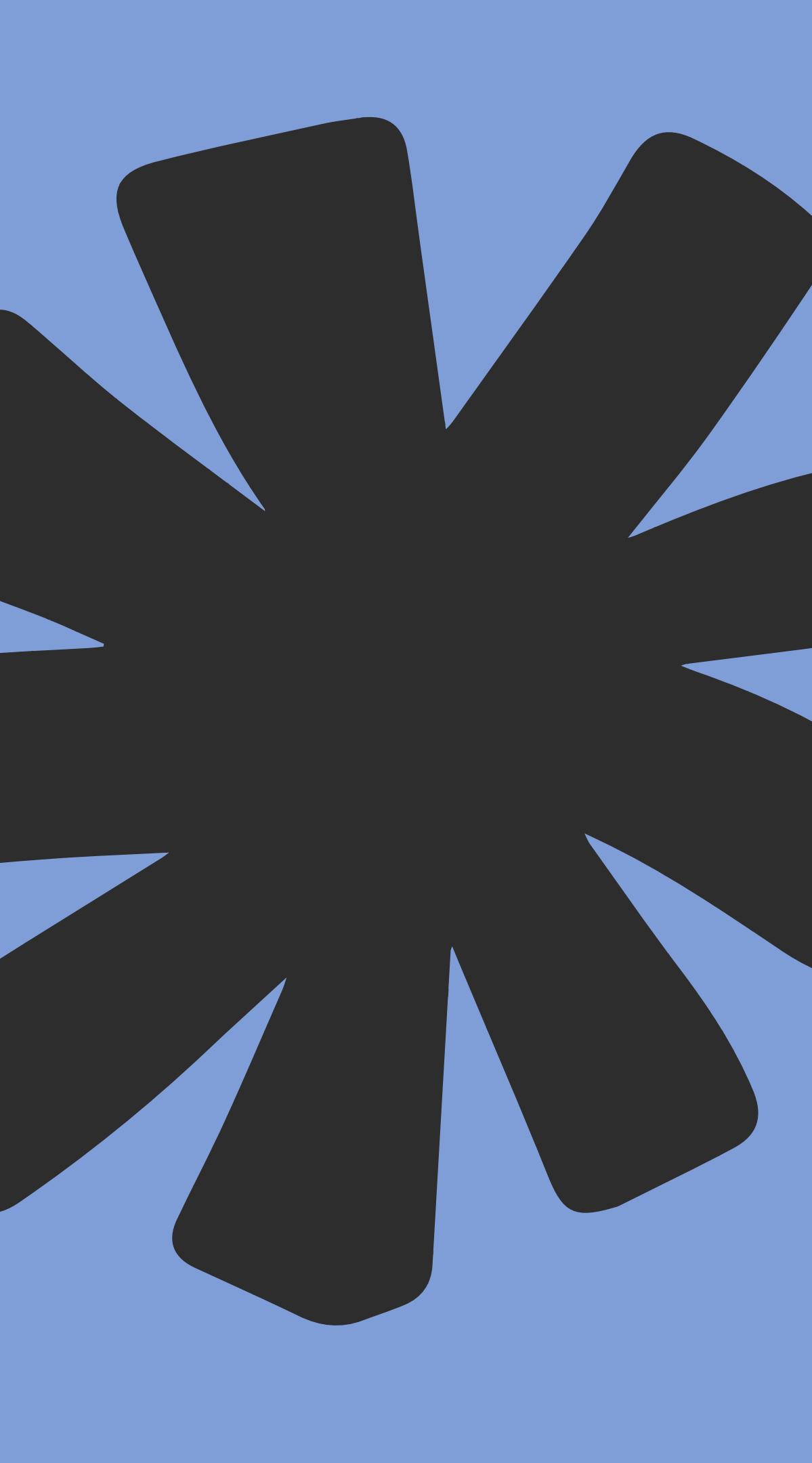


AlgoBounty

Trustless Bounties for Open-Source
Issues, Powered by Algorand.

