Solmate Project Solution

Since: Saturday, December 11, 2021
Version: 1

Table of Content

1. **Facts**

- Given a time slot(in this project, the time unit of emitting is day), calculating one NFT's $CIETY requires calculating all NFTs' veNFT, what's more, it has to calculate on all slots during last_slot throug current_slot, it's computationally heavy.

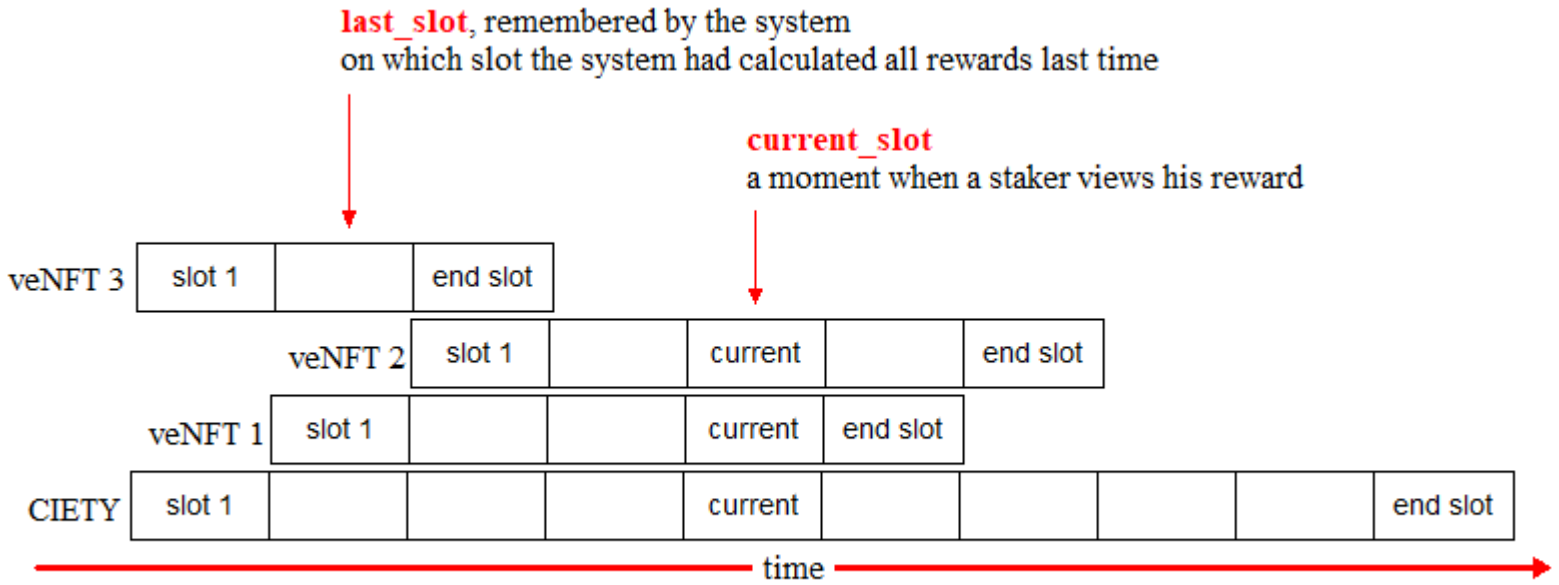

Figure 1 Timeline Model of Emission of $CIETY and evNFT

Logically there is a aligning process, since initializing and on each time of staking, because every NFT's timeline slots has to be aligned with the $CIETY emitting timeline slots, otherwise no way to calculate rewards, surely it's not right to rewad twice for an NFT on each slot.

- Solana smart contract have limits:
  - Space: 10mb/account
  - Compute unit: 200k/execution/contract (this is the problem we are facing right now and what this doc is focusing on)

