# Treasury Withdrawal - Pallas: Sustaining Critical Rust Tooling for Cardano

## **Title**

Withdraw ₳220,914 for Pallas: Sustaining Critical Rust Tooling for Cardano

## **Abstract**

This treasury withdrawal funds **Pallas: Sustaining Critical Rust Tooling for Cardano** which will provide the following services:

This request for support focuses on Pallas (github.com/txpipe/pallas), a collection of Rust-native building blocks for the Cardano blockchain ecosystem. Pallas provides reusable components, such as cryptographic primitives and CBOR encoding, to enable the development of higher-level use cases like explorers and wallets. The project remains open-source, and actively welcomes contributions from the broader Cardano developer community. With 591 commits, over 430 pull requests, and 39 contributors, it stands as a strong example of open-source collaboration in the ecosystem. Pallas is being used by key projects in the ecosystem such as Aiken, Lucid, Mithril, Amaru, and many others. To ensure its continued evolution and maximize its impact, Pallas requires ongoing maintenance, updates, and new functionalities. We are requesting funding to secure the following roles assigned to the project: 0.5 FTE blockchain developer and 0.125 FTE tech lead.

This Treasury Withdrawal is submitted by Intersect on behalf of the vendor. The following sections; Abstract, Motivation, Rationale and Vendor Profile have been sourced from the approved proposal submitted by the Vendor as part of the Intersect budget process.

This treasury withdrawal funds one of 39 proposals to give effect to the approved budget info action for ₳275,269,340, administered by Intersect via gov\_action1u9x73kwufaxa70lfy59g4ynwyrcsaxdcd0gxzzmh67s9fxq4j8hqqk2phgh. The information provided herein is intended to fulfill the spirit of the constitutional requirement for a treasury withdrawal info action by also providing the details of the proposed solution, alignment to the budget, and amount to be withdrawn from the Cardano Treasury.

## **Motivation**

This proposal aims to solve the following problem:

Pallas has become a critical Rust-native foundation for Cardano development, providing essential blockchain primitives such as cryptographic primitives and CBOR encoding. However, maintaining an open-source project of this scale requires continuous effort—bug fixes, updates to support protocol changes, performance optimizations, and new feature development. Without dedicated resources, Pallas risks stagnation, which could impact the many projects and developers who rely on it. This proposal seeks funding to ensure Pallas remains well-maintained, up-to-date, and capable of evolving alongside the broader Cardano ecosystem.

## **Rationale**

### Project Solution

Securing ongoing funding for Pallas directly benefits the Cardano developer community by ensuring its continued reliability and performance as a key open-source resource for the Cardano ecosystem. Projects like Aiken, Lucid, Mithril, and Amaru already utilize Pallas to streamline development and reduce infrastructure complexity. By maintaining Pallas, we help developers focus on building applications rather than managing low-level blockchain details. This support strengthens the ecosystem’s developer tooling, makes it easier for new developers to contribute, and fosters innovation across various Cardano-based projects.

### Vendor Profile

TxPipe is an active member of the Cardano ecosystem

TxPipe has been developing open-source tools for the Cardano ecosystem for over 3 years and we're not going anywhere. Evidence of our commitment can be found by evaluating the continuous activity of our public code repositories.

Experience developing in the Cardano ecosystem

TxPipe has helped developed several dApps for the Cardano ecosystem. This experience allows us to evaluate the feasibility of the project and its potential benefit from a developer's perspective. We've also participated in development of chains using the released version of the IOG's Partnerchain SDK.

Successful Catalyst proposals

We have successfully completed several Catalyst proposals. This may serve as evidence that our team has the required capabilities to fulfill these type of projects.

Development process will be public and open-source

Both the output and the development process will be public and open-source. This approach provides an easy way for the Catalyst team and the Cardano community to evaluate the progress at each step of the process.

### Contract Management

A written off-chain Legal Contract will be created between the Vendor and the Cardano Development Holdings (CDH), as mandated by the constitution, and will be administered by Intersect. This will include details of the project delivery schedule and dispute resolution.

### Project Delivery

All milestones, acceptance criteria, payment amounts and expected delivery dates will be agreed between the Vendor and Intersect, acting on behalf of the CDH. The vendor will deliver according to the agreed-upon project schedule within the Legal Contract, of which the necessary information will be made public via the budget management platform via transaction metadata.

Defined by the milestones within a Legal Contract, the vendor will submit and attest milestone acceptance to the community, Intersect or 3rd Party Assurer.

Project progress will be monitored via Intersect’s delivery assurance function which will be communicated to the community.

