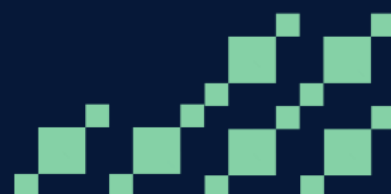




CARDANO ブロックチェーン
エコシステム予算
提案詳細

2025年5月6日



目次

ビジョン.....	3
別のノードを構築する理由.....	3
概要.....	5
留意点：前提条件とバイアス.....	5
類似プロジェクトとの比較.....	6
RETH.....	6
Firedancer.....	6
Cardano-node.....	7
見積もりと貢献.....	8
プロジェクト全体の規模.....	8
すでにプロジェクトに割り当てられたリソース.....	9
国庫要求リソースの内訳.....	10
プロジェクトの規模と運営体制.....	12
Amaruの開発手法：ビジョンを実現可能な形に.....	12
設計フェーズの成果と製品アーキテクチャ.....	13
製品設計から実践へ：チーム編成とスコープの決定.....	14
境界づけられた責任範囲とオーナーシップ.....	14
ガバナンス体制.....	17
資金管理：オンチェーンとオフチェーンの連携.....	17
オンチェーンの仕組み.....	17
オフチェーンでの取り組み.....	20
PRAGMAによる四半期レビュー.....	20
デモ重視の開発方針.....	21
Amaruプロジェクトの詳細.....	22
カルダノ台帳（レジャー）の実装.....	23
実施範囲とスケジュール.....	23
必要なリソース.....	23
既に確保しているリソースと資金.....	23
追加で必要となるリソース.....	23
ウロボロスコンセンサス.....	24
実施範囲とスケジュール.....	24
必要なリソース.....	25
すでに確保しているリソースと資金.....	25
追加で必要となるリソース.....	25
プロジェクトマネジメント・広報&マーケティング活動.....	25
実施範囲とスケジュール.....	25
必要なリソース.....	26
すでに確保しているリソースと資金.....	26
追加で必要となるリソース.....	26
境界づけられていない活動.....	27

Amaru製品の統合.....	27
Cardanoの研究開発と仕様策定.....	27
個別のユースケースへの対応、問題解決、バグ修正.....	28
アドホック開発チームの編成.....	28
メモリプール(アドホック開発).....	28
実施範囲とスケジュール.....	28
必要なリソース.....	29
すでに確保しているリソースと資金.....	29
追加で必要となるリソース.....	29
まとめ.....	30

Tables

Table 1: RETH プロジェクト概算.....	7
Table 2: Firedancer プロジェクト概算.....	8
Table 3: Haskell Cardano-node プロジェクト概算.....	8
Table 4: Amaruプロジェクト全体の規模 (スコープ概要).....	9
Table 5: 既に確保済のリソースの項目 (スコープ概要).....	10
Table 6: 推定リソース、確保済のリソース、要求リソースのリスト (スコープごと).....	11
Table 7: 各スコープのオーナーと米ドルでの要求額.....	12

Figures

Figure 1: Amaruプロジェクト全体の規模.....	9
Figure 2: すでにプロジェクトに割り当てられたリソースのリスト.....	10
Figure 3: 国庫要求リソースのリスト (スコープごと).....	11
Figure 4: Amaru 開発フレームワーク.....	13
Figure 5: 境界づけられた責任範囲のリスト.....	16
Figure 6: 管理されるインターフェースのリスト.....	17
Figure 7: Amaru オンチェーン財務管理.....	19
Figure 8: PRAGMA 現状の構成と関係性.....	23

ビジョン

Amaruは、Cardanoネットワークの安全性とセキュリティを守りながら、より使いやすく強固なシステムを実現するための新しいブロック生成ノードを開発するオープンソースプロジェクトです。ステークプール運営者や開発者に新しい選択肢を提供し、使いやすさを重視した柔軟な設計を特徴としています。主にRustプログラミング言語を採用することで、エコシステムの基盤開発に新たな人材の参加を促すことを目指しています。

この提案書では、Amaruの目的、範囲、計画、予算について詳しく説明します。また、プロジェクトの全体像を示し、開発資金としてCardanoの国庫からの支援を受けることについて議論を行います。

別のノードを構築する理由

Cardanoの大きな強みの一つは、その安定性と信頼性です。これは主に現在のHaskellノードによるもので、2020年2月のByron再起動から約5年間、大きな問題なく稼働しています。

そんな中で、なぜ新しいノードを開発するのでしょうか？理由は以下の通りです：

1. リソースの効率的な活用

ソフトウェアは様々な設計上の判断とトレードオフの組み合わせです。現在のHaskellノードは、メモリ使用量などのリソース消費が比較的高く、これを抑えるためにmin-utxo-valueなどの制限が設けられています。私たちは、異なるアプローチを取ることで、より効率的なリソース利用が可能だと考えています。

2. 優れた監視機能

システムの監視は常に課題となりますが、[Amaru](#)では業界標準の監視、トレース、ログ機能を採用することで、ノードの運用管理を格段に分かりやすくします。

3. 高い相互運用性

[Amaru](#)はRust言語で開発されているため、既存のブロックチェーン・エコシステムとの連携が容易で、WebAssemblyやRISC-Vなど、様々なプラットフォームにも対応できます。これにより、多くの開発者が参加しやすくなり、Haskellノードでは実現が難しかった新しい使い方も可能になります。

4. より確実な品質保証

既存のソフトウェアを新しく実装し直すことで、元のシステムを細部まで見直すことができます。この過程で、バグや仕様の不一致が見つかり、ドキュメントの不足も明らかになります。[Amaru](#)の開発でも、すでにこのような成果が得られています。長期的には、複数のノード実装があることで、より良い設計と堅牢な実装への収束が早まり、開発効率も向上します。

5. 真の分散化

ビザンチン障害耐性を重視し、分散化を最も大切にする私たちにとって、単一のノード実装は単一障害点であり、中央集権化のリスクとなります。新しいノードの導入は、ネットワーク全体の耐障害性を高めるだけでなく、将来の発展性も広げます。

[Amaru](#)は、Cardanoの重視するセキュリティと信頼性を維持しながら、これらの目標を実現できると考えています。RustはHaskellと同様、静的型付け言語として強力な型チェックと効率的なメモリ管理を提供します。また、Rustのエコシステムは高品質なソフトウェア開発に適しており、ブロックチェーンの実装に最適な選択肢となっています。

概要

この提案書は長文となるため、まずプロジェクトの全体像を簡潔に説明し、その後で[Amaruプロジェクト](#)の詳細について解説します。

留意点: 前提条件とバイアス

本提案は大規模な予算を必要とするプロジェクトです。また、その実現には多くの課題やリスクが伴うことを私たちは十分に認識しています。その上で、以下の強みを活かして取り組んでいきます:

- ゼロから始めるわけではありません: 既存のHaskellノードは、多くの課題を解決し、高いパフォーマンスと信頼性を実証してきた優れた基盤です。
- プロジェクトには、Cardanoエコシステムで長年の実績を持つ一流の人材が集まっています: 各メンバーは、それぞれの専門分野でリーダーシップを発揮してきました。
- 効率性: Amaruは私たちの複数のプロジェクトの一つであり、効率的で無駄のない開発アプローチを採用します。
- 私たちは実践的な開発者です: 小さな成果を積み重ね、実際に動くソフトウェアを提供することを重視します。
- オープンソースの精神を大切にし、建設的な意見を歓迎します: プロジェクトに貢献してくれる方々と積極的に協力していきます。

また、本提案には以下の前提条件があります:

- 各担当分野で既に進行中の作業はありますが、専任で取り組んでいる部分は限られています。今回の見積もりは、専任チームを前提としています。
- プロジェクト期間は1年以上を予定しており、段階的な成果物の提供を行います。2025年末までにブロック生成可能なノードの完成を目指し、その過程で実用的な機能を順次リリースします。
- ガバナンスアクションの進行スケジュールを考慮し、この文書の見積もりは2025年6月(資金引き出し開始予定)から6ヶ月間の作業を対象としています。2025年第1・第2四半期のマイルストーンは、この人員見積もりには含まれていません。
- 予算はフルタイム従業員(FTE)単位で見積もっています。RustとDLT(分散型台帳技術)に精通したシニアソフトウェアエンジニア1名あたり年間20万ドルを想定しています(米国在住のリモートワーク、採用費用、有給休暇等の諸経費を含む)。
- プロジェクトメンバーへの支払いの価値を安定させるため、予備費を除く全ての資金はステーブルコインでの引き出しを予定しています。
- 楽観的な見積もりに注意: プロジェクトの複雑さや課題を過小評価し、必要な時間とコストを低く見積もってしまう可能性があることを認識しています。

最後に、私たちは5月8日に情報アクションを提出し、6月9日に資金引き出しを実施する予定です。2025年第1・第2四半期に予定している成果物は、この提案の採決時には既に完了しており、今回の人員見積もりには含まれていません。

類似プロジェクトとの比較

RETH

Project	RETH
Description	イーサリアムネットワークの速度、効率、クライアントの多様性を強化するためにRustで開発された、高性能、モジュール式、オープンソースのイーサリアム実行レイヤークライアント。
Timeline	メインネットまでに2年の開発期間
Dedicated team	コア開発者8名、貢献者90名
Roughly estimated overall cost	>3M\$

Table 1: RETH プロジェクト概算

Firedancer

Project	Firedancer
Description	トランザクションのスループットを最適化し、システムのボトルネックを軽減することで、スケーラビリティ、分散化、およびネットワーク効率を向上させるSolana向けの高性能なC++ベースのバリデータークライアント
Timeline	2年間の開発（現在も継続中）メインネットは2025年を予定
Dedicated team	公表されている数字なし
Roughly estimated overall costs	>5M\$

Table 2: Firedancer プロジェクト概算

Cardano-node

Project	Cardano-node
Description	Haskellで書かれたCardanoのブロックチェーンのコア実装で、Cardanoエコシステム内での分散型トランザクション処理、コンセンサスメカニズム、スマートコントラクトの実行を可能にする。
Timeline	2017年のByronリリースで開始、現在も継続中
Dedicated team	公表されている数字なし
Roughly estimated overall costs	>20M\$

Table 3: Haskell Cardano-node プロジェクト概算

注: 私たちのプロジェクトはRETHIにセットアップや環境が近いので、最終的な見積もり比較の際にはこれを目標にする。

見積もりと貢献

プロジェクト全体の規模

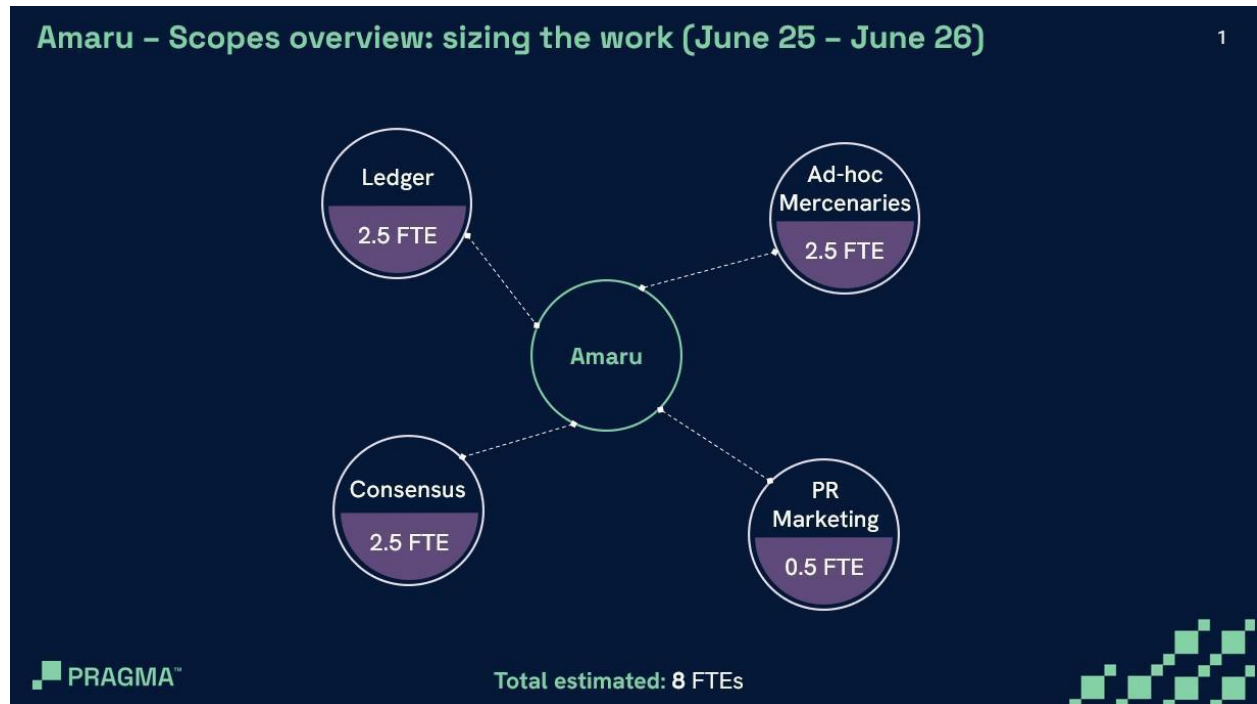


Figure 1: Amaruプロジェクト全体の規模

見積もり結果をご覧ください：

スコープ	リソース見積もり
台帳	2.5 FTEs
コンセンサス	2.5 FTEs
アドホック開発	2.5 FTEs
プロジェクトマネジメント・広報&マーケティング	0.5 FTE

Table 4: Amaruプロジェクト全体の規模 (スコープ概要)

