ecoSave

ACT. SAVE. GROW.

Ecosave Token Smart Contract

Table of Contents

1. Introduction

The EcoSaveToken is an ERC-20 compliant token contract designed to facilitate eco-friendly initiatives, rewards, and platform governance through a structured minting and locking mechanism. This contract is built on top of OpenZeppelin libraries and incorporates advanced security features, such as signature verification and time-lock mechanisms. The contract includes functionalities such as minting new tokens based on community voting (but with a maximum cap of 1 billion tokens), locking tokens for specified durations, and claiming locked tokens.

2. Contract Overview

Contract Name: EcoSaveToken
Inheritance:
- ERC20: Inherits standard ERC-20 functionality from OpenZeppelin.
- Ownable: Ensures owner-restricted functions.
- ReentrancyGuard: Protects against reentrancy attacks.
- SafeMath: Provides safe arithmetic operations, avoiding overflows and underflows.
- ECDSA: Enables signature verification for minting and voting.

3. Key Features and Functionalities

I. Tokenomics

- Max Supply: 1,000,000,000 tokens (1 billion).
- Initial Supply: 425,000,000 tokens minted at deployment.
- Reward Pool: 150,000,000 tokens reserved for reward-related purposes.
- Eco-Friendly Pool: 150,000,000 tokens reserved for eco-friendly initiatives.
- Owner Supply: The portion of the initial supply reserved for the contract owner, intended for various purposes including pre-sale distribution, ICO participation, strategic partnerships, airdrops, and team allocations.

II. Minting Tokens

Purpose: Allows minting of new tokens based on a community-approved voting process. The voting signature is verified using ECDSA. Once the ecosystem reaches its maximum supply of 1 billion tokens, the smart contract will automatically block any further minting attempts.
Key Features: Minting follows a structured time-based minting schedule over an 8-year period, and the tokens are distributed across the reward pool, eco-friendly pool, and owner.
Security: Uses `onlyValidSignature` to ensure that only valid, signed minting requests from the platform are executed. Implements nonce validation to prevent replay attacks.

III. Locking and Claiming Tokens

Locking Tokens: Locks tokens for a specified duration, typically 1 year for ICO transfers.
Claiming Locked Tokens: Allows users to claim their locked tokens after the lockup period.

Batch Claiming Locked Tokens: Allows multiple users to claim their locked tokens in one transaction.

IV. Platform Public Key Update

Initiating Public Key Update: Initiates a public key update that is subject to a 3-day time-lock.
Finalizing Public Key Update: Finalizes the update after the time-lock period.
Canceling Public Key Update: Cancels a pending public key update before the time-lock expires.

V. Burning Tokens

Purpose: Allows token holders to burn their tokens, permanently reducing the total supply.

4. Security Mechanisms

I. Reentrancy Protection

All critical functions that involve token transfers or external calls are protected by the `nonReentrant` modifier, preventing reentrancy attacks.

II. Signature Verification

The platform uses signatures for minting and voting, ensuring that only authorized requests are processed. The signature verification process uses ECDSA to recover the signer's address and compare it with the platform's public key.

III. Time-Lock for Public Key Updates

A time-lock mechanism ensures that the platform's public key cannot be immediately updated, providing a 3-day window for stakeholders to review the change.

IV. Nonce Management

Nonce management is implemented to prevent replay attacks by requiring a unique nonce for each minting request.

V. No Ether Handling

The contract explicitly rejects any Ether sent to it by reverting transactions in both the `receive()` and `fallback()` functions, preventing unintended Ether deposits.

5. Events for Transparency

TokensTransferred: Emitted on every token transfer, with additional metadata.
LockedTokensClaimed: Emitted when locked tokens are claimed by a user.
BatchLockedTokensClaimed: Emitted when multiple users claim locked tokens in a batch.
MintWithVotingApproval: Emitted when tokens are minted following community voting.
PlatformPublicKeyUpdateInitiated: Emitted when a public key update is initiated.

PlatformPublicKeyUpdated: Emitted when the public key is updated after the time-lock period.
TokensBurned: Emitted when tokens are burned by a user.

6. Conclusion

The EcoSaveToken smart contract incorporates robust tokenomics and eco-friendly features while leveraging security measures such as reentrancy guards, ECDSA-based signature verification, and time-locked updates. With clear mechanisms for token minting, transfer, and locking, the contract ensures a transparent, secure, and eco-conscious token economy.

Token Address:

