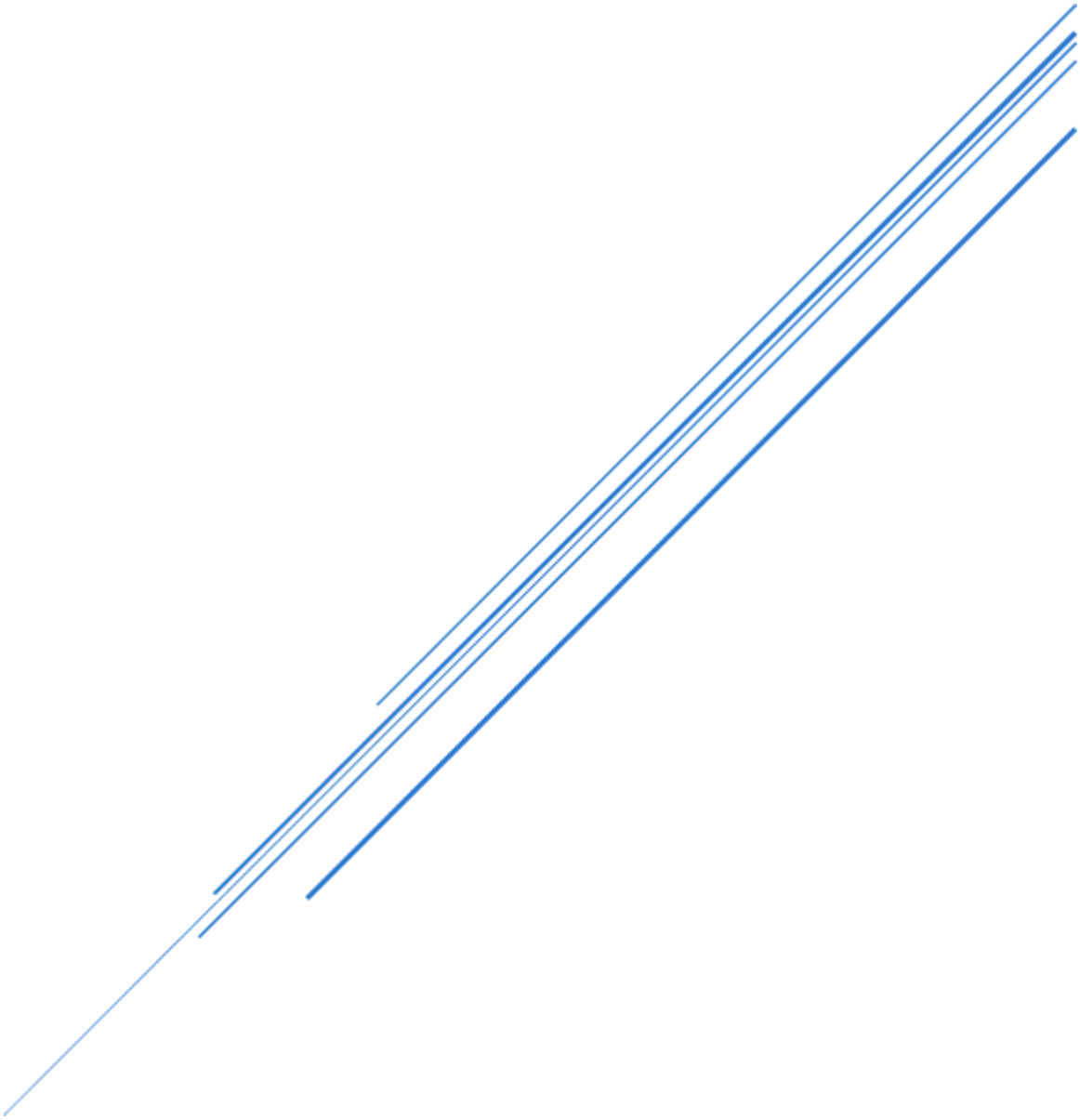
DECENTRALIZED RAFFLE SYSTEM

CS-411 Blockchain



Hamza Hasan Ellahie 2021197
Muhammad Zulfiqar Ali 2021493

Research and System Design: Web3 Lottery DApp

Overview

The Web3 Lottery DApp is a sophisticated decentralized application engineered to leverage the Polygon blockchain network for transparent and secure lottery operations using cryptocurrency. By automating key functionalities such as ticket sales, lottery draws, and prize distributions via smart contracts, this system eliminates the need for intermediaries, ensuring trustless interactions and immutable record-keeping. This decentralized design addresses the inefficiencies and trust deficits of traditional lottery systems, while providing scalability and cost-efficiency through blockchain technology.