2. System Use Cases

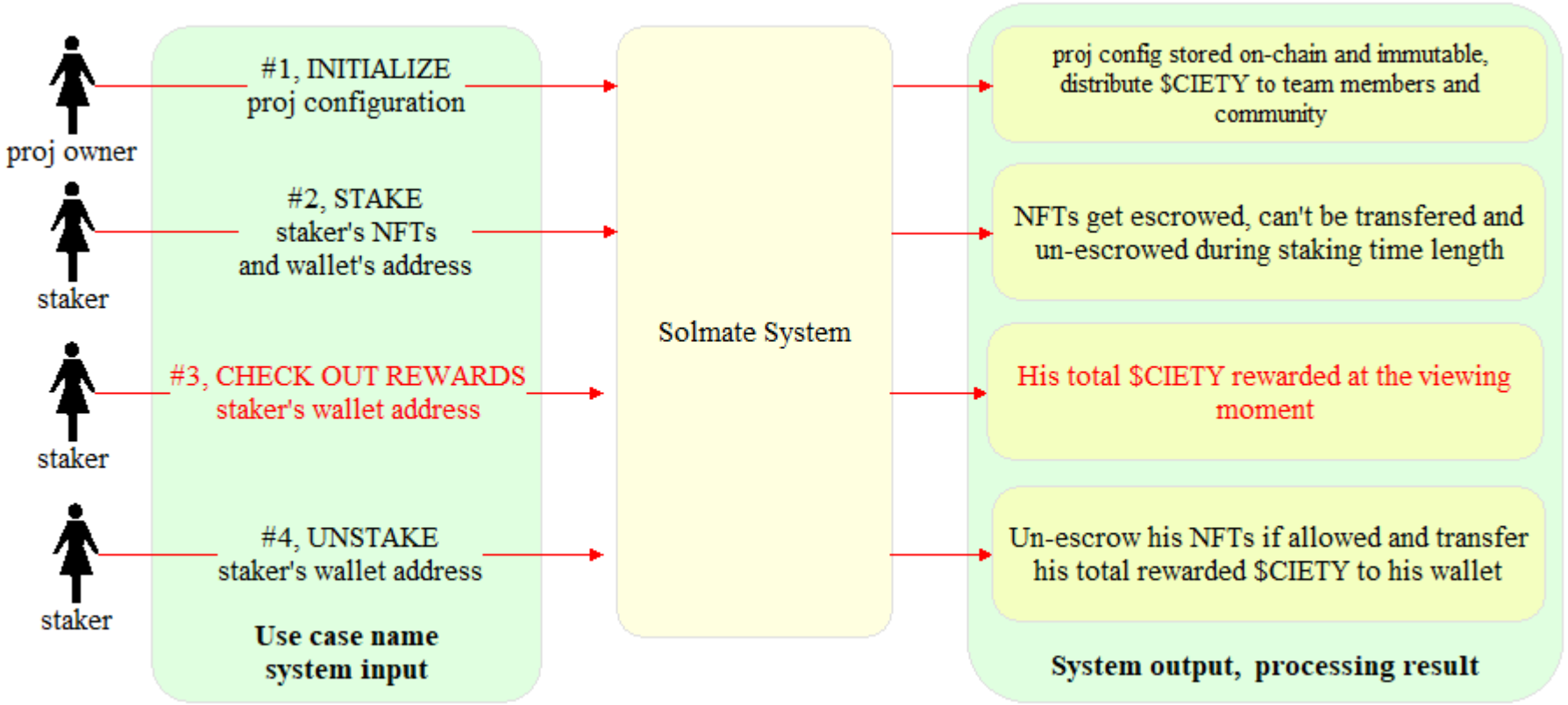

Figure 2 System Use Cases

3. New Solution of Calculating $CIETY Reward

Every use case is okay, except for use case #3 CHECK OUT REWARDS due to Solana chain's restrictions, the spaces taken by storing staking info of all NFTs won't be a problem, because we can tackle it by sharding the storage into multiple Solana accounts. But, the compute unit limit is a problem, here is the solution with no ompromise on fairness.

