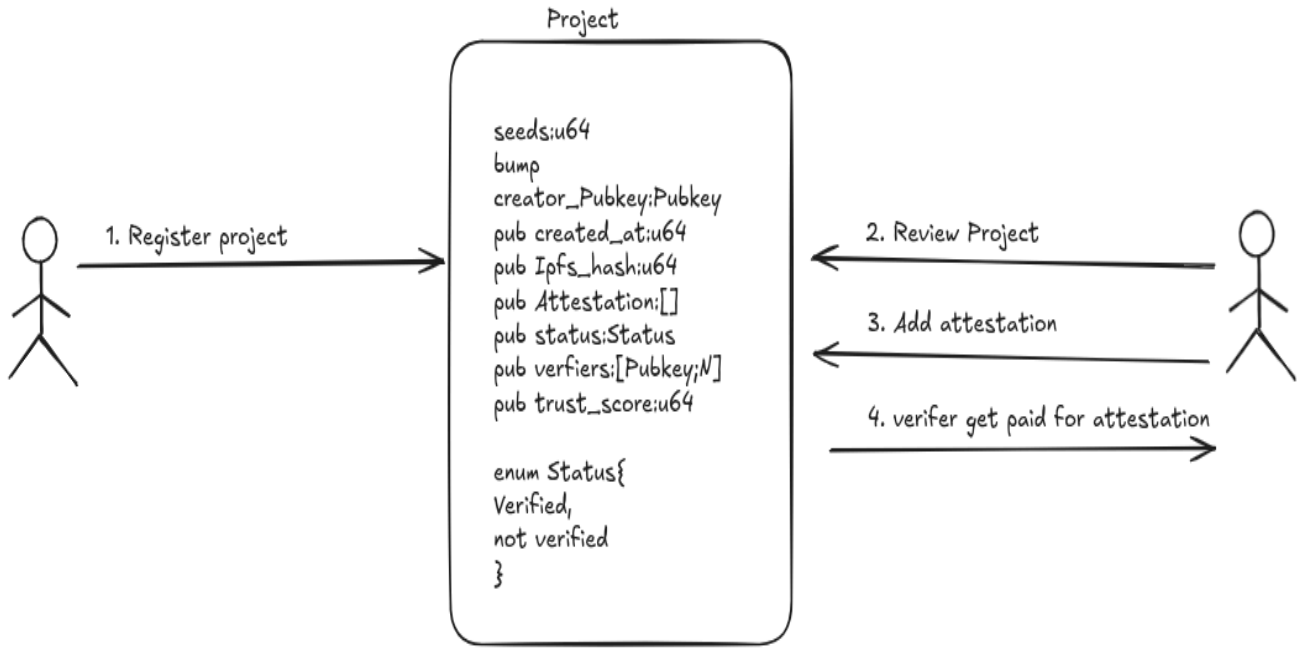
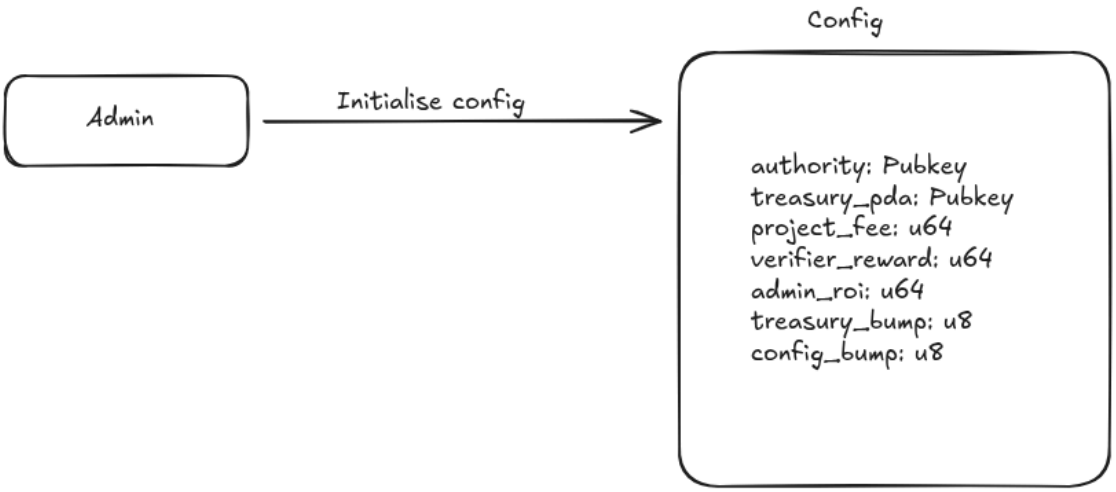
Assignment: Architecture Diagram