0x3aB279CF87d9Ebe968A9a2ea7fF3A67BC000b7f0

7. Code

```
// SPDX-License-Identifier: MIT
pragma solidity 0.8.26;

import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v4.9.2/
contracts/token/ERC20/ERC20.sol";
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v4.9.2/
contracts/access/Ownable2Step.sol";
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v4.9.2/
contracts/utils/cryptography/ECDSA.sol";
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v4.9.2/
contracts/security/ReentrancyGuard.sol";

contract EcoSaveToken is ERC20, Ownable2Step, ReentrancyGuard {
  using ECDSA for bytes32;

  // Custom Errors
  error InvalidPlatformKey();
  error InvalidSignature();
  error MaxSupplyExceeded();
  error MintingNotAllowed();
  error InvalidNonce();
  error IncorrectMintAmount();
  error InvalidRecipient();
  error TransferFailed();
  error SameRewardPoolAmount();
  error SameEcoFriendlyPoolAmount();
  error NoMoreMintingPeriods();
  error NoPendingUpdate();
  error TimelockNotFinished();
  error SamePlatformPublicKey();
  error EtherNotAccepted();

  uint256 private constant _TOKEN_DECIMALS = 18;
  uint256 private constant _MAX_SUPPLY = 1_000_000_000 * 1e18; // 1 billion tokens
  uint256 private constant _INITIAL_SUPPLY = 425_000_000 * 1e18; // Adjusted
initial supply
  uint256 private constant _REWARD_POOL = 150_000_000 * 1e18;
  uint256 private constant _ECO_FRIENDLY_POOL = 150_000_000 * 1e18;
  uint256 private constant _OWNER_SUPPLY = _INITIAL_SUPPLY - _REWARD_POOL -
_ECO_FRIENDLY_POOL;
  uint256 private constant _TIMELOCK_DURATION = 3 days;

  uint256 public immutable deploymentTime;
  address public platformPublicKey;
  uint256 public rewardPoolBalance;
  uint256 public ecoFriendlyPoolBalance;
```

```
    address public pendingPlatformPublicKey;
    uint256 public timelockEndTime;

    mapping(address user => uint256 nonce) public nonces;

    event TokensBurned(address indexed burner, uint256 amount);
    event MintWithVotingApproval(address indexed owner, uint256 amount, bytes32
votingId, uint256 currentNonce);
    event PlatformPublicKeyUpdateInitiated(address indexed oldPublicKey, address
indexed newPublicKey);
    event PlatformPublicKeyUpdateCancelled(address indexed owner);
    event PlatformPublicKeyUpdated(address indexed oldPublicKey, address indexed
newPublicKey);
    event RewardsPoolUpdated(address indexed owner, uint256 newAmount);
    event EcoFriendlyPoolUpdated(address indexed owner, uint256 newAmount);

    constructor(address _platformPublicKey) ERC20("EcoSaveToken", "ECOSAVE") {
        if (_platformPublicKey == address(0)) {
            revert InvalidPlatformKey();
        }
        platformPublicKey = _platformPublicKey;

        _mint(msg.sender, _OWNER_SUPPLY); // Mint initial tokens to owner
        _mint(msg.sender, _REWARD_POOL);  // Mint reward pool tokens to owner
        _mint(msg.sender, _ECO_FRIENDLY_POOL);  // Mint eco-friendly pool tokens to
owner

        rewardPoolBalance = _REWARD_POOL;
        ecoFriendlyPoolBalance = _ECO_FRIENDLY_POOL;
        deploymentTime = block.timestamp;
    }

    function decimals() public pure override returns (uint8) {
        return uint8(_TOKEN_DECIMALS);
    }

    modifier validRecipient(address to) {
        if (to == address(0)) {
            revert InvalidRecipient();
        }
        _;
    }

    modifier onlyAfterTimelock() {
        if (block.timestamp < timelockEndTime) {
            revert TimelockNotFinished();
        }
        _;
    }
```

```
  modifier pendingUpdateExists() {
    if (pendingPlatformPublicKey == address(0)) {
      revert NoPendingUpdate();
    }
    _;
  }

  function _verifySignature(
    address sender,
    uint256 amount,
    bytes32 votingId,
    uint256 currentNonce,
    bytes memory signature
  ) internal view {
    bytes32 messageHash = keccak256(abi.encodePacked(sender, amount, votingId,
currentNonce));
    bytes32 ethSignedMessageHash = messageHash.toEthSignedMessageHash();
    address _platformPublicKey = platformPublicKey;
    address recoveredAddress = ethSignedMessageHash.recover(signature);
    if (recoveredAddress != _platformPublicKey) {
      revert InvalidSignature();
    }
  }

  function mint(
    uint256 amount,
    bytes32 votingId,
