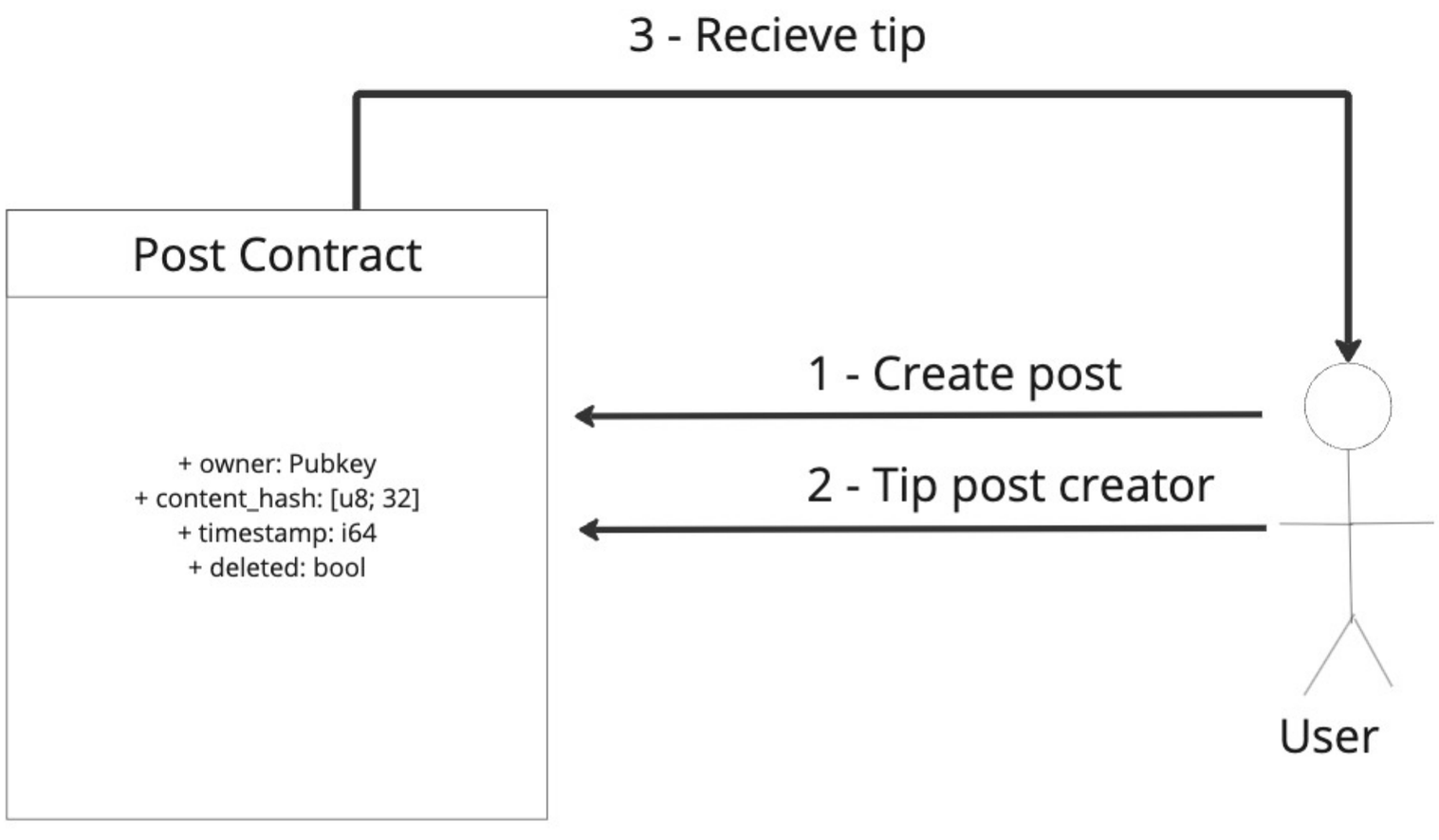
Protocol POC Reqs

- Store each post as a compressed NFT on Solana
- Use **Concurrent Merkle Trees** to enable **cheap, on-chain verification** while keeping post content **off-chain**
- Utilize **Solana's high throughput** and **Metaplex Bubblegum** for efficient NFT
- Allow **users to tip posts** using **Solana** (SOL) through **smart contracts**.management.

Overview



🔷 **Smart Contract Responsibilities**

1. **Post Creation (Posting)**
    a. Users submit a tweet **off-chain** (e.g., IPFS, Arweave, or a centralized DB).
    b. A **SHA-256 hash** of the tweet content is stored **on-chain** in a **Concurrent Merkle Tree**.
    c. The contract appends the new tweet hash into the Merkle Tree and records **metadata** (e.g., timestamp, author).
2. **Reading Post**
    a. The frontend fetches tweets from the **off-chain database**.
    b. Users can verify tweet authenticity using **Merkle Proofs**.
    c. The contract **does not store full post content**, only the **root hash** of the Merkle Tree.
3. **Updating Post**
    a. Users modify an existing tweet.
    b. A **new hash replaces the old hash** in the Merkle Tree.
    c. The previous tweet content remains unverifiable unless the original is saved off-chain.
4. **Deleting Post**
    a. The off-chain tweet data is deleted or marked as **inactive**.
    b. The on-chain **Merkle hash remains**, proving that the tweet **once existed**.
5. **Tipping Mechanism (Solana Pay)**
    a. Users can send **SOL (or SPL tokens)** as tips to tweet authors.
    b. The contract **executes a direct transfer** from the tipper's wallet to the tweet author's wallet.
    c. Optionally, a **small platform fee** can be taken from each tip.

