**Prize Targets:**

* ***Flow***
* ***Walrus***
* *Nora (be ready to show traces, etc.)*
* *Hardhat*

**Potential Project Description**

Your onchain bug bounty platform runs on Flow’s EVM, where Solidity contracts manage bounty creation, escrow, submission, review, and automated payouts. You write and test everything in Solidity with Hardhat, so the core logic stays fully on-chain and auditable end-to-end.

Sponsors post bounties and lock funds in escrow on Flow EVM; researchers submit findings by referencing an encrypted report instead of raw details. The contracts track states (OPEN → REVIEW → AWARDED/REJECTED) and release funds automatically when reviewers (sponsor) approve, creating a transparent, tamper-resistant trail for all decisions and transfers.

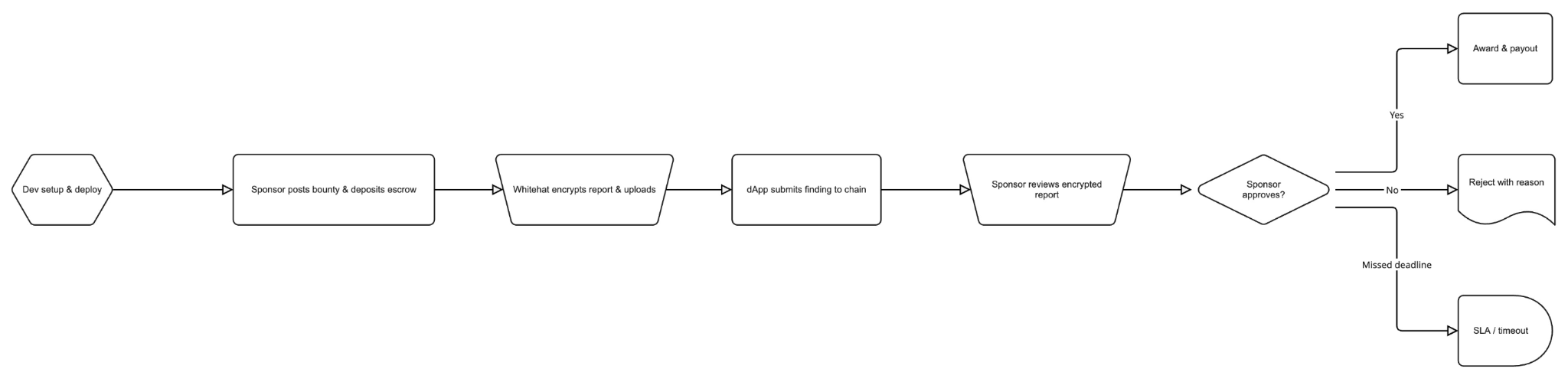
ETHGlobal

All sensitive artifacts (write-ups, PoCs, traces) are client-side encrypted and stored on Walrus, a decentralized storage/data-availability network. On-chain you store only the Walrus content IDs and hashes; reviewers decrypt off-chain with the right keys, and you can later publish a redacted version for public disclosure. This keeps exploits private during triage while remaining verifiable and robust.

Hardhat powers the dev/test loop (compile, unit/integration tests, scripted deployments), so you can iterate quickly and safely before shipping to Flow’s EVM. The same toolchain also supports clean migration scripts and reproducible deployments for hackathon judging.

Throughout, Nora (the “Solidity Cursor”–style AI agent) assists with contract scaffolding, audits-in-the-loop, and refactors—speeding up implementation and helping catch issues earlier. It complements your tests rather than replacing them, letting two developers deliver a polished, prize-aligned MVP fast.

**Graph**



**PRD:**

*Concept:* A bug bounty submission platform

*Value Prop:* Provides an anonymous but traceable way for ethical hackers to submit bug and vulnerability alerts to developers

*Users:*

* Developer: The original developer of an application, who are concerned that it may have unknown bugs or vulnerabilities and are willing to pay a “bounty” if someone finds one. They want a way to securely solicit this information.
* Hacker: An individual or group acting as “ethical hackers”, searching for bugs and vulnerabilities in hopes of submitting them to receive a bounty. They want a way to anonymously and securely submit information on any bug they’ve found.