    uint256 currentNonce,
    bytes memory signature
  ) external nonReentrant onlyOwner {
    _verifySignature(msg.sender, amount, votingId, currentNonce, signature);

    if (currentNonce != nonces[msg.sender]) {
      revert InvalidNonce();
    }
    nonces[msg.sender] = currentNonce + 1;

    uint256 period = (block.timestamp - deploymentTime) / 365 days;
    uint256 mintAmount;
    if (period == 1) mintAmount = 10_000_000 * 1e18;  // Year 1
    else if (period == 2) mintAmount = 15_000_000 * 1e18;  // Year 2
    else if (period == 3) mintAmount = 20_000_000 * 1e18;  // Year 3
    else if (period == 4) mintAmount = 30_000_000 * 1e18;  // Year 4
    else if (period == 5) mintAmount = 45_000_000 * 1e18;  // Year 5
    else if (period == 6) mintAmount = 60_000_000 * 1e18;  // Year 6
    else if (period == 7) mintAmount = 75_000_000 * 1e18;  // Year 7
    else if (period == 8) mintAmount = 95_000_000 * 1e18;  // Year 8
    else if (period == 9) mintAmount = 110_000_000 * 1e18;  // Year 9
```

```
    else if (period == 10) mintAmount = 115_000_000 * 1e18;  // Year 10
    else {
      revert NoMoreMintingPeriods();
    }

    if (mintAmount != amount) {
      revert IncorrectMintAmount();
    }

    if (totalSupply() + amount > _MAX_SUPPLY) {
      revert MaxSupplyExceeded();
    }

    uint256 rewardShare = (mintAmount * 30) / 100;
    uint256 ecoFriendlyShare = (mintAmount * 30) / 100;
    rewardPoolBalance = rewardPoolBalance + rewardShare;
    ecoFriendlyPoolBalance = ecoFriendlyPoolBalance + ecoFriendlyShare;

    _mint(msg.sender, mintAmount - rewardShare - ecoFriendlyShare);
    _mint(msg.sender, rewardShare);
    _mint(msg.sender, ecoFriendlyShare);

    emit MintWithVotingApproval(msg.sender, amount, votingId, currentNonce);
  }

  function transfer(address to, uint256 value) public override validRecipient(to)
returns (bool) {
    return super.transfer(to, value);
  }

  function transferFrom(address from, address to, uint256 value) public override
validRecipient(to) returns (bool) {
    return super.transferFrom(from, to, value);
  }

  function burn(uint256 amount) external {
    _burn(msg.sender, amount);
    emit TokensBurned(msg.sender, amount);
  }

  function updateRewardPool(
    uint256 newAmount,
    bytes32 id,
    uint256 currentNonce,
    bytes memory signature
  ) external nonReentrant onlyOwner {
    _verifySignature(msg.sender, newAmount, id, currentNonce, signature);
    if (rewardPoolBalance == newAmount) {
      revert SameRewardPoolAmount();
```

```
    }
    rewardPoolBalance = newAmount;
    emit RewardsPoolUpdated(msg.sender, newAmount);
  }

  function updateEcoFriendlyPool(
    uint256 newAmount,
    bytes32 id,
    uint256 currentNonce,
    bytes memory signature
  ) external nonReentrant onlyOwner {
    _verifySignature(msg.sender, newAmount, id, currentNonce, signature);
    if (ecoFriendlyPoolBalance == newAmount) {
      revert SameEcoFriendlyPoolAmount();
    }
    ecoFriendlyPoolBalance = newAmount;
    emit EcoFriendlyPoolUpdated(msg.sender, newAmount);
  }

  function initiatePlatformPublicKeyUpdate(address newPlatformPublicKey) external
onlyOwner {
    if (newPlatformPublicKey == address(0)) {
      revert InvalidPlatformKey();
    }
    if (pendingPlatformPublicKey == newPlatformPublicKey) {
      revert SamePlatformPublicKey();
    }
    pendingPlatformPublicKey = newPlatformPublicKey;
    timelockEndTime = block.timestamp + _TIMELOCK_DURATION;
    emit PlatformPublicKeyUpdateInitiated(platformPublicKey, newPlatformPublicKey);
  }

  function cancelPlatformPublicKeyUpdate() external onlyOwner pendingUpdateExists {
    pendingPlatformPublicKey = address(0);
    timelockEndTime = 0;
    emit PlatformPublicKeyUpdateCancelled(msg.sender);
  }

  function finalizePlatformPublicKeyUpdate() external onlyOwner onlyAfterTimelock
pendingUpdateExists {
    address oldPublicKey = platformPublicKey;
    platformPublicKey = pendingPlatformPublicKey;
    pendingPlatformPublicKey = address(0);
    timelockEndTime = 0;
    emit PlatformPublicKeyUpdated(oldPublicKey, platformPublicKey);
  }

  receive() external payable {
    revert EtherNotAccepted();
```

```
    }

    fallback() external payable {
        revert EtherNotAccepted();
    }
}
```

