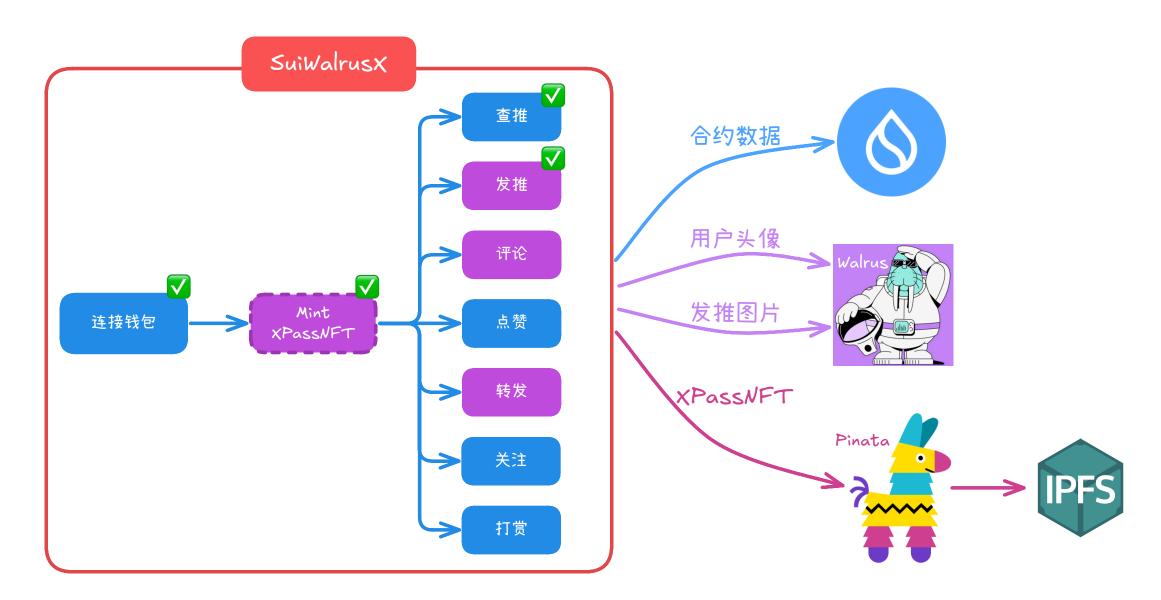
SuiWalrusX

基于 Sui + Walrus 的去中心化X

rzexin

2024.12.17

核心功能



合约设计

数据结构

推文 public struct Tweet has store walrusx.move id: u64. owner: address, 核心结构 content: String, public struct WalrusX has key { media_blob_id: String, id: UID. likes: u64, /// 推文ID的计数器 retweets: u64, tweet_counter: u64, is_retweet: bool, /// 评论ID的计数器 original_tweet_id: u64, comment_counter: u64, created_at: u64, /// 存储所有推文, 推<u>文ID=</u>>Tw tweets: Table<u64, Tweet 用户资料 /// 存储用户发布的所有推文ID, 用户地址=>推文ID user_tweets: Table<address, vector<u64>>, public struct Profile has key, store /// 存储用户资料信息, 用户地址=>用户资料 id: UID, user_profile: Table<address, Profile owner: address, /// 存储推文的所有评论,推文ID=>评论数组 nickname: String, tweet_comments: Table<u64, vector<Comment>> email: String, /// 存储某个用户被哪些用户关注 bio: String, followers: Table<address, VecSet<address> image_blob_id: String, /// 存储某个地址关注了哪些用户 ipfs_nft_url: String, following: Table<address, VecSet<address>>, created_at: u64, /// 存储推文点赞, 推文ID=>点赞用户地址数组 tweet_likes: Table<u64, VecSet<address>>, 评论 public struct Comment has store { id: u64, tweet_id: u64, owner: address, content: String, created_at: u64,

主要接口

```
mint_profile 创建用户并铸造XPassNFT
     create_tweet 创建新推文(需持有XPassNFT)
      add_comment 添加评论 (需持有XPassNFT)
         retweet 转发推文(需持有XPassNFT)
      like_tweet 点赞推文
      follow_user 关注用户
       get_tweet 获取推文信息
  get_user_tweets 获取用户的所有推文ID
get_tweet_comments 获取推文的所有评论
          x_pass_nft.move
                  核心结构
        public struct XPassNFT has key, store {
           id: UID,
           name: String,
           image_url: String,
             mint 为当前用户铸造XPassNFT
```

mint_and_transfer 为指定用户铸造XPassNFT

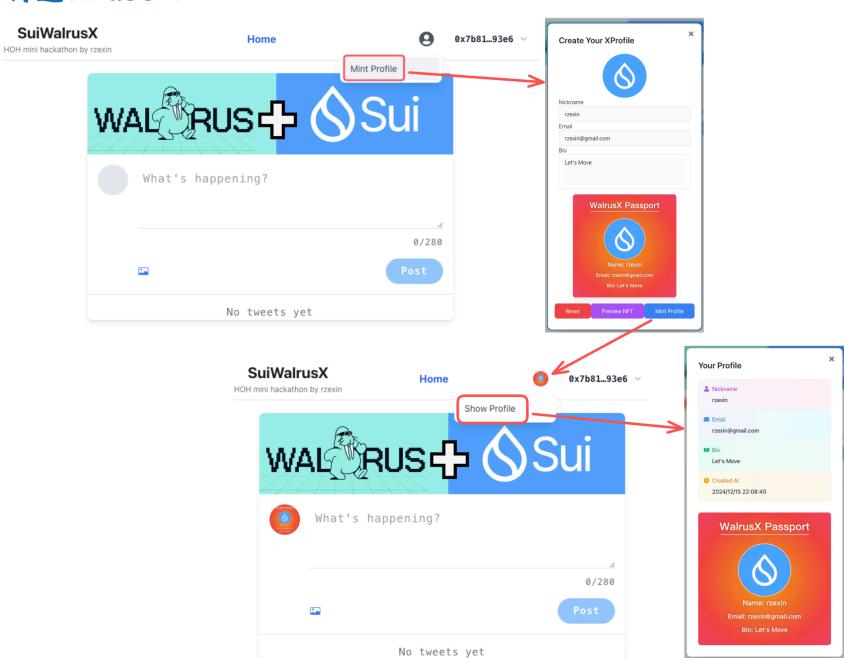
功

铸造XPassNFT

能

演

示



功

发推及查看

能

演

示