*Hacker Side:*

* Should provide a way to provide bug info, including reproduction steps or details (text, photo, video)
* Fully anonymous, but verifiable/traceable, and some level of assurance that they’ll receive payment

*Developer Side:*

* Should provide alerts when a bug has been submitted
* Allows a way to verify bug and judge severity before releasing any payment to hacker

*Other Platform Requirements:*

* Should provide secure and decentralized storage for bug data [probably Walrus]
* Should provide a way for companies to prove their identity [Mint NFTs for them? Link to their sites?]
* Developers should have to stake a certain amount to participate

*Tech Stack:*

* Flow - used to deploy
* [Stablecoin TBD, likely USDF] - used to pay bounties
* Walrus - used to securely **store bug data** + **host our site**
  + Can host [next.js](http://next.js) site
* [Dev only] Nora - used for solidity development
* [Dev only] Hardhat 3 (MUST BE 3, NOT 2)... Maybe not if we use cadence?

Workflows:

*Bounty Report/Payment:*

1. Hacker submits a bug report via a web portal
2. Report is minted as some sort of token
3. Developer is alerted and receives some very basic info on the bug

Options:

1. *Bug contains an “expiration date” when the bug goes public*
2. *Developer “unlocks” the bug, which remains private until the expiration date.*
3. *Developer can pay the hacker whenever/whatever they want* [do Flow actions fit here?]
4. *Bug goes public on the expiration date [Flow agent???] (maybe not viable)*
5. *[Flow chaining to revert payment if the report is fake???]*

OR



*Bounty Negotiation:*

1. Hacker provides an initial price, lead time before publication, and impact assessment

—----------------------------------------------------------------------------------------------------------------------------

