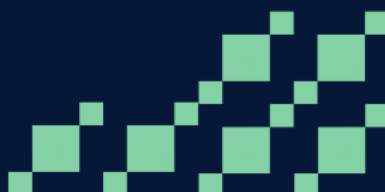
Amaru

CARDANO BLOCKCHAIN ECOSYSTEM BUDGET DETAILED PROPOSAL

06 MAY 2025



Contents

Tables

Figures

Vision

Amaru is an open-source project that aims to build a new fully interoperable block-producing node for improving the overall accessibility and robustness of the Cardano network without compromising on its safety and security. Amaru provides another perspective and solution for stake pool operators and developers alike, prioritizing a modular approach and a seamless user experience. The project is implemented mainly in Rust and aims to attract new contributors to the core maintenance of the ecosystem.

This proposal is designed to breakdown the objectives, scopes, timelines and costs pertaining to Amaru. It delivers the full vision behind the project and open the discussion regarding a Cardano treasury withdrawal to sustain its development.

Why build another node?

One of Cardano's key asset comes from its legendary robustness and stability, which is mainly attributed to the existing Haskell node that has been working without significant disruption for about 5 years now (counting since Byron-reboot, circa February 2020).
So why bother with another node? The answer is, of course, plural:

1. **Better resource usage**

   A software is ultimately a collection of design decisions and trade-offs. Some of those trade-offs in the Haskell node have led to rather high resource usages (e.g. resident memory) thus leading to the creation of specific protocol parameters (e.g. min-utxo-value) as counter-measures to ensure resource usage remains somewhat bounded. Our analysis is that different (albeit acceptable) trade-offs would yield better resources usage.

2. **Industry-grade observability**

   Observability is hard, and it is even harder when rolling your own solution. With Amaru, we intend to embrace industry standards when it comes to monitoring, tracing & logging to drastically enhance the user experience for operators seeking insights in the behavior of their node.

3. **Enhanced interoperability**

   Amaru being primarily written in Rust means that we can more easily leverage an existing and thriving ecosystem of blockchain solutions, and aim at multiple target platforms (e.g. web-assembly, RISC-V). This opens up not only to a wide variety of potential contributors, but also to many new use-cases that are much harder (not to say unrealistic) to achieve with a Haskell node.

4. **Higher-assurance & robustness**

   Developing a new implementation of existing software means dissecting every part of the original implementation. Along the way, bugs and discrepancies with the specification are commonly found through conformance testing. Gaps in the documentation can also be identified and addressed. The development of Amaru has already proven this on several occasions. In the long run, more node implementations can be a synonym for reduced development time by allowing all implementations to converge faster towards better designs and more robust implementations.

5. **Decentralization**

   In a world where Byzantine fault-tolerance is at the centre and where decentralisation is a paramount value, we believe that a single node implementation present a single-point-of-failure and a central point of control. A new node not only increases the overall resilience of the network, but also provides perspectives in terms of roadmap and use-cases.

With Amaru, we believe that we can reconcile with those goals, while preserving Cardano's original focus on security and robustness. Like Haskell, Rust is a statically typed language that provides strong compile-time guarantees as well fine-grained memory management. Its ecosystem is also well-equipped for high-assurance; all-in-all making it a sane choice for implementing blockchain solutions.

Executive summary

Given the length of our proposal, we first provide a high-level overview of the project. The second part of the document goes into more details about the specifics of the Amaru project.

Disclaimer: proposal assumptions and bias

We consider that the current ask is a significant amount of money for a project proposal. We are also well-aware of the challenge (and therefore risks) that the project represents. However:

- we don't start from scratch: the Haskell node is an excellent basis which has already solved numerous challenges. It also provides a solid baseline for performance and reliability.
- we have gathered our best people around the project: we are all actively contributing to the Cardano ecosystem for many years and are recognized leaders in our respective fields.
- lean environment: Amaru isn't our only project, so we need ways of working that are efficient and lightweight.
- we are builders: we deliver in small steps, showcased in demos. Our main goal is to ship a working software.
- we breathe open-source: we seek constructive feedback and actively engage with contributors who are willing to help us in our endeavour.

In addition, the sizing of this proposal stems from a few assumptions:

- There's already ongoing work for most of the scopes, but rarely as a primary priority for people involved. The estimation presented in this proposal assumes a dedicated team;
- This proposal spans over a year, throughout which there will be several deliveries of Amaru. We aim to deliver a capable block-producing node by end of 2025, with several useful increments in-between.
- Given the timeline of the governance actions that are linked to this proposal, everything that is mentioned in this document is an estimate of 6 months of work that starts from June 2025 (our realistic timeline to get the treasury withdrawal active); The milestones described in Q1 and Q2 2025 are not taken into account in the FTE asked by each scope.
- All the scopes described here are estimated in FTE (Full Time Employee) for the sake of evaluating the cost of our project we took the assumption that one senior Software Engineer with expertise in Rust and DLT (Distributed Ledger Technology) is valued at

$200k (remote profile working from the US, recruitment, onboarding and extra costs related to working -- i.e paid vacations, sick leave -- are considered included in the $200k figure)
- We will secure the treasury withdrawal with stablecoins whenever possible to keep the value until final payment to contributors is done (main assumption: withdraw everything but the Contingency in stablecoins)
- Optimism bias: we are aware that while ongoing the forecast and constructing our proposal we are subject to underestimating complexity, overlooking challenges and undervaluing time and cost required to deliver: we will add a 25% contingency fund

One final assumption for our proposal: we expect to be able to publish the info action on the 8th of May and go through with the treasury withdrawal on the 9th of June. Everything mentioned as Q1 or Q2 2025 will be already delivered by the time this proposal is voted upon and is not considered in the estimates of each scope.

Comparison to similar projects

RETH

Project	**RETH**
Description	A high-performance, modular, and open-source Ethereum execution layer client developed in Rust to enhance speed, efficiency, and client diversity for the Ethereum network
Timeline	2 years of development before mainnet
Dedicated team	8 core developers; 90 contributors
Roughly estimated overall cost	>3M$

Table 1: RETH project estimates

Firedancer

Project	**Firedancer**
Description	A high-performance, C++-based validator client for Solana, to enhance scalability, decentralization, and network efficiency by optimizing transaction throughput and reducing system bottlenecks
Timeline	2 years of development (still ongoing) mainnet targetted for 2025
Dedicated team	no public figures available
Roughly estimated overall costs	>5M$

Table 2: Firedancer project estimates

Cardano-node

Project	**Cardano-node**
Description	The core implementation of Cardano's blockchain, written in Haskell, enabling decentralized transaction processing, consensus mechanisms, and smart contract execution within the Cardano ecosystem
Timeline	Started with the Byron release on 2017, still ongoing
Dedicated team	no public figures available
Roughly estimated overall costs	>20M$

Table 3: Haskell Cardano-node project estimates

Note: As our project is closer to RETH in its setup and environment this will be our target when coming down to the final estimate comparison.

