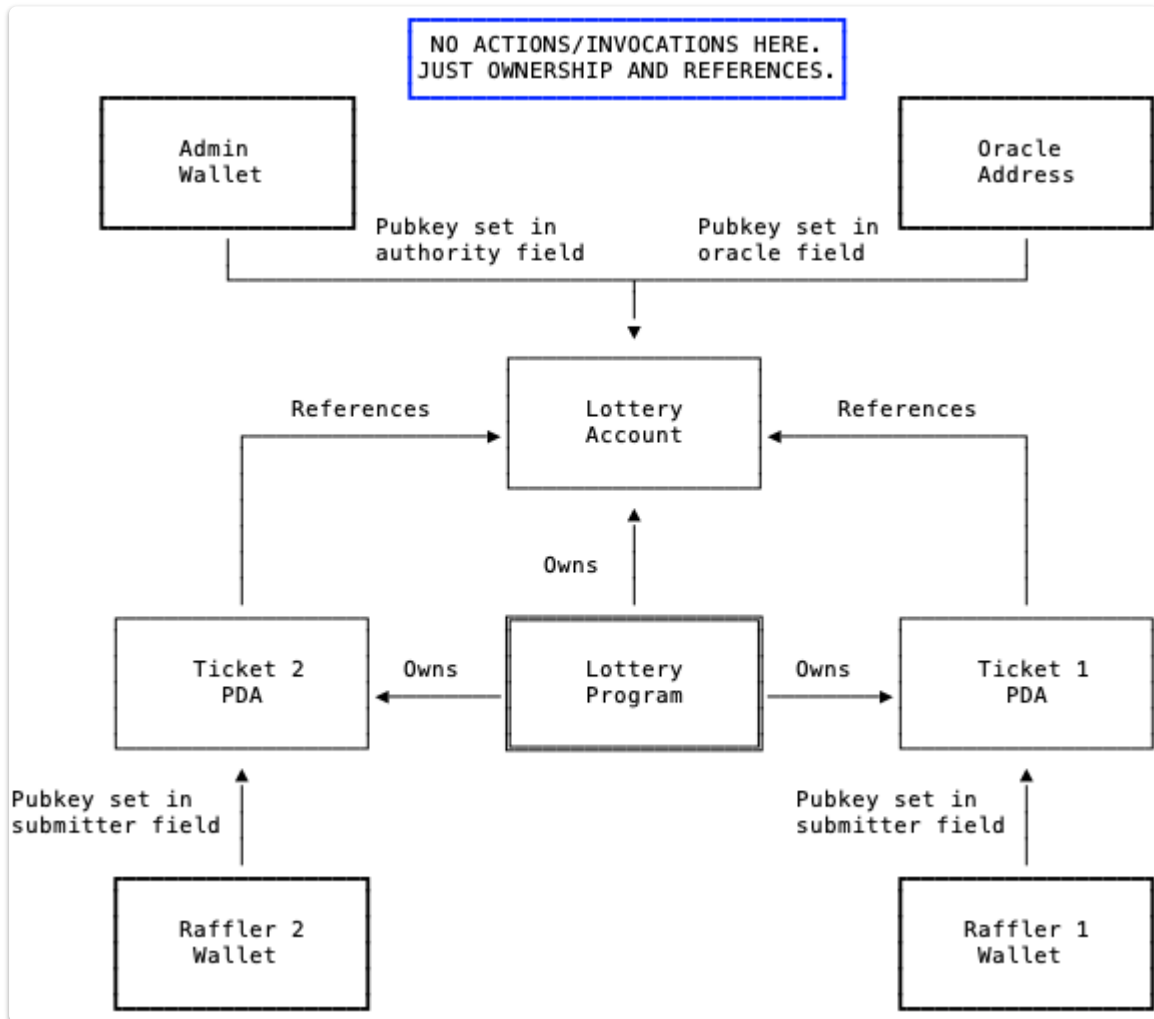


example1-lottery

Service:

Program that implements a lottery.

Account diagram:



Flow:

1. Authority creates lottery account: `initialise_lottery`

```
#[account]
pub struct Lottery {
    pub authority: Pubkey,
    pub oracle: Pubkey,
    pub winner: Pubkey,
    pub winner_index: u32,
    pub count: u32,
    pub ticket_price: u64,
}
```

This account will have lamports deposited to it when a user buys a ticket and withdrawn when the winner gets the payout. Additionally:

- caller set as the authority for this lottery account
- ticket price is set, provided as argument
- oracle is set (ie address allowed to pick the winner), provided as argument
- ticket counter is initialised to zero

2. Raffle buys a ticket: `buy_ticket`

```
#[account]
#[derive(Default)]
pub struct Ticket {
    pub submitter: Pubkey,
    pub idx: u32,
}
```

Ticket PDA is created using the counter and the lottery public keys.

Raffler deposits money into the main lottery account

Lottery ticket counter is incremented

3. Oracle picks the winner: `pick_winner`

It picks index in the range of 0 to `ticket.counter`.

This is saved at the `winner.idx` field

4. Any account can invoke the payout provided they supply winners account: `pay_out_winner`

Lottery account balance is transferred to the winning address