

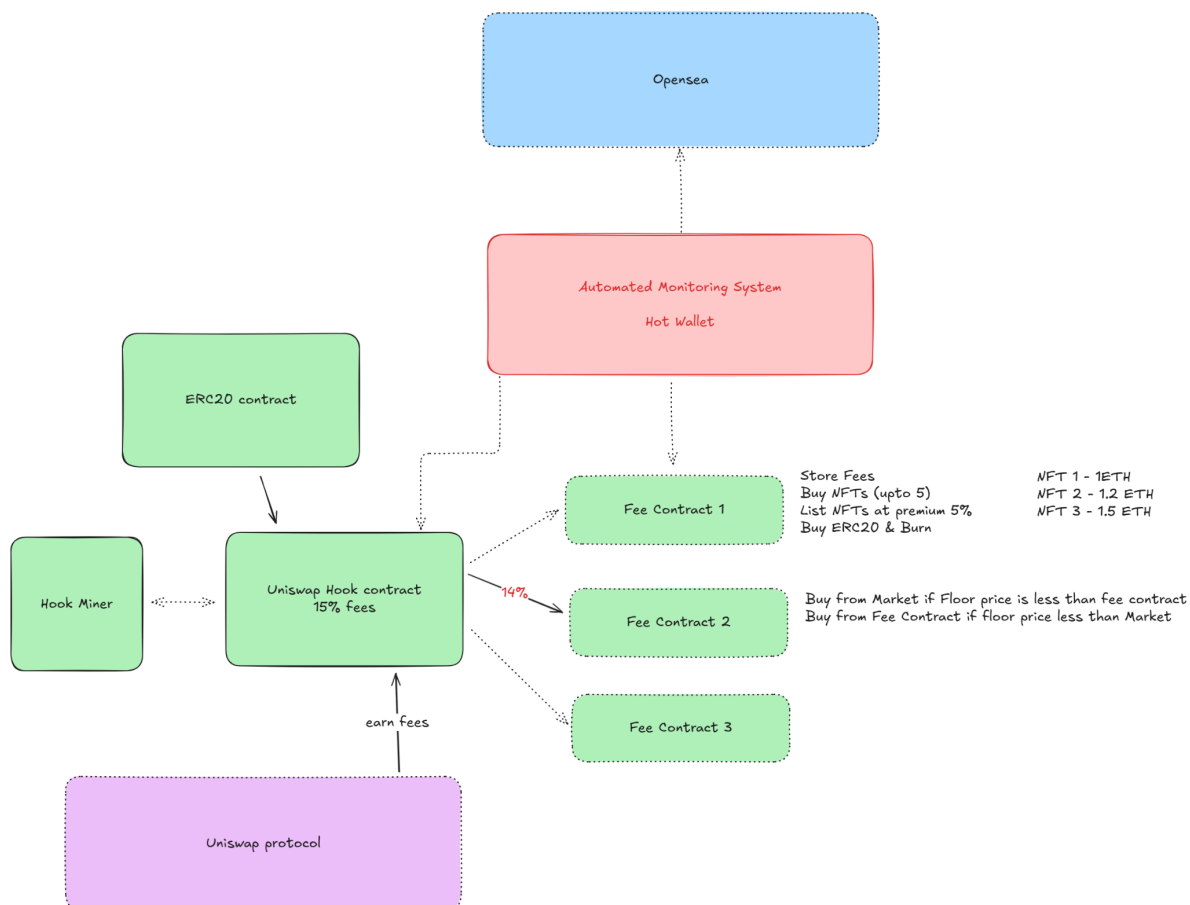
Rarity Town Protocol – Smart Contract & System Architecture

Overview

The RARITY Protocol is a decentralized ecosystem designed to create deflationary pressure on the RARITY Token through automated NFT trading and token burn mechanisms.

Users buy and sell RARITY Token via the project's own dApp or through Uniswap. A 15% transaction fee is collected automatically.

The ETH is pooled into the active **fee contract**, and when sufficient balance is reached, the protocol automatically purchases **floor NFTs** (e.g., BAYC). NFTs are listed at a **5% premium**. When sold, the proceeds are used to **buy back and burn RARITY tokens**, creating long-term value for holders.



System Components

1. ERC20 Contract (RARITY Token)

Purpose:

Core token of the ecosystem. Designed to interact with the Uniswap Hook Contract for fee collection and burning logic.

Key Features:

- ERC20 standard (OpenZeppelin base)
 - `transfer()` is disabled so holders cannot transfer to anyone else
 - `burn()` function callable only by the **Fee Contracts** or **Automated System**
 - Supply tracking and event emission on burn
-

2. Uniswap Hook Contract

Purpose:

Acts as the liquidity and fee routing layer for RARITY Token trades on Uniswap V4. It captures transaction fees and directs them to the Fee Contracts.

Key Features:

- Integrated with Uniswap V4 Hooks
- Enforces 15% transaction fee
- Splits collected fees as:
 - 14% to fee contract
 - 1% to founder wallet / Buy & burn brand asset token
- Creates Fee Contracts to buy NFTs
- Routes the collected ETH in the active Fee contract
- Triggers events for fee distribution

- Configurable fee percentage via admin control
-

3. Hook Miner

Purpose:

Rewards active liquidity miners who interact with the Hook Contract.
It ensures fair incentive distribution to early participants and liquidity providers.

Key Features:

- Tracks mining activity and earned fees
 - Smart reward claim system
 - Integrates with the Uniswap Hook Contract for performance tracking
-

4. Fee Contract

Each Fee Contract represents a vault-like ETH pool responsible for NFT purchases and token burns.

Responsibilities:

- Accumulate ETH from swap fees
- Execute NFT purchase (up to 5 NFTs)
- Purchase logic:
 - Buy from **market** if NFT floor < Fee Contract NFT value
 - Buy from **Fee Contract** if NFT floor > Fee Contract listing
- List NFTs on OpenSea at **5% premium price**
- On NFT sale:
 - Receive ETH
 - **buyback and burn RARITY token**

5. Automated Monitoring System (Backend)

Purpose:

A Node.js-based system continuously monitoring ETH balances, NFT prices, and OpenSea listings. It acts as the **automation engine** and **hot wallet operator**.

Key Features:

- Monitors ETH balance in Fee Contracts
- Tracks BAYC floor price via OpenSea APIs
- Triggers NFT buy/sell when thresholds are met
- Manages relayer transactions (hot wallet)
- Trigger RARITY buyback and burn after NFT sales
- Logs transaction history to MongoDB
- Emits events to frontend (React dashboard)

6. OpenSea Integration

Purpose:

Interface for NFT trading automation.

Key Features:

- Fetch BAYC floor price (API)
 - Automated NFT purchase (via OpenSea Seaport contract)
 - Automated relisting at 1.2x price
 - Sale detection and callback to backend system
-

System Workflow

1. User trades RARITY Token on dApp or Uniswap → 15% fee collected
2. ETH routed to Fee Contract
3. Automated Monitoring detects sufficient ETH to buy BAYC NFT
4. Fee Contract buys NFT via OpenSea → lists for 5% profit
5. NFT sells → ETH received
6. System triggers buyback & burn of RARITY Token
7. Dashboard updates NFT and burn stats