8. ABI


```
[
 {
  "inputs": [
   {
    "internalType": "address",
    "name": "_platformPublicKey",
    "type": "address"
   }
  ],
  "stateMutability": "nonpayable",
  "type": "constructor"
 },
 {
  "inputs": [],
  "name": "EtherNotAccepted",
  "type": "error"
 },
 {
  "inputs": [],
  "name": "IncorrectMintAmount",
  "type": "error"
 },
 {
  "inputs": [],
  "name": "InvalidNonce",
  "type": "error"
 },
 {
  "inputs": [],
  "name": "InvalidPlatformKey",
  "type": "error"
 },
 {
  "inputs": [],
  "name": "InvalidRecipient",
  "type": "error"
 },
 {
  "inputs": [],
  "name": "InvalidSignature",
  "type": "error"
```

```
},
{
  "inputs": [],
  "name": "MaxSupplyExceeded",
  "type": "error"
},
{
  "inputs": [],
  "name": "MintingNotAllowed",
  "type": "error"
},
{
  "inputs": [],
  "name": "NoMoreMintingPeriods",
  "type": "error"
},
{
  "inputs": [],
  "name": "NoPendingUpdate",
  "type": "error"
},
{
  "inputs": [],
  "name": "SameEcoFriendlyPoolAmount",
  "type": "error"
},
{
  "inputs": [],
  "name": "SamePlatformPublicKey",
  "type": "error"
},
{
  "inputs": [],
  "name": "SameRewardPoolAmount",
  "type": "error"
},
{
  "inputs": [],
  "name": "TimelockNotFinished",
  "type": "error"
},
{
  "inputs": [],
  "name": "TransferFailed",
```

```
  "type": "error"
 },
 {
  "anonymous": false,
  "inputs": [
   {
    "indexed": true,
    "internalType": "address",
    "name": "owner",
    "type": "address"
   },
   {
    "indexed": true,
    "internalType": "address",
    "name": "spender",
    "type": "address"
   },
   {
    "indexed": false,
    "internalType": "uint256",
    "name": "value",
    "type": "uint256"
   }
  ],
  "name": "Approval",
  "type": "event"
 },
 {
  "anonymous": false,
  "inputs": [
   {
    "indexed": true,
    "internalType": "address",
    "name": "owner",
    "type": "address"
   },
   {
    "indexed": false,
    "internalType": "uint256",
    "name": "newAmount",
    "type": "uint256"
   }
  ],
  "name": "EcoFriendlyPoolUpdated",
```

```
  "type": "event"
 },
 {
  "anonymous": false,
  "inputs": [
   {
    "indexed": true,
    "internalType": "address",
    "name": "owner",
    "type": "address"
   },
   {
    "indexed": false,
    "internalType": "uint256",
    "name": "amount",
    "type": "uint256"
   },
   {
    "indexed": false,
    "internalType": "bytes32",
    "name": "votingId",
    "type": "bytes32"
   },
   {
    "indexed": false,
    "internalType": "uint256",
    "name": "currentNonce",
    "type": "uint256"
   }
  ],
  "name": "MintWithVotingApproval",
  "type": "event"
 },
 {
  "anonymous": false,
  "inputs": [
   {
    "indexed": true,
    "internalType": "address",
    "name": "previousOwner",
    "type": "address"
   },
   {
    "indexed": true,
```

```
      "internalType": "address",
      "name": "newOwner",
      "type": "address"
    }
  ],
  "name": "OwnershipTransferStarted",
  "type": "event"
},
{
  "anonymous": false,
  "inputs": [
    {
      "indexed": true,
      "internalType": "address",
      "name": "previousOwner",
      "type": "address"
    },
    {
      "indexed": true,
      "internalType": "address",
      "name": "newOwner",
      "type": "address"
    }
  ],
  "name": "OwnershipTransferred",
  "type": "event"
},
{
  "anonymous": false,
  "inputs": [
    {
      "indexed": true,
      "internalType": "address",
      "name": "owner",
      "type": "address"
    }
  ],
  "name": "PlatformPublicKeyUpdateCancelled",
  "type": "event"
},
{
  "anonymous": false,
  "inputs": [
    {
```

```
      "indexed": true,
      "internalType": "address",
      "name": "oldPublicKey",
      "type": "address"
    },
    {
      "indexed": true,
      "internalType": "address",
      "name": "newPublicKey",
      "type": "address"
    }
  ],
  "name": "PlatformPublicKeyUpdateInitiated",
  "type": "event"
},
{
  "anonymous": false,
  "inputs": [
    {
      "indexed": true,
      "internalType": "address",
      "name": "oldPublicKey",
      "type": "address"
    },
    {
      "indexed": true,
      "internalType": "address",
      "name": "newPublicKey",
      "type": "address"
    }
  ],
  "name": "PlatformPublicKeyUpdated",
  "type": "event"
},
{
  "anonymous": false,
  "inputs": [
    {
      "indexed": true,
      "internalType": "address",
      "name": "owner",
      "type": "address"
    },
    {
```

```
      "indexed": false,
      "internalType": "uint256",
      "name": "newAmount",
      "type": "uint256"
    }
  ],
  "name": "RewardsPoolUpdated",
  "type": "event"
},
{
  "anonymous": false,
  "inputs": [
    {
      "indexed": true,
      "internalType": "address",
      "name": "burner",
      "type": "address"
    },
    {
      "indexed": false,
      "internalType": "uint256",
      "name": "amount",
      "type": "uint256"
    }
  ],
  "name": "TokensBurned",
  "type": "event"
},
{
  "anonymous": false,
  "inputs": [
    {
      "indexed": true,
      "internalType": "address",
      "name": "from",
      "type": "address"
    },
    {
      "indexed": true,
      "internalType": "address",
      "name": "to",
      "type": "address"
    },
    {
```

```
      "indexed": false,
      "internalType": "uint256",
      "name": "value",
      "type": "uint256"
    }
  ],
  "name": "Transfer",
  "type": "event"
},
{
  "stateMutability": "payable",
  "type": "fallback"
},
{
  "inputs": [],
  "name": "acceptOwnership",
  "outputs": [],
  "stateMutability": "nonpayable",
  "type": "function"
},
{
  "inputs": [
    {
      "internalType": "address",
      "name": "owner",
      "type": "address"
    },
    {
      "internalType": "address",
      "name": "spender",
      "type": "address"
    }
  ],
  "name": "allowance",
  "outputs": [
    {
      "internalType": "uint256",
      "name": "",
      "type": "uint256"
    }
  ],
  "stateMutability": "view",
  "type": "function"
},
```

```
{
  "inputs": [
    {
      "internalType": "address",
      "name": "spender",
      "type": "address"
    },
    {
      "internalType": "uint256",
      "name": "amount",
      "type": "uint256"
    }
  ],
  "name": "approve",
  "outputs": [
    {
      "internalType": "bool",
      "name": "",
      "type": "bool"
    }
  ],
  "stateMutability": "nonpayable",
  "type": "function"
},
{
  "inputs": [
    {
      "internalType": "address",
      "name": "account",
      "type": "address"
    }
  ],
  "name": "balanceOf",
  "outputs": [
    {
      "internalType": "uint256",
      "name": "",
      "type": "uint256"
    }
  ],
  "stateMutability": "view",
  "type": "function"
},
{
```

```
  "inputs": [
   {
    "internalType": "uint256",
    "name": "amount",
    "type": "uint256"
   }
  ],
  "name": "burn",
  "outputs": [],
  "stateMutability": "nonpayable",
  "type": "function"
},
{
  "inputs": [],
  "name": "cancelPlatformPublicKeyUpdate",
  "outputs": [],
  "stateMutability": "nonpayable",
  "type": "function"
},
{
  "inputs": [],
  "name": "decimals",
  "outputs": [
   {
    "internalType": "uint8",
    "name": "",
    "type": "uint8"
   }
  ],
  "stateMutability": "pure",
  "type": "function"
},
{
  "inputs": [
   {
    "internalType": "address",
    "name": "spender",
    "type": "address"
   },
   {
    "internalType": "uint256",
    "name": "subtractedValue",
    "type": "uint256"
   }
```

```
  ],
  "name": "decreaseAllowance",
  "outputs": [
   {
    "internalType": "bool",
    "name": "",
    "type": "bool"
   }
  ],
  "stateMutability": "nonpayable",
  "type": "function"
 },
 {
  "inputs": [],
  "name": "deploymentTime",
  "outputs": [
   {
    "internalType": "uint256",
    "name": "",
    "type": "uint256"
   }
  ],
  "stateMutability": "view",
  "type": "function"
 },
 {
  "inputs": [],
  "name": "ecoFriendlyPoolBalance",
  "outputs": [
   {
    "internalType": "uint256",
    "name": "",
    "type": "uint256"
   }
  ],
  "stateMutability": "view",
  "type": "function"
 },
 {
  "inputs": [],
  "name": "finalizePlatformPublicKeyUpdate",
  "outputs": [],
  "stateMutability": "nonpayable",
  "type": "function"
```

```
},
{
  "inputs": [
    {
      "internalType": "address",
      "name": "spender",
      "type": "address"
    },
    {
      "internalType": "uint256",
      "name": "addedValue",
      "type": "uint256"
    }
  ],
  "name": "increaseAllowance",
  "outputs": [
    {
      "internalType": "bool",
      "name": "",
      "type": "bool"
    }
  ],
  "stateMutability": "nonpayable",
  "type": "function"
},
{
  "inputs": [
    {
      "internalType": "address",
      "name": "newPlatformPublicKey",
      "type": "address"
    }
  ],
  "name": "initiatePlatformPublicKeyUpdate",
  "outputs": [],
  "stateMutability": "nonpayable",
  "type": "function"
},
{
  "inputs": [
    {
      "internalType": "uint256",
      "name": "amount",
      "type": "uint256"
```

```
    },
    {
      "internalType": "bytes32",
      "name": "votingId",
      "type": "bytes32"
    },
    {
      "internalType": "uint256",
      "name": "currentNonce",
      "type": "uint256"
    },
    {
      "internalType": "bytes",
      "name": "signature",
      "type": "bytes"
    }
  ],
  "name": "mint",
  "outputs": [],
  "stateMutability": "nonpayable",
  "type": "function"
},
{
  "inputs": [],
  "name": "name",
  "outputs": [
    {
      "internalType": "string",
      "name": "",
      "type": "string"
    }
  ],
  "stateMutability": "view",
  "type": "function"
},
{
  "inputs": [
    {
      "internalType": "address",
      "name": "user",
      "type": "address"
    }
  ],
  "name": "nonces",
```

```
  "outputs": [
   {
    "internalType": "uint256",
    "name": "nonce",
    "type": "uint256"
   }
  ],
  "stateMutability": "view",
  "type": "function"
 },
 {
  "inputs": [],
  "name": "owner",
  "outputs": [
   {
    "internalType": "address",
    "name": "",
    "type": "address"
   }
  ],
  "stateMutability": "view",
  "type": "function"
 },
 {
  "inputs": [],
  "name": "pendingOwner",
  "outputs": [
   {
    "internalType": "address",
    "name": "",
    "type": "address"
   }
  ],
  "stateMutability": "view",
  "type": "function"
 },
 {
  "inputs": [],
  "name": "pendingPlatformPublicKey",
  "outputs": [
   {
    "internalType": "address",
    "name": "",
    "type": "address"
```

```
    }
   ],
   "stateMutability": "view",
   "type": "function"
  },
  {
   "inputs": [],
   "name": "platformPublicKey",
   "outputs": [
    {
     "internalType": "address",
     "name": "",
     "type": "address"
    }
   ],
   "stateMutability": "view",
   "type": "function"
  },
  {
   "inputs": [],
   "name": "renounceOwnership",
   "outputs": [],
   "stateMutability": "nonpayable",
   "type": "function"
  },
  {
   "inputs": [],
   "name": "rewardPoolBalance",
   "outputs": [
    {
     "internalType": "uint256",
     "name": "",
     "type": "uint256"
    }
   ],
   "stateMutability": "view",
   "type": "function"
  },
  {
   "inputs": [],
   "name": "symbol",
   "outputs": [
    {
     "internalType": "string",
```

```
      "name": "",
      "type": "string"
    }
  ],
  "stateMutability": "view",
  "type": "function"
},
{
  "inputs": [],
  "name": "timelockEndTime",
  "outputs": [
    {
      "internalType": "uint256",
      "name": "",
      "type": "uint256"
    }
  ],
  "stateMutability": "view",
  "type": "function"
},
{
  "inputs": [],
  "name": "totalSupply",
  "outputs": [
    {
      "internalType": "uint256",
      "name": "",
      "type": "uint256"
    }
  ],
  "stateMutability": "view",
  "type": "function"
},
{
  "inputs": [
    {
      "internalType": "address",
      "name": "to",
      "type": "address"
    },
    {
      "internalType": "uint256",
      "name": "value",
      "type": "uint256"
```

```
    }
   ],
   "name": "transfer",
   "outputs": [
    {
     "internalType": "bool",
     "name": "",
     "type": "bool"
    }
   ],
   "stateMutability": "nonpayable",
   "type": "function"
  },
  {
   "inputs": [
    {
     "internalType": "address",
     "name": "from",
     "type": "address"
    },
    {
     "internalType": "address",
     "name": "to",
     "type": "address"
    },
    {
     "internalType": "uint256",
     "name": "value",
     "type": "uint256"
    }
   ],
   "name": "transferFrom",
   "outputs": [
    {
     "internalType": "bool",
     "name": "",
     "type": "bool"
    }
   ],
   "stateMutability": "nonpayable",
   "type": "function"
  },
  {
   "inputs": [
```

```
    {
      "internalType": "address",
      "name": "newOwner",
      "type": "address"
    }
  ],
  "name": "transferOwnership",
  "outputs": [],
  "stateMutability": "nonpayable",
  "type": "function"
},
{
  "inputs": [
    {
      "internalType": "uint256",
      "name": "newAmount",
      "type": "uint256"
    },
    {
      "internalType": "bytes32",
      "name": "id",
      "type": "bytes32"
    },
    {
      "internalType": "uint256",
      "name": "currentNonce",
      "type": "uint256"
    },
    {
      "internalType": "bytes",
      "name": "signature",
      "type": "bytes"
    }
  ],
  "name": "updateEcoFriendlyPool",
  "outputs": [],
  "stateMutability": "nonpayable",
  "type": "function"
},
{
  "inputs": [
    {
      "internalType": "uint256",
      "name": "newAmount",
```

```
      "type": "uint256"
     },
     {
      "internalType": "bytes32",
      "name": "id",
      "type": "bytes32"
     },
     {
      "internalType": "uint256",
      "name": "currentNonce",
      "type": "uint256"
     },
     {
      "internalType": "bytes",
      "name": "signature",
      "type": "bytes"
     }
    ],
    "name": "updateRewardPool",
    "outputs": [],
    "stateMutability": "nonpayable",
    "type": "function"
   },
   {
    "stateMutability": "payable",
    "type": "receive"
   }
]
```