全体の見積もり: 8 FTEs

すでにプロジェクトに割り当てられたリソース

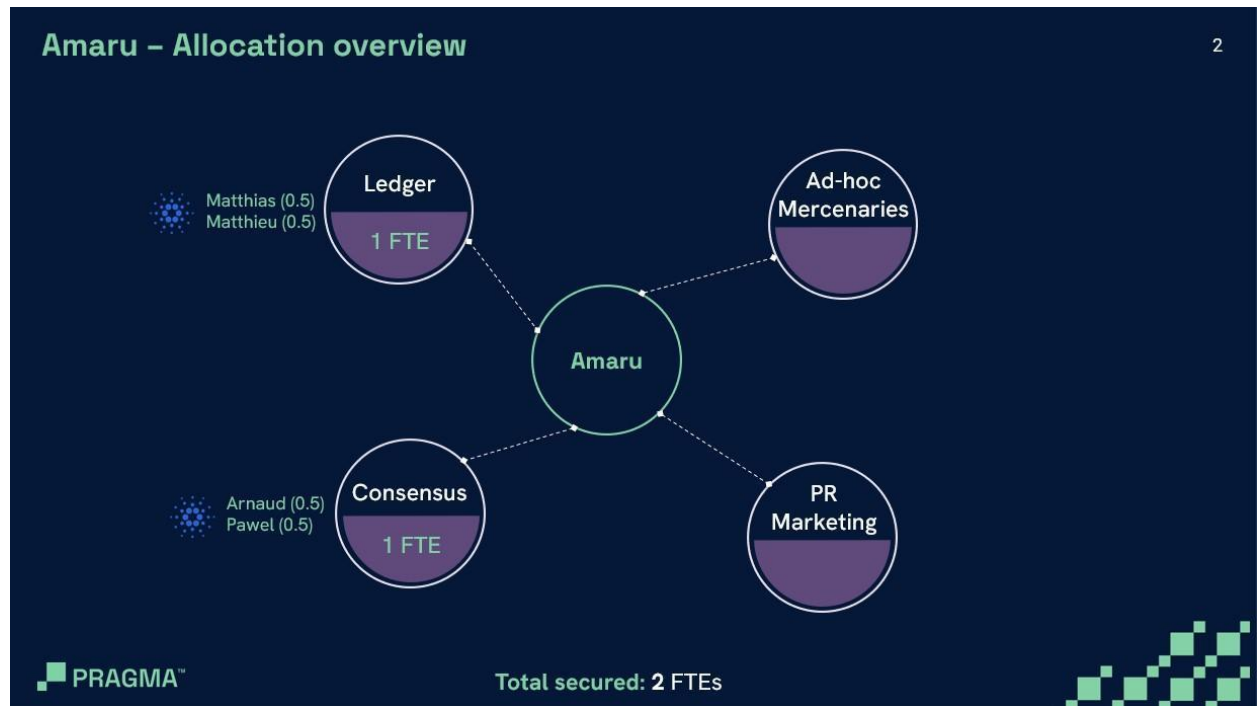


Figure 2: すでにプロジェクトに割り当てられたリソースのリスト

以下は、すでにリソースを確保及び資金調達済の項目です：

スコープ	リソース確保済
台帳	1 FTE
コンセンサス	1 FTE
アドホック開発	0 FTEs
プロジェクトマネジメント・PR & マーケティング	0 FTEs

Table 5: 既に確保済のリソースの項目 (スコープ概要)

すでに利用可能または資金提供されているリソースの合計：カルダノ財団の従業員から**2 FTE**。

注：スコープオーナーは、プロジェクトでの活動を管理する時間も割くこととなりますが、これは上記の FTE 評価には反映されていません。

国庫要求リソースの内訳

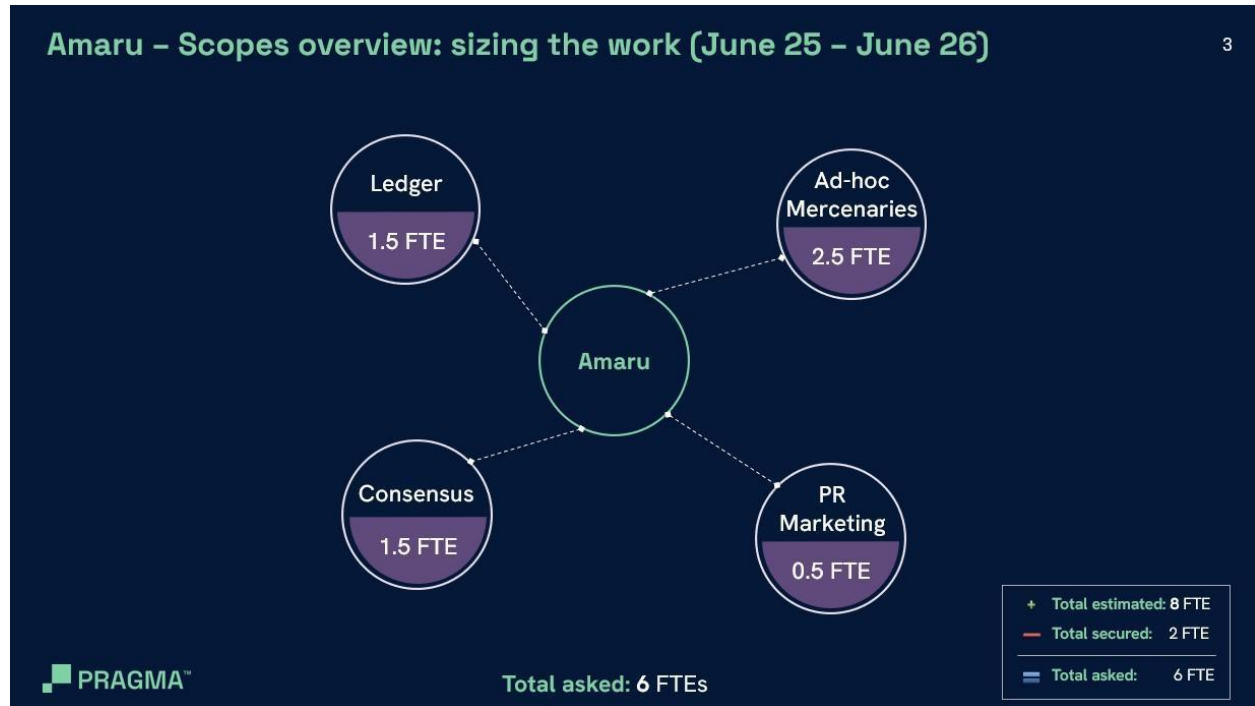


Figure 3: 国庫要求リソースのリスト (スコープごと)

各スコープの国庫要求リソースです:

スコープ	推定リソース	確保済のリソース	要求リソース
台帳	2.5 FTEs	1 FTE	1.5 FTEs
コンセンサス	2.5 FTEs	1 FTE	1.5 FTEs
アドホック開発	2.5 FTEs	0 FTEs	2.5 FTEs
プロジェクトマネジメント・広報 & マーケティング	0.5 FTEs	0 FTEs	0.5 FTEs

Table 6: 推定リソース、確保済のリソース、要求リソースのリスト (スコープごと)

私たちの提案の合計要求リソース: **6 FTEs**.

1人年あたり20万ドルで計算すると、年間予算は以下のようになります:

$$\Rightarrow 6 \text{ FTEs} * \$200\text{k} = \$1200\text{k}$$

ただし、公式のネットチェンジリミットが2025年までとなっているため、実際の申請金額は以下となります:

$$\Rightarrow 6 \text{ FTEs} * 200k / 2 = *600k$$

さらに、予期せぬ事態に備えて25%の予備費を計上することを考慮し、各スコープの最終予算と予備費は以下の通りとなります:

スコープ	要求リソース	スコープオーナー
台帳	\$150k	Matthias Benkort
コンセンサス	\$150k	Arnaud Bailly
アドホック開発	\$250k	Pi Lanningham
プロジェクトマネジメント・広報 & マーケティング	\$50k	Damien Czapla
予備費	\$150k	Amaru保守管理委員会 (AMC)
---	---	---
合計	\$750k	

Table 7: 各スコープのオーナーと米ドルでの要求額

本提案では、Amaruプロジェクトの範囲内で**75万ドル**の予算を申請いたします。

以下の内容について詳しく説明します:

- 採用した開発手法とフレームワーク
- 初期設計段階での成果とプロジェクト範囲の区分け
- Amaruの運営における基本方針
- プロジェクトの範囲、スケジュール、予算の詳細分析
- これまでに完了した成果物の概要

プロジェクトの規模と運営体制

Amaruの開発手法：ビジョンを実現可能な形に

私たちチームの最大の課題は、カルダノノードの構築という目標を明確にし、それを製品の各モジュールとして実装可能なコンポーネントに分解することでした。

数多くの開発手法を検討しましたが、私たちのプロジェクトのビジョンと実際の開発環境に完全に適合するものは見つかりませんでした。

そこで、以下の2つの主要フェーズからなる独自の開発フレームワークを策定しました：

1. 設計フェーズ：解決すべき課題を明確化し、製品のアーキテクチャを設計した上で、Amaruのビジョンの核となるモジュール性とパフォーマンスの実現に向けた作業範囲を定めます。
2. 開発・デリバリーフェーズ：各アーキテクチャコンポーネント間のデモとインターフェースを軸に進める基本的な開発サイクルを実施します。

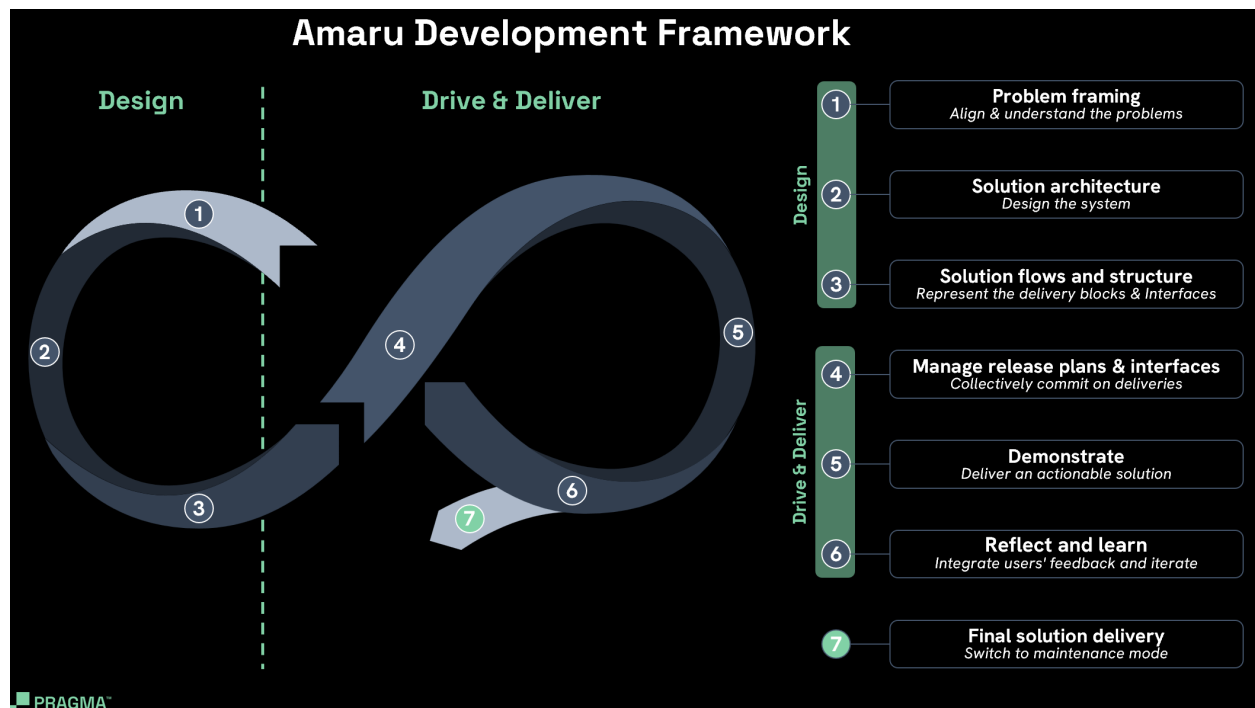


Figure 4: Amaru 開発フレームワーク

以下に、Amaruプロジェクトの設計から実装までの全体的な流れをまとめました：

1. 問題の整理:[DDDの手法を用いて](#)、課題、関係者、目標、制約条件を明確化します
2. 解決策の設計:[C4モデル](#)を使用して、システム全体の構造と各要素間のつながりを設計します
3. 実装の計画:[DDDの考え方に基づき](#)、具体的なユースケースを中心に、システムの各部分の役割と開発単位を決めていきます
4. リリースの管理:[エクストリームプログラミング](#)の手法を取り入れ、定期的なデモを軸にした開発サイクルを組み立てます
5. 検証:[エクストリームプログラミング](#)の考え方に従い、具体的な基準とKPIを設定して機能の完成度を確認します
6. 振り返り:[リーンスタートアップ](#)の手法を用いて、実際の利用環境やユーザーの声を踏まえた改善を行います
7. 成果のまとめ:[リーンスタートアップ](#)の「構築・計測・学習」サイクルに基づき、主な成果と発見を文書化し、今後の保守計画に活かします

現在(2025年4月)は、各領域の責任者がフレームワークのステップ4～6に沿って開発を進めており、4～8週間ごとのデモを通じて進捗を確認しています。

設計フェーズの成果と製品アーキテクチャ

この設計フェーズでは、以下の3点を明確にすることを目指しました:

- プロジェクトが影響を与えるステークホルダー
- プロジェクトの成功基準
- 考慮すべき制約事項

設計フェーズには、Blink Labs、Cardano Foundation、dcSparks、IOHK、Sundae Labs、TxPipeの6社から、CEO、COO、CTO、シニアデベロッパー、プロジェクトマネージャー、SPOなど、様々な立場の方々が参加しました。

プロジェクトの詳細については、[設計フェーズの成果物](#)をご参照ください。

この文書には、課題の分析結果、主要ステークホルダー、そして全体的なアーキテクチャ設計が含まれています。なお、今後のニーズやユースケースに応じて内容が更新される可能性があります。

