Time-Limited NFT Membership with Gasless Renewals

A Lightweight Web3 System with Meta-Transactions and RBAC

Ferial Najiantabriz
University of Oklahoma

Project Overview

- **NFT Membership Pass:** ERC-721 token with on-chain expiration logic
- **Renewal Support:** Users can renew even after expiration
- **Gasless Renewal:** EIP-712 signatures submitted by a relayer using EIP-2771 MinimalForwarder
- **Role-Based Access Control (RBAC):** Admin-only price control with secure role grant/revoke
- **Simple, Secure, and Forkable:** No proxies, SDKs, or third-party dependencies

Deployed on Sepolia Testnet using Brownie and Web3.py

Problem and Motivation

- **Why this project?**
  - Unlock and Lit Protocols are powerful, but often complex and overbuilt
  - Smaller communities (DAOs, clubs, research groups) need something lightweight and self-hosted
- **Our Goals:**
  - Easy to deploy and test
  - Minimal gas cost
  - No front-end dependency
  - Support for casual users (no ETH needed)

Motivation & Use Case

- **Why this problem matters:**
  - Communities like DAOs, student clubs, and online groups need access control
  - Centralized platforms are limited and not trustless
- **Who needs this system:**
  - Student organizations offering membership access
  - Research teams, event groups, or private Discord/Zoom users
  - Any group needing short-term or renewable Web3 access
- **Why Unlock/Lit are not ideal:**
  - Unlock uses factory + proxy contracts — complex to deploy or audit
  - Lit is built for off-chain encryption — not focused on time-based access
  - Both assume advanced frontend/backend integration
- **Our Solution:**
  - Simple, on-chain, gas-efficient, secure with RBAC
  - Designed for small groups with low cost and easy setup

Technical Architecture Overview

- **NFTMembership Contract:**
  - Based on ERC-721 standard
  - Adds `validUntil` field for expiration logic
  - Includes `mintMembership`, `renewMembership`, and `metaRenewMembership`
- **MinimalForwarder Contract:**
  - Implements EIP-2771 (meta-transaction forwarder)
  - Verifies EIP-712 signature and nonce
  - Appends real user to calldata and calls NFT contract
- **Gasless Flow:**
  - User signs off-chain message (EIP-712)
  - Relayer submits signed message to forwarder
  - Forwarder calls contract as if user submitted
- **Role-Based Access Control (RBAC):**
  - Only `ADMIN_ROLE` can update pricing or policies
  - Roles managed via Brownie script (`manage_roles.py`)

Meta-Transaction Flow (EIP-712 + EIP-2771)

- **Step 1:** User signs a structured message off-chain (EIP-712)
- **Step 2:** Relayer sends message to MinimalForwarder
- **Step 3:** Forwarder verifies signature + nonce
- **Step 4:** Forwarder appends real sender, calls NFTMembership
- **Step 5:** NFTMembership renews token + emits event
- **Key Benefit:** User does not need ETH to renew