Blockchain Engineer(EVM) - Assignment

What happens when you increase the number of wallets on the network to 1000? Does a smart contract only approach still work? If not, what alternative solution do you propose? Highlight the changes required to implement the additional functionality on the application side and relevant smart contract changes if any.

Solution:

Challenges with Current Smart Contract Approach at Scale:

A. **Gas Costs:**

- When blacklisting/whitelisting a wallet, we need to update all interacted peers
- With 1000 wallets, a single wallet could have hundreds of interactions
- The gas cost would grow exponentially, potentially hitting block gas limits
- Each storage operation (updating whitelist status) costs 20k+ gas

B. **Storage Issues:**

- Storing interaction history for 1000 wallets on-chain is expensive
- Reading large arrays of interactions becomes costly
- Potential for n^2 storage complexity in worst case
    - Number of wallets = n
    - Each wallet interacts with every other wallet
    - Total possible unique interactions = $^nC_2$ = n * (n-1) / 2

For Example:

For n = 5 wallets (A, B, C, D, E):
A → B, A → C, A → D, A → E (4 interactions)
B → C, B → D, B → E (3 interactions)
C → D, C → E (2 interactions)
D → E (1 interaction)
Total = 10 interactions

For n = 10 wallets:
Total = 45 interactions

For n = 100 wallets:
Total = 4,950 interactions

For n = 1000 wallets:
Total = 499,500 interactions

Proposed Alternative Solution

A. Smart Contract Changes:

- Remove on-chain interaction tracking
- Add batch operations with size limits
- Simplify storage to just whitelist status

B. Backend Service:

- Event Listener: Monitors and stores all transfer events
- Database: Stores interaction history and temporary status changes
- API Service: Provides network analysis and status information

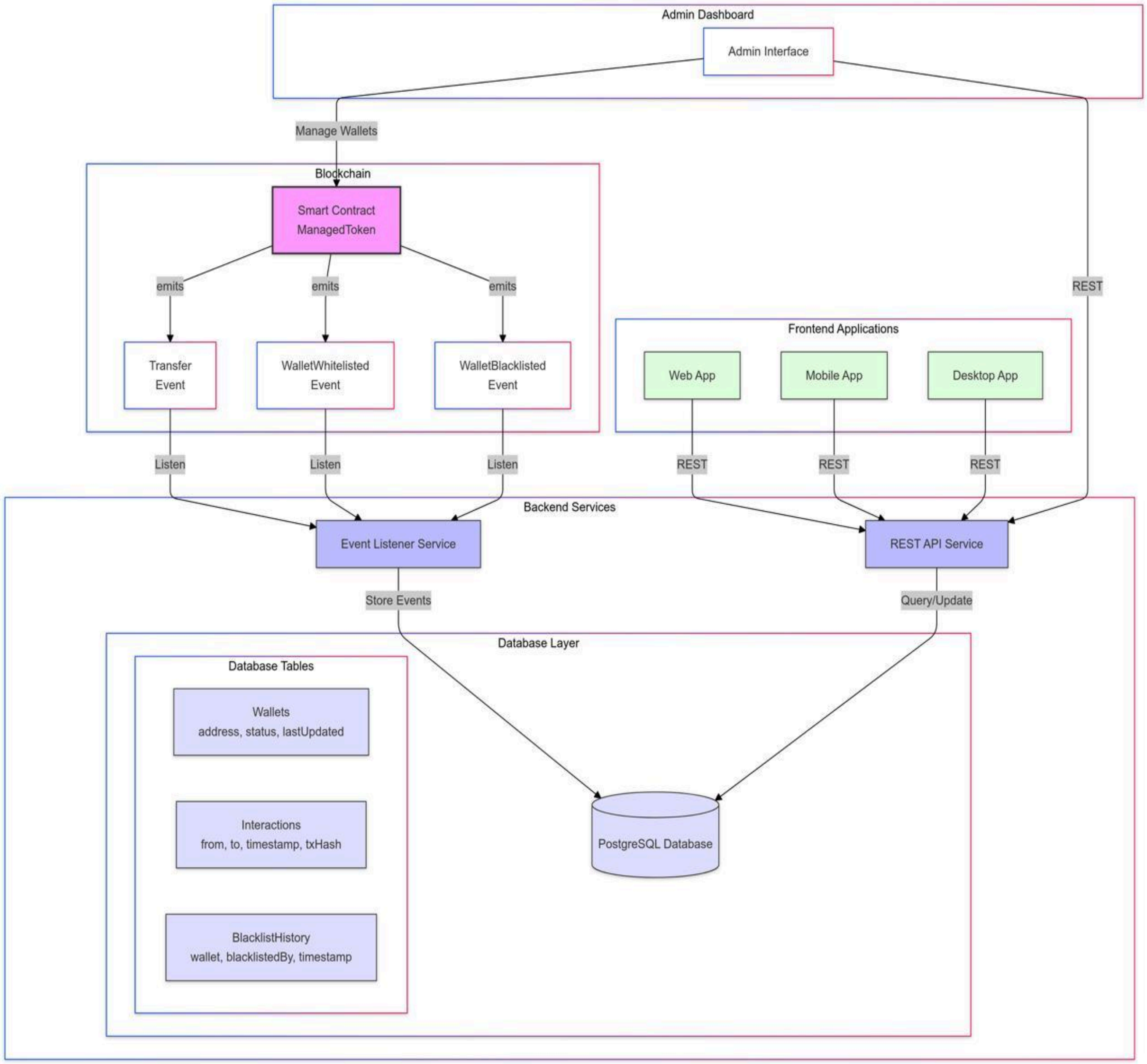
C. Database Schema:

- Interactions collection: Records all transfers
- Wallet status collection: Tracks whitelist/blacklist status

D. Optimizations:

- Batch processing for contract updates
- Caching of frequently accessed network data
- Indexed queries for rapid relationship lookups

UML diagram



Edge Case Scenarios

1. **Scenario:**

- When blacklisting wallet A
- A has interacted with B, C, and D
- If D is not updated along with others, it might allow it to send its balance

Solution: Batch update needs to be atomic