**Title:** Decentralized Arbitration System

**Subtitle:** Optimizing Token-Based Dispute Resolution

**Presented by:** DecentralEyes

**Date:** 20/10/24

# Introduction to Decentralized Arbitration

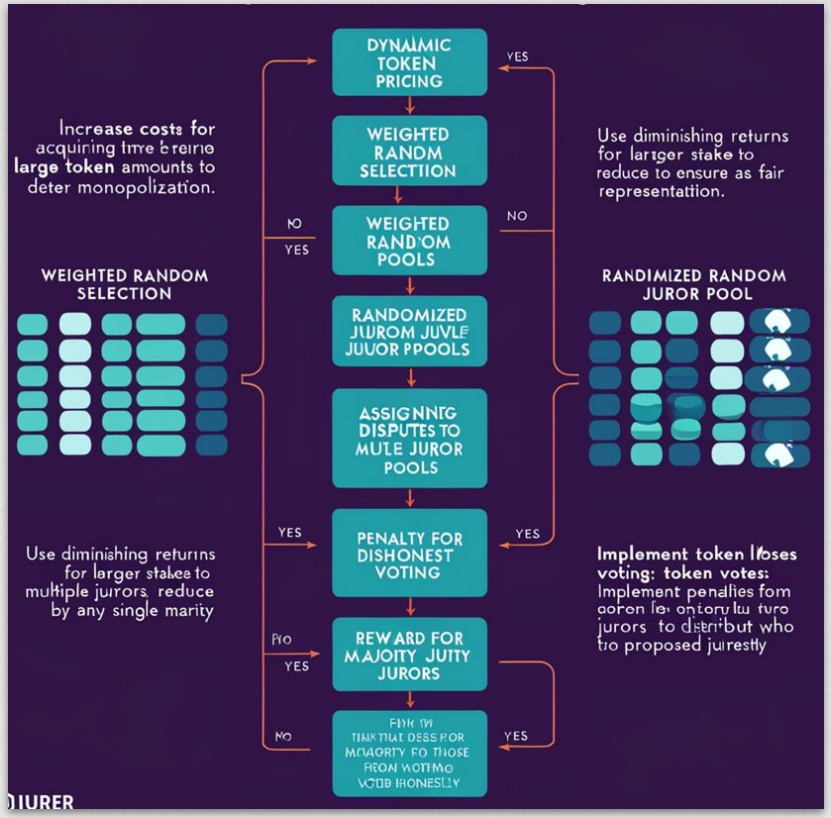
- **Definition:** A dispute resolution process where jurors are selected through a decentralized mechanism.
- **Importance:** Fosters trust among users in online marketplaces and platforms.
- **Challenges:** Ensuring fairness, transparency, and efficiency amidst increasing global interactions.
- **Role of Tokens:** Jurors stake tokens (e.g., GRULL) to participate, enhancing engagement.
- **Objective:** Create a system that is impartial, cost-effective, and scalable.

# Key Challenges

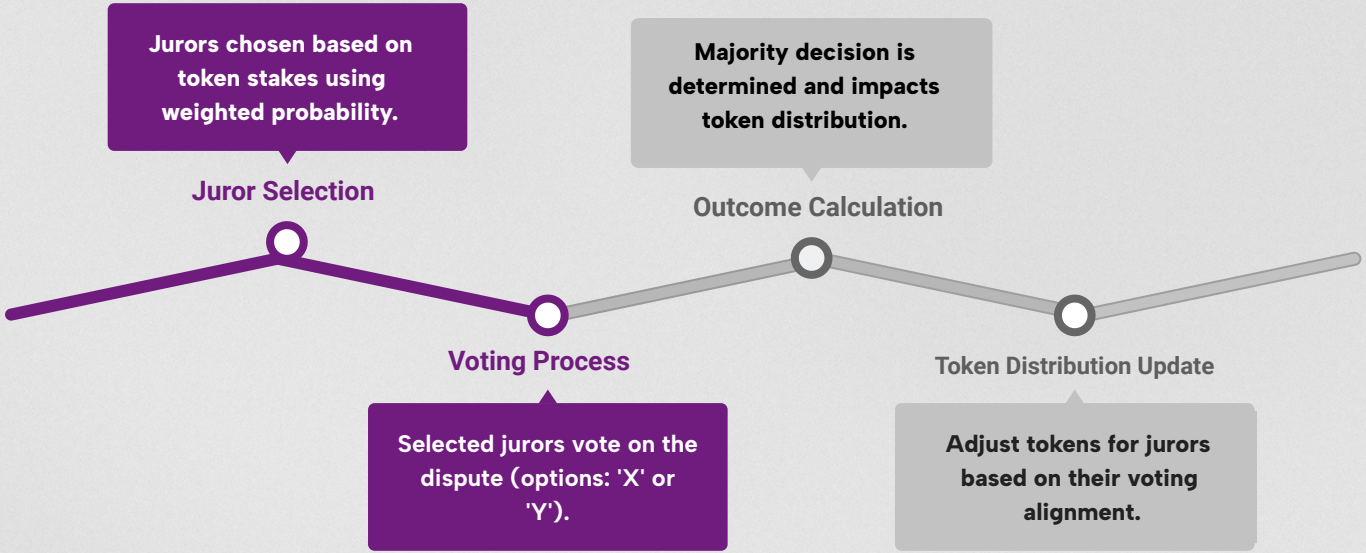
- **Efficient Juror Selection:** Engage only active jurors to prevent inactive participants.
- **Fairness Optimization:** Balance selection probability to avoid monopolization by large stakeholders.
- **Sybil Attack Prevention:** Protect against bad actors creating multiple accounts.
- **Token Accumulation Risk:** Prevent single entities from controlling the selection process.
- **Incentive Misalignment:** Ensure jurors have the right economic motivations to vote honestly.