以下では、これらの成果を踏まえ、プロジェクトの具体的な実施内容について説明します。

製品設計から実践へ：チーム編成とスコープの決定

これまでの設計段階で、解決すべき課題と製品の全体像が見えてきました。次は、各チームの役割を明確にし、プロジェクトの実現に向けた体制を整えていきます。

境界づけられた責任範囲とオーナーシップ

私たちは「境界づけられた責任範囲（バウンデッド・コンテキスト）」という考え方を採用しています。これは、製品の全体設計の中で、独立して機能する一つの領域を指します。各領域には、その活動を主導し、関係者との調整や他の領域との連携を担当するオーナーを置いています。

各オーナーは、担当領域の開発からプロジェクトへの統合まで、すべての責任を負います。また、割り当てられた予算に関するすべての決定権も持ちます（詳細は後述）。

このように各領域にオーナーを置くことで、全体の目標に沿いながらも、各チームが独立して開発を進められる体制を実現しています。

現在、以下のような境界づけられた責任範囲とオーナーを設定しています:



Figure 5: 境界づけられた責任範囲のリスト

- Amaruの統合:Amaru 保守管理委員会 (最終決定メンバー:[Matthias Benkort](#))
- 台帳オーナー:[Matthias Benkort](#)
- ウロボロス・コンセンサス所有者:[Arnaud Bailly](#)
- メンプールおよびブロックフォージ所有者:[Pi Lanningham](#)
- ピア2ピア (P2P) オーナー:[Santiago Carmuega](#)
- Operator Interface Management: スcopeの内容とスケジュールを調整中。

各領域のオーナーの責任範囲を超える事項については、Amaruの品質管理を担当する保守管理委員会と協議を行います:

- デモンストレーションの内容と期待される機能の一致性確認
- 統合範囲の内容確認
- 製品全体の性能評価と目標達成状況の監視

また、私たちのプロジェクトでは、Amaruのリポジトリ外にある以下の製品との連携が重要となるため、各製品の担当者を指名して密接な連携を図っています。



Figure 6: 管理されるインターフェースのリスト

- Dingo (GO Node) インターフェース: [Chris Gianelloni](#)参照: [Dingo Github](#)
- Dolos インターフェース: [Santiago Carmuega](#)参照: [Dolos Github](#) [Dolos プロポーザル](#)
- UTxO RPCインターフェース: [Santiago Carmuega](#)参照: [UTxO RPC Github](#) [UTxO RPCの提案](#)
- Pallasインターフェース: [Santiago Carmuega](#)参照: [Pallas Github](#) [Pallas提案](#)
- UPLC VMインターフェース: [Lucas Rosa](#)参照: [UPLC Githubを参照](#)

ガバナンス体制

Amaruプロジェクトでは、「決定できることは即実行」という基本方針のもと、意思決定を行っています。

各部門が自身の担当範囲内で自由に判断できる体制を整えています。ただし、以下の事項については保守管理委員会との協議が必要です：

- 他システムとの相互運用性
- 他部門との連携方法
- 製品全体への統合方針

これらの調整は、2週間ごとに開催される保守管理委員会で行われています。この会議では、システム設計に関する重要な決定と、進行中の活動の進捗確認を行います。

基本的な意思決定方式として、私たちは[遅延コンセンサス方式](#)を採用しています。これは、関係者全員で議論・説明を行い、会議で合意が得られた事項については、特に反対意見がなければ決定とみなす方式です。保守管理委員会で合意に至らない場合は、主任アーキテクトの[Matthias Benkort](#)が最終判断を下します。

資金管理：オンチェーンとオフチェーンの連携

Cardanoの財務システムを利用するオープンソースプロジェクトにとって、最も重要な課題の一つが、資金の適切な管理と透明性の確保です。当プロジェクトでは、資源配分と運営管理をAmaru保守管理委員会（AMC）が担当し、定期的にPRAGMA取締役会へ報告する体制を取っています。

この課題に対応するため、私たちは主要なリスクを軽減できる独自のシステムを構築しました。このシステムは、オンチェーンとオフチェーンの両方の要素で構成され、AMCの役割とGitHub上の透明性の高い開発プロセスと連携しています。このシステムは、Cardanoエコシステム予算の引き出しが承認された時点で稼働を開始します。

オンチェーンの仕組み

この提案の詳細な検討に基づき、[各部門の責任者](#)は、定められた範囲内で資金を引き出すことができます。

私たちは資金に関する意思決定を極力シンプルにしたいと考えています。資金の引き出しには、すべての部門責任者の同意が必要です。

取引に関する情報はすべて一箇所に集約し、判断の根拠と影響範囲を明確にします。各提案には、プロジェクト全体の計画に沿った具体的な成果物に対する責任者のコミットメントを明記します。

上述の通り、このプロセスはシンプルであることを重視し、すべての情報をブロックチェーン上で確認できるようにします:

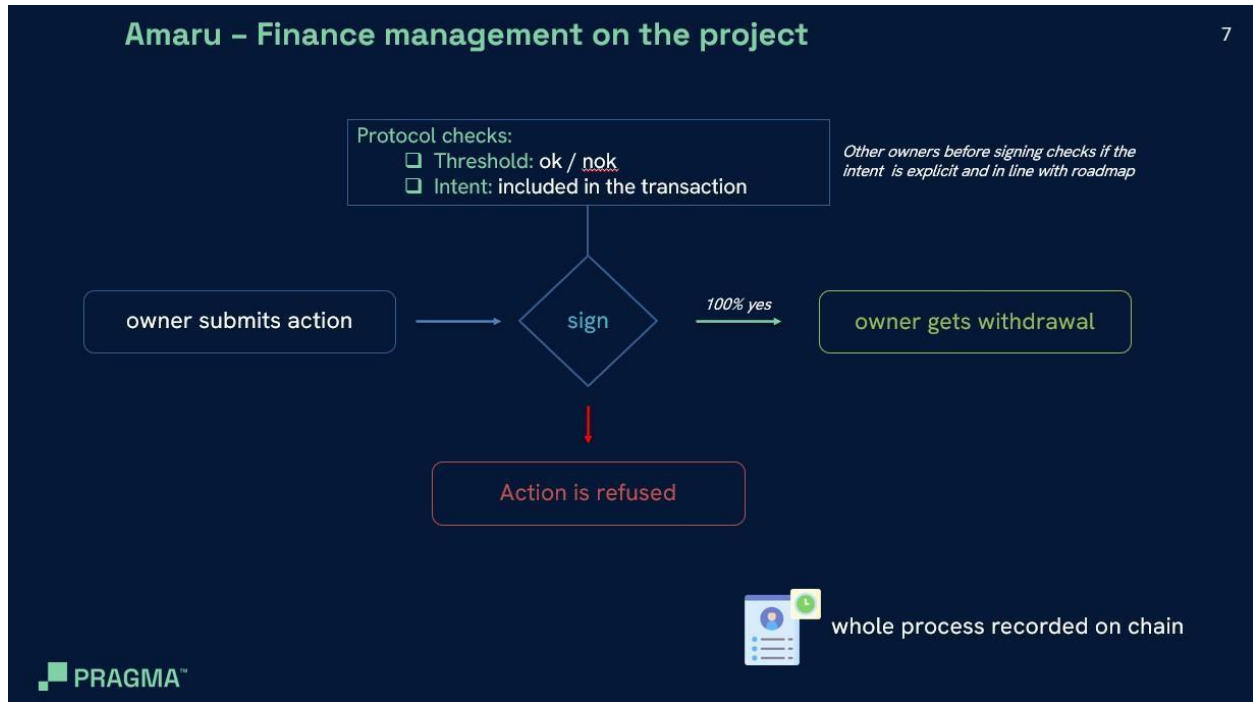


Figure 7: Amaru オンチェーン財務管理

現在、プロジェクトでは以下の財務管理機能を備えています:

- 資金の引き出し: スcopeオーナーは、担当部門の予算から資金を引き出す際に他のオーナーの承認を得る必要があります
- 予備費の使用: スcopeオーナーは、予備費を使用する際に他のオーナーに承認を求めます
- オーナーの変更: スcopeオーナーを変更する場合や、承認に必要な人数を変更する場合は、他のオーナーの同意が必要です
- 国庫への返還: 緊急時には、スcopeオーナーの判断でCardano国庫に予備費を返還することができます

注: 国庫から資金を受け取ってから12ヶ月後、未使用の予備費はCardano国庫に返還されます。予備費を継続して使用する必要がある場合は、成果物と連動した新たな提案を行う必要があります。なお、各部門および全体の予算執行状況は毎月確認しています。

実装予定の機能

私たちは、プロジェクトの財務管理システムにおいて、全員の合意による意思決定を実現したいと考えています。そのために、以下の機能を実装します:

第一の機能: 担当範囲と予算管理

財務管理を効率的かつ安全に行うため、プロジェクトの予算は複数の担当範囲に分けられ、それぞれに責任者と独自の予算が設定されます。各責任者には予算の使用限度額が定められ、その範囲内で管理を行います。

第二の機能: 支出管理のルール

予算の使用には以下の厳格なルールが適用されます:

- 支出には担当範囲の責任者の承認が必要です
- すべての取引について、その目的と必要性を明確に示す必要があります
- 使用限度額内の支出は、各担当範囲の責任者が承認できます
- 限度額を超える支出(予備費の使用)には、全責任者の承認が必要です

第三の機能: 認証情報の管理

マルチシング設定ではNFTによる認証を採用し、必要に応じて認証情報を更新できます。責任者の認証情報が失われた場合は、過半数の承認により新しい認証情報に変更できます。その後、一定期間の異議申立期間が設けられ、この間に元の認証情報が回復された場合は、変更を取り消すことができます。

第四の機能: 緊急時の安全対策

認証情報が漏洩するような重大な事態が発生した場合、1名を除く全責任者の承認により、資金をCardano国庫または指定口座に移動できる緊急対策を用意しています。

第五の機能: 透明性の確保

Amaruプロジェクトの資金状況はすべてブロックチェーン上で確認可能です。各取引は理由とともに記録され、関係者全員が確認できる形で管理されます。これにより、常に監査可能な記録が残ります。

第六の機能: 成果物の提出

成果物の提出はGitHubのプルリクエストを通じて行います。提出された成果物は、以下のプロセスで確認します: ・コードがAmaruリポジトリの適切な場所に組み込まれていることを確認 ・修正済み

環境でのテスト実施・コア開発者による確認とプルリクエストへの承認・コード以外の成果物の場合は、文書として提出

第七の機能：貢献者への報酬支払い

報酬を受け取る貢献者は、支払い用のウォレットアドレスを登録する必要があります。ウォレット情報は非公開とし、取引記録のみを公開します。

報酬の支払いは以下のルールで行われます：金額の30%を即時支払い、残りは成果物の提出予定日に応じて段階的に支払います。

なお、以下の場合にはAMCメンバーの過半数による決定で処理を進めることができます：・AMCメンバーの承認が客観的に不可能な場合・AMCメンバーの不適切な行動により、3回連続で手続きが進まない場合

これらの機能により、プロジェクトの柔軟な運営を維持しながら、適切な管理体制と資金の責任ある運用を実現します。

オフチェーンでの取り組み

Cardano国庫から報酬を受け取る貢献者との間で、より確実な協力体制を構築するため、AMCは保守管理委員会のフレームワークを策定しました。このフレームワークには、作業範囲の割り当て方法、貢献の受け入れ手順、紛争解決を含む各種規定が含まれています。

このフレームワークは、各担当範囲の責任者と報酬を受ける貢献者が個別に同意する必要があります。

PRAGMAによる四半期レビュー

[PRAGMA](#)は、Blink Labs、Cardano Foundation、dcSparks、Sundae Labs、Txpipeの5社が設立した、ブロックチェーンソフトウェアのための非営利のオープンソース団体です。

PRAGMAは以下の2つの組織で構成されています：・創設メンバーの代表で構成される総会（戦略と組織構成の決定）・総会が選出する運営委員会（運営管理とプロジェクト監督）

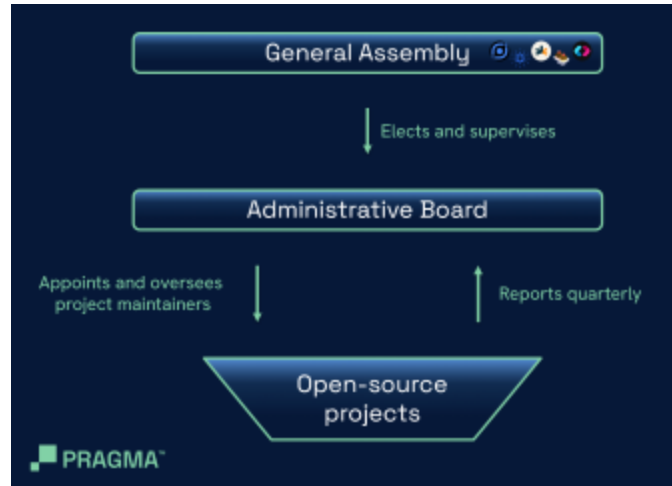


Figure 8: PRAGMA 現状の構成と関係性

PRAGMAの運営委員会は、各プロジェクトと保守管理の体制を調整し、四半期ごとにプロジェクトの進捗状況を確認します。この確認は[Pragma Decision Record](#)に基づいて行われます。

報告書には以下の項目が含まれます：

- 理事会での検討が必要な課題（インフラや商標に関する問題など）
- プロジェクトの現状（進行中、停止中、開始予定など）
- コミュニティの変更点（新しいコミッターの追加など）
- 今後の予定（新機能のリリース、主要な変更など）
- これまでの成果と実績の評価

この定期的な確認により、プロジェクトの全体像を長期的に把握し、優先順位の見直しや今後の計画策定にも役立てています。

レビューの内容はすべてAmaru Githubリポジトリで公開されています。

デモ重視の開発方針

Amaruでは、すべての計画や議論を実際のデモを通じて進めています。私たちは常に最終的な成果物を意識し、掲げる目標やマイルストーンは必ずデモで示せるものとしています。