9. Bytecode

60a060405234801561000f575f80fd5b50604051611ff5380380611ff583398101604081905261002e916102a3565b60405180
60400160405280600c81526020016b22b1b7a9b0bb32aa37b5b2b760a11b8152506040518060400160405280600781526020
20016645434f5341564560c81b81525081600390816100829190610367565b50600461008f8282610367565b5050506100a8
6100a361016c60201b60201c565b610170565b600160075560016001600160a01b0381166100d457604051635d3d682f60e01b8
152600401604051809103 90fd5b60088054600160016001600160a01b0319166001600160a01b038316179055610124336a7c13bc4b
2c133c560000006101158 16b015f8d402a5236804900000061043556 5b61011f9190610435565b61018c565b610139336a7c
13bc4b2c133c5600000061018c565b61014e336a7c13bc4b2c133c5600000061018c565b506a7c13bc4b2c133c5600000060
09819055600a554260805261046 1565b3390565b60068054600160016001600160a01b0319169055610189816102 4d565b50565b60
01600160a01b0382166101e65760405162461bcd60e51b815260206004820152601f60248201527f45524332303a206d696e
7420746f20746865207a65726f206164647265737300604482015260640160405180910390fd5b8060025f8282546101f7919
061044e565b90915550506001600160a01b0382165f8181526020818152604080832080548601905551848 1527fddf252ad1
be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef9101604051809103 90a35050565b600580546001600160
a01b038381166001600160a01b0319831681179093556040519116919082907f8be0079c531659141344cd1fd0a4f2841949
7f9722a3daafe3b4186f6b6457e0905f90a35050565b505050565b5f602082840312156102b3575f80fd5b81516001600160a
01b03811681146102c9575f80fd5b9392505050565b634e487b7160e01b5f52604160045260245ffd5b600181811c90821680
6102f857607f821691505b602082108103610316 31657634e487b7160e01b5f52602260045260245ffd5b50919050565b601f821
11561029e57805f5260205f20601f840160051c810160208510156103415750805b601f840160051c820191505b8181101561
0360575f815560010161034d565b5050505050565b81516001600160401b038111156103805761038 06102d0565b6103948
161038e84546102e4565b8461031c565b6020601f8211600181146103c6575f83156103af5750848201515b5f19600385901b
1c1916600184901b178455610360565b5f848152602081206 01f198516915b828110156103f55787850151825560209485 01
946001909201910161 03d5565b5084821015610412578684 01515f19600387901b60f8161c191681555b505050506001908
11b01905550565b634e487b7160e01b5f52601160045260245ffd5b8181038181111561044857610448610421565b9291505
0565b8082018082111561044857610448610421565b608051611b756104805f395f8181610574015261 0be60152611b755ff
3fe608060405260043610610 1d0575f3560e01c80637a5c08ae116100f6578063a9059cbb11610094578063e30c3978116100
63578063e30c397814610527578063e41813c014610544578063ecda10f514610563578063f2fde38b14610596576101ee56
5b8063a9059cbb146104b5578063b313f4f8146104d4578063ba78a98a146104f3578063dd62ed3e14610508576101ee565b
806387ff8ff5116100d057806387ff8ff514610415780638da5cb5b1461046557806395d89b4114610482578063a457c2d714
610496576101ee565b80637a5c08ae146103f25780637d51e4851461040757580 7ecebe0014610426576101ee565b806339
5093511161016e57806342f7e8c51161013d57806342f7e8c51461035f57806370a0823114610396578063715018a6146103c
a57806379ba5097146103de576101ee565b8063395093511461 02ee57806339cfc26c1461030d5780634040f1451461032c5
7806342966c6814610340576101ee565b806323b872dd116101aa57806323b872dd1461027e5780632433daf41461029d57
8063313ce567146102b257806334c30712146102cd576101ee565b806306fdde0314610207578063095ea7b314610231578
06318160ddd14610260576101ee565b366101ee57604051630dccd20d60e21b8152600401604051809103 90fd5b60405163
0dccd20d60e21b8152600401604051809103 90fd5b348015610212575f80fd5b5061021b6105b5565b60405161022891906
11886565b60405180910390f35b34801561023c575f80fd5b5061025061024b3660046118d6565b610645565b60405190151
581526020016102285 65b34801561026b575f80fd5b506002545b604051908152602001610228565b348015610289575f80f
d5b506102506102983660046118fe565b61065e565b3480156102a8575f80fd5b50610270600a5481565b3480156102bd57
5f80fd5b506040516012815260200161022856 5b3480156102d8575f80fd5b506102ec6102e7366004611938565b61069b56
5b005b3480156102f9575f80fd5b506102506103083660046118d6565b61075f565b348015610318575f80fd5b506102ec610
32736600461196c565b610780565b348015610337575f80fd5b506102ec610809565b34801561034b575f80fd5b506102ec6
1035a366004611a3b565b6108c0565b34801561036a575f80fd5b5060085461037e906001600160a01b031681565b604051
6001600160a01b0390911681526020016102285 65b3480156103a1575f80fd5b506102706103b0366004611938565b60016
00160a01b03165f9081526020819052604090205490565b3480156103d5575f80fd5b506102ec610902565b3480156103e95
75f80fd5b506102ec610915565b3480156103fd575f80fd5b5061027060095481565b348015610412575f80fd5b506102ec61
042136600461196c565b610994565b348015610431575f80fd5b5061027061044036600461193856 5b600d6020525f908152
604090205481565b34801561045c575f80fd5b506102ec610a0a565b348015610470575f80fd5b5060055460016001600160a01b
031661037e565b34801561048d575f80fd5b5061021b610a7b565b3480156104a1575f80fd5b506102506104b03660046118
d6565b610a8a565b3480156104c0575f80fd5b506102506104cf3660046118d6565b610b0f565b3480156104df575f80fd5b5

0600b5461037e906001600160a01b031681565b3480156104fe575f80fd5b50610270600c5481565b348015610513575f80fd
5b506102706105223660046l1a52565b610b4a565b348015610532575f80fd5b506006546001600160a01b031661037e565
b34801561054f575f80fd5b506102ec61055e36600461196c565b610b74565b34801561056e575f80fd5b506102707f000000
0000000000000000000000000000000000000000000000000000000000081565b3480156105a1575f80fd5b506102ec6105b0
366004611938565b610e6e565b6060600380546105c490611a83565b80601f0160208091040260200160405190810160405
280929190818152602001828054610 5f090611a83565b801561063b5780601f1061061257610100808354040283529160200
19161063b565b820191905f5260205f20905b815481529060010190602001808311610 61e57829003601f168201915b50505
05050905090565b5f33610652818585610edf565b60019150505b92915050565b5f826001600160a01b03811661068757604
051634e46966960e11b815260040160405180910390fd5b6106928585856110 03565b95945050505050565b6106a361101b
565b6001600160a01b0381166106ca57604051635d3d682f60e01b815260040160405180910390fd5b600b5460016001600160a
01b038083169116036106f857604051632b9fa3f960e21b815260040160405180910390fd5b600b80546001600160a01b031
9166001600160a01b038316179055610720620 3f48042611acf565b600c556008546040516001600160a01b0380841692169
07fb3f5b421963d2604e55a02812a08e54093392ea1416a0aaf4f147bcb882edc79905f90a350565b5f3361065281858561077
18383610b4a565b61077b9190611acf565b610edf565b610788611075565b61079061101b565b61079d33858585856110ce
565b83600a54036107be5760405162dba0db60e21b815260040160405180910390fd5b600a849055604051848152339 07f9
6a71e6abc43b11734bebed318ccc22bd4b244aa1062e845ae3b8f8019482c92906020015b60405180910390a261080360016
00755565b50505050565b61081161101b565b600c5442101561083457604051635192dd5560e01b8152600401604051809
10390fd5b600b546001600160a01b031661085d576040516319b2c6ef60e31b815260040160405180910390fd5b600880546
00b80546001600160a01b0319808416600160016 0a01b038381169182179096559116909155 5f600c819055604051939092
1692909183917f753797208a3bb557f39345fc0474e4f697a5e8ed80379c9462d42cbc4f55e4e19190a350565b6108ca33826
111ac565b60405181815233907ffd38818f5291bf0bb3a2a48aadc06ba8757865d1dabd804585338aab3009dcb6906020016
0405180910390a250565b61090a61101b565b6109135f6112d4565b565b60065433906001600160a01b0316811461098857
60405162461bcd60e51b815260206004820152602960248201527f4f776e61626c6532537465703a2063616c6c6572206973
206e6f742074686520604820152683732bb9037bbb732b960b91b60648201526084015b60405180910390fd5b610991816
112d4565b50565b61099c611075565b6109a461101b565b6109b133858585856110ce565b83600954036109d3576040516
370e96a1d60e11b815260040160405180910390fd5b60098490556040518481523390 7f66f2b450621a7e7985f865e4ce94f9
a4b74e346b2609e45701e1f2ce002b1cfc906020016107f1565b610a1261101b565b600b546001600160a01b0316610a3b57
6040516319b2c6ef60e31b815260040160405180910390fd5b600b80546001600160a01b03191690555f600c819055604051
33917f393179a8ff62549879cc0bc81c4c23c21485eb3cabdf6f4272cef8d249e279f791a2565b6060600480546105c490611a8
3565b5f3381610a978286610b4a565b905083811015610af75760405162461bcd60e51b815260206004820152602560248
01527f45524332303a20646563726561736564206l6c6c6f77616e63652062656c6f77604482015264207a65726f60d81b606
48201526084016l097f565b610b048286868403610edf565b5060019493505050565b5f826001600160a01b038116610b3
857604051634e46966960e11b815260040160405180910390fd5b610b4284846112ed565b949350505050565b6001600160
a01b039182165f908152600160209081526040808320939094168252919091522054905 65b610b7c611075565b610b84611
01b565b610b9133858585856110ce565b335f908152600d602052604090205482146 10bbf57604051633ab3447f60e11b815
260040160405180910390fd5b610bca826001611acf565b335f908152600d602052604081209190915563 01e13380610c0b7f
0000000000000000000000000000000000000000000000000000000000000042611ae2565b610c159190611af5565b905
05f81600103610c3257506a084595161401484a000000610d35565b81600203610c4c57506a0c685fa11e01ec6f000000610d
35565b81600303610c6657506a108b2a2c28029094000000610d35565b81600403610c8057506a18d0bf423c03d8de00000
0610d35565b81600503610c9a57506a25391ee35a05c54d000000610d35565b81600603610cb457506a31a17e847807b1bc
000000610d35565b81600703610cce57506a3e09de2596099e2b000000610d35565b81600803610ce857506a4e950851be0
c2ebf000000610d35565b81600903610d0257506a5afd67f2dc0e1b2e000000610d35565b81600a03610d1c57506a5f20327
de60ebf53000000610d35565b604051633d55769d60e11b815260040160405180910390fd5b858114610d555760405163a7
dcb93b60e01b815260040160405180910390fd5b6b033b2e3c9fd0803ce800000086610d6c60025490565b610d769190611
acf565b1115610d9557604051638a164f6360e01b815260040160405180910390fd5b5f6064610da383601e611b14565b610
dad9190611af5565b90505f6064610dbd84601e611b14565b610dc79190611af5565b905081600954610dd79190611acf565
b600955600a54610de8908290611acf565b600a55610e093382610dfa8587611ae2565b610e049190611ae2565b6112fa565
b610e1333836112fa565b610e1d33826112fa565b6040805189815260208101899052908101879052 33907f05e2aa3a37e7d
cf3a610602926e17ccf96f61b2b2eb3d4c240bfb881eb4045439060600160405180910390a2505050506108036001600755565
5b610e7661101b565b60068054600160016 0a01b038316600160016 0a01b0319909116817909155610ea76005546001600
160a01b031690565b6001600160a01b03167f38d16b8cac22d99fc7c124b9cd0de2d3fa1faef420bfe791d8c362d765e22700