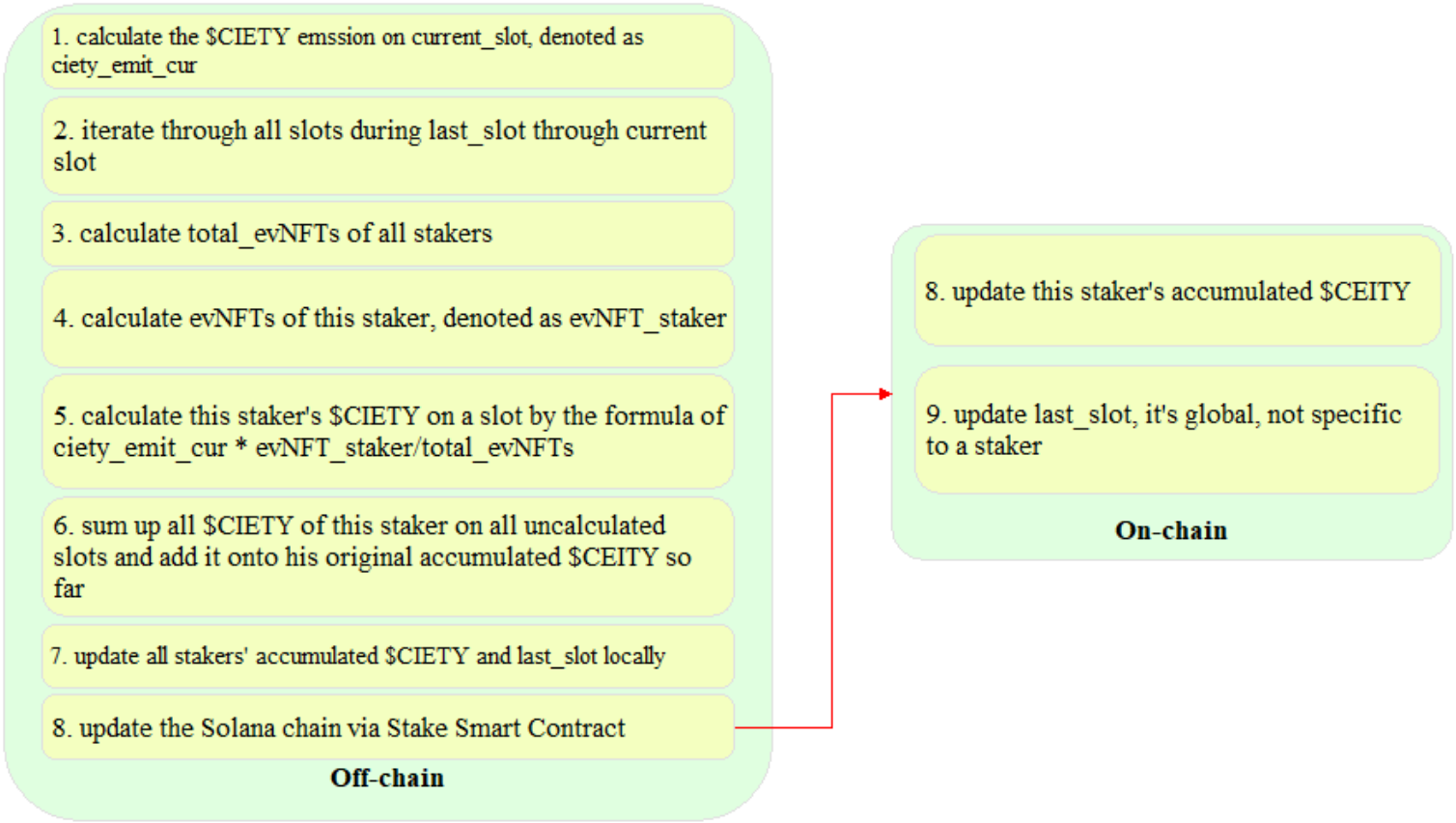

Figure 3 System Behavior Model When A Staker Checks Out His Total Rewards

Key points:

- If you don’t like to read through all steps of off-chain, it’s fine, simply put, just move over the reward calculations off-chain and store the result on-chain without compromise on the fairness.
- This off-chain action will be performed atomically via a distributed queue, since last_slot can only be updated atomically.
- When a staker wants to check out his total $CIETY at a moment of time, the system will determine if current_slot == last_slot first to see if it’s necessary to calute, if not necessary just return his current accumulated $CIETY from the local storage, this reduces the system burden.
- A staker’s on-chain accumulated $CIETY can only be updated under his permission, sure, he is the payer, because updating an account’s data is not free on Solana chain, someone has to pay, meanwhile the proj config is immutalbe, so it’s fair. To be more fair, if you like, you can get the code of reward calculation and proj config open source.
- The accumulated $CIETY and all other staking info(when staked, time length) on each staked NFT of a given staker is accesible, so a staker can audit, that’s fair.