Research Insights

Problem Domain

Traditional lottery systems are plagued by issues such as lack of transparency, centralized control, and vulnerability to fraud. These issues undermine user trust and often result in inefficiencies. Blockchain technology addresses these shortcomings by introducing immutable transaction records, publicly verifiable outcomes, and decentralized operations. These features form a robust foundation for a modernized, efficient lottery ecosystem.

Existing Solutions

1. **Traditional Lotteries**: Operated by centralized entities, these systems often suffer from limited transparency and demand blind trust in the integrity of the organizers. Fraud and mismanagement risks are significant.
2. **Blockchain-Based Lotteries**: While blockchain solutions use smart contracts to enhance transparency and automate operations, they face challenges such as scalability limitations, energy inefficiency, and the complexity of onboarding non-technical users.

Research Findings

- **Transparency**: Immutable blockchain ledgers guarantee publicly verifiable records for all transactions and lottery outcomes, removing ambiguity.
- **Efficiency**: The low transaction fees and rapid processing times of the Polygon network make micro-transactions and high-frequency interactions feasible.
- **Decentralization**: By eliminating centralized intermediaries, the system fosters trust and reduces operational risks.
- **User Accessibility**: Wallet integration and user-friendly interfaces simplify the onboarding process, broadening accessibility even to non-technical users.

Blockchain Architecture

Blockchain Type

- **Private vs Public**: This system employs a public blockchain, specifically the Polygon network, to maximize transparency, scalability, and interoperability within the Ethereum ecosystem.

Platform Choice

- **Polygon**: As a Layer-2 solution, Polygon was selected for its high transaction throughput, low fees, and compatibility with Ethereum development tools.
    - **Rationale**: Polygon offers a balanced trade-off between scalability and security, addressing network congestion and high gas fees inherent to Ethereum's Layer-1 network.

Consensus Mechanism

- **Proof of Stake (PoS)**:
    - **Advantages**:
        - **Security**: PoS incentivizes validator honesty through staking, reducing the risk of malicious behavior.
        - **Energy Efficiency**: Drastically reduces computational energy requirements compared to Proof of Work systems.
        - **Scalability**: Provides faster transaction finality, essential for seamless ticket purchasing and prize distributions.
    - **Project Integration**:
        - Supports high-frequency transactions and ensures quick user feedback.
        - Enhances user experience by minimizing confirmation delays.
    - **Challenges and Mitigation**:
        - **Validator Centralization**: Mitigated through incentives for distributed validator participation.
        - **Stake Slashing Risks**: Comprehensive user education campaigns build confidence in the staking model.

Data Structures to Be Used

Blocks

- **Purpose**: Record and secure all transactions, including ticket purchases, draw results, and prize distributions.
- **Structure**:
    - **Header**: Contains metadata such as timestamps, block hashes, and references to previous blocks.
    - **Body**: Stores transactional data, ensuring traceability and accountability.

Merkle Trees

- **Purpose**: Enable efficient verification of transactions while preserving data integrity and privacy.
- **Application**:
    - Organize and secure ticket purchase records, enabling rapid verification.
    - Facilitate proofs of transaction inclusion without exposing sensitive user data.

Mappings

- **Purpose**: Simplify data retrieval and association by linking user addresses to transaction data.
- **Example**:

```
mapping(address => uint) public ticketCounts;
```

- **Usage**: Maintain a dynamic record of the number of tickets purchased by each user.