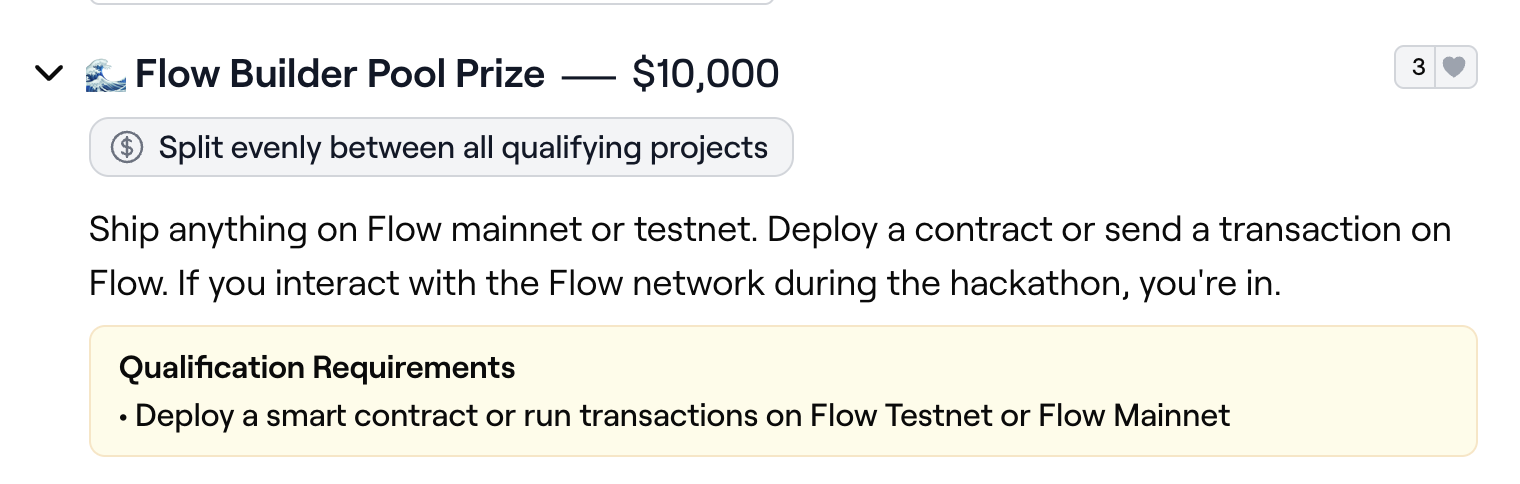
**Tracks to explore**

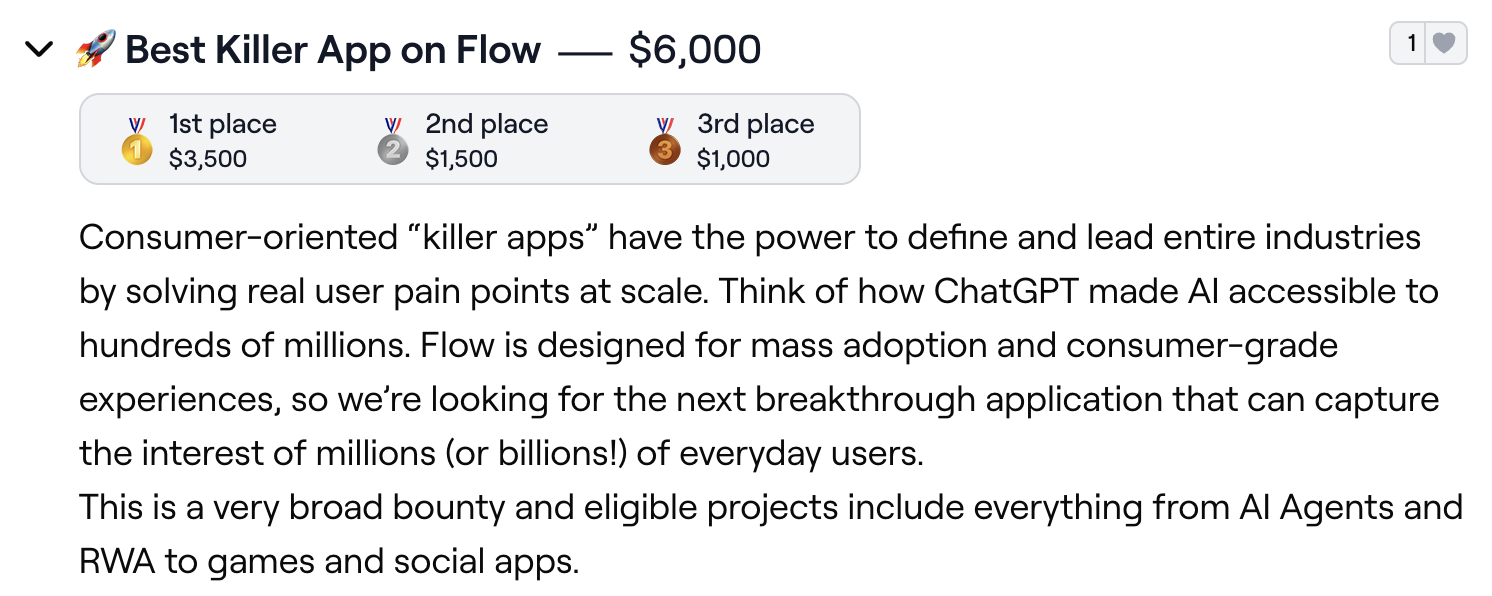
[**Nora**](https://ethglobal.com/events/newyork2025/prizes#nora) **- $1k:**

* AI coding assistant
* $1k to vibe code smart contracts?!?

[**Flow**](https://ethglobal.com/events/newyork2025/prizes#flow) **- up to $3.5k + $10k builder pool**:

* EVM who wants cool apps in their ecosystem
* Build whatever we want? (we want to stay open-minded about project direction)
* Sounds like free money
  + We might only have to swap an RPC endpoint to get the app on flow





[**Coinbase Developer Platform**](https://ethglobal.com/events/newyork2025/prizes#coinbase-developer-platform) **- $5k:**

* Tools to interact with the chain (these would be the most useful to us)
  + Coinbase Onramp
  + CDP Wallets (Server or Embedded)
  + CDP Data APIs (Token Balance, Event, SQL API)
* Make onboarding super easy for those who use our platform?

[**PayPal USD**](https://ethglobal.com/events/newyork2025/prizes#paypal-usd) **- Up to $4.5k:**

* Wants real-world applications
* Could be super nifty if we implement a payout mechanism
* Caveat: they definitely want more than just a payout mechanism :(
  + Fast UI?

[**Zircuit**](https://ethglobal.com/events/newyork2025/prizes#zircuit) **- $3k:**

* Secure, automated finance
* Supposedly just swap an RPC endpoint

[**Walrus**](https://ethglobal.com/events/newyork2025/prizes#walrus) **- $2k:**

* On-chain multimedia storage
* Whistleblowers can upload video and audio?
* Caveat: Sounds complex especially with storing and retrieving media
  + Stored media might not be private either

**Meeting notes**

Aug 14, 2025: <https://notes.granola.ai/d/8281d400-9404-4378-a4d1-b88778f8bc90>

**Operational Workflow**

Flow ($ 3.5k) → deploy smart contracts, guaranteed money track

Nora ($ 5k) → Solidity Cursor, AI agent tuned to smart contracts

Walrus ($ 2k) → data stored onchain, we stored all the encrypted files(e.g. reports)

\*HardHat → testing smart contracts

ZK IDs → proving that you put enough time into the application without revealing what you did to find the exploit. We’ll leave it to the end