Estimates and contributions

Overall sizing of the project

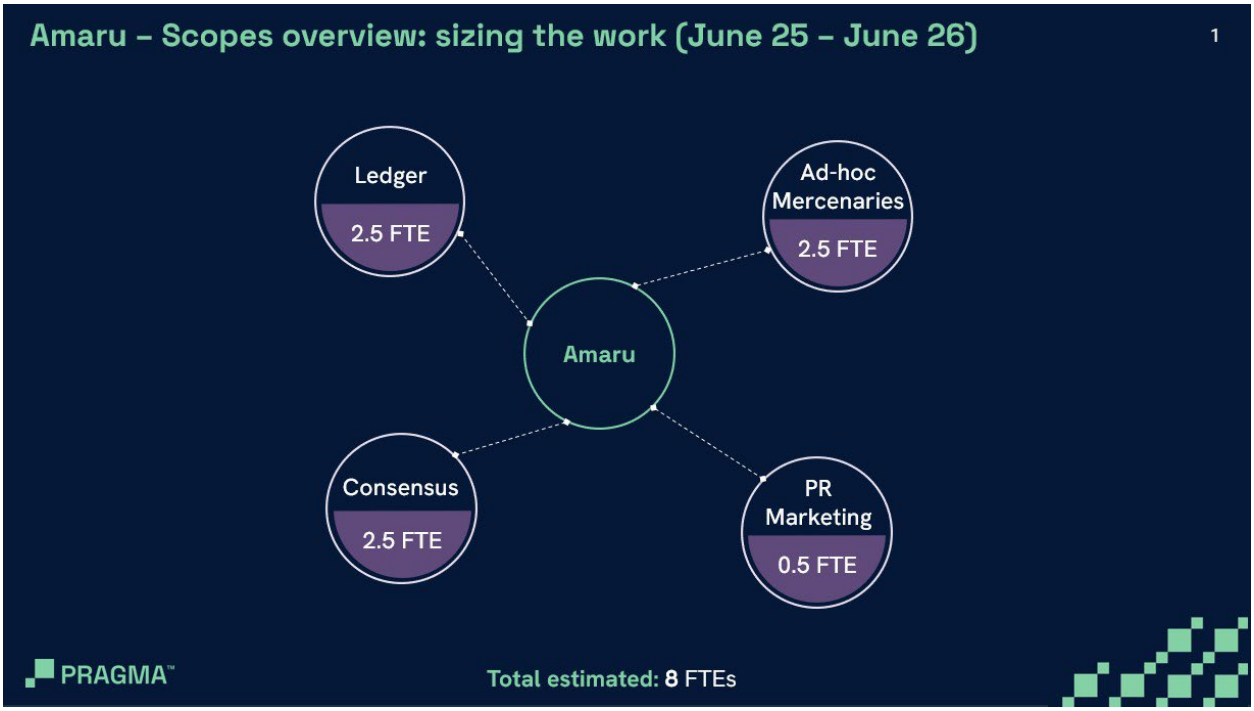


Figure 1: Total sizing of the Amaru project

You will find here the result of the estimation:

Scope	Resource estimated
Ledger	2.5 FTEs
Consensus	2.5 FTEs
Ad-hoc Mercenaries	2.5 FTEs
Project Management, Public Relations & Marketing	0.5 FTE

Table 4: Total sizing of the Amaru project (scope overview)

Total size estimated: **8 FTEs**

Already secured resources allocated to the project

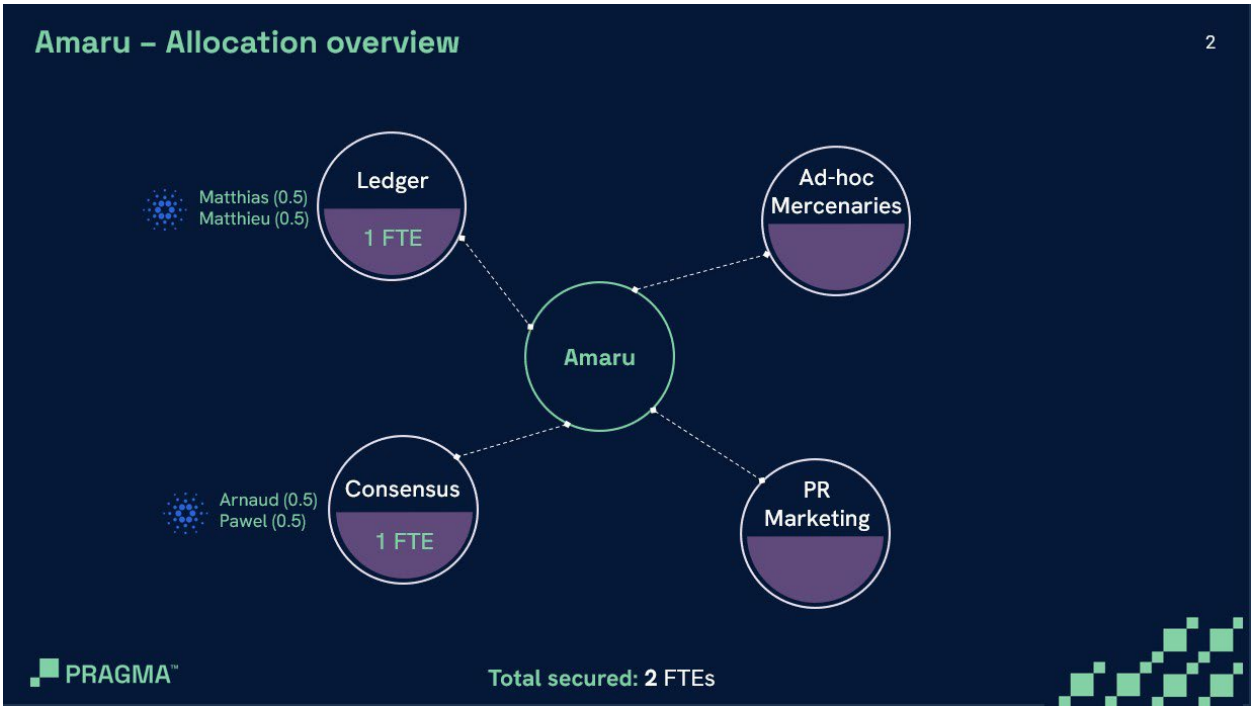


Figure 2: List of resources already allocated to the project

Here are the already funded that have resources allocated or a funding secured:

Scope	Resources already secured
Ledger	1 FTE
Consensus	1 FTE
Ad-hoc Mercenaries	0 FTEs
Project Management, Public Relations & Marketing	0 FTEs

Table 5: List of resources already allocated to the project (scope overview)

Total of the resources already available or funded: **2 FTEs** from people that are employees of the Cardano Foundation.