Smart Contract Design

Purpose

The smart contracts underpin the entire system, automating core operations such as ticket purchases, random winner selection, and secure prize disbursement. They are designed for transparency, reliability, and robustness, ensuring seamless user interaction while mitigating risks.

Functionality

1. **Buy Tickets**:
    - Allows users to purchase lottery tickets by sending funds to the smart contract.
    - Updates internal mappings to track the number of tickets purchased per user.
2. **Draw Winner**:
    - Utilizes randomness oracles like Chainlink VRF to ensure fair and unbiased winner selection.
    - Logs the winning address and triggers the prize distribution process.
3. **Withdraw Winnings**:
    - Enables the lottery winner to securely claim their prize through a dedicated contract function.

Security Considerations

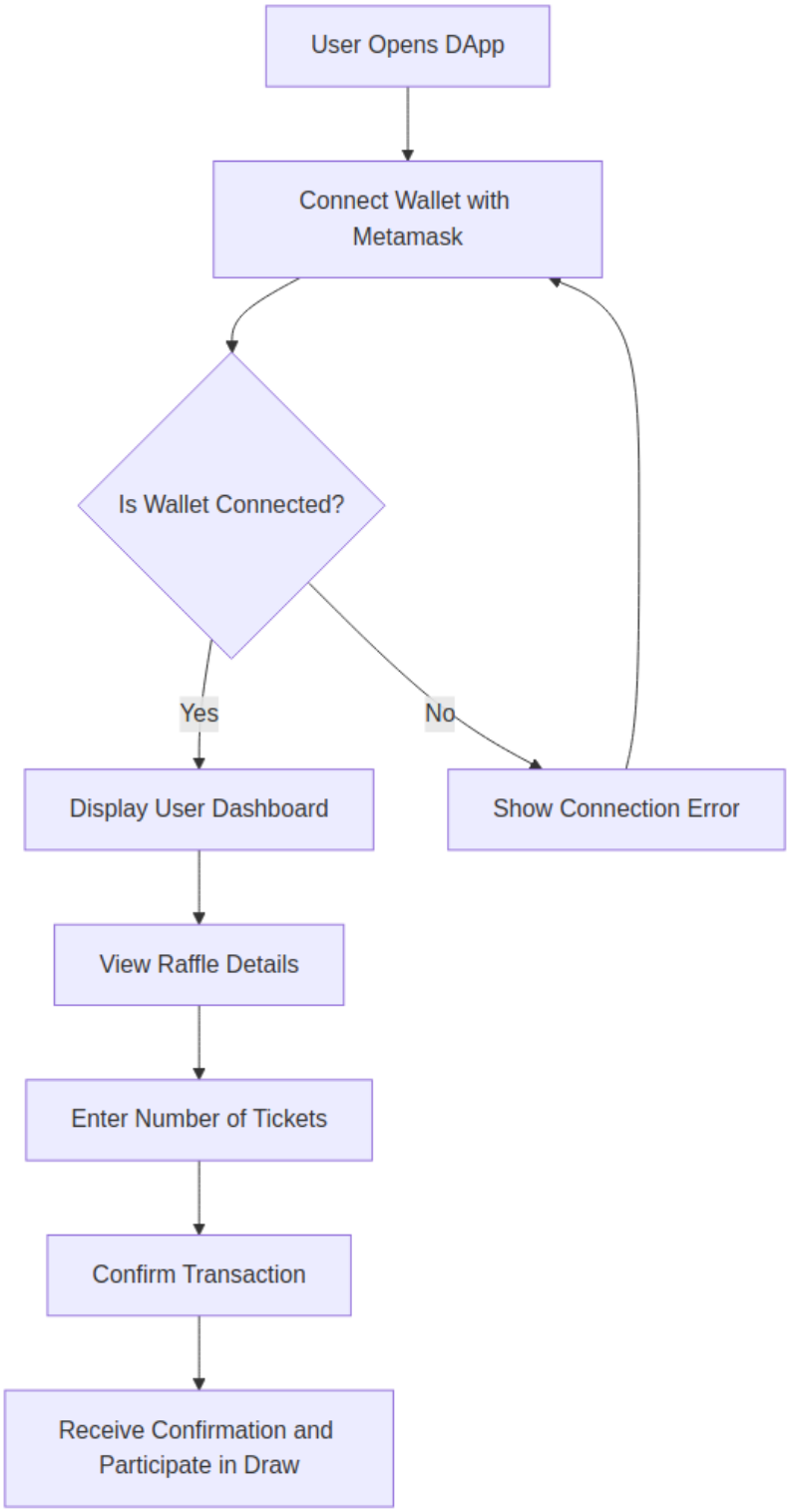
- **Randomness**:
    - Chainlink VRF is integrated to provide verifiably random numbers, ensuring tamper-proof outcomes.
- **Access Control**:
    - Implements role-based permissions, restricting critical administrative actions to authorized addresses.
- **Reentrancy Protection**:
    - Follows the `checks-effects-interactions` pattern to prevent potential reentrancy attacks.
- **Comprehensive Testing**:
    - Rigorous auditing, stress testing, and deployment simulations ensure system robustness and resilience against vulnerabilities.