* AlgoBounty

Hayden Bradley
EASYA x ALGORAND HACKATHON



A quick introduction



Full Stack Dev **Hayden Bradley**

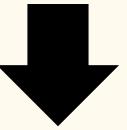
A master's student at KCL, studying advanced software engineering after spending 4 years in industry.

Using my expertise in product management and full-stack development, I have tried to put together a product I had been wanting to make for a long time.



The Open Source Problem: Unfunded Issues & Trust Barriers

- Maintainers face **massive backlogs** of issues but have limited time and resources
- Volunteer-driven fixes **stall** due to **lack of incentives**
- Paid open-source efforts are rare, with **low trust** payments will be honoured



There is no seamless, trustworthy way to reward open-source contributions

Introducing my solution: AlgoBounty

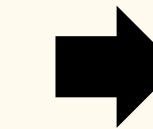
AlgoBounty aims to empower open-source maintainers to attach trustless, verifiable bounties to their GitHub issues, leveraging Algorand's blockchain for escrowed payments.

Our mission is to empower open-source innovation by creating trustless, accessible opportunities for developers worldwide to contribute, collaborate, and earn fairly.

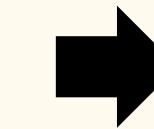
So how does this work?

All it takes is three simple steps

Our GitHub App
automatically adds “**Fund
Issue with AlgoBounty**”
button on a GitHub issue.



User's can then contribute
into an **escrow smart
contract** on Algorand for
the issue and transfer ALGO
into it.



Once the issue is closed, the
maintainer can **attribute the
payment** on the blockchain
and that user can then **claim
their reward**.

Multiple people can fund the same issue bounty, crowdfunding the reward pool to increase incentive.

Why use AlgoBounty as a Developer

01.

Get Paid Fairly, Instantly

Earn verified rewards in ALGO through trustless smart contracts – no middlemen, no delays.

02.

Build Reputation & Impact

Contribute to real open-source projects while showcasing your skills and contributions on-chain.

03.

Global, Frictionless Access

Join from anywhere – no bank accounts, no barriers, just code, contribute, and collect.

Why use AlgoBounty as a Maintainer

01.

Accelerate Issue Resolution

Attract skilled contributors with transparent, escrow-backed bounties that motivate faster fixes.

02.

Earn Community Support

Let your users and sponsors directly fund the issues that matter most to your project.

03.

Trustless, Hassle-Free Payments

Securely manage and distribute rewards without handling funds or paperwork — blockchain does the work.

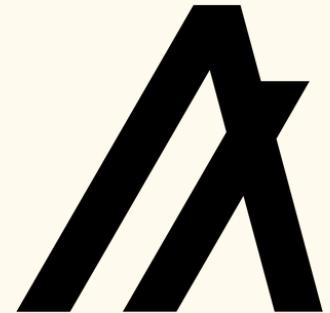
What made Algorand so great for this?

Speed

3.8 second finality on average

Trustless Execution with Global Reach

No need to worry about currency conversion or bank integrations



Low Fees

\$0.001–\$0.005 per transaction

Consistent and no forking

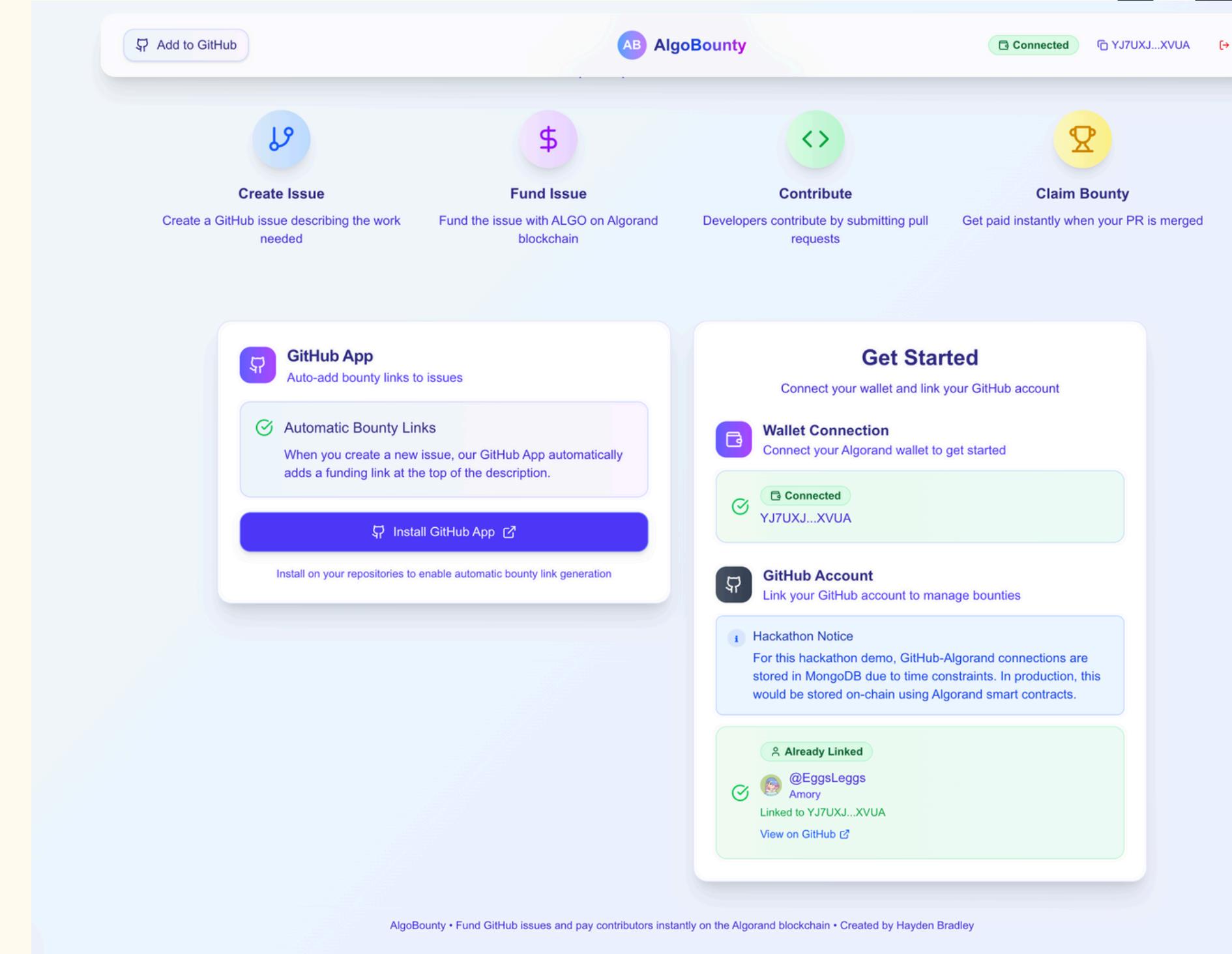
The outcome of the bounty contract is always clear and cannot be reverted

What does this look like?

Stage 1: Link your GitHub

To automatically post AlgoBounty links to your repository's issues, connect the GitHub App. It listens for issue creation and invite events, then posts verified bounty links directly to the corresponding GitHub issues.

To link your wallet with your GitHub account, a `github_link` smart contract offers a `link()` method using Ed25519 attestations and box storage to securely verify your identity, prevent replay attacks, and ensure scalability. This creates a trustless, verifiable bridge between GitHub identities and Algorand wallets.



* AlgoBounty

What does this look like?

Stage 2: Posting an issue

When a new issue is posted, the webhook picks up the request and the description is prefixed with a link to the issue.

Behind the scenes, the IssueEscrow smart contract powers this process. When a new bounty is initialised via `create_bounty()`, the contract sets up a dedicated escrow tied to the GitHub issue ID. Users can then contribute funds through the `fund_bounty()` method, which aggregates ALGO deposits into the escrow's total balance

* AlgoBounty

The image displays two screenshots illustrating the AlgoBounty process. The top screenshot shows a GitHub issue titled "Feature - Create AlgoBounty #6" with the status "Open". The description includes a link to "Fund this issue with AlgoBounty". The bottom screenshot shows a web-based "Fund Issue" interface for AlgoBounty. It features a "GitHub Issue URL" input field containing the URL of the GitHub issue, a summary of the issue, and a "Current Bounty Balance" of "0.0001 ALGO". Below this, there are sections for "Add to Bounty (ALGO)", "ESCROW ALGO (Native)", "PAYOUTS On-chain split", and "SPEED ~4s finality". At the bottom, there are notes about "Low fees, instant finality" and "Trustless escrow", and a prominent "Add to Bounty >" button.

What does this look like?

Stage 3: Claim your bounty

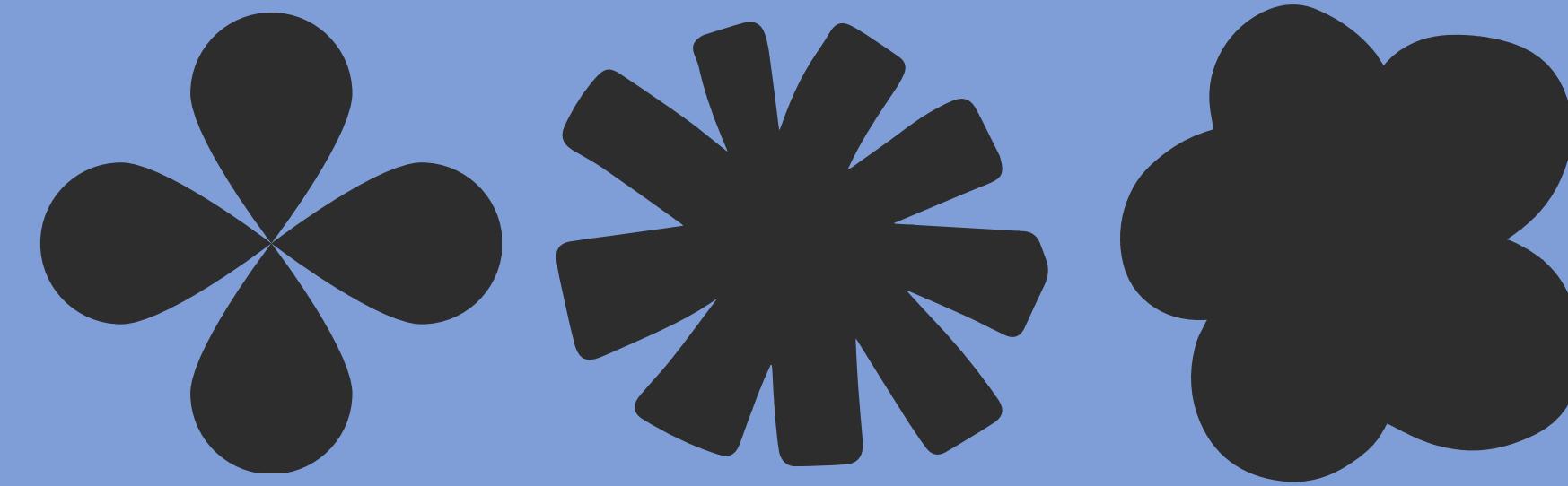
Once the issue is resolved, maintainers can trigger `distribute_payout()` to automatically release rewards to contributors, or invoke `refund()` if the issue remains unresolved. All transactions are atomic, transparent, and verifiable on the Algorand blockchain, ensuring complete trust without intermediaries.

The authenticity of a maintainer, including verifying that they are allowed to distribute payments, is handled similarly to account linking by signing their requests with the same Ed25519 attestation methods on the server.

* **AlgoBounty**



AlgoBounty
on GitHub



**Together, we can make open source
more accessible, equitable, and
rewarding for everyone.**

* AlgoBounty

LinkedIn