> *Note: The scope owners will also dedicate time managing their activity on the project but this has not been reflected in the above FTE evaluation.*

Treasury ask breakdown

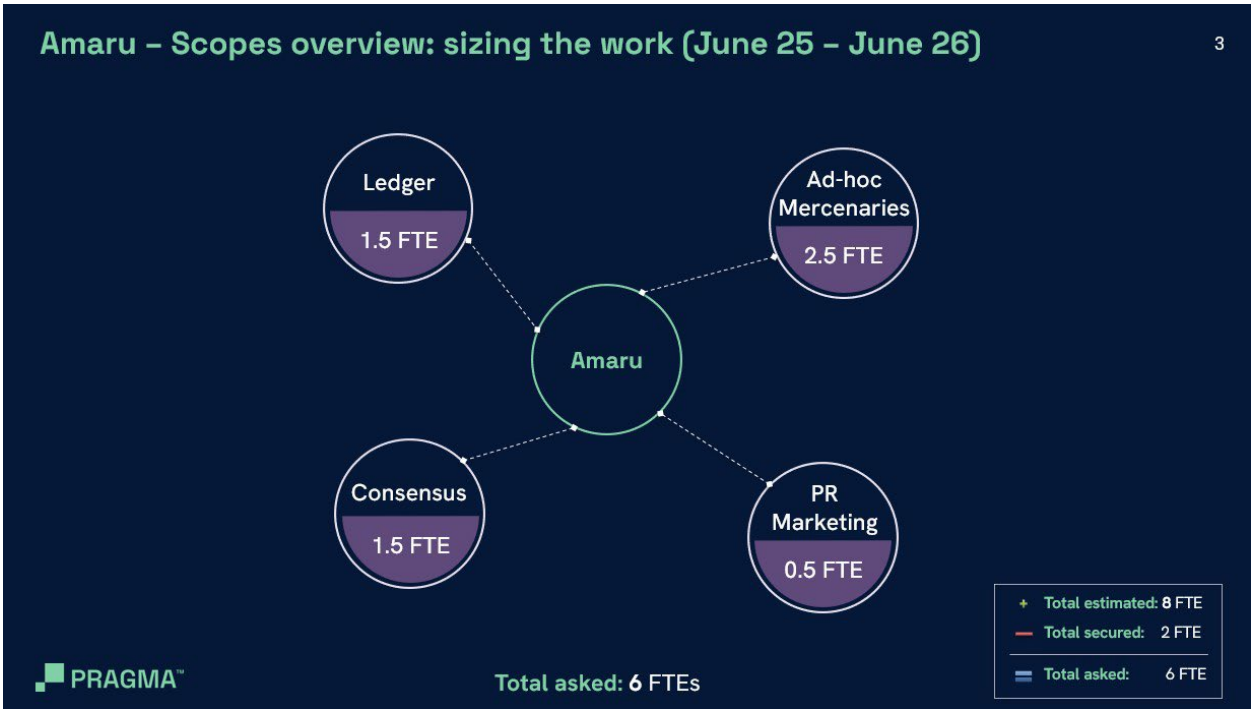


Figure 3: List of resources asked from the treasury (by scope)

Here are the treasury ask for each scope:

Scope	Estimated effort	Resources already secured	Resources asked
Ledger	2.5 FTEs	1 FTE	1.5 FTEs
Consensus	2.5 FTEs	1 FTE	1.5 FTEs
Ad-hoc Mercenaries	2.5 FTEs	0 FTEs	2.5 FTEs
Public Relations & Marketing	0.5 FTEs	0 FTEs	0.5 FTEs

Table 6: Recap of effort required, resources allocated and resources asked (by scope)

Total of the asked for our proposal: **6 FTEs**.

Given our assumptions the yearly valuation of this proposal based on a $200k comes down to:

⇒ 6 FTEs * $200k = $1200k

With the official Net Change Limit being active for 2025 only, the final amount considered for our proposal is:

⇒ 6 FTEs * 200k / 2 = *600k

Given our assumption regarding our optimism bias (*add 25% of contingency*), here are the final figure for each scope and the contingency amount that will be available:

Scope	Resources asked	Scope owner
Ledger	$150k	Matthias Benkort
Consensus	$150k	Arnaud Bailly
Ad-hoc Mercenaries	$250k	Pi Lanningham
Project Management, Public Relations & Marketing	$50k	Damien Czapla
Contingency	$150k	Amaru Maintainer Committee (AMC)
---	---	---
TOTAL	**$750k**	

Table 7: List of owners and amounts in US Dollars for each scope

The proposal submitted will be targeting: **$750k** as a result of the scopes covered in Amaru.

Below in this document you will find details on:

- The framework, methodology used
- The result of the first Design phase and the scope split
- The principles shared on managing Amaru
- Deep dives on scopes, timelines, estimations
- Already delivered scope overview

Sizing and managing the project

Amaru development framework: turning vision into achievable goals

Our main challenge as a team was to put into perspective what it means to build a Cardano node and dissecting this meaning into building components that are bringing the modular aspect of our product to life.

We looked into many methodologies and frameworks, and none of them really fit with both the vision and the environment of the project.

What we designed is a customised framework that has 2 major phases:

1. **Design:** Formalising the main problems we are solving and building the architecture of the product and allocating the scopes behind the vision of Amaru (modularity, performance)
2. **Drive and Deliver:** A basic development cycle that is driven by demos and interfaces between each part of the architecture

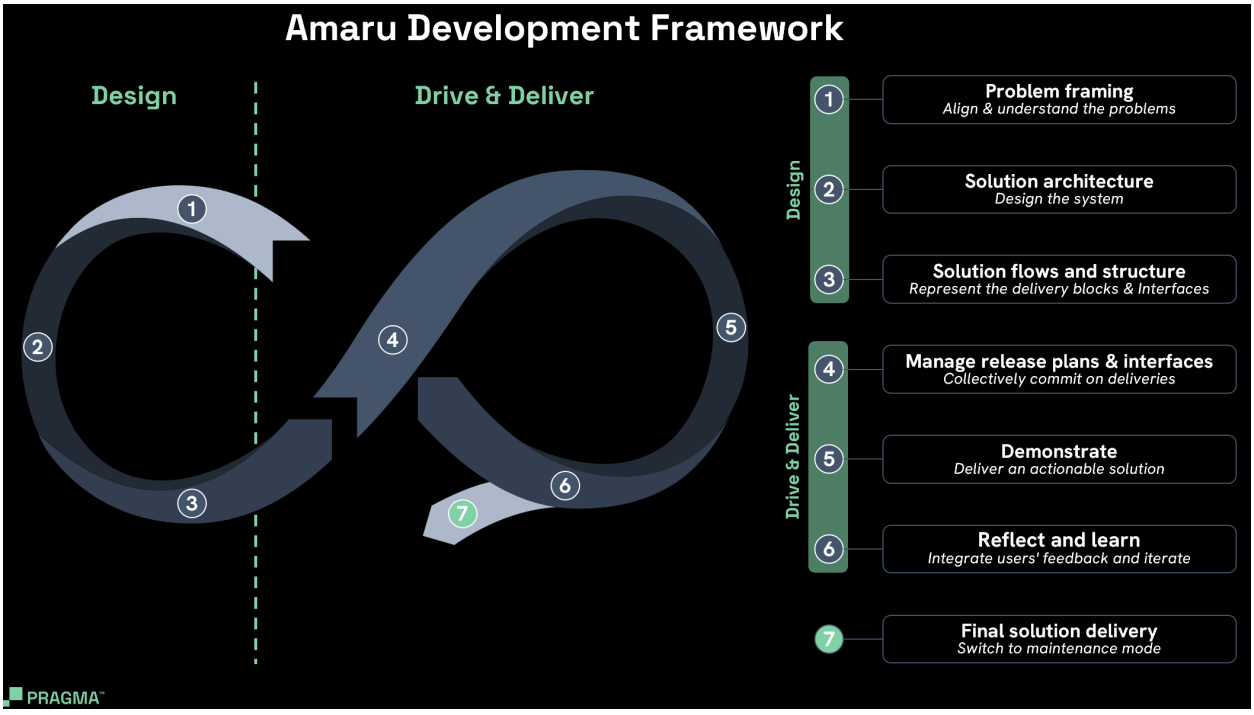


Figure 4: Amaru Development Framework

This representation encapsulates all the phases imagined for designing and running the Amaru project:

1. Problem framing : DDD : Framing the problem linking together Problem, People, Outcome and Constraints
2. Solution architecture : C4 Architecture representing the system and the interfaces between each key components
3. Solution flows and structure : DDD : Bounded context focusing on use cases and structuring the delivery blocks of the solution
4. Manage release plans & interfaces : Extreme programming build a cycle with a release plan directed towards delivering a demo
5. Demonstrate : Extreme programming deliver on the features with acceptance criteria and KPIs to measure
6. Reflect and learn : Lean Startup : Validated learning allocate time to confront the solution to the problem environment and its user
7. Final solution delivery : Lean Startup : Build, Measure, Learn document the main delivery and discoveries related to the problems, prepare the next maintaining cycle

Currently (April 2025) we are in a cycle where scope owners manage its demo driven release plan aligned with the project's objectives (Steps 4; 5; 6 of the framework) and use demos that happen every 4 to 8 weeks to showcase progress.

Design phase results: Architecture of the product

Our objective during that phase was to create clear and compelling problem statements that establishes a starting point to quickly understand:

- Who is impacted by our project,
- The desired outcomes that define success for our project
- The constraints that will need to be taken into account

While starting out the discussion one topic was clear, we need diversity in the people contributing right from the start. The initial team that contributed to this design phase came from 6 companies: Blink Labs, Cardano Foundation, dcSparks, IOHK, Sundae Labs, TxPipe. The positions included were also of a varied range: CEO, COO, CTO, Senior developer, Project manager, SPO.

If you want to better understand the extent of the project and see what we went through you can have a look into the results of the design phase here: Amaru documentation: Design phase results

This document includes an overview of the problems statements that were identified, the people that we chose as stakeholders and the overall architecture setup we drafted. Keep in mind this is the current results, the document is subject to change depending on the direction we want to take and the use case we will be facing.

For this proposal we will take into account the results and focus on the execution of the project.

Turning product design into actionable scopes and teams

Now that we have a better understanding of the problems we want to solve, the global architecture of our product and have identified the scopes that each team will deliver, let's look into how we will organize the project to achieve everything we want to do.

Bounded context ownership

Our definition of a bounded context in our project is: "An autonomous scope included in the overall architecture of the product. The bounded context owner who has the responsibility of driving the activity and aligning with stakeholders the deliveries and managing interfaces."

Each bounded context owner has the responsibility of managing everything in order to deliver an integrated scope for the project. They also have ownership of decision making for everything related to their allocated budget. (that will be described later on in that document)

Having those scope owners will enable asynchronous development while being driven by the projects common targets: demos.

We currently establish the following bounded contexts, and their respective owners:

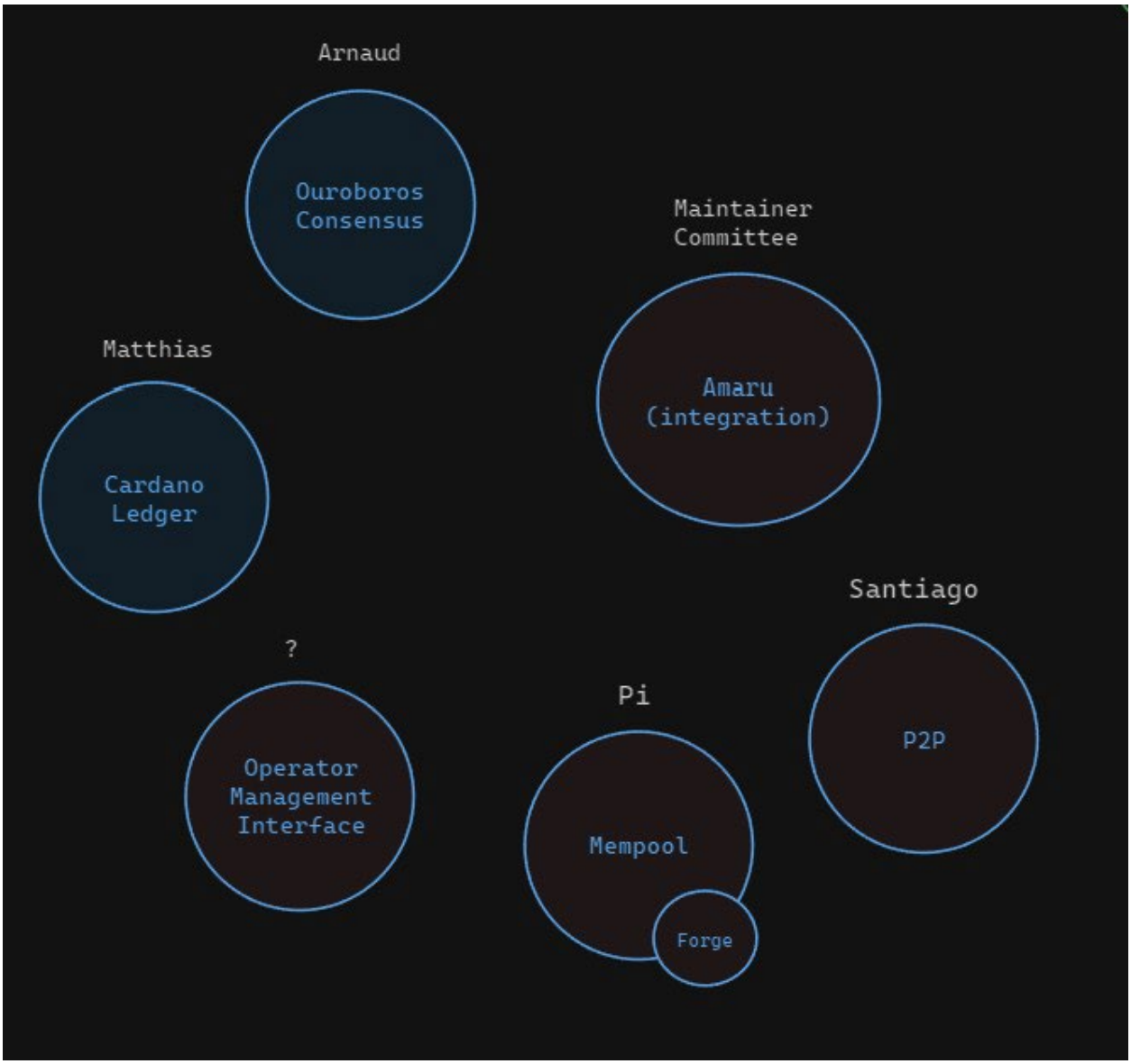


Figure 5: List of bounded context of the project

- Amaru integration: Amaru Maintainers Committee (with final deciding member: Matthias Benkort)
- Ledger owner: Matthias Benkort
- Ouroboros Consensus owner: Arnaud Bailly
- Mempool and block forge owner: Pi Lanningham
- Peer 2 peer (P2P) owner: Santiago Carmuega
- Operator Interface Management: refining the scope content and timeline, no owner yet nominated

Anything that comes out of the scope of each owner will be discussed with the maintainer committee who has a role of sanity check for Amaru:

- Aligning the demo content with the expected features
- Reviewing the content of the integrated scope
- Measuring the product overall performance and keeping up with expected targets

As an additional part of our setup, we identified strong interfaces with the following products (that are not hosted under the Amaru repo) and nominated a contact person for each of these scopes to align with

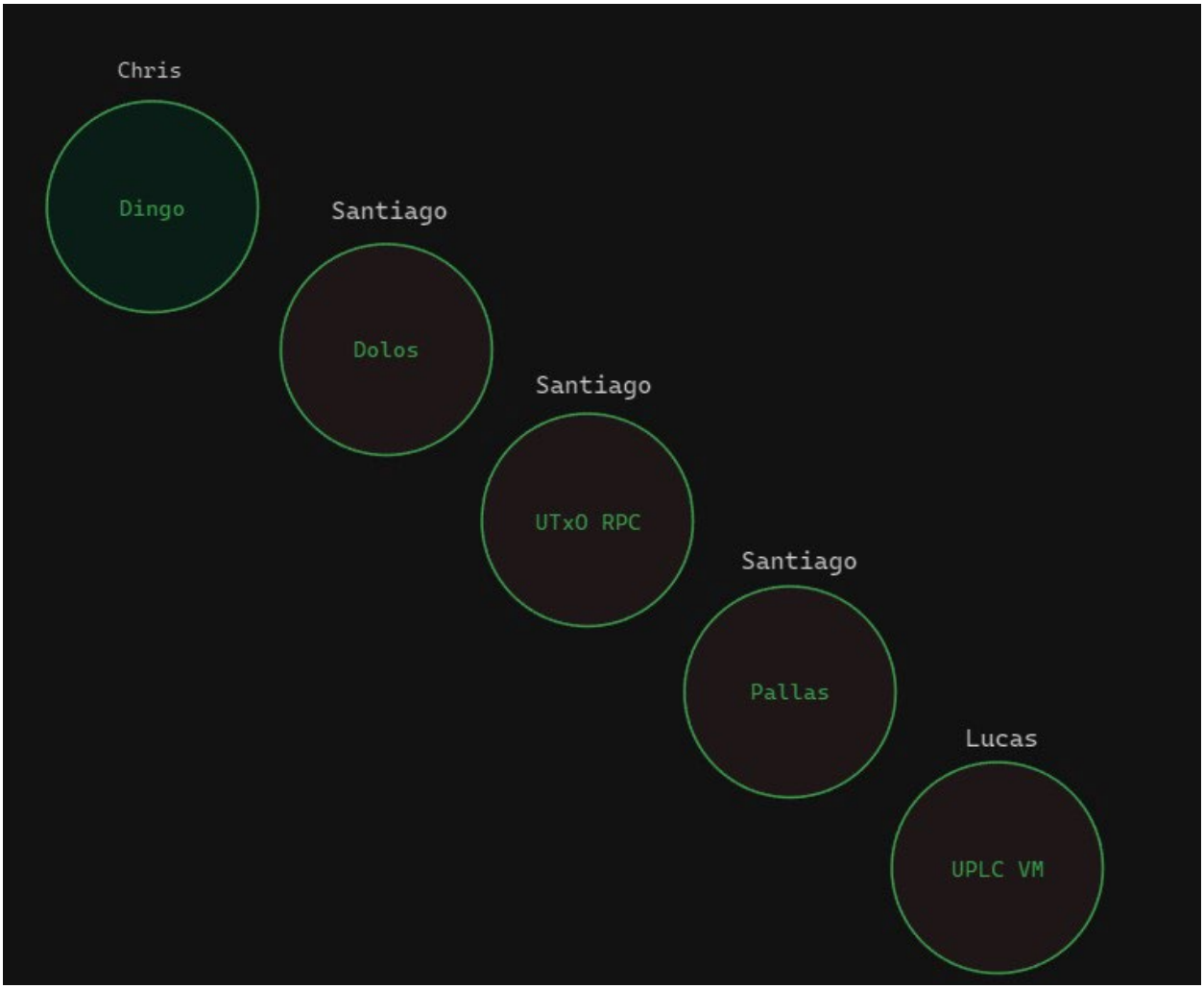


Figure 6: List of interfaces to be managed

- Dingo (GO Node) interface: Chris Gianelloni see: Dingo Github
- Dolos interface: Santiago Carmuega see: Dolos Github Dolos proposal
- UTxO RPC interface: Santiago Carmuega see: UTxO RPC Github UTxO RPC proposal
- Pallas interface: Santiago Carmuega see: Pallas Github Pallas proposal
- UPLC VM interface: Lucas Rosa see: UPLC Github

Governance

The decision making on the Amaru project function with a simple rule: "if you can decide, do it"

The project is structured in a way that allows each bounded context to have freedom in its decision making within the boundaries of its scope. The only topics where decisions have to be addressed with the maintainers committee are:

- Interoperability
- Interfaces with other scopes
- Integration with the global product

Today this alignment is a recurring Maintainer Committee meeting every two weeks. Its purpose is to align the key decisions to be made regarding architecture and alignment of all the ongoing streams of activity.

The way we reach a decision by default is by lazy consensus meaning that once something is discussed with all the owners that are impacted by it, explained and an agreement is reached during the meeting mentioned above: it is a decision if no objections are made. For any objections that aren't settled in the Maintainer Committee the final approval will be settled with our main architect: Matthias Benkort.

Administration of funds with on-chain decision making & off-chain contracting

One of the core challenges faced by any open source project that is requesting a Cardano Treasury withdrawal is ensuring the sound, transparent and accountable management of the allocated resources. Oversight of the project including resource allocation and management rests with the Amaru Maintainer Committee (“AMC”) which in turn provides regular reports and is accountable to the board of Pragma.

To this effect, we have created a system to address these challenges and mitigate key risks that consist of an on-chain component and an off-chain component which both refer to and tie back into the AMC roles and transparent development process on GitHub. The described

processes and systems will be fully functional and ready to go at the time that the Cardano Ecosystem Budget withdrawal request is committed on-chain.

On-Chain component

Based on this proposal deep dives, each scope owner that has allocated resources will have access to a threshold that they can withdraw money from, each scope has only 1 owner and only 1 threshold (valued in $).

We want the decision making to be very simple and straightforward: every scope owner must be in agreement when the withdrawal is made.

We want to have all the information of the transaction at the same place so that it's easier to understand why the decision has been made and what's the extent of it. We will include in all propositions an intent and a document that states the commitment of the owner to a specific delivery that goes in line with the roadmap of the whole project.

