Miras: Trustless Inheritance Protocol

Abstract

Miras is a decentralized inheritance protocol that enables verifiable, trustless, and automated asset transfer upon the verified death or incapacity of an owner. Built on Ethereum and compatible EVM chains, it leverages a multi-layered attestation system, encrypted safe architecture, and UUPS upgradeable smart contracts to ensure continuity of digital and financial assets without centralized custