**1. Core Token Contract**

**Responsibilities:**

* Manage token balances, transfers, and total supply.
* Handle basic ERC20 functionality.
* Integrate with the Reward/Burn Contract for seamless interactions.

**Key Features:**

* Compatibility with modular external contracts.
* No dependency on reflective functions, keeping it clean and efficient.

**Next Steps:**

* Define the basic ERC20 implementation.
* Determine how it interacts with the Reward/Burn Contract without overriding core ERC functions.

**2. Reward/Burn Contract**

**Responsibilities:**

* Automate daily reward and burn cycles.
* Coordinate with off-chain data for wallet eligibility.
* Use Chainlink VRF for randomness in wallet selection.

**Key Features:**

* Hybrid approach for wallet filtering (off-chain) and randomness (on-chain).
* Transparent on-chain reward and burn transactions.

**Next Steps:**

* Complete logical flowchart (as started above).
* Draft the smart contract to handle on-chain functions.

**3. Cooldown Contract**

**Responsibilities:**

* Enforce cooldown periods for wallets after sales.
* Log and monitor transactions to determine cooldown status.

**Key Features:**

* Record "last sold timestamp" for each wallet.
* Allow querying cooldown status on-chain.

**Next Steps:**

* Define cooldown parameters based on sell percentages.
* Write logic for cooldown status tracking and requalification.

**4. Governance Contract**

**Responsibilities:**

* Enable proposal creation and voting by token holders.
* Leverage AI for summarizing proposals and ensuring fair governance.

**Key Features:**

* Weighted voting (e.g., based on holdings or activity).
* Transparent on-chain records of voting outcomes.

**Next Steps:**

* Map out proposal lifecycle (creation → voting → execution).
* Integrate AI tools to analyze and summarize proposals.

**5. AI Middleware**

**Responsibilities:**

* Connect blockchain data to off-chain resources for advanced analytics.
* Dynamically adjust reward/burn parameters based on community activity and market conditions.
* Automate tasks like engagement and governance facilitation.

**Key Features:**

* Use APIs for real-time data feeds and decision-making.
* Automate notifications and updates via Telegram and Discord bots.

**Next Steps:**

* Define integration points (e.g., with Chainlink, off-chain scripts).
* Specify AI models and workflows for dynamic adjustments.

**6. Off-Chain Analytics Script**

**Responsibilities:**

* Collect and process wallet data for eligibility filtering.
* Identify trends and insights to inform on-chain decisions.

**Key Features:**

* Interact with blockchain explorers (e.g., Shidoscan) to pull wallet data.
* Apply filters for eligibility based on project-defined parameters.

**Next Steps:**

* Draft a script to handle data collection and filtering.
* Develop APIs for seamless data transfer to on-chain contracts.

**7. Community Engagement Tools**

**Responsibilities:**

* Facilitate trivia events, polls, and educational content through AI agents.
* Provide leaderboards and rewards for active participants.

**Key Features:**

* Use AI to automate event creation and notifications.
* Integrate with governance to promote community-driven initiatives.

**Next Steps:**

* Map out interactive features (e.g., leaderboards, trivia mechanics).
* Develop a roadmap for AI-driven engagement expansion.