Protocol POC Requirements

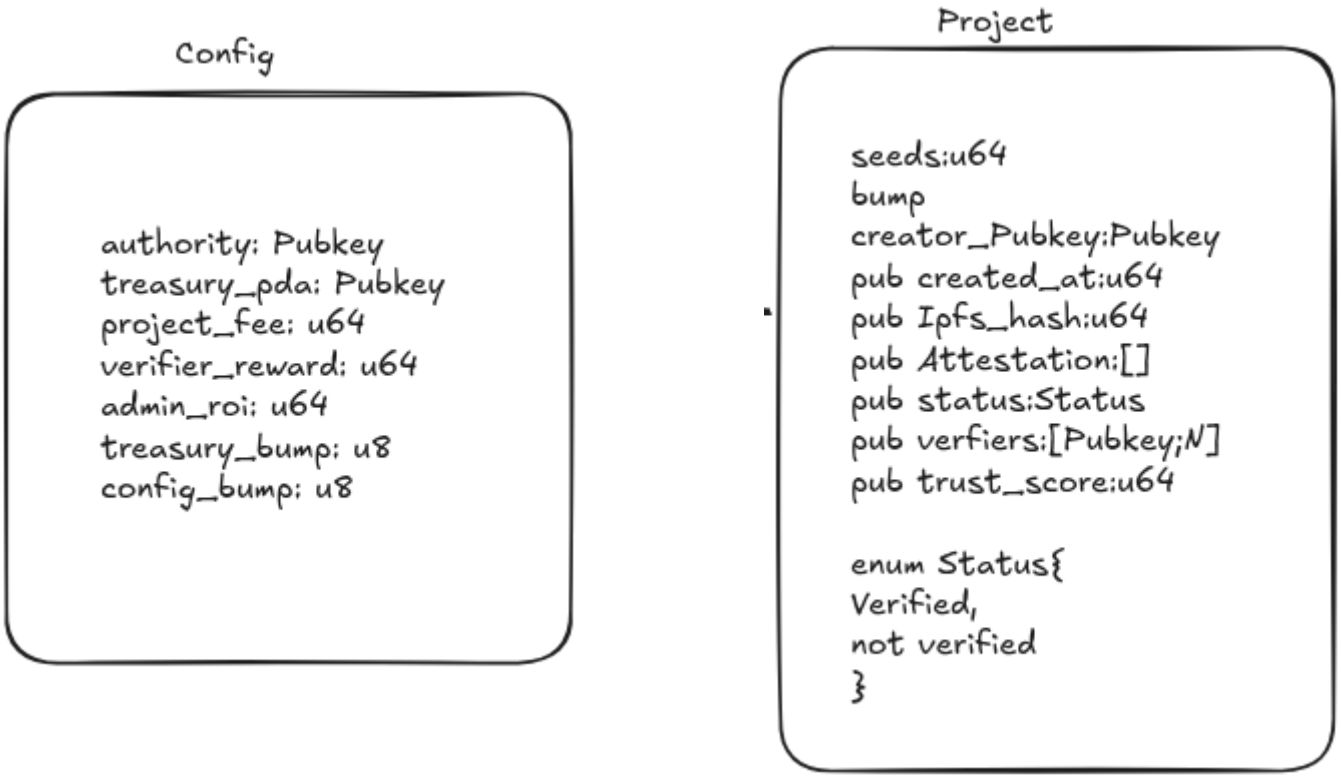
- The protocol should allow users to register projects on-chain, creating PDAs to store project state.
- The Protocol should enable trusted auditors to submit attestations linked to projects via their wallets, with legitimacy confirmed.
- The Protocol should allow document hashes to be stored on-chain while actual files reside on IPFS .
- The protocol should calculate and update trust scores based on attestations quantity and credibility.
- A frontend interface must enable exploring projects, viewing attestations, verifier identities, and accessing audit documents via on-chain hashes.
- The protocol should handle rejection, missing documentation, and authenticity error cases, and avoid storing sensitive data on-chain by using hashed references.
- The protocol enables manual and automatic updates for trust scores, document statuses, and verifier reputations.

