GhostCheck : PoC and Arch Diagram

POC Overview — ZK GitHub Reputation Protocol :
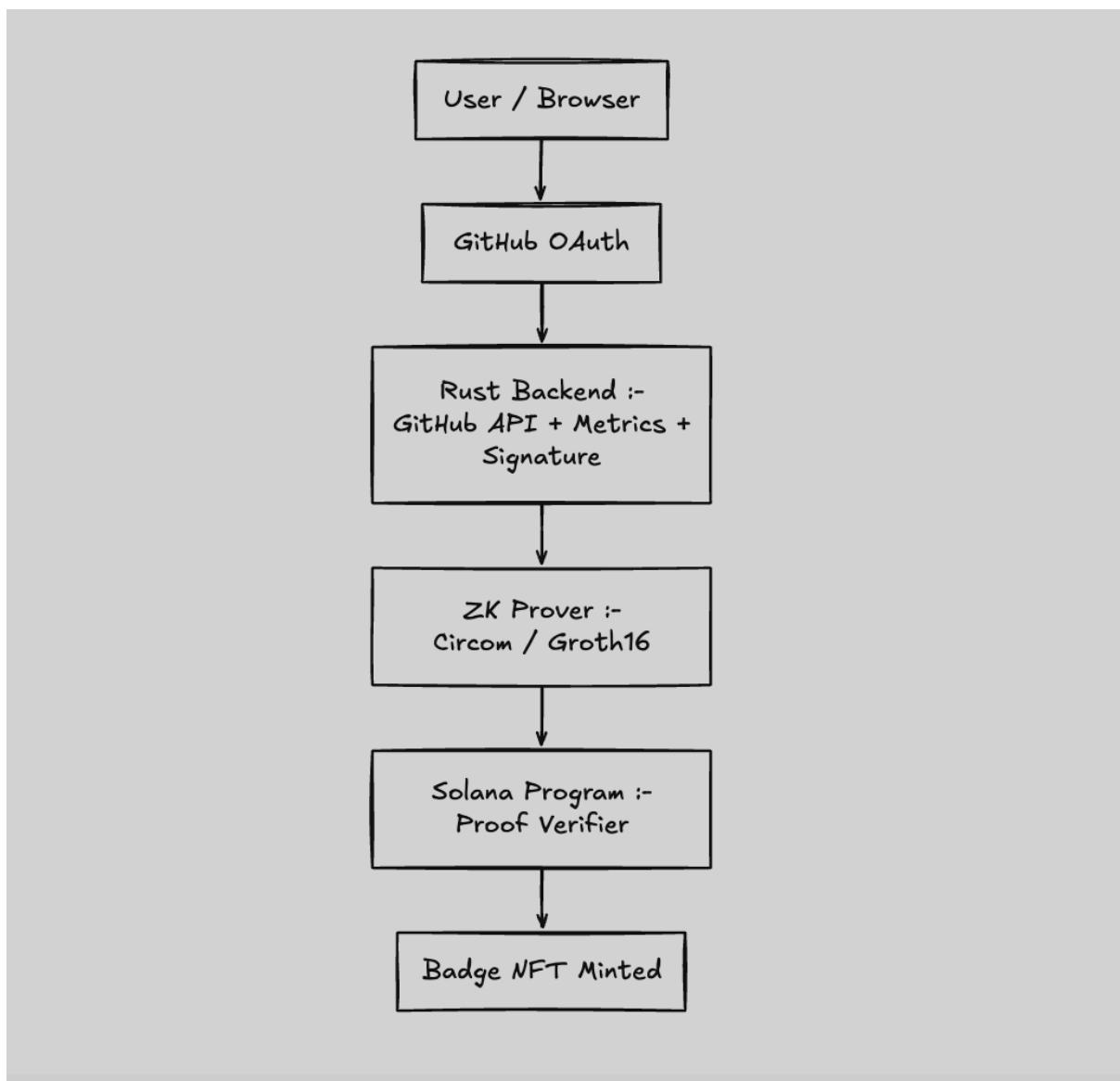
The protocol shall allow a user to:

- Authenticate with GitHub via OAuth
- Fetch verified GitHub statistics through backend
- Generate a Zero-Knowledge proof from those stats
- Submit proof to Solana program
- Mint an on-chain **Dev Badge**
- Mint an on-chain **Repo Badge** for a specific repository
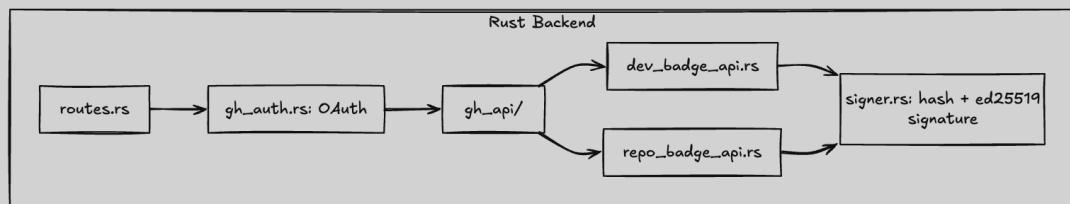
The protocol shall ensure:

- No GitHub username is revealed on-chain
- All stats are cryptographically signed by backend
- On-chain program trusts only the ZK proof
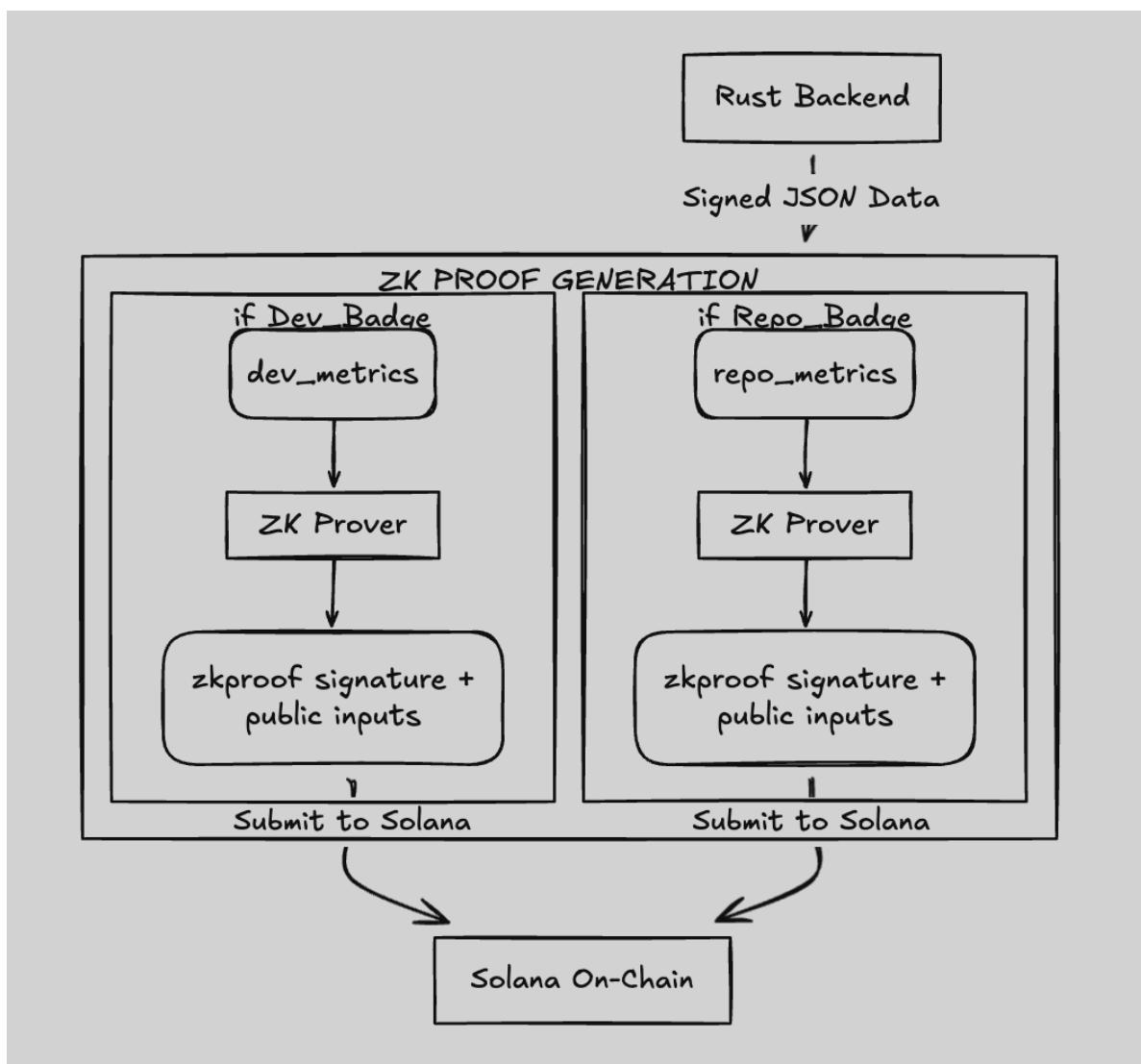
GhostCheck END-TO-END Overview :