System Workflows

User Interaction Flow

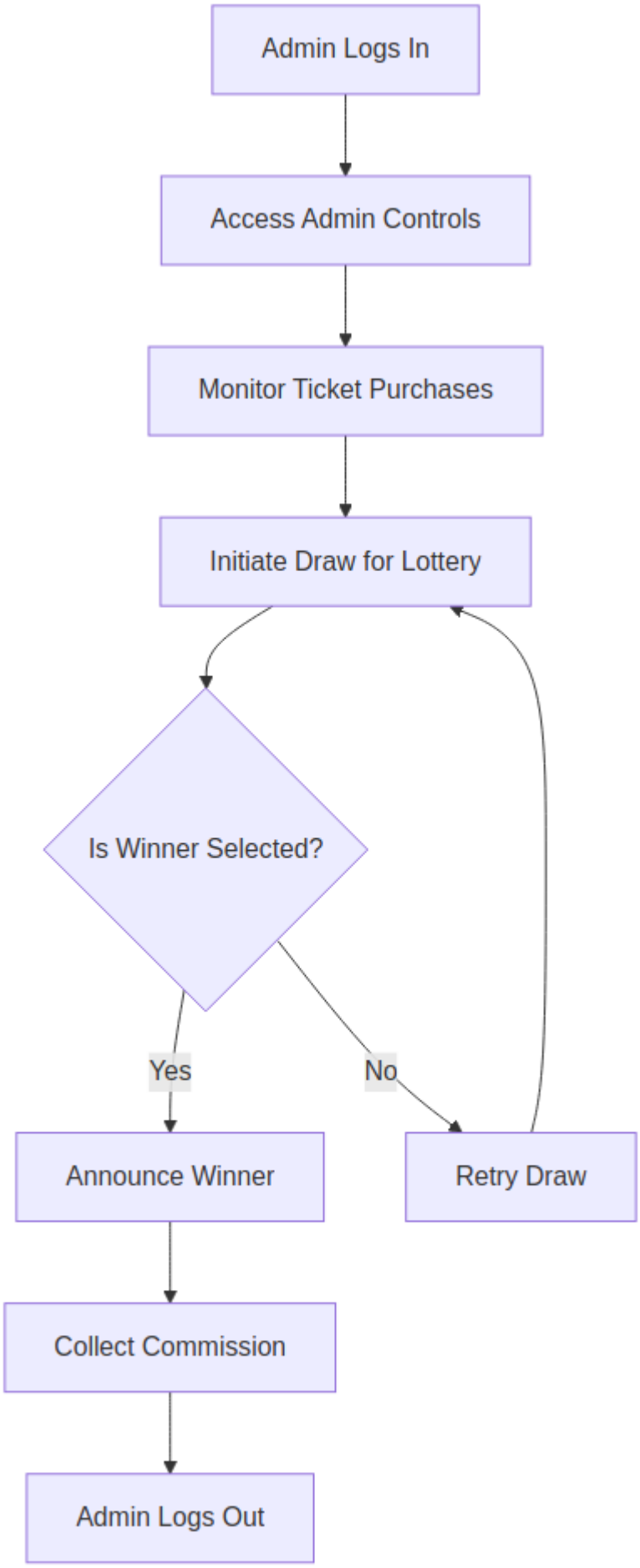
1. Connect a compatible wallet (e.g., Metamask) to the application.