Overview

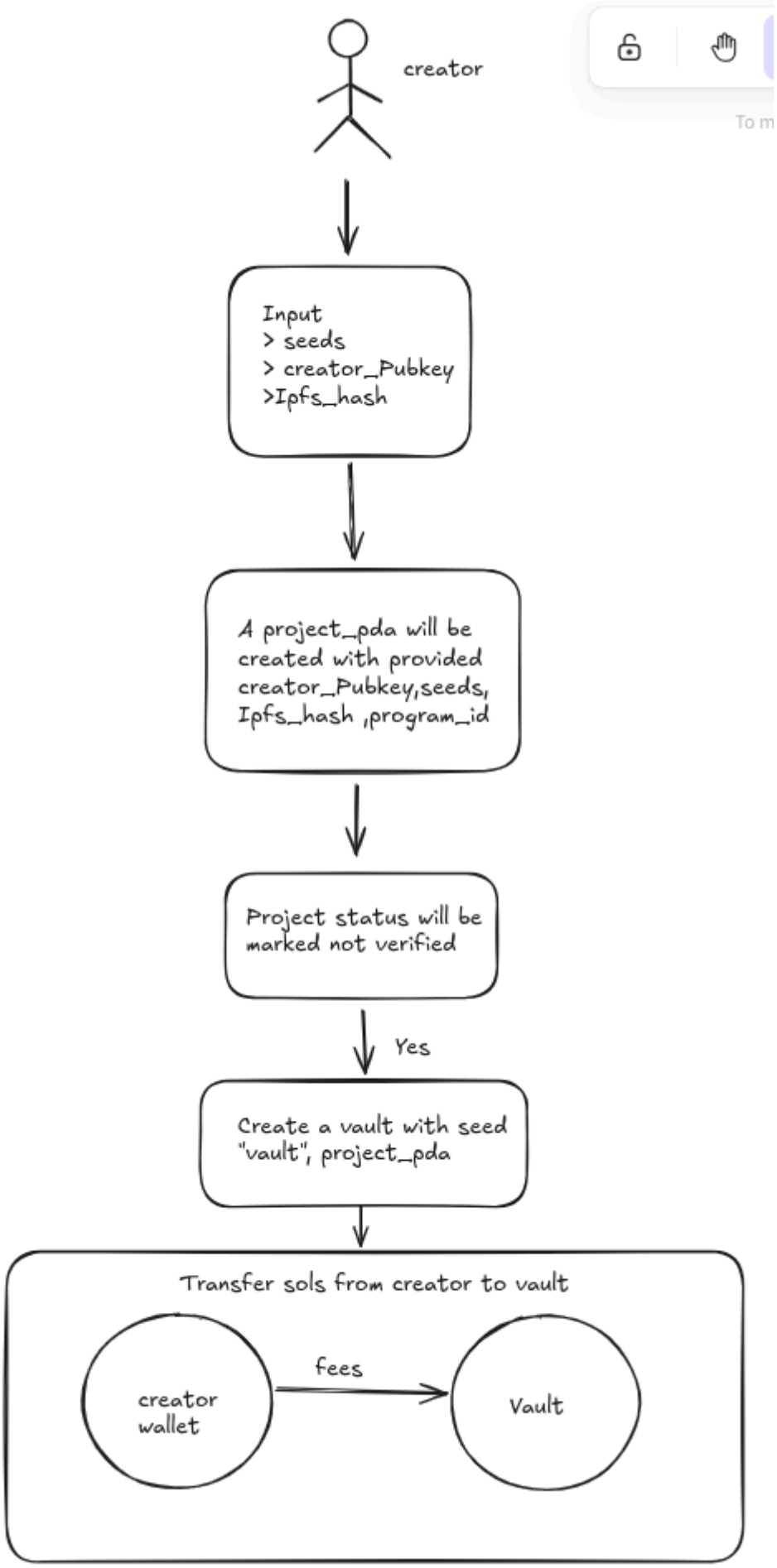



Config
Admin
Initialise config
authority: Pubkey
treasury_pda: Pubkey
project_fee: u64
verifier_reward: u64
admin_roi: u64
treasury_bump: u8
config_bump: u8