そのため、各担当者は自分の担当範囲の成果をAmaruのデモに組み込み、保守管理委員会と協力して、実施時期や内容、必要な機能について調整を行っています。

これまでに実施したAmaruのデモは以下の通りです：

- [第1回：単一のブロック・プロデューサーによるP2Pネットワークの基本実験](#)（2024年10月22日）
- [第2回：台帳状態の保存と監視機能のデモ](#)（2024年12月20日）
- [第3回：ステーキング配分、マルチチェーン合意形成、シミュレーションテスト](#)（2025年2月7日）
- [第4回：ブロック検証、P2P通信、台帳状態、シミュレーション](#)（2025年3月28日）

今後予定しているデモの範囲：

- [2025年第2四半期：リレーノードの公開（プレビューとプリプロド対応）](#)
- [2025年第3四半期：ブロックプロデューサー実装版](#)
- [2025年第4四半期：メインネット対応版](#)

すべてのデモは[PRAGMAウェブサイトのAmaruセクション](#)でご覧いただけます。

また、Amaruの開発範囲は[ホワイトボード](#)で管理・調整しています。

Amaruプロジェクトの詳細

このセクションでは、私たちのプロジェクトのビジョンと目標に基づき、これまでの計画内容を詳しく説明します。また、実現済みの範囲や方針の変更点についても触れていきます。

各項目は以下の構成で説明します：

- 実施範囲とスケジュール
- 必要なリソース（人員数、インフラ費用、ライセンス料、その他経費）
- すでに確保しているリソースと資金
- 追加で必要となるリソース

以下では、[境界づけられた責任範囲](#)の一覧を説明していきます。

前提条件：

- 各項目には2025年6月から半年間（2025年第2四半期末まで）に必要なリソースをすべて記載しています。
- それぞれの項目について、統合作業とリサーチ活動の2つの追加タスクも考慮に入れています。
- 現時点での見積もりは控えめな数字となっています。必要最小限の工数で計算していますが、実施段階で比較検証を行い、予期せぬ事態も考慮して全体の調整を行う予定です。

カルダノ台帳(レジヤー)の実装

実施範囲とスケジュール

実施範囲

このプロジェクトでは、カルダノと互換性のある台帳システムの開発と運用を行います。具体的には以下の段階で進めていきます:

- 第1段階: プライベートテストネットでの試作と検証
- 第2段階: 基本的な台帳状態の管理: プライベートテストネット環境で実施、信頼できるノード間での動作確認
- 第3段階: 台帳の完全実装: プライベートテストネットとプレビュー環境での検証、信頼関係のないノード間での動作確認
- 第4段階: 性能最適化、台帳状態の追跡と検証の効率化: プライベートテストネット環境での検証、メインネット展開に向けた準備、信頼関係のないノード間での動作確認
- ソフトフォーク対応: システムを止めることなく新機能を追加できる仕組みの実装
- 継続的な性能改善: 台帳の検証処理の最適化

スケジュール

- 第1段階: 完了
- 第2段階: 完了
- 第3段階: 2025年第2四半期
- 第4段階: 2025年第3四半期
- ソフトフォーク機能: 2025年第4四半期

必要なリソース

Rust開発者: 2.5 FTE

既に確保しているリソースと資金

Rust開発者: 1 FTE (Matthias: 0.5、Matthieu: 0.5)

追加で必要となるリソース

Rust開発者: 1.5 FTE

以上が台帳システム開発の概要です

オーナー：Matthias Benkort

ウロボロスコンセンサス

実施範囲とスケジュール

実施範囲

このプロジェクトでは、ブロックチェーンのコンセンサスプロトコル「ウロボロス」とその派生プロトコルの開発・保守を行います。現在開発中のOuroboros Praosに加え、将来的にはGenesis、Peras、Leiosなどの拡張も含まれます。主な要件は以下の通りです：

1. 正確性：最新のプロトコルを使用するノードが、カルダノネットワークにクライアント、リレー、ブロック生成ノードとして完全に参加できること
2. 安全性：プロトコルの理論的な性質に基づいて、コンセンサスソフトウェアの安全性が証明できること
3. 可観測性 (o11y)：ノード運用者や開発者が、システムの動作を完全に把握できるよう、必要な情報をすべて提供すること
4. 保守性：コードが理解しやすく、分析や拡張が容易であること
5. パフォーマンス：一般的なハードウェアでも効率的に動作すること

スケジュール

以下が主な開発項目とスケジュールです(3ヶ月ごと)。なお、各項目の依存関係は、カルダノネットワークでの動作検証に必要な範囲に限定しています。

1. 調査フェーズ(完了済)
 - Praos (Genesisを含む?)の動作モデルの構築
 - 他の境界コンテキスト(ネットワーク、メモリプール、台帳)とのインターフェースと関係性の定義
2. テストフェーズ(完了済)
 - コンセンサス仕様と既存の実装に基づく適合性テストツールと関連テストスイートの構築
 - 既存のouroboros-consensusやAmaruの将来実装を含む、任意のOuroborosの実装に対して適合性テストスイートを実行可能にする
3. 検証フェーズ(2025年第2～3四半期)
 - チェーンをフォローしブロックを検証できるコンセンサスバージョンの構築(ヘッダーとトランザクション検証を含む)

- 既存のノードとネットワークに対する検証ノードの適合性の実証
- 依存関係: 台帳の状態アクセスとトランザクション検証、ネットワークn2nプロトコル
- 4. ブロック生成フェーズ(2025年第3~4四半期)
 - VRFベースの事前定義されたスケジュールに基づき、メモリプールからトランザクションを含むブロックを生成できるコンセンサスバージョン
 - 既存ノードに対するブロック生成スケジュールとトランザクション選択ロジックの適合性スイートの検証
 - 依存関係: 台帳、ネットワーク+メモリプール
- 5. ギャップ分析と構成(2025年第4四半期)
 - Amaruの現在の実装とGenesisおよびLeiosの必要構成間のギャップのレビュー
 - これらの要件に準拠したAmaruのセットアップ構築

必要なリソース

Rust開発者: 2.5 FTEs

すでに確保しているリソースと資金

Rust開発者: 1 FTE (Arnaud: 0.5、Pawel: 0.5)

追加で必要となるリソース

Rust開発者: 1.5 FTEs

以上がコンセンサス開発の概要です

オーナー: **Arnaud Bailly**

プロジェクトマネジメント・広報&マーケティング活動

実施範囲とスケジュール

実施範囲

目標: 主要な取り組みを推進し、実際のユーザーの声を製品開発に活かすサイクルを確立します。

具体的な活動計画は以下の通りです:

- SPO(ステークプール運営者)との協力体制を構築し、Amaruの運用面での改善を進めます
- 製品の重要機能に関するワークショップを企画・実施します

- マーケティング活動の展開、導入支援のための提携、特定用途向けの協業を進めます
- 開発ロードマップの調整と範囲の明確化を行います
- バグ報奨金制度:テストネットとプリプロダクション環境でのトラフィック創出と、開発者の参加を促進します