# Proposed Solutions

- **Dynamic Token Pricing:** Increase costs for acquiring large token amounts to deter monopolization.
- **Weighted Random Selection:** Use diminishing returns for larger stakes to ensure fair representation.
- **Randomized Juror Pools:** Assign disputes to multiple juror pools to reduce control by any single party.
- **Penalty for Dishonest Voting:** Implement token losses for jurors voting against the majority.
- **Reward for Majority Jurors:** Distribute penalties from dishonest voters to those who voted honestly.



# System Workflow



# Example Scenario

**Participants:** Six token owners with a total of 12,000 tokens.
**Voting Results:** Majority decision in favor of 'X' with various owners voting.
**Outcome:**

- Majority gains tokens from the losing party.
- Token redistribution based on voting alignment.

### Final Token Distribution for All Jurors:

Owner A: 2000 tokens

Owner B: 1450 tokens

Owner C: 2800 tokens

Owner D: 3000 tokens

Owner E: 950 tokens

Owner F: 2000 tokens

### Selected Jurors and Votes:

Owner C: 2800 tokens (Vote: X)

Owner C: 2800 tokens (Vote: X)

Owner E: 950 tokens (Vote: Y)

Owner C: 2800 tokens (Vote: X)

Owner B: 1450 tokens (Vote: Y)

### Majority Decision: X

### Token Adjustments:

Owner C gains 100 tokens for voting with the majority.

Owner C gains 100 tokens for voting with the majority.

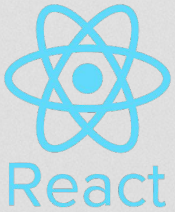
Owner E loses 50 tokens for voting against the majority.

Owner C gains 100 tokens for voting with the majority.

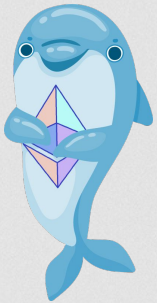
Owner B loses 50 tokens for voting against the majority.

# Tech Stack And Outputs

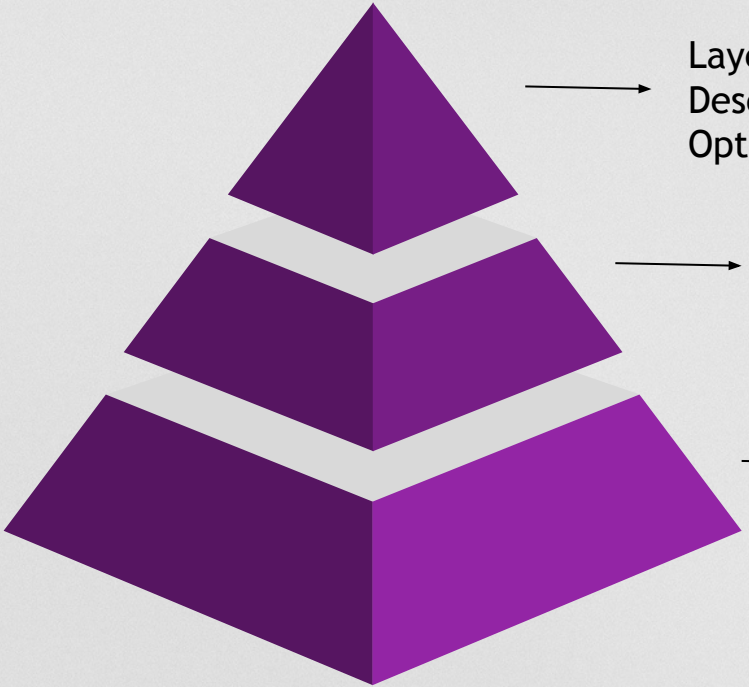
- ➔ **React**: Frontend library for building dynamic user interfaces.
- ➔ **Solidity**: Language for writing Ethereum smart contracts.
- ➔ **Hardhat**: Development environment for compiling and testing smart contracts.
- ➔ **Ether.js**: JavaScript library for blockchain interactions.
- ➔ **Sepolia Testnet**: Safe environment for testing Ethereum contracts.







# Future Enhancement



Layer 2 Solutions:
Description: Integrate Layer 2 protocols like Optimistic Rollups or zk-Rollups.

Zero-Knowledge Proofs (ZKP):
Description: Use ZKP protocols like zk-SNARKs or zk-STARKs for privacy and verification.

Work to Earn:
Description: Implement a reward mechanism where users earn tokens or NFTs by completing tasks or contributing to the network.

# THANK YOU