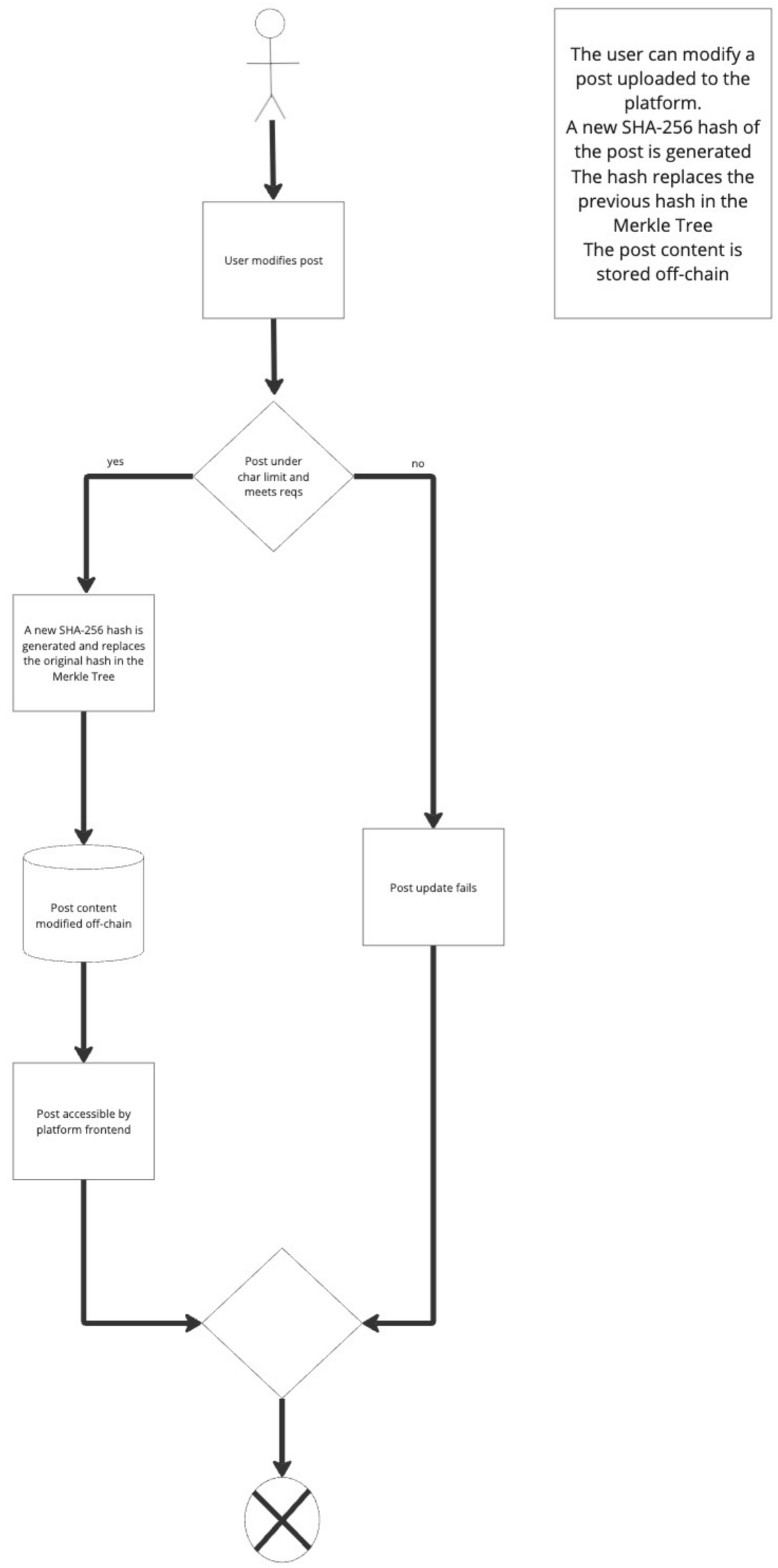
🔷 **Smart Contract Structure**

1. This **single smart contract** would have:
    a. **Post Storage & Verification**
        i. Uses **Concurrent Merkle Trees** to track compressed tweets.
    b. **Tipping Logic**
        i. Facilitates **peer-to-peer payments** for tipping.
    c. **Access Control**
        i. Ensures that only the **tweet author** can update or remove a tweet.

Create Post

The user will create a post to be uploaded to the platform.
A SHA-256 hash of the tweet is generated
The hash is stored in the Merkle Tree via smart contracts
The tweet content is stored off-chain

Update Post

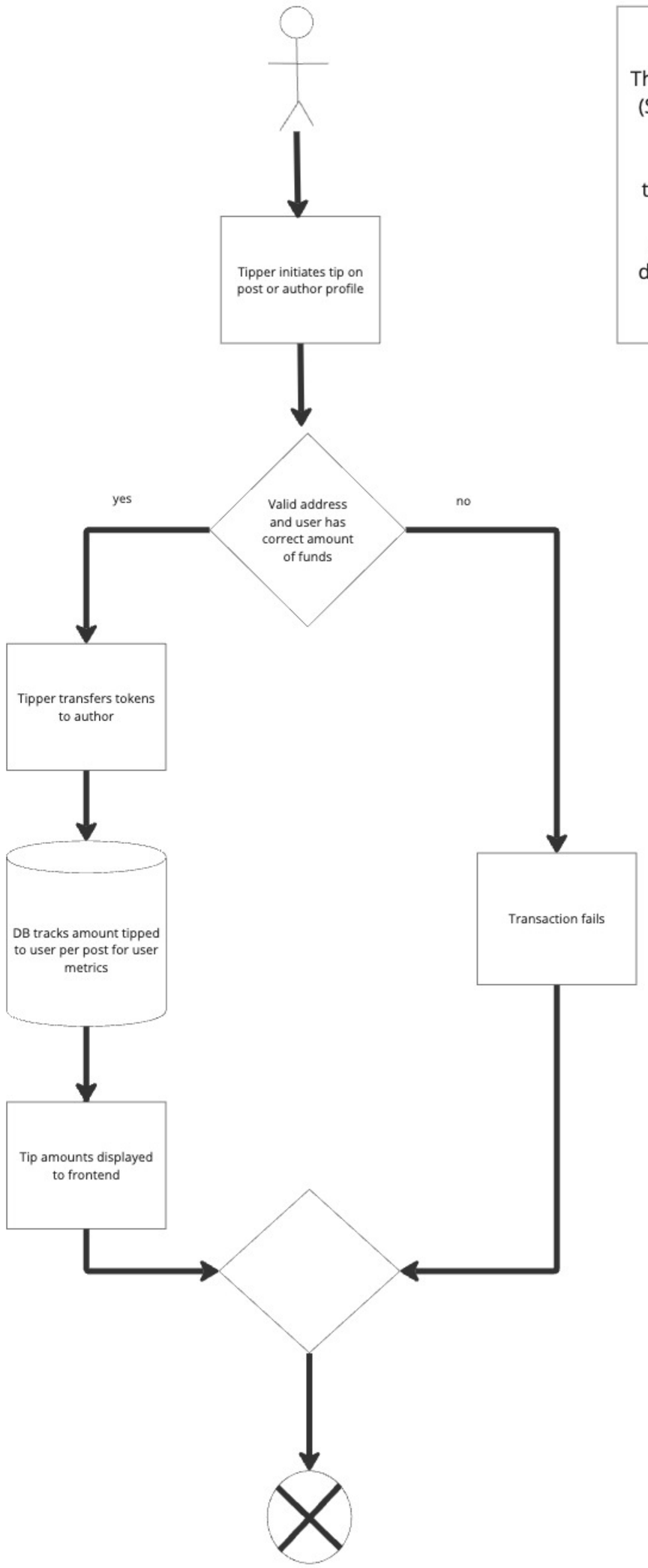


The user can modify a post uploaded to the platform.
A new SHA-256 hash of the post is generated
The hash replaces the previous hash in the Merkle Tree
The post content is stored off-chain

Delete Post

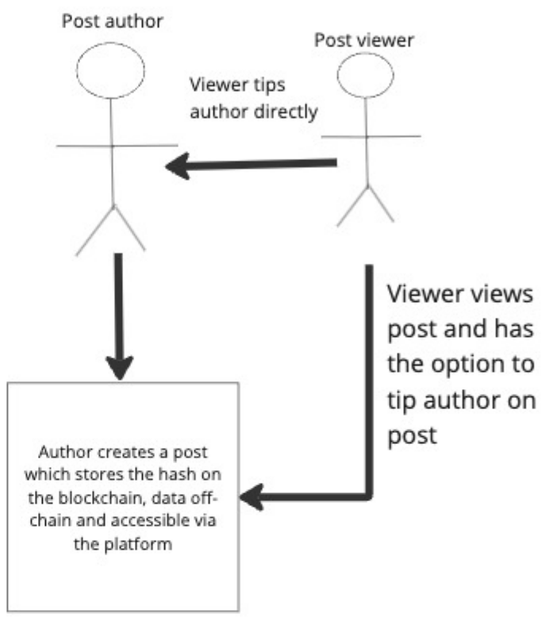
The user can delete a post uploaded to the platform.
The off-chain post data is deleted or marked as inactive
The on-chain hash remains for proof of existence

Tip Post



The user can send a tip (SOL/SPL Token) via a transaction
A smart contract transfers funds from tipper to author
Author recieves tip directly in their wallet

End to end overview



1. User creates or modifies post on the platform
2. Other users can view interact and tip the author on the post
3. Tip gets deposited directly into the author's account