テストネット開発者向け:Amaruチームが定めた要件に沿って、ネットワーク上でトランザクションを生成し、実際の利用シナリオを構築

テストネットSPO向け:上記の利用シナリオを含むブロックの生成

Amaru開発貢献者向け:3件以上のコードが正式に採用されること

現在の目標は、テストネット環境での実験、適合性テスト、シミュレーションを可能な限り実施し、初期ユーザーからのフィードバックを活かしながら、私たちの全体目標を達成できる信頼性の高いシステムを構築することです。

各部門は、上記の活動を円滑に進めるため、Amaruプロジェクトの関係者全員と協力して必要なリソースを共有します。

スケジュール

- SPOワーキンググループの立ち上げと主要活動:2025年第3四半期
- バグ報奨金プログラムの開始:2025年第3四半期
- Amaruワークショップ:3ヶ月ごとに開催

必要なリソース

Rust開発者:0.5 FTE

すでに確保しているリソースと資金

0 FTE

追加で必要となるリソース

0.5 FTE

以上がプロジェクトマネジメント・広報&マーケティング活動の概要です

オーナー:Damien Czapla

境界づけられていない活動

このセクションでは、プロジェクトのソフトウェア開発における基本的な要件と、境界づけられていない活動の特徴について説明します。

Amaru製品の統合

各開発チームは、担当する機能をAmaruに組み込める形で提供する必要があります。具体的には以下の点を満たす必要があります：

- リファクタリング: 開発したコードを現在のAmaruのコード構造に合わせて最適化すること
- ベンチマーク: 機能の性能が既存の標準と同等以上であること
- テスト: Amaruに実装後、想定環境での動作確認レポートを提出すること
- パフォーマンス: 期待される性能を測定し、保守管理チームと協議すること

各チームに求められること：

- 方向性の一致: 成果物はデモで示された予測と整合し、保守管理チームの承認を得ること
- 品質管理: 各実装は明確な方法論に基づき、Amaruの基準を満たすこと
- 問題解決: 担当範囲で発生した問題やバグの解決に責任を持つこと

プロジェクト管理の主な課題は、チーム間の連携を図り、メンバーの貢献を確保しながら、リスクに備えて予期せぬ事態に対応できる体制を整えることです。これは各チームリーダーと保守管理チームにとって重要な責務となります。

Cardanoの研究開発と仕様策定

シニア開発者にとってソフトウェアの実装は自然なことです。私たちのプロジェクトはCardanoエコシステムに関する最先端の水準に見合うものであることを目指しています。

各境界コンテキストに関する期待事項は以下の通りです：

- 仕様の参照: 実装される各コンポーネントは、その範囲をカバーする仕様を参照する必要があります。
- 仕様のクロスチェック: コンポーネントの統合版の機能とパフォーマンスは、仕様に含まれる基準値または目標値によって検証される必要があります。もし仕様がパフォーマンスの結果と一致しない場合は、仕様を更新するか実装を修正する必要があります。
- 仕様の更新: 仕様に含まれていないパフォーマンス境界に影響を与える機能実装やリリース

は、追跡され説明されなければなりません。それによって文書を更新するか、存在しない場合は新たに文書を作成します。

- 研究との整合性：実装される範囲が既存のホワイトペーパーや研究文書の状態に疑問を投げかける場合は、研究カテゴリーに関わる人々と共に対処する必要があります。

個別のユースケースへの対応、問題解決、バグ修正

私たちはノードを機能ごとに分割し、様々な使用方法に対応できるようにします。各チームリーダーには、担当する機能について、これらのニーズに応える責任があります。

実装やユースケースが増えるにつれて発生する問題やバグについても適切に対処し、Amaruの安定した動作を確保していきます。

アドホック開発チームの編成

上記の活動と2025年第3四半期以降の不確実性を考慮し、「アドホック開発チーム」として2名分の人員を確保することにしました。

アドホック開発チームの役割について

このチームは、必要に応じて各開発チームを支援し、具体的な成果を出すための即応部隊として機能します。「この実装を手伝ってほしい」といった要請に、どのチームからでも柔軟に対応できる体制を整えます。

アドホック開発チームのオーナー：Pi Lanningham

メモリプール（アドホック開発）

プロジェクトの他の部分と密接に連携する必要があるため、アドホック開発に割り当てる作業範囲の例を説明します。

実施範囲とスケジュール

実施範囲

この部分では、Cardanoと互換性のあるメモリプールの実装と保守を行います。具体的には以下の内容が含まれます：

- メモリプール・ライブラリ：これらのトランザクションの抽出を取得するインターフェースを備えた独立したライブラリ（メモリ内のトランザクションを表すデータ構造）を構築する

- 「シンプルな」メモリプール実装: トランザクションの追加、削除、収集、排出を行い、新しい台帳状態を公開する基本的なメモリプールを構築する
- 「複雑な」メモリプール実装: 機能を異なる方法で処理し、全体的なリソース消費を最適化するメモリプールのより洗練されたバージョンを構築する
- メモリプール・ツールとAPI: メモリプールとモジュール性を管理できるツールを作成する
ノード管理リモートプロシージャコール (RPC): Amaruノードの操作における「オペレーターの視点」を処理するソフトウェアを構築する
- ブロックフォージ: ブロックをフォージできるコンポーネントを開発する

スケジュール

- メモリプールライブラリ: 開発完了済み
- 「シンプルな」メモリプール実装: 2025年第2四半期
- ブロックフォージ: 2025年第3四半期
- 「複雑な」メモリプール実装: 2025年第4四半期

必要なリソース

Rust開発者: 0.5 FTE

すでに確保しているリソースと資金

0 FTE

追加で必要となるリソース

0.5 FTE

以上がアドホック開発チームの実施範囲です。

メモリプールのオーナー: Pi Lanningham

まとめ

本提案は、見積もりと貢献の詳細に示す通り、Amaruプロジェクトの開発範囲全体で75万ドルの予算を必要としています。

Amaruプロジェクトは、モジュール式で高性能、かつ相互運用可能なブロック生成ノードを開発することで、Cardanoエコシステムのノードの多様性を高め、システム全体を強化することを目指しています。

す。私たちは、オープンで革新的、そして柔軟な運用を重視しながら、セキュリティや運用性を損なうことなく、ブロックチェーン技術の発展に貢献したいと考えています。

私たちは、綿密な設計と実装計画に基づき、現実的な開発範囲とスケジュール、そして必要なリソースを明確にしました。すでに確保している開発範囲、資金、人材についても詳細を示し、透明性の高いプロジェクト運営を心がけています。今後の見通しについては楽観的ですが、予想される課題にも備え、適切な対策を講じています。

今回申請する75万ドルの資金は、これまでの基礎的な開発を実用的なシステムへと発展させるために必要不可欠です。優れた技術力を持つチームと確かな開発体制により、プロジェクトの成功を確信しています。

この提案は単なる資金調達の要請ではなく、革新的な開発への参加の呼びかけです。Amaruプロジェクトは野心的な目標を掲げていますが、貢献者とCardanoコミュニティ全体の力を結集することで、必ず達成できると信じています。開発と実証を重ねることで、Amaruは分散化されたCardanoエコシステムの実現に貢献し、長期的な信頼性とイノベーションを確保していきます。

一緒にCardanoの未来を創造しましょう。