ZK-Whistle

A Decentralized, Trust-Minimized Architecture for Confidential Whistleblowing

Kerimberdi Agayev · **Giorgiana Violeta Vlăsceanu** · Charles Lee
POLITEHNICA Bucharest · Brown University · RoEduNet 2026, 19–21 November

What happens to the evidence if the source is silenced before it is published?

Where sources are weakest



The vulnerable interval

Evidence endangers its holder most while it is unpublished: silencing the source before release stops the disclosure.



Credible anonymity

An anonymous source cannot prove insider access, and any reward leaves a financial trail back to them.

Custodial platforms such as SecureDrop and GlobaLeaks secure transmission, but concentrate trust in one organization.

ZK-Whistle in numbers

4

cryptographic building blocks composed

3

Solidity smart contracts deployed

91

contract-level automated tests

6

design principles (P1–P6)

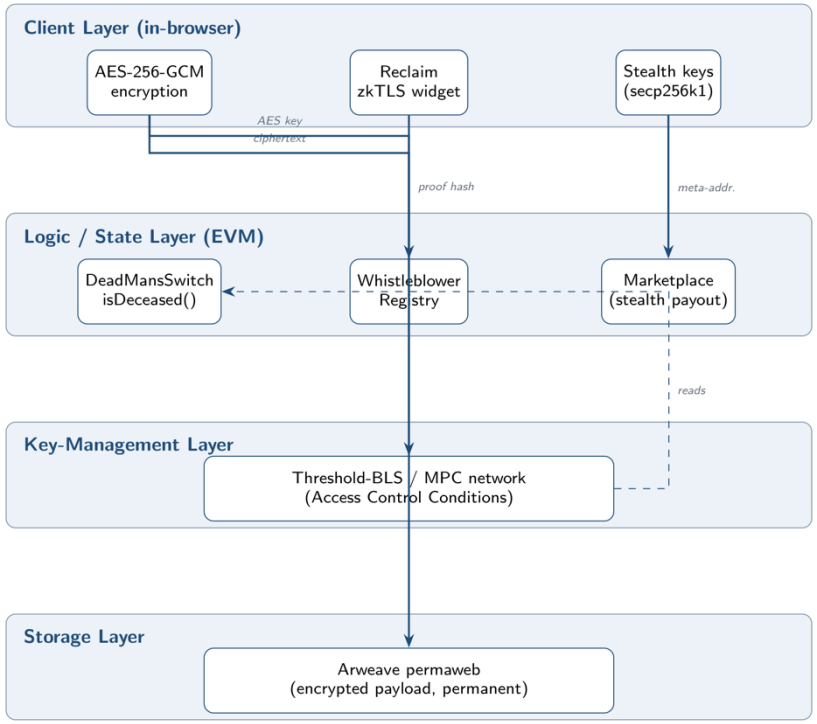
100 KiB

payload cap on the free storage tier

2.5%

fixed marketplace fee per accepted bid

Four layers, four separate systems



1

Client (in-browser)

All encryption and decryption; zkTLS widget; stealth keys

2

Logic / state (EVM)

3 contracts holding coordination data only

3

Key management (Lit)

