**Interdependencies Between Modular Components**

Understanding the interactions between modular components ensures a seamless, flexible, and scalable framework. Below are the key interdependencies:

**1. Core Token Contract**

* **Depends on:**
  + **Reward/Burn Contract**: To initiate token transfers during rewards and burns.
  + **Governance Contract**: To allocate token-based voting rights.
* **Required by:**
  + **Cooldown Contract**: For tracking token balances and transactions.
  + **AI Middleware**: For real-time monitoring of supply and activity.

**2. Reward/Burn Contract**

* **Depends on:**
  + **Off-Chain Analytics Script**: To filter wallet eligibility.
  + **Chainlink VRF**: For executing randomness in wallet selection.
* **Required by:**
  + **Core Token Contract**: To initiate rewards and burns.
  + **AI Middleware**: To dynamically adjust reward and burn parameters.

**3. Cooldown Contract**

* **Depends on:**
  + **Core Token Contract**: For transaction tracking, including timestamps and wallet balance updates.
* **Required by:**
  + **Reward/Burn Contract**: To validate wallet eligibility.
  + **Governance Contract**: To ensure wallets in cooldown cannot participate in governance proposals.

**4. Governance Contract**

* **Depends on:**
  + **Core Token Contract**: To allocate voting rights and determine token-weighted votes.
  + **AI Middleware**: For proposal analysis and decision facilitation.
* **Required by:**
  + **Community Engagement Tools**: For automating and summarizing governance updates.
  + **Off-Chain Analytics Script**: For tracking feedback and participation.

**5. AI Middleware**

* **Depends on:**
  + **Off-Chain Analytics Script**: For data processing and dynamic adjustments.
  + **Governance Contract**: To facilitate governance-related decisions.
* **Required by:**
  + **Reward/Burn Contract**: For real-time adjustment of parameters.
  + **Community Engagement Tools**: To drive participation and engagement.

**6. Off-Chain Analytics Script**

* **Depends on:**
  + **Blockchain Explorer APIs (e.g., Shidoscan)**: For wallet data extraction.
* **Required by:**
  + **Reward/Burn Contract**: For filtering eligible wallets.
  + **AI Middleware**: To analyze market and user data for decision-making.

**7. Community Engagement Tools**

* **Depends on:**
  + **AI Middleware**: For event creation, leaderboard tracking, and notifications.
  + **Governance Contract**: For participation in polls and events.
* **Required by:**
  + **Token Holders**: For interactive tools like trivia, leaderboards, and updates.

**Prioritized Development Plan**

1. **Core Token Contract**
   * Foundation for all functionality; critical for basic token operations.
2. **Reward/Burn Contract**
   * Enables daily reward and burn cycles, a core feature of the project.
3. **Cooldown Contract**
   * Ensures disciplined behavior and fairness in rewards.
4. **Off-Chain Analytics Script**
   * Supports wallet filtering and enhances reward/burn functionality.
5. **AI Middleware**
   * Adds intelligence to reward/burn cycles and engagement tools.
6. **Governance Contract**
   * Finalizes community involvement and decentralized decision-making.
7. **Community Engagement Tools**
   * Enhances user interaction and education, boosting project appeal.

**Logical Flowchart: Reward/Burn Wallet Process**

Below is a high-level breakdown of the Reward/Burn Wallet process:

**1. Daily Trigger**

* **Input:** Current day type (reward or burn).
* **Logic:** The system alternates between reward and burn days.

**2. Wallet Eligibility**

* **Process:**
  + Off-chain script collects all wallet addresses.
  + Filters applied:
    - Minimum balance requirement.
    - Cooldown status check.
    - Exclusion of team wallets and known exploit wallets.
* **Output:** A list of eligible wallets.

**3. Random Wallet Selection**

* **Process:** Chainlink VRF selects up to 10% of eligible wallets (maximum of 1,000 wallets).
* **Output:** Selected wallets for rewards.

**4. Reward/Burn Execution**

* **Reward Day:** Distribute 1% of the reward wallet evenly to selected wallets.
* **Burn Day:** Burn 1% of the reward wallet to reduce supply.
* **Output:** On-chain transaction log.

**5. Post-Execution Adjustments**

* **Process:**
  + Update wallet balances.
  + Record transactions on-chain.
  + Prepare analytics for community updates.

**Final Notes**

This document ensures modular components are designed to complement each other while maintaining scalability and efficiency. The logical flowchart outlines the reward/burn process in a step-by-step manner, emphasizing the hybrid on-chain and off-chain approach for optimal performance.