6040516040518091039a350565b6001600160a01b038316610f415760405162461bcd60e51b81526020600482015260248
08201527f45524332303a20617070726f76652066726f6d20746865207a65726f206164646044820152637265737360e01b6
0648201526084016109 7f565b6001600160a01b038216610fa2576040516246 1bcd60e51b815260206004820152602260 24
8201527f45524332303a20617070726f766520746f20746865207a65726f206164647265604482015261737360f01b6064820
15260840161097f565b6001600160a01b038381165f818152600160209081526040808320948716808452948252918290208
5905590518481527f8c5be1e5ebec7d5bd14f71427d1e84f3dd0314c0f7b2291e5b200ac8c7c3b92591015b60405180910390
a3505050565b5f336110108582856113b7565b610b0485858561142956 5b6005546001600160a01b0316331461091357604
05162461bcd60e51b8152602060048201819052602482015 27f4f776e61626c653a2063616c6c6572206973206e6f74207468
65206f776e65726044820152606401610 97f565b6002600754036110c7576040516246 1bcd60e51b8152602060048201526 01
f60248201527f5265656e7472616e637947756172643a207265656e7472616e742063616c6c006044820152606401610 97f5
65b6002600755565b6040516bfffffffffffffffffffffffff19606087901b166020820152603481018590526054810184905260748101
8390525f906094016040516020818303038152906040528051906020012090505f611153827f19457468657265756d205369
676e6564204d6573736167653a0a3332000000005f908152601c91909152603c902090565b60085490915060016001600160a01
b03165f61116d83866115cb565b9050816001600160a01b0316816001600160a01b0316146111a157604051638baa579f60
e01b81526004016040518091039 0fd5b505050505050505050565b6001600160a01b03821661120c5760405162461bcd60e
51b8152602060048201526021602 4820152 7f45524332303a206275726e2066726f6d20746865207a65726f2061646472657
360448201526073 60f81b6064820152608401610 97f565b6001600160a01b0382165f908152602081905260409020548181 1
01561127f5760405162461bcd60e51b8152602060048201526022602 4820152 7f45524332303a206275726e20616d6f756e7
420657863656564732062616c616e6044820152616360 f01b6064820152608401610 97f565b6001600160a01b0383165f8
181526020818152604080832086860390556002805487900390555185815291929 17fddf252ad1be2c89b69c2b068fc378da
a952ba7f163c4a11628f55a4df523b3ef9101610ff6565b60068054600160016 00a01b0319169055610991816115ed565b5f336
106528185856114295 65b6001600160a01b0382166113505760405162461bcd60e51b8152602060048201526 01f60248201
527f45524332303a206d696e7420746f20746865207a65726f206164647265737300604482015260640161097f565b806002
5f828254613619190611acf565b909155505060016001600160a01b0382165f818152602081815260408083208054860190555 1
8481527fddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef9101604051809103 90a35050565b5f
6113c28484610b4a565b90505f198114610803578181101561141c5760405162461bcd60e51b8152602060048201526 01d6
0248201527f45524332303a20696e7375666669636965 6e7420616c6c6f77616e6365000006044820152606401610 97f565
b6108038484848403610edf565b6001600160a01b03831661148d5760405162461bcd60e51b8152602060048201526025 60
248201527f45524332303a207472616e736665722066726f6d20746865207a65726f206164604482015264647265737360d8
1b6064820152608401610 97f565b6001600160a01b0382166114ef5760405162461bcd60e51b8152602060048201526023 60
0248201527f45524332303a207472616e7366657220746f20746865207a65726f206164647260448201526265737360e81b6
0648201526084016109 7f565b6001600160a01b0383165f908152602081905260409020548181101561156657604051624 61
bcd60e51b8152602060048201526026602 4820152 7f45524332303a207472616e7366657220616d6f756e742065786365656
473206260448201526 5616c616e636560d01b6064820152608401610 97f565b6001600160a01b038481165f8181526020818
1526040808320878703905593871680835291849020805487019055925185815290927fddf252ad1be2c89b69c2b068fc378
daa952ba7f163c4a11628f55a4df523b3ef9101604051809103 90a3610803565b5f805f6115d8858561163e565b9150915061
15e581611680565b509392505050565b6005805460016001600160a01b0383811660016001600160a01b0319831681179093556040 5
19116919082907f8be0079c531659141344cd1fd0a4f28419497f9722a3daafe3b4186f6b6457e0905f90a35050565b5f80825
160410361167257602083015160408401516060850 1515f1a61166687828585611 7c9565b9450945050505050611679565b50
5f905060025b9250929050565b5f816004811115611693576 11693611b2b565b0361169b5750565b6001816004811115611 6
6af576116af611b2b565b036116fc5760405162461bcd60e51b8152602060048201526018602482015 27f45434453413a206
96e76616c6964207369676e6174757265000000000000000060448201526064016109 7f565b6002816004811115611711057
611710611b2b565b0361175d5760405162461bcd60e51b8152602060048201526 01f60248201527f45434453413a20696e7
6616c6964207369676e6174757265206c656e6774680060448201526064016109 7f565b6003816004811115611771771
1611b2b565b0361099157604051624 61bcd60e51b8152602060048201526022602 4820152 7f45434453413a20696e76616c
6964207369676e6174757265202773272076616c6044820152615 61756560f01b6064820152608401610 97f565b5f807f7fffffffff
ffffffffffffffffffffffff5d576e7357a4501ddfe92f46681b20a08311156117 fe57505f9050600361187d565b604080515f8082526020
8201808452899052 60ff8816928201929092526060810186905260808101859052600 19060a0016020604051602081039080
840390855afa15801561184f573d5f803e3d5ffd5b5050604051601f190151915050600 1600160a01b038116611877575f6001
9250925050611 87d565b91505f90505b9450949250505056 5b602081525f8251806020840152806020850160408501 5e5f60
4082850101526040601f19601f830116840101915050929 15050565b80356001600160a01b03811681146118 d1575f80fd5b

919050565b5f8060408385031215611 8e7575f80fd5b6118f0836118bb565b9460209390930135935050505 65b5f805f6060848603121561191057 5f80fd5b611919846118bb565b9250611927602085016118bb565b92959294505050604091909101359056 5b5f602082840312156119485 75f80fd5b611951826118bb565b9392505050565b634e487b7160e01b5f52604160045260245ffd5b5f805f80608085870312156119 7f575f80fd5b84359350602085013592506040850135915060608501356 7fffffffffffffff8111156119aa575f80fd5b8501601f810187136119ba575f80fd5b803567ffffffffffffffff8111156119d4576119d461195856 5b604051601f8201601f19908116603f0116810167ffffffffffffffff8111828210171561 1a0357611a03611958565b6040528181528282016020018910 15611a1a575f80fd5b8160208401602083013 75f6020838301015280935050505092959194509250565b5f60208284031215611a4b575f80fd5b5035919050565b5f806040838503121561 1a63575f80fd5b611a6c836118bb565b9150611a7a602084016118bb565b9050925092905056 5b600181811c90821680611a9757607f821691505b602082108103611ab557634e487b7160e01b5f52602260045260245ffd5b50919050565b634e487b7160e01b5f52601160045260245ffd5b8082018082111561065857610658611abb565b8181038181111561065857610658611abb565b5f82611b0f57634e487b7160e01b5f52601260045260245ffd5b500490565b8082028115828204841417610658576106 58611abb565b634e487b7160e01b5f5260216004526 0245ffdfea2646970667358221220 9b72cd6fdeb205f14ffe90e87fbd00a329843ffe7d43ec15a178af0225e2af1364736f6c634300081a0033