ZCG Grant Proposal: ZEC Order Pay

Shielded ZEC checkout, refunds and creator payouts for service marketplaces · Category: Non-Wallet Applications

Applicant: Hennadii Kovalenko · hennadii@titanhold.app · Telegram @Barinoff88 · October 2026

Project	ZEC Order Pay: MIT-licensed Python library + Rust helper + reference integration
Requested amount	**USD 36,000**, four milestones; milestone 1 (startup) = USD 6,000
Applicant	Hennadii Kovalenko, solo founder and developer, relocating to Portugal
Legal entity	TITANHOLD LLC formed 14 October 2026 (Wyoming, USA).
Previous ZCG funding	None (first application)
License	MIT (as required by the ZCG Grant Agreement)
Payout	Shielded ZEC, USD-denominated per milestone
Website	https://titanhold.app · X: @Titanhold_app

1. Summary

An MIT-licensed Python library and reference integration that lets any web marketplace accept shielded ZEC per order, detect payment with a viewing key only, issue refunds and pay out creators, demonstrated live on titanhold.app.

2. Problem

Integrating shielded ZEC into a small service marketplace is still too hard: there is no lightweight library for Python web stacks, no ready model for escrow-like order flows (refunds, acceptance, payouts), and keeping spending keys on a web server is a security risk. Most indie marketplaces are Python/FastAPI or Django apps with order states (paid → delivered → accepted/refunded) that general payment processors do not model.

3. Solution

A drop-in module: create an order → get a fresh shielded Unified Address and ZIP-321 payment URI/QR → payment detected via UFVK (no spending keys on the server) → order states, partial payments, expiry and refund-address handling → batched creator payouts prepared as unsigned transactions and signed offline. It ships with a reference checkout on titanhold.app so real buyers can pay in shielded ZEC.

How it differs from existing work: ZCG already funds the BTCPay Server Zcash plugin, and CipherPay exists as a self-hosted shielded processor. ZEC Order Pay is not another general payment processor. It is a small library for service marketplaces with acceptance, refund and payout logic, and it can use CipherPay or zcash-walletd as backends instead of duplicating them.

4. Technical approach

- Diversified Unified Addresses (Orchard + Sapling receivers) derived from a UFVK per order.
- Compact-block scanning with trial decryption in a Rust helper built on `zcash_client_backend`, `zcash_keys`, `zcash_address`, exposed to Python via a small CLI/gRPC; optional CipherPay / zcash-

walletd backend.

- Confirmation thresholds per order value; exchange-rate lock at invoice time; idempotent webhooks.
- Refunds via a refund UA collected at checkout; creator payouts exported as PCZT / unsigned transactions for offline signing.
- Extensive regtest/testnet tests; lightwalletd or zaino endpoint (public or self-hosted).
- No fork of core Zcash repos. Any fixes found (e.g. ZIP-321 parsing) go upstream as separate PRs following librustzcash CONTRIBUTING rules.

5. Milestones

#	Deliverables and acceptance	Due	Amount
M1	Public MIT repo with CONTRIBUTING.md; Rust helper (UFVK scan) + Python package alpha; testnet demo page. *Accepted when* community members pay the testnet demo from Zashi/Ywallet and see the order marked paid.	2026-12-15	**$6,000**
M2	Order state machine (pending, paid, underpaid, expired, refunded), partial/over-payment handling, expiry, refund-address flow, webhooks, regtest test suite, PyPI beta. *Accepted when* CI passes and an end-to-end refund is demonstrated on testnet.	2027-01-31	$10,000
M3	Batched creator payouts exported as unsigned transactions for offline signing; FastAPI reference app; docs site. *Accepted when* a batch payout to ≥5 recipients is prepared by the library and signed offline on testnet.	2027-03-15	$10,000
M4	Mainnet checkout live on titanhold.app; external security review + fixes; merchant guide for ZecHub; v1.0 release. *Accepted when* a community member completes a live mainnet shielded payment and the review report is published in the repo.	2027-04-30	$10,000
	Total	**~5 months**	**$36,000**

6. Budget

Item	USD	Justification
Hardware/Software	1,800	Always-on mini PC with 2 TB NVMe for a full node + lightwalletd/zaino for testnet/mainnet scanning tests ($1,500); domain/TLS/monitoring for the public demo ($300).
Services	3,000	Independent security review of key-handling and payout code by an external Zcash/Rust developer (≈3 days).
Compensation	31,200	1 developer × ~693 h at $45/h over ~5 months.
Total	**36,000**	

7. Risks and success metrics

Risks: scanning performance on small servers; wallet compatibility with ZIP-321 multi-receiver URIs; exchange-rate volatility during payment windows; solo-developer bus factor (mitigated by docs, tests and a public repo from day 1).

Success metrics: library on PyPI with ≥90% test coverage of the state machine; live shielded-ZEC checkout on titanhold.app; ≥25 real shielded payments in the 3 months after launch; ≥2 other merchants/projects integrating; merchant guide published on ZecHub.

8. Team

Hennadii Kovalenko, founder and sole developer. Builds and runs titanhold.app, a marketplace for AI-made work (images, video, bots, websites) generated on own hardware: Python/FastAPI backends, local LLM and media pipelines, Telegram bots; delivered paid client projects (e.g. Kraavi, Estonia, 2026). An external reviewer for the security pass will be named in milestone 4.

> **Current status (honest):** titanhold.app is live with the AI assistant and order intake; no Zcash code exists yet, and this would be the applicant's first Zcash project. Milestone 1 is deliberately small. TITANHOLD LLC formed 14 October 2026 (Wyoming). Other funding: parallel applications to Aleo and Oasis for separate, non-overlapping modules on their chains; no other funding for this Zcash work.