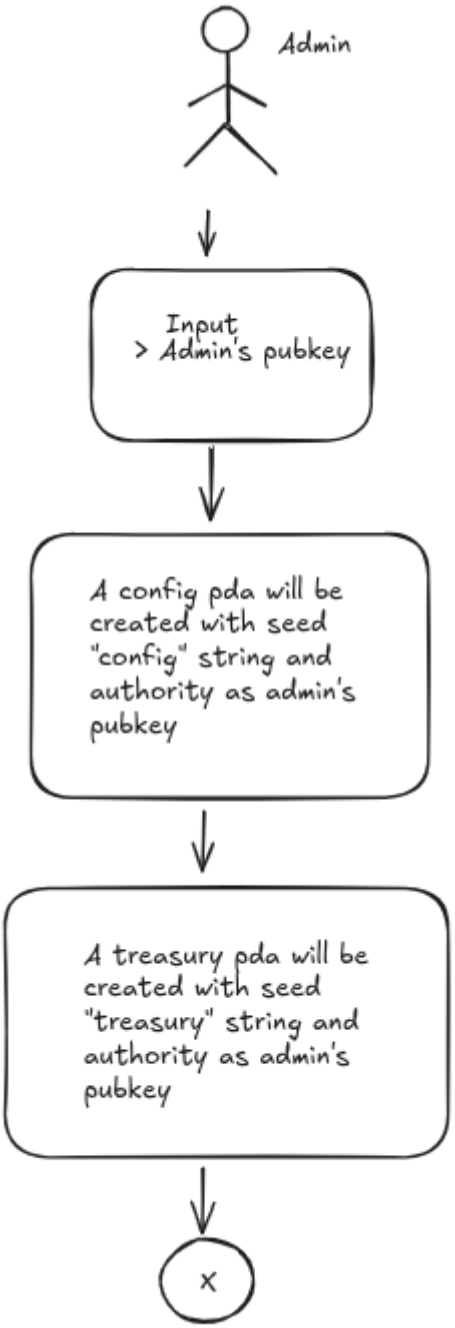
States


Config
authority: Pubkey
treasury_pda: Pubkey
project_fee: u64
verifier_reward: u64
admin_roi: u64
treasury_bump: u8
config_bump: u8
Project
seeds:u64
bump
creator_Pubkey:Pubkey
pub created_at:u64
pub Ipfs_hash:u64
pub Attestation:[]
pub status:Status
pub verfiers:[Pubkey;N]
pub trust_score:u64
enum Status{
Verified,
not verified
}