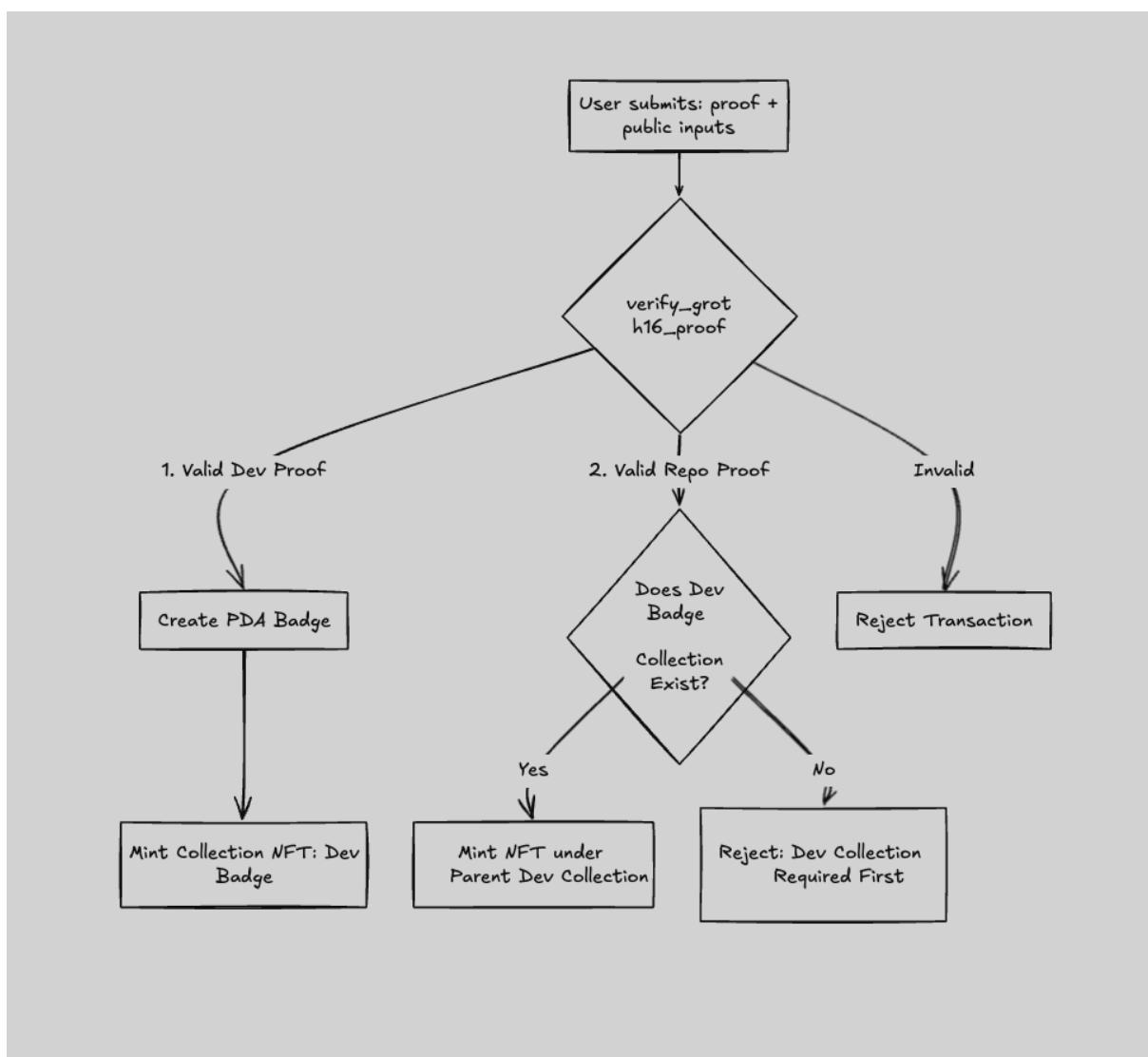
Server Backend Overview
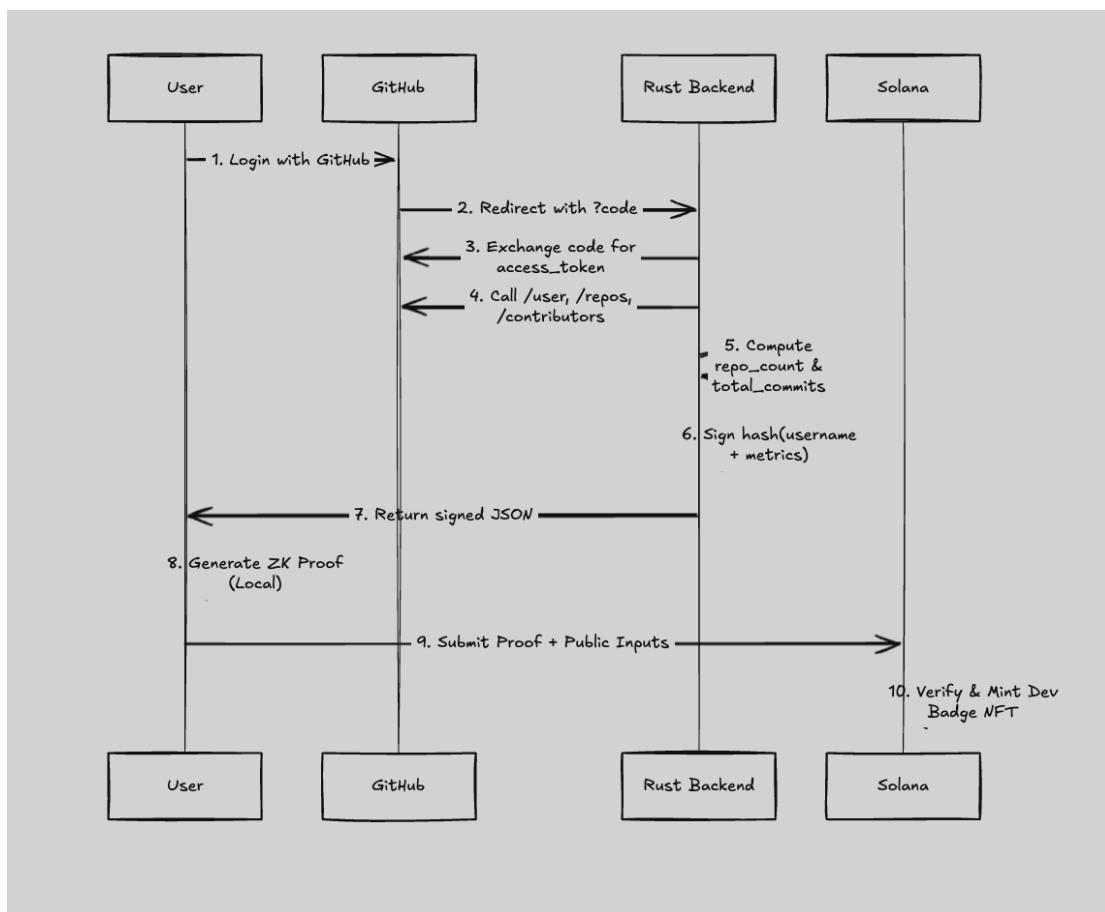
ZK-Proof Overview :

Solana On-Chain Overview :

User submits: proof + public inputs

verify_groth16_proof

1. Valid Dev Proof → Create PDA Badge → Mint Collection NFT: Dev Badge

2. Valid Repo Proof → Does Dev Badge Collection Exist?
- Yes → Mint NFT under Parent Dev Collection
- No → Reject: Dev Collection Required First

Invalid → Reject Transaction

Complete Dev_Badge Overview :

User — GitHub — Rust Backend — Solana

1. Login with GitHub
2. Redirect with ?code
3. Exchange code for access_token
4. Call /user, /repos, /contributors
5. Compute repo_count & total_commits
6. Sign hash(username + metrics)
7. Return signed JSON
8. Generate ZK Proof (Local)
9. Submit Proof + Public Inputs
10. Verify & Mint Dev Badge NFT

Complete Repo_Badge Overview :

User — Rust Backend — GitHub — Solana

1. Enter repo: owner/repo
3. Use stored access_token
4. Call /repos/{owner}/{repo} & /contributors
5. Compute is_contributor & user_commits
6. Sign Data (Ed25519)
7. ZK Proof Generated
8. Submit Proof to Solana
9. Verify & Mint Repo Badge NFT