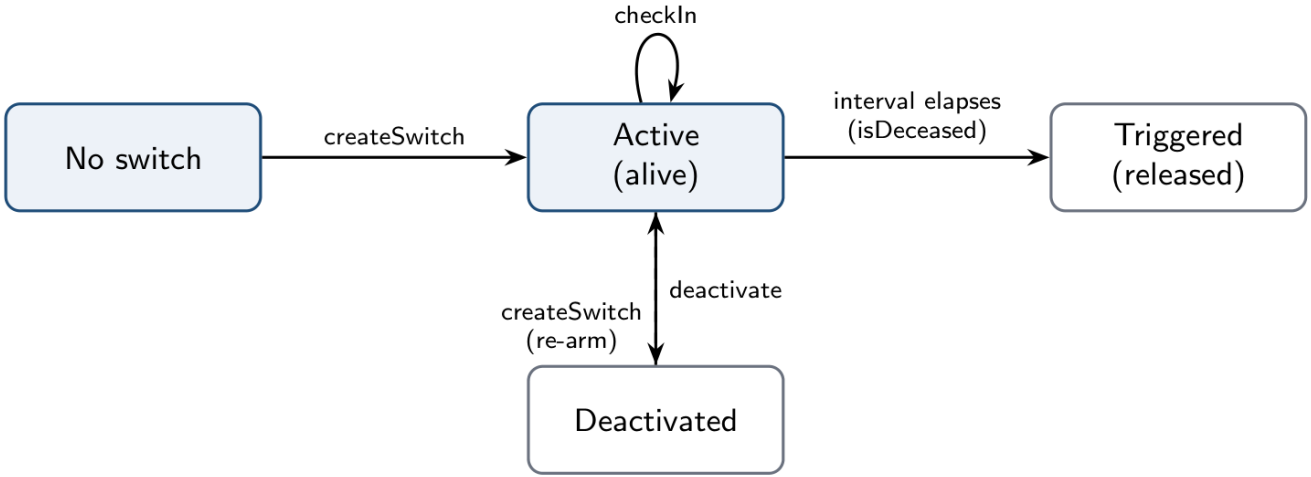
A t-of-n threshold quorum seals and releases the AES key

4

Storage (Arweave)

Permanent, public ciphertext

The switch: eligibility, not publication



t_now > t_last + Δ

Strict: at exactly t_last + Δ the vault is not yet eligible

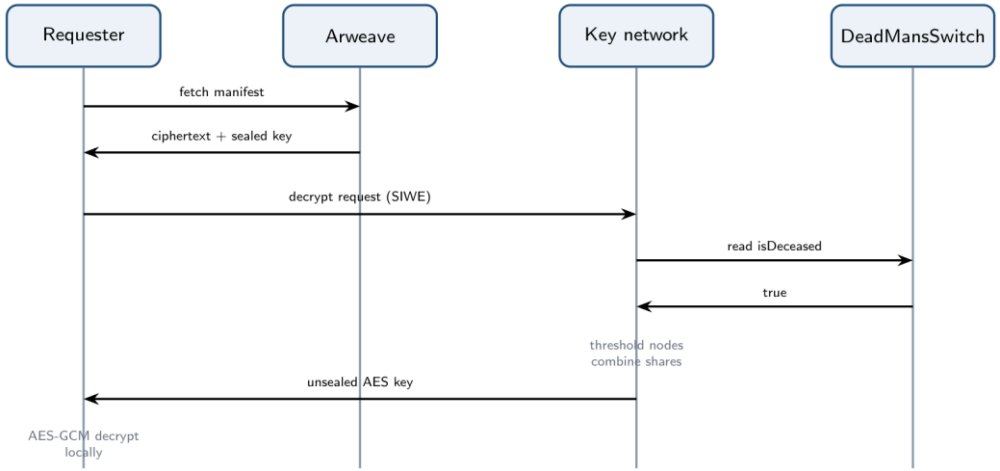
0 transactions

needed for a vault to become eligible; time alone does it

≠ proof of death

A missed check-in cannot tell detention from inconvenience

Release takes four steps after expiry



1. Requester fetches the manifest from Arweave
2. Requester authenticates (SIWE) and asks for the key
3. A quorum checks isDeceased() and returns shares
4. Payload is decrypted on the requester's device

Chain, Arweave, and the key network must all be available for release to succeed.

From one enclave to a t-of-n quorum

Single enclave

Prior design: Intel SGX via iExec

Assumption: One chip, one vendor

Failure: A single side-channel break exposes every key

Threshold quorum

This work: Lit SDK v8, Naga network

Assumption: Fewer than t of n operators collude; operators are independent

Failure: An attacker must compromise t operators

$\geq t$ compromised nodes expose the evidence early (ciphertext is public)

$> n - t$ nodes offline block release without compromising anything

Confidentiality and availability fail under different conditions.

