Once your project went live, how did you progress? What learnings and challenges did you find along the way? Were you able to keep the project within the scope of your application? What milestones/KPIs did you set at the outset, and did you achieve them? Importantly, what were the fund challenges, and did you achieve them? What did you not achieve? What other major achievements were there?

   Partially Met: Discussed progress and outcomes, but lacks detailed learnings, specific challenges, and KPI achievements. No mention of fund challenges.

5. Please include a demonstration of the project outputs, such as the software or dapp product or service, and how it works. If the project is non-technical, please provide evidence of the deliverables and work that you have completed.

   Met: Provided links to GitHub repositories for each data structure, demonstrating project outputs.

6. What next for the product/service you have developed? If appropriate, please tell us about any commercialization/exploitation plans you have now that your funded project is complete, particularly whether you are seeking further funding and what the funding would be used for.

   Met: Discussed future prospects, ongoing improvements, and long-term commitment to the project. No specific mention of commercialization or further funding plans.

Recommendations for Full Compliance:

- Approach Submitted in Proposal: Include a brief summary of the approach you submitted in your proposal and how it was eventually funded.
- Funding Amount: Mention the amount of funding received for better transparency.
- Learnings and Challenges: Add details about specific learnings, challenges faced during the project, and how they were addressed.
- Milestones and KPIs: Specify the milestones/KPIs set at the outset and discuss if they were achieved.
- Fund Challenges: Include any challenges related to funding and how they were overcome