As mentioned, the process should be straightforward and everything has to be available on-chain:

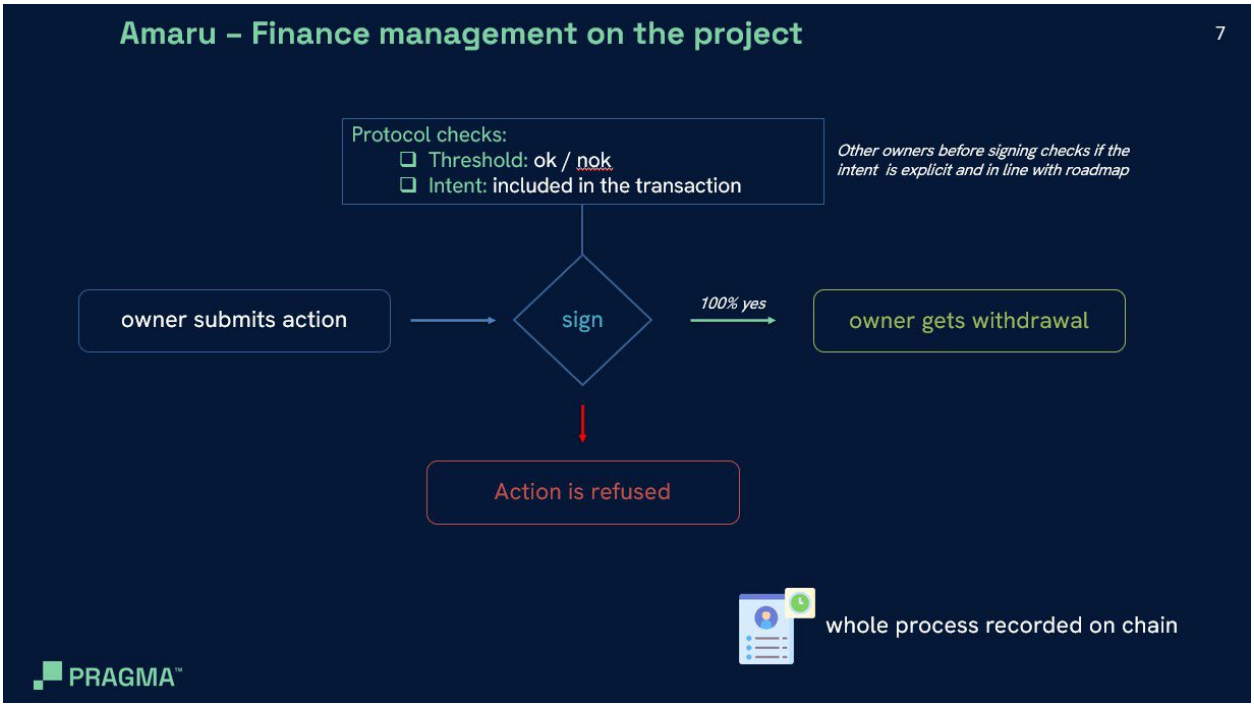


Figure 7: Amaru on chain finance management

The current features related to the finance management on the project are:

- Withdraw money: scope owner asks other scope owners for an amount of money to be withdrawn from his scope
- Contingency withdraw: scope owner asks other scope owners for an amount to be withdrawn from the contingency funds
- Change of ownership: scope owner asks other scope owners for a change of ownership of a scope (or a reallocation of threshold)
- Contingency back to the Cardano Treasury: scope owner asks that the contingency be sent back to the Cardano Treasury

Note: 12 months after the treasury withdrawal what's left within the contingency account will be transferred back to the Cardano Treasury.
We believe that if we need the contingency to carry over then we need another proposal to be discussed linked with deliveries.
We will track the budget status of each scope and of the overall project on a monthly basis.

Features to be implemented

We want to make the system to manage the finance of the project to be using a full consensus for decision making. This would require adding the following features:

First feature: Scopes and budget
The treasury proposal is organized into different "scopes," each with its own credentials and rules. Each scope has designated owners and its own budget, making the system more manageable and secure. This allows different aspects of the project to maintain separate financial controls. We will set a threshold for each scope owner with an amount that can be withdrawn and a limit.

Second feature: Spending rules
All spending from the treasury must follow strict rules:

- Any spending requires authorization from the scope owner
- Every transaction must include a clear rationale
- Spending within limits can be approved by the scope owner
- Spending above limits (using the contingency funds) requires unanimous approval from the scope owners

Third feature: Recovery and rotation of credentials
We will use NFT credentials stored in a multisig setup, which makes it easy to rotate credentials if needed. If a scope owner's credentials are lost, there's a recovery mechanism

that allows credential changes with majority approval, followed by a contestation period. During this period, any scope owner can cancel the credential change if the original credentials are recovered.

Fourth feature: Failsafe for compromised credential
In the extreme case where a scope is compromised, a failsafe mechanism can be triggered. A donation to the Cardano Treasury or to a pre-determined account is possible with approval from all but one (N-1) scope owners.

Fifth feature: Transparency and verification
The treasury's state of the Amaru project will be fully transparent and verifiable on-chain. Each operation will be recorded with its rationale, ensuring accountability and allowing for oversight by all stakeholders. This creates an audit trail that can be reviewed at any time.

Sixth feature: Delivery
We want to have a very straightforward method when it comes to managing deliveries: use Github Pull Requests to formalise the final delivery. The acceptance process of the deliveries will be using our day to day regimen: Make sure the codebase proposed is integrated on the right part of the Amaru repository
Include testing results from the modified environment Validate with the core maintainers and have them sign off the Pull request Push the pull request to the active codebase If the activity is not delivering code then the documentation should be the delivery.

Seventh feature: disbursing funds to remunerated contributors.
Remunerated contributors will have to specify a wallet address in order to receive the funds from the withdrawal. The wallet information will be kept private and only the transaction will be visible to the public.

The withdrawal will be done with a basic rule on our project: 30% of the sum required will be withdrawn directly and the rest will be allocated to the forecast of delivery dates where the code is pushed to the repository as mentioned above in the Sixth safeguard.

In the event of (i) objective impossibility of an AMC member to sign or (ii) severe adversarial or dishonest behaviour of an AMC member, leading to three consecutive rounds of on chain activity with negative outcome, a fourth round of on chain decision with a simple majority of the AMC members shall be held.

All of these features work together to ensure that funds are managed responsibly, with appropriate checks and balances, while maintaining the flexibility needed for the project to function smoothly.

Off-Chain Component

In order to create a stronger and more traditionally enforceable work framework, particularly with Remunerated Contributors that receive Cardano Treasury resources, the AMC has created a *Maintainer Committee Framework* that describes the work scope allocations process, the contribution process and acceptance procedures and other obligations of Contributors, incl. dispute resolution.

The Framework will be signed by the Scope Owner and any Remunerated Contributors individually for their scope.

PRAGMA policies applied to Amaru: quarterly review

PRAGMA is a member-based, not-for-profit open-source association for blockchain software projects created by 5 companies: Blink labs, Cardano Foundation, dcSparks, Sundae Labs, Txpipe.

PRAGMA is composed of two entities: the General assembly and the Administrative board.

The General assembly is composed of a representative of each founding member and decides the strategy and the composition of PRAGMA.

The Administrative board is elected by the General assembly, manages the operational side of PRAGMA and oversees each of the project.