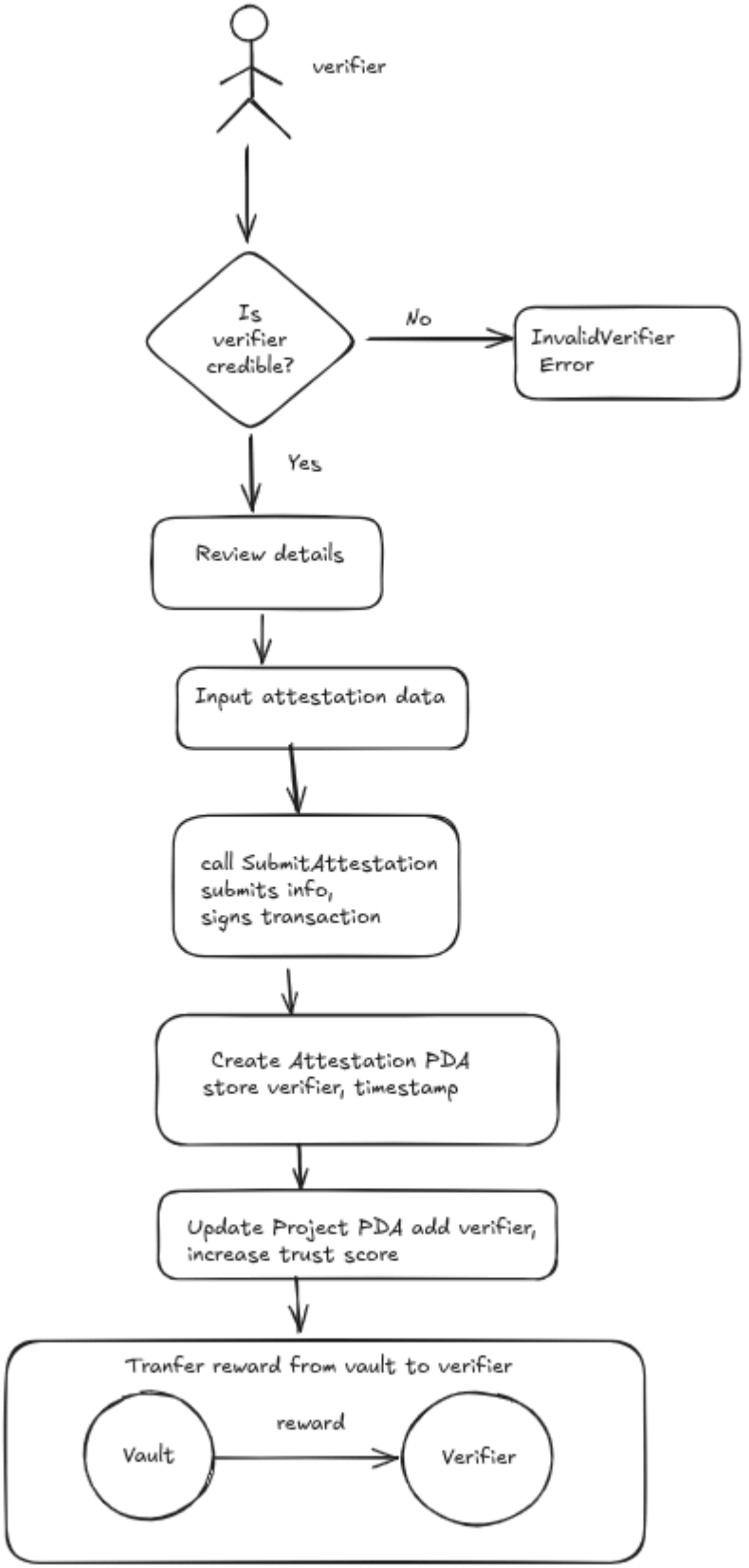
Register a project &deposit fees



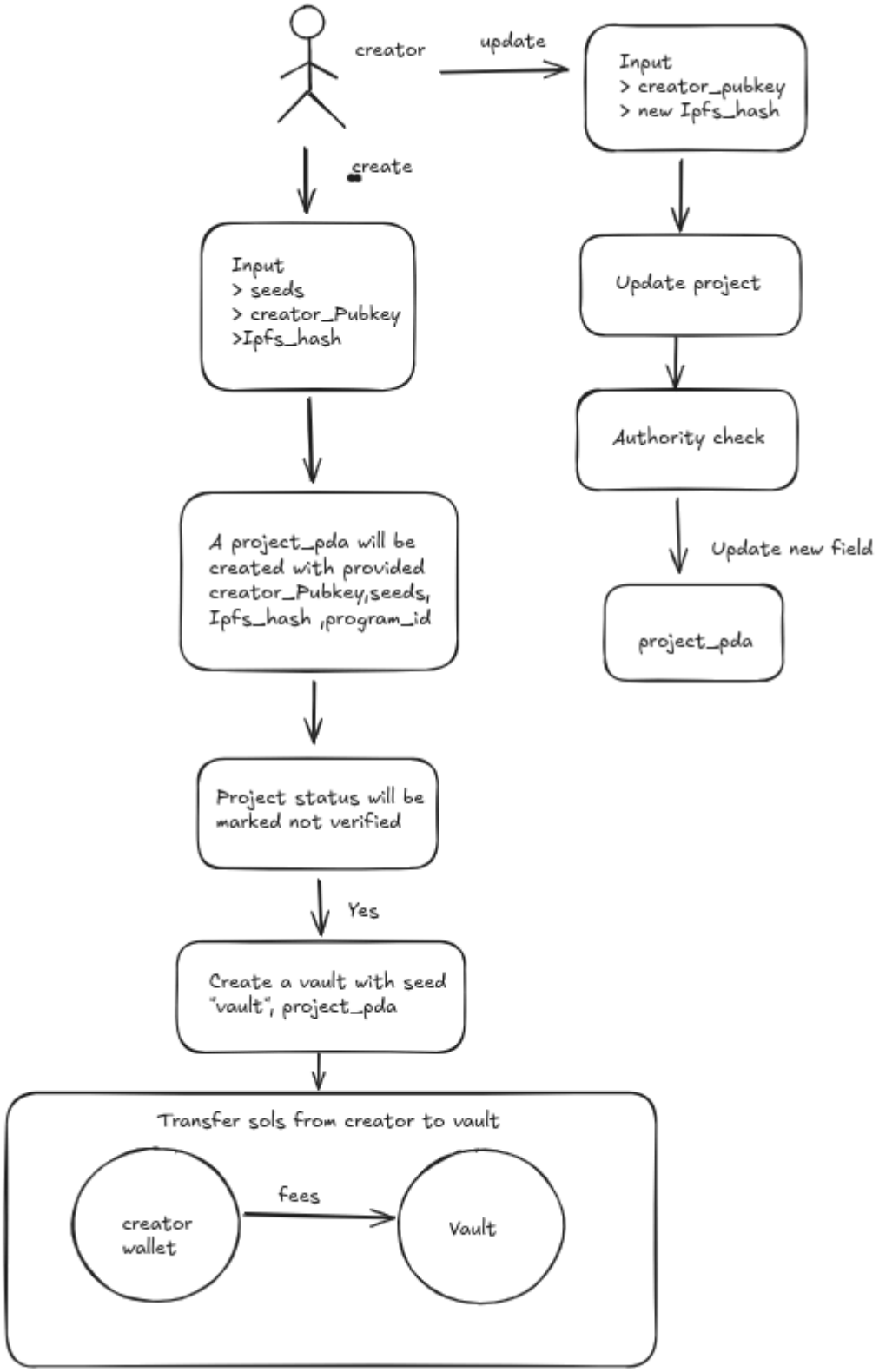
Initialise config(by admin)



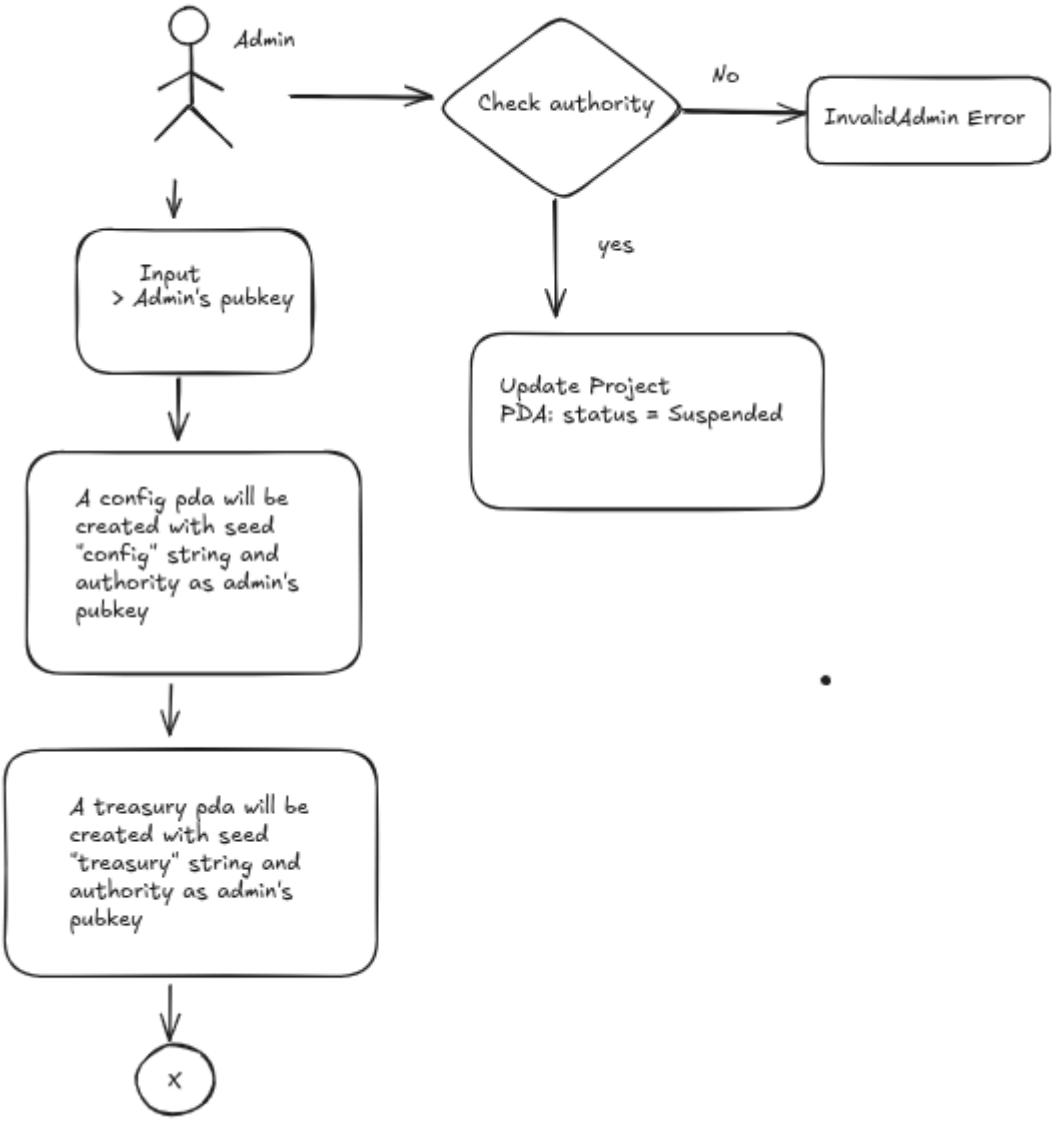
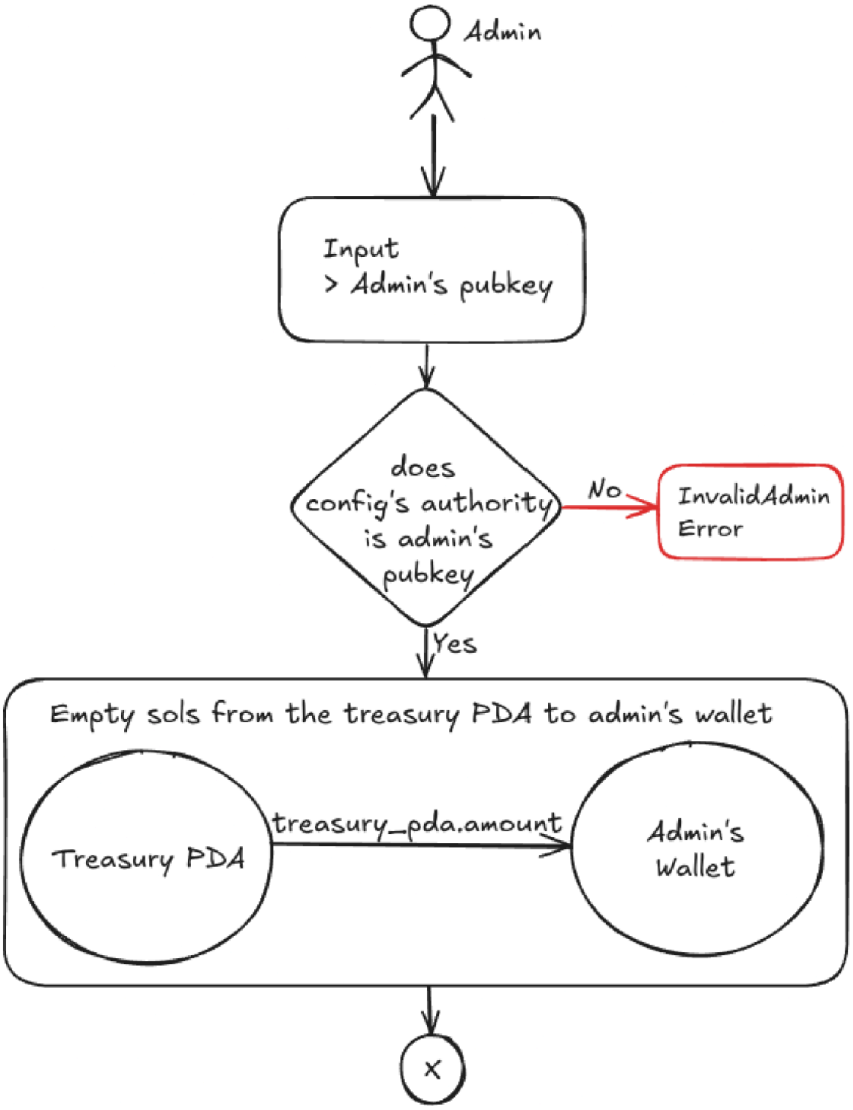
Verify a project


verifier
Is verifier credible?
No
InvalidVerifier Error
Yes
Review details
Input attestation data
call SubmitAttestation
submits info,
signs transaction
Create Attestation PDA
store verifier, timestamp
Update Project PDA add verifier,
increase trust score
Tranfer reward from vault to verifier
Vault
reward
Verifier

Update project



Suspend project for fraud



Withdraw from Treasury(by Admin)