

Meme Battle DApp

Meme Battle is a decentralized application where users can participate in daily meme competitions using their NFT memes. Users can submit memes, vote for their favorites, and compete for prize pools. The platform uses a combination of blockchain technology for transparency and off-chain processing for efficient data management.

Key Features:

1. NFT Integration

- Users can register their meme NFTs for battles
- Each NFT can participate in battles based on its category
- Smart contract verification of NFT ownership

2. Battle System

- Daily battle schedule with fixed timings
- Registration period: 10 AM - 12 PM IST
- Battle period: 12 PM - 8 AM IST (next day)
- Automated winner declaration and prize distribution
- AI-generated creative battle names

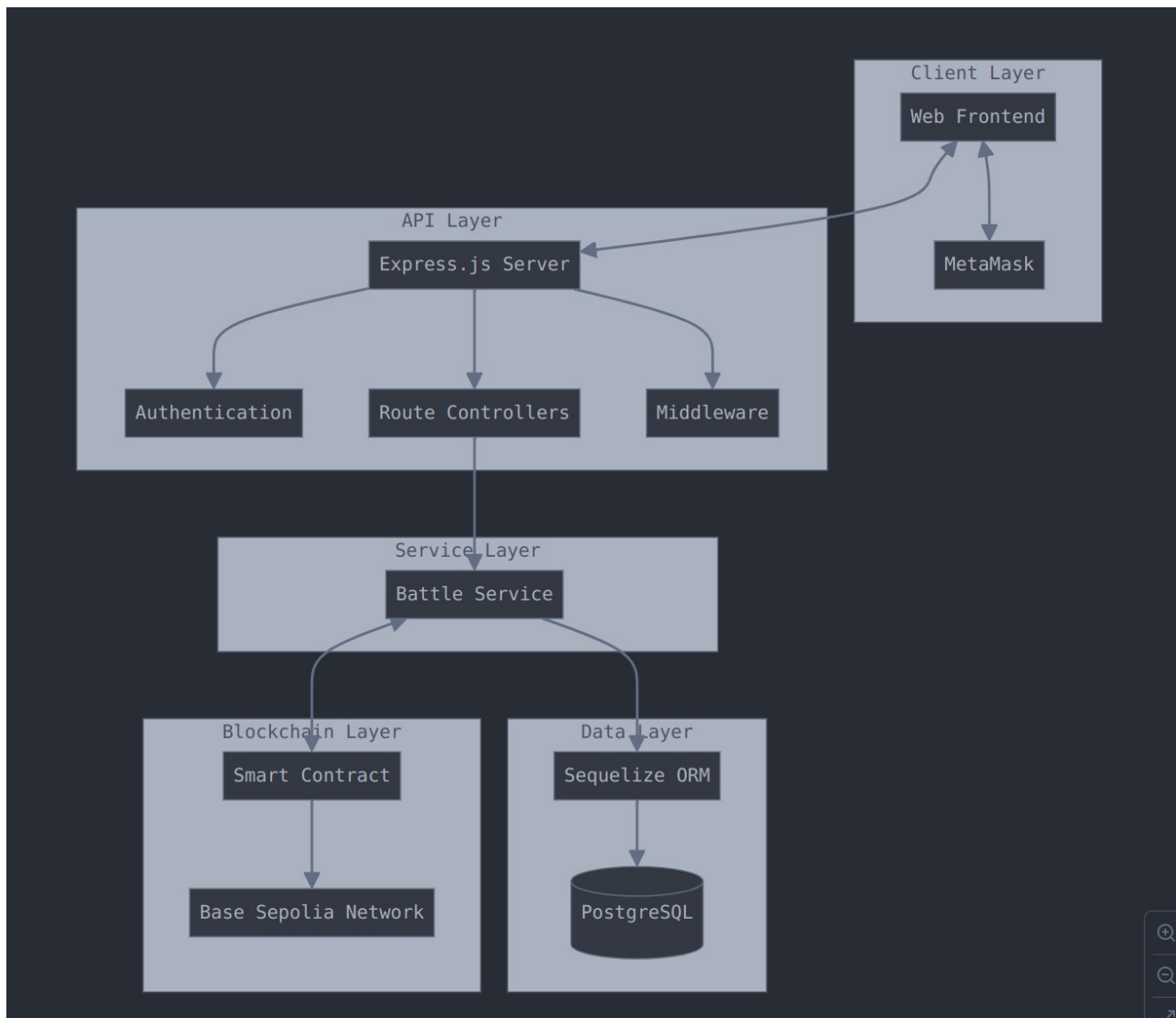
3. Voting Mechanism

- On-chain voting system
- One vote per wallet address per battle
- Real-time vote counting and updates

4. User Dashboard

- View owned meme NFTs
- Track battle participation history
- Monitor winning statistics
- View global leaderboard

Architecture:



System Interactions:

- The Web Frontend communicates with MetaMask to handle blockchain-related user interactions.
- Users interact with the frontend, which sends requests to the Express.js Server.
- The server's authentication and middleware process these requests.
- Route controllers direct requests to the Battle Service.
- The Battle Service interacts with the Sequelize ORM to perform database operations.
- The ORM manages interactions with the PostgreSQL database.
- The Battle Service can also directly interact with the Smart Contract for blockchain-specific functionalities.

MemeBattle Smart Contract Description

Overview

The MemeBattle smart contract is a decentralized application (dApp) designed to host competitive meme battles on the blockchain. It allows users to:

- Create meme battles
- Join battles with their meme tokens
- Vote on memes
- Determine winners based on upvote count

Key Structures

1. MemeTokens

Represents an individual meme in a battle:

- **reator**: Address of the meme's creator
- **meme**: Address of the meme token
- **NoUpvotes**: Number of upvotes received

2. BattleDetails

Defines the details of a specific meme battle:

- **participantsCount**: Number of memes in the battle
- **Memes**: Array of MemeTokens participating
- **MemeId**: Mapping to track meme IDs within a battle
- **Winner**: Address of the winning meme

3. UserDetails

Tracks a user's battle history:

- `NoOfWins`: Number of battle wins
- `NoOfBattles`: Total battles participated in
- `battleCodes`: List of battle IDs the user has joined

4. UserBattle

Tracks whether a user has joined a specific battle:

- `isAlreadyJoined`: Boolean indicating battle participation

Key Mappings

- `MemeJoined`: Tracks which memes have joined a battle
- `Battles`: Stores battle details by battle ID
- `userInfo`: Stores user participation details
- `isVoted`: Prevents multiple votes from the same user

Core Functions

1. JoinBattle

- Allows a user to join a battle with a meme
- Requires the meme not to have already joined the battle
- Charges a participation fee
- Increments participant count
- Adds meme to battle details
- Tracks user's battle participation

2. Upvote

- Users can upvote a meme in a specific battle

- Prevents multiple votes from the same user
- Increments upvote count for the meme

3. DeclareWinner

- Determines the winner based on the highest number of upvotes
- Iterates through all memes in the battle
- Sets the winning meme address

4. Utility Functions

- `viewWinner`: Retrieves the winner of a specific battle
- `viewVotes`: Shows the number of votes for a specific meme
- `payFee`: Internal function to collect participation fees
- `getBattleId`: Generates a unique battle ID using block timestamp

Fee Mechanism and Prize Distribution

- A fixed fee of 0.25 ETH is charged when joining a battle
- Fee is transferred to the contract deployer (FeeReceiver)
- Prize is 95% of aggregated fees of battle and Prize is distributed to winner