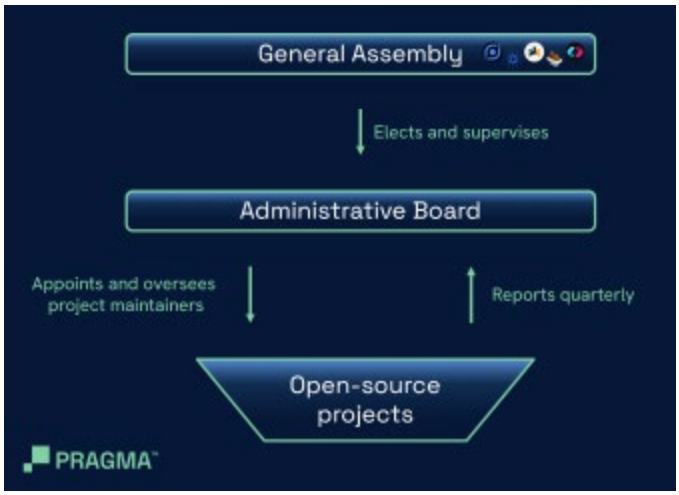


Figure 8: PRAGMA existing bodies and relationships

PRAGMA's Administrative board has the responsibility of aligning with each of the projects that are part of PRAGMA the composition of the maintainers and reviews every quarter the progress of each project as defined in the Pragma Decision Record regarding project reporting.

The content of these reports are straightforward and include:

- Problems that require board attention (infrastructure, trademarks, etc.)
- Project status (active, inactive, initiating, etc.)
- Community changes (last committer addition)
- Next planned steps (new releases, significant changes, etc.)
- Reflection & Performance

This recurrent interaction will provide an overview of the project with a long term perspective and support the maintainers committee on aligning priorities and drawing out next steps.

The outcome of these reviews will be publicly available as documentation in the Amaru Github repository.

Demo driven approach

Everything that happens when planning and discussing future implementation has to be demo driven by our product: Amaru. We want to have an approach to always put into perspective the outcome of what we are trying to do. Every goals and milestones that we want to deliver has to be linked to something that can be run in an integrated demo.

As a result of this mindset, each bounded context owner has to integrate the deliveries of his scope into a demo for Amaru and align with the maintainer committee timing, content and the extent of the features required.

The achieved demos for Amaru so far are:

- First Amaru node experiment : Simple peer-to-peer network with a single block producer (22nd October 2024)
- On-disk Ledger State & Observability (20th December 2024)
- Stake distribution, multi-chain consensus & simulation testing (7th February 2025)
- Block validation, P2P, Ledger state, Simulation (28th March 2025)

The scopes that will be covered by demos:

- Amaru Q2 25 : Relay node release (preview and preprod compatible)
- Amaru Q3 25 : Block producer implementable version
- Amaru Q4 25 : Mainnet Ready node

All the demos can be watched on the Amaru section of the PRAGMA website

The scopes covered by Amaru are being aligned using a whiteboard approach

Delivering Amaru: deep dives

This section of the proposal will cover everything that has been planned so far related to the vision and objectives we set for our project. And also go through already delivered scopes and changes of strategy.

Each scope will be described with the following pattern:

- Scope to be covered and associated timeline
- Estimated resources required: number of FTE (Full time employee), Infrastructure costs, Licenses and additional costs
- Resources and funding already secured contribution to the scope
- Additional effort required to be added to the proposal

The following summarizes the list of bounded contexts that will be detailed in the document.

Assumption for this deep dive:

- The below scopes describe every resources needed for 6 months starting from June 2025 (end of Q2 2025)
- In each scope two additional activities are to be considered: integration and research we will illustrate the extent of this at the end of our deep dive
- Our estimates are optimistic by nature and we try to limit ourselves to the "just necessary" workload in order to be as lean as we can, we will use a comparison after the deep dive to reflect upon our estimates and adjust the overall picture with contingencies

Cardano ledger

Scope and timeline

Scope

The scope of this bounded context is to implement and maintain a ledger compatible with Cardano. This includes:

- Phase 1: Prototype/exploration/proof of concept -private testnet
- Phase 2: Tracking of ledger state (restricted to anything but transaction validation and governance) -private testnet; trusted peers
- Phase 3: Complete ledger rules implementation -private testnet; preview; trustless
- Phase 4: Performance optimisation of tracking ledger state and full validation -private testnet; mainnet ready; trustless
- Soft-fork capabilities: shipping new features without breaking consensus
- Ledger validation performance enhancements

Timeline

- Phase 1: done
- Phase 2: done
- Phase 3: Q2 2025
- Phase 4: Q3 2025
- Soft-fork capabilities shipping new features without breaking consensus: Q4 2025

Estimated resources required

Rust developers: 2.5 FTE

Resources and funding already secured

Rust developers: 1 FTE (Matthias 0.5) (Matthieu 0.5)

Additional effort required

Rust developers: 1.5 FTE

Included in the Ledger scope

The owner of the Ledger will be **Matthias Benkort**.

Ouroboros Consensus

Scope and timeline

Scope

The scope of this bounded context is to implement and maintain software implementing the Ouroboros family of blockchain consensus protocols. This includes Ouroboros Praos, the ongoing incarnation, and all future extensions like Genesis, Peras, Leios... The core requirements of the consensus are:

1. *correctness*: Whatever version of the protocol is current, a node using it can participate fully in the Cardano network, as a client, relay, or block forging node,
2. *security*: Consensus software provides *evidences* it is safe and secure in accordance with the theoretical properties expected from current version of the protocol and situation of the network,
3. *observability* (o11y): Consensus software affords all information needed for node operators and developers to fully understand how it operates, both at build and run time
4. *maintainability*: Consensus code strives to be easy to understand, analyze, and extand now and in the future
5. *performance*: consensus should require limited resources and run on commodity hardware

Timeline

We give a rough (with a 3 months granularity) timeline of the main deliverables for the Amaru project below. Note that dependencies are only needed to demonstrate the deliverable runs on a full Cardano network.

1. *Explore* (done)
    - Build a model of Praos (w/ Genesis?) that exhibits expected properties
    - Define interfaces and relationships with other bounded contexts (networking, mempool, ledger)
2. *Testing* (done)
    - Build a conformance testing tool and related suite of tests based on consensus specification and existing implementation
    - Be able to run conformance testing suite against any implementation of Ouroboros, including existing ouroboros-consensus and future implementation for Amaru
3. *Chain validation* (Q2-Q3 2025)
    - Build a version of consensus that can follow the chain and validate blocks (include headers and transactions validation)
    - Demonstrate conformance of validating node against existing node and network
    - **Dependencies**: ledger state access and transactions validations, networking n2n protocols
4. *Block production* (Q3-Q4 2025)
    - A version of consensus that can forge blocks with transactions from the mempool according to the predefined VRF-based schedule
    - Conformance suite valiging block production schedule and transaction selection logic w.r.t existing node
    - **Dependencies**: ledger, network + mempool
5. *Gap Analysis and configuration* (Q4 2025)
    - Review the gap between the current implementation of Amaru and the required configuration of Genesis and Leios
    - Build a setup of Amaru that is compliant with those requirements

Estimated resources required

Rust developer: 2.5 FTEs

Resources and funding already secured

Rust developers: 1 FTE (Arnaud 0.5) (Pawel 0.5)

Additional effort required

Rust developers: 1.5 FTE

Included in the Consensus scope

The owner of the consensus will be **Arnaud Bailly**.