2. Interact with the smart contract to purchase tickets using cryptocurrency.
3. Participate in the lottery draw conducted by the system.
4. Claim winnings if selected as the lottery winner.



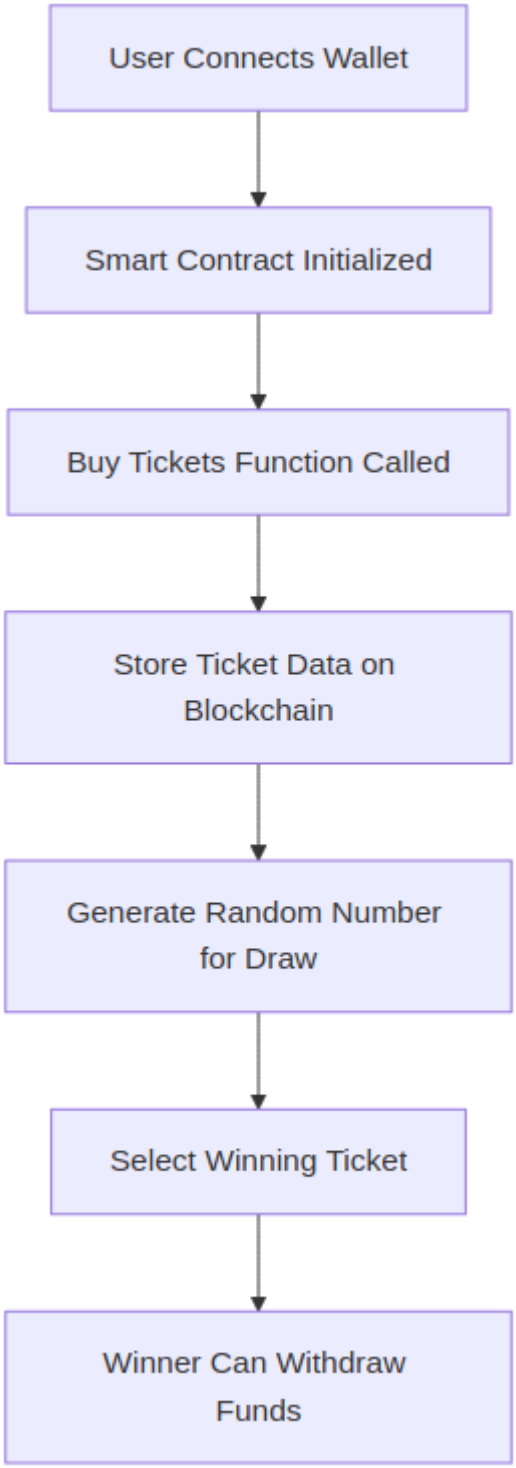
Administrator Workflow

1. Connect an administrative wallet to the system.
2. Trigger the `drawWinner` function to initiate the lottery draw.
3. Oversee system operations, including commission management and ensuring contract integrity.