Acceptance of the above work is expected to be supported by a 3rd Party Assurer, who will be responsible for reviewing and signing off the work completed at each project milestone against the corresponding milestone deliverables detailed within the Legal Contract.. This work is funded from a portion of this treasury withdrawal.

### Budget Management Tooling

To administrate treasury funds on-chain, Intersect will utilize the treasury management smart contract framework developed by Sundae Labs. The smart contracts have been [extensively tested](https://github.com/SundaeSwap-finance/treasury-contracts/tree/main/offchain/tests) including audits from TxPipe and MLabs. Examples of the usage of these contracts can be seen across mainnet described across Intersect authored [Blog 1](https://www.intersectmbo.org/news/smart-contract-mainnet-demo-a-step-toward-on-chain-treasury-withdrawals), [Blog 2](https://www.intersectmbo.org/news/smart-contract-mainnet-demo-day-two-update) and [Blog 3](https://www.intersectmbo.org/news/smart-contract-mainnet-demo-day-three-update).

Final mainnet validation test can be seen via the Disburse action within transaction: 0f591dc544ae14102dbb4a74d5311a6acffc1772b163d8b7a9656b9525950b17

With the confirmed treasury reserve contract address being: stake17xzc8pt7fgf0lc0x7eq6z7z6puhsxmzktna7dluahrj6g6ghh5qjr

#### Specifics

Intersect will utilize a single Treasury Reserve Smart Contract (TRSC), with many Project-Specific Smart Contracts (PSSC), managed by Intersect. Intersect’s management consists of three ‘admin’ and two Intersect ‘leadership’ roles. An Oversight Committee consisting of five external, independent third-party entities will provide checks and balances on Intersect, and safeguard against errors and unilateral control. The administration of both TRSC and PSSCs will be managed by Intersect, with external oversight on certain actions from the Oversight Committee.

The Oversight Committee consists of Sundae Labs, Cardano Foundation, Dquadrant, Xerberus and NMKR. Their role is to independently verify key administrative actions using on-chain logic, ensuring accuracy and consistency without exercising discretion over governance decisions.

For all details on Intersect’s configuration please see the [**Smart Contract Guide**](https://docs.intersectmbo.org/cardano-facilitation-services/cardano-budget/intersect-administration-services/smart-contracts-as-part-of-our-administration) on the knowledge base.

The high level permissions are as follows:

* TRSC Fund and PSSC Modify
  + Two of the three Intersect admins, two of the five trusted entities and one of the two Intersect leadership sign-off must authorize
* TRSC Disperse
  + Two of three Intersect admins, three of five trusted entities and two of two Intersect leadership sign-off must authorize
* TRSC Pause and Resume
  + Two of three Intersect admins, and one of two Intersect leadership sign-off must authorize
* TRSC Sweep
  + One of three Intersect admins, and one of two Intersect leadership sign-off must authorize
* TRSC Reorganize
  + Two of three Intersect admins and three of five trusted entities must authorize

#### Processes

Upon enactment of this governance action, funding for this project will be directed into the TRSC’s stake account. The TRSC and PSSC can not be staked with a SPO and will be delegated to the auto-abstain predefined DRep. From here funds will be withdrawn into a UTxO remaining at the TRSC.

When the Legal contract is prepared and the vendor is ready, funding for this project will be transferred using the Fund action to a PSSC. All milestones will be outlined within the metadata.

A dashboard will be available for the community to audit the TRSC or PSSC and track metrics related to this withdrawn ada as well as being immutably verifiable on chain.

## **References**

TxPipe Pallas Github Repository

* <https://github.com/txpipe/pallas>

Project Proposal In Ekklesia

* <https://2025budget.intersectmbo.org/ballots/680d1b63565577986442d123/proposals/680d1b64565577986442d296>

Approved Budget Info Action submitted by Intersect via GovTool

* https://gov.tools/outcomes/governance\_actions/e14de8d9dc4f4ddf3fe9250a8a926e20f10e99b86bd0610b77d7a054981591ee#0

Details of all successful proposals (CSV)

* ipfs://bafybeicwrop4q7xvnyjdd5drumbe56sqtm5lbe2ul3c262zt4hgguzdycm

Automating Accountability: Cardano’s Smart Contract Framework Blog

* ipfs://bafybeihqx4ae72z7suqfnxrpqpqithp43cai7o2uuewnqtezgaoyc3ptyq

Sundae Labs Budget Management Smart Contracts Github Repository

* https://github.com/SundaeSwap-finance/treasury-contracts

Budget Management Smart Contracts TxPipe Audit Report

* ipfs://

Budget Management Smart Contracts MLabs Audit Report

* ipfs://

## **Authors**

* Intersect