Project Management, Public Relations & Marketing

Scope and timeline

Scope

Purpose: drive key initiatives and manage use cases and end users feedback loops.

The current set of activities forecasted are:

- Creating and managing SPO working groups to actively improve the operator side of Amaru
- Organizing and facilitating workshops related to key explorations of the product
- Marketing, implementation partnerships alignment, use case specific collaborations
- Roadmaps interfaces and scopes alignment synchronization
- Bug bounty: create an incentive to generate traffic on the testnet preprod and bring contributors to the project

  Testnet developer bounty: Create on preprod the use cases defined by the Amaru team by building the transactions on the network and generate the conditions needed
  Testnet SPO bounty: Produce a block that includes the above use case
  Amaru development bounty: be involved in at least 3 contributions accepted by maintainers

Our current ambition is to experiment with Amaru as much as we can with testnet activity, conformance tests, simulations and build (with the feedback of early adopters) a reliable setup that fulfils all our global targets

Each bounded context will share ownership over a resource that will interface with all the stakeholders of the Amaru project to facilitate the activities mentioned above.

Timeline

- SPO working group setup and key initiatives: Q3 2025
- Bug bounty setup and start: Q3 2025
- Amaru workshops: Every quarter

Estimated resources required

Rust developer: 0.5 FTE

Resources and funding already secured

0 FTE

Additional effort required

0.5 FTE

Included in the Project management, PR Marketing scope

The owner of the Project management PR & Marketing will be **Damien Czapla**.

Unbounded activities

This section describes the expected rigorous development expectations that comes with delivering software on the project and the activities that aren't specific to a bounded context.

Amaru product integration

Each bounded context has to deliver an implementable version of his scope for Amaru. This means that the following has been achieve:

- Refactoring: the structure of the developed code has been optimized to fit with the current structure of the Amaru code base
- Benchmarking: the targets set for the features has been compared to the standards and is up to par with what's available
- Testing: once implemented in the Amaru context, reports are available to show that the expected features are working within the targeted environment.
- Performance: expected performance is tracked and aligned with the Amaru maintainers

Expectations regarding each bounded context are:

- Vision & direction alignment: the deliveries are linked to demos forecast and aligned during maintainer committees
- Quality assurance: each implementation comes with a clear methodology and respects the standards set for Amaru
- Troubleshooting: the scope owner has a responsibility of resolving issues and bugs when pinned down to a specific scope

The main responsibilities of managing a bounded context comes with aligning forecasts between multiple teams, securing commitments from contributors all while taking into account risks and managing contingencies that can help delivering. This will also be a significant workload for each scope owners and the maintainers committee of the project.

Cardano research, specifications

Delivering an implemented software comes naturally to senior developers, although we want our project to be up to par with the state of the art regarding the Cardano ecosystem.

Expectations regarding each bounded context are:

- Specifications referencing: each implemented component will have to reference the specifications that cover their scope
- Specifications cross check: the features and the performance of the integrated version of the component has to be validated by a threshold or a target included in a specification. If the said specification does not align with the outcome of the performance, either the specification has to be updated or the implementation has to be corrected.
- Specifications updates: any feature implementation or release that impacts a performance boundary that isn't contained in a specification has to be traced and explained to update the documentation or create documentation if none exist
- Research alignment: if the implemented scope challenges the state of Whitepapers or research documents that are existing it has to be addressed with the people involved in the research category

Use case specific workload, troubleshooting, bug fixes

Part of our journey will involve making our node modular and welcome many different use cases. We intent that each scope owner has a responsibility towards making evolutions towards fulfilling these use cases.

We are aware that bringing more implementations and various use cases will come with a higher number of bugs and troubleshooting tasks towards making sure Amaru runs smoothly with a high performance.

Resulting workload: Ad-hoc Mercenaries

Given the mentioned above activities and a set of uncertainty regarding the overall scope of the project after Q3 2025 the other scope owners decided to allocate 2 FTE to a bounded context called "Ad-hoc Mercenaries".

> Included in the Ad-hoc Mercenaries scope

The purpose of that team is to be a lean surgical team that can help any bounded context in delivering actionable results. They will function as a straightforward "help us deliver this" and can be sollicitated by any bounded context owner.

The owner of the Ad-hoc Mercenaries will be **Pi Lanningham**.

Mempool (Ad-hoc Mercenaries)

Here is an example of a scope allocated to the "Ad-hoc Mercenaries" of the project as it will require strong interfacing with the other scopes of the project.

Scope and timeline

Scope

The scope of this bounded context is to implement and maintain a mempool compatible with Cardano. This includes:

- Mempool library: Build a standalone library (data structure that represents transactions in memory) with an interface to get an extract of these transactions
- "Simple" mempool implementation: Build a basic mempool that adds, remove, gather, drain transactions and exposes a new ledger state
- "Complex" mempool implementation: Build a more refined version of the mempool that handles features differently and optimises the overall resources consumption
- Mempool tooling and API: Create a tool able to manage the mempool and the modularity
- Node management Remote Procedure Call (RPC): Build a software that handles the "operator perspective" on operating the Amaru node
- Block forging: Develop a component able to forge blocs

Timeline

- Mempool library: done
- "Simple" mempool implementation: Q2 2025
- Block forging: Q3 2025
- "Complex" mempool implementation: Q4 2025

Estimated resources required

Rust developers: 0.5 FTE

Resources and funding already secured

0 FTE

Additional effort required

0.5 FTE

> Included in the Ad-hoc Mercenaries scope

The owner of the Mempool will be **Pi Lanningham**.

Conclusion

The proposal submitted will be targeting: $750k as a result of the scopes covered in Amaru, as summarized in Estimates and contributions.

The Amaru project represents an opportunity to bring node diversity through the development of a modular, high-performance, and interoperable block-producing node for enhancing the Cardano ecosystem. Our approach, rooted in openness, innovation, and operational resilience, aims to deliver significant advancements in blockchain technology without compromising security or operator experience.

Through rigorous design and execution phases, we have outlined actionable scopes, realistic timelines, and comprehensive resource allocations to achieve our ambitious goals. The detailed breakdown of scopes, funding and resources already secured reflects our commitment to transparency and efficiency. While we remain optimistic about our forecasts, we acknowledge the complexities involved and have incorporated contingencies to address unforeseen challenges.

The requested treasury funding of $750k will empower us to build on the foundational work started, enabling us to transition from vision to reality. We firmly believe that the exceptional talent and dedication of our team, combined with the robust framework, will ensure the project's success.

This proposal is more than just a funding request; it is an invitation to collaborate and innovate together. The Amaru project is ambitious, but it is also achievable with the collective effort of our contributors and the broader Cardano community. By focusing on building, learning, and demonstrating tangible progress, we are confident that Amaru will pave the way for a more resilient and decentralized Cardano ecosystem, ensuring long-term reliability and innovation.

Let’s build the future of Cardano together.