Smart Contract Interaction

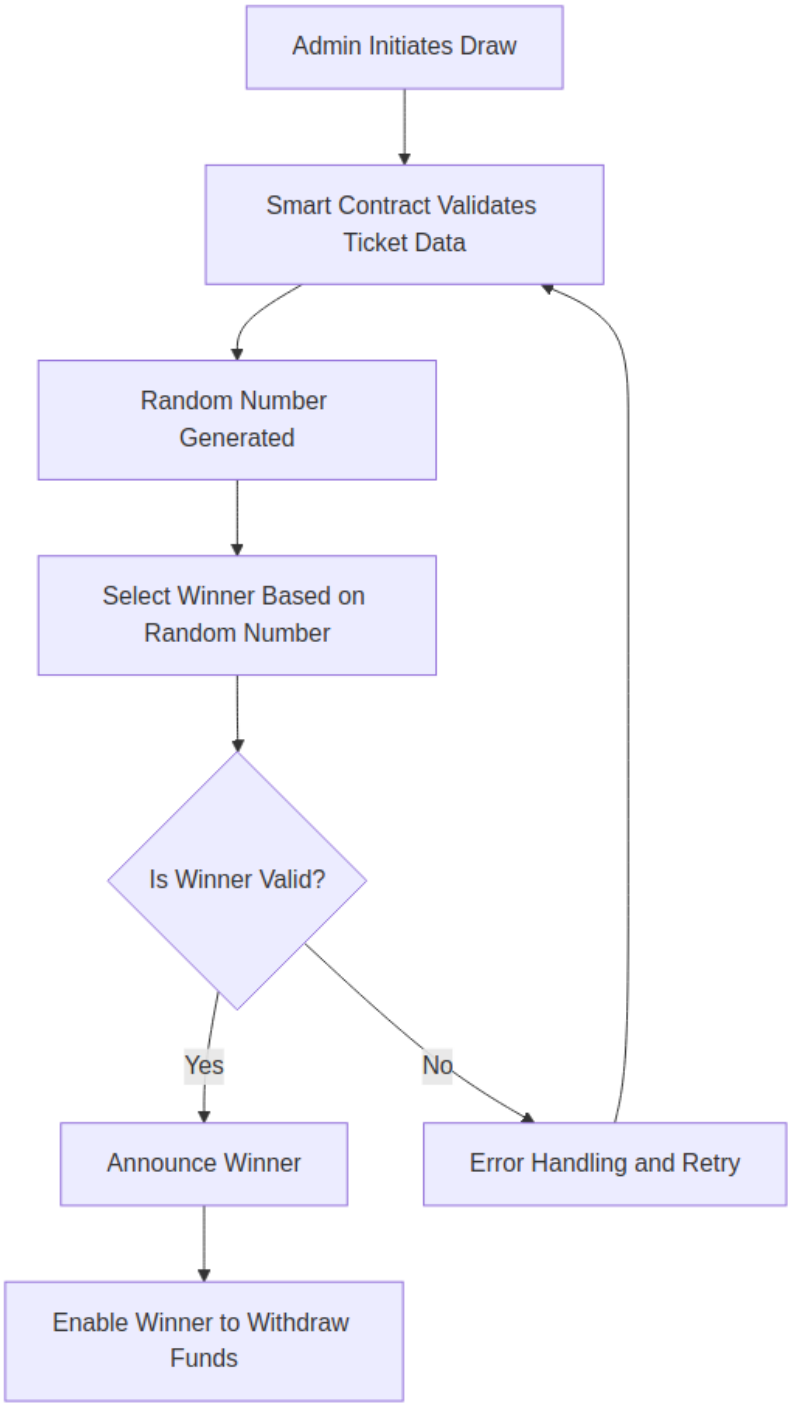
1. Users invoke the `buyTickets` function, sending the required funds and incrementing their ticket counts.
2. Administrators trigger the `drawWinner` function, initiating the randomness oracle for winner selection.
3. The selected winner calls the `withdrawWinnings` function to claim their prize securely.



Lottery Draw Process

1. The administrator initiates the lottery draw by calling the smart contract's draw function.
2. Chainlink VRF generates a verifiably random number.

3. The smart contract determines the winner based on the random output and logs the result for transparency.
4. The prize distribution process is automatically executed, ensuring timely and secure payouts.



Conclusion

The Web3 Lottery DApp epitomizes the transformative potential of blockchain technology in revolutionizing lottery systems. By harnessing the transparency, security, and automation capabilities of the Polygon blockchain, the system addresses the limitations of traditional lottery mechanisms. Its decentralized, trustless architecture, coupled with innovative features like Chainlink VRF and user-friendly wallet integration, ensures a fair and efficient lottery experience. Through meticulous design and robust implementation, this DApp establishes a new benchmark for modern, blockchain-powered lottery solutions.