**The 1000 Project**

**Overview**

The 1000 Project is an ambitious, AI-driven cryptocurrency initiative designed to promote responsible investing and innovative tokenomics. Built on the Shido blockchain, this project integrates hybrid on-chain/off-chain workflows, advanced AI middleware, and sustainable community governance mechanisms.

**Features**

* **Hybrid Workflow**: Combines on-chain operations with off-chain analytics for efficiency.
* **Reward and Burn Cycles**: A 1000-day system alternating between reward distribution and token burns.
* **AI Middleware**: Optimizes wallet eligibility, monitors cooldown periods, and prevents exploits.
* **Chainlink VRF Integration**: Ensures fair, randomized wallet selection for rewards.
* **Community Engagement Tools**: Encourages participation through governance, trivia, and real-time updates.

**File Structure**

**Scripts**

1. **setupanddependencies.py**: Installs necessary libraries and configures environment settings.
2. **mainfunction.py**: Orchestrates daily reward and burn processes, managing calls to sub-scripts.
3. **fetch\_wallet\_data.py**: Fetches wallet metadata from the blockchain for processing.
4. **filter\_eligible\_wallets.py**: Applies eligibility filters such as minimum balance, cooldown periods, and blacklist status.
5. **detect\_p2p\_transfers.py**: Detects and flags anomalous peer-to-peer transfers to prevent system exploitation.
6. **export\_selected\_wallets.py**: Compiles the eligible wallet list and prepares it for Chainlink VRF integration.
7. **call\_chainlink\_vrf.py**: Handles communication with Chainlink VRF for random wallet selection.

**Documentation**

* **README.md**: Project overview, file structure, and setup instructions.
* **workflow\_diagrams.svg**: Visual representation of hybrid workflows and middleware logic.
* **reward\_burn\_logic.txt**: Text-based pseudocode of reward and burn processes for clarity and reference.

**Workflow Overview**

**Daily Reward/Burn Process**

1. **Fetch Wallet Data**:
   * Pull wallet metadata (addresses, balances, transaction history) from the blockchain.
   * Store data in the middleware for processing.
2. **Filter Ineligible Wallets**:
   * Apply minimum balance requirements, cooldown flags, and blacklist exclusions.
   * Output: A list of eligible wallets.
3. **Chainlink VRF Integration**:
   * Submit the eligible wallet list to Chainlink VRF for random selection (10% or max 1,000 wallets).
4. **Execute Operations**:
   * On reward days: Distribute 1% of the reward wallet equally among selected wallets.
   * On burn days: Burn 1% of the reward wallet to reduce supply.
5. **Log Transactions**:
   * Record all reward/burn transactions on-chain for transparency.

**Setup Instructions**

1. **Install Dependencies**: Run the setupanddependencies.py script to install required Python libraries.

python3 setupanddependencies.py

1. **Run the Main Workflow**: Execute the mainfunction.py script to start the daily reward and burn process.

python3 mainfunction.py

1. **View Logs**: Check transaction and middleware logs to verify execution.

**Roadmap**

**Phase 1: Foundation**

* Finalize middleware scripts and hybrid workflows.
* Begin testing reward and burn functionality on the Shido testnet.

**Phase 2: Community Engagement**

* Deploy basic Telegram/Discord bots for notifications.
* Share updates and progress via X and Telegram.

**Phase 3: AI Middleware Integration**

* Add adaptive reward/burn logic and advanced wallet analytics.

**Phase 4: Governance and Launch**

* Finalize community-driven governance workflows.
* Launch token with a fair distribution model and full functionality.

**Contributing**

Contributions are welcome! For any ideas or suggestions, please create an issue or submit a pull request. Let’s innovate together.

**Contact**

* **Telegram**: [The 1000 Project](https://t.me/The1000Project)
* **X (formerly Twitter)**: [@1000CryptoAI](https://x.com/1000CryptoAI)
* **Email**: 1000cryptoai@gmail.com