Implementation in eight lines



Contracts

Solidity, Hardhat, OpenZeppelin



Client crypto

AES-256-GCM, fresh 96-bit IV per payload



Key custody

Lit SDK v8 on Naga (v7/Datil sunset 25 Feb 2026)



Provenance

Reclaim zkTLS, 3 verification tiers



Payments

ERC-5564 stealth addresses, @noble/secp256k1



Storage

Arweave via Turbo, ≤ 100 KiB free tier



Front end

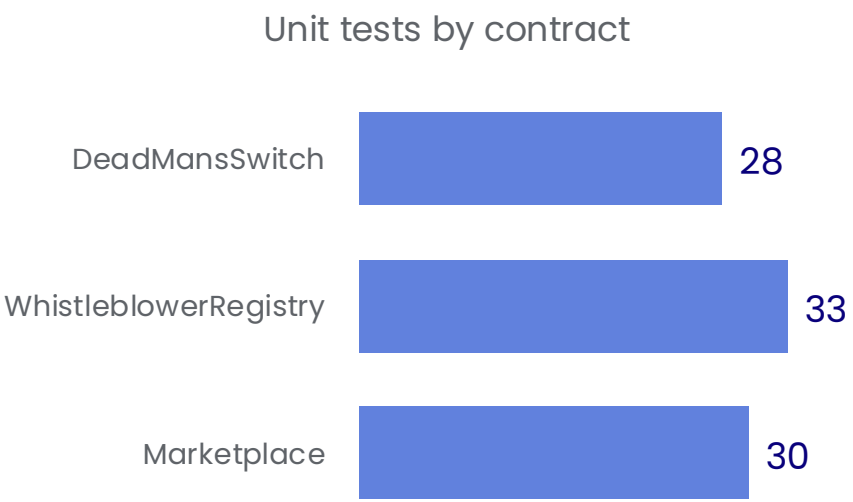
Next.js, wagmi, viem



Testnet

Base Sepolia, chain ID 84532

91 contract-level tests, all passing



What the tests assert

- isDeceased() is true at Δ + 1 s
- A check-in 10 s before expiry keeps the vault non-eligible
- A relayed proof credits only its owner
- Payout is exactly the bid minus 2.5%

The tests cover contracts only, not the client, Lit, Arweave, or Reclaim.

What is validated, and what is not

Contracts
Unit-tested and deployed on Base Sepolia

Client encryption
Exercised in the arming path

Key sealing
Exercised in the arming path

Arweave upload
Exercised in the arming path

On-chain zkTLS check
Mock verifier; no live proof reported

Stealth payout
Contract tested; no live payment trace

Key unsealing
Not validated live

Expired vs. unexpired test
Not yet performed

Recipient restriction
Not implemented

4 exercised or deployed

2 tested, not live

3 open

A fresh address hides where, not who

Linked to A

- Listing owner
- Verified flag
- Every check-in

Linked A → S

- acceptBid input carries S
- BidAccepted event: S and payout
- Payout = bid - 2.5%

Public on Arweave

- File name and MIME type
- Size and timestamp
- Condition naming A

A = source's registered address; S = one-time payout address.

System-wide unlinkability is a design objective, not a demonstrated property.

Six principles, three fully met

P1 — No single point of trust
Partially met

P2 — Client-side confidentiality
Implemented for file contents

P3 — No plaintext on-chain
Implemented

P4 — Autonomy
Live release not validated

P5 — Credible anonymity
Partially met

P6 — Graceful degradation
Implemented

3 implemented · 2 partially met · 1 awaiting live validation

Limitations and next steps

Limitations

- Live release not yet validated
- Recipient not enforced after expiry
- Payout linkable through acceptBid
- File name and type unencrypted
- Depends on Lit's operators and roadmap

Next steps

- Live expired vs. unexpired release test
- Testnet payment trace
- Recipient-restricted conditions
- Encrypt manifest metadata
- Lift the 100 KiB cap; measure gas and latency

Longer term: a professional security audit and formal verification of the switch and settlement logic.

Raising the bar from one party to t

1. ## Buildable now

   4 existing primitives, 3 contracts, deployed on a public testnet

2. ## Honest about scope

   91 contract tests; live threshold release still to be validated

3. ## Explicit trust

   Quorum, independence, and availability assumptions stated, not hidden



This work was supported by DACISLab: Virtual Laboratory on Open Data and Open Science in the New Generation of Continuum Computing Systems, project number PN-IV-PCB-RO-MD-2024-0364, within PNCDI IV.

Let's connect

kerimberdi.agayev@stud.acs.upb.ro · giorgiana.vlasceanu@upb.ro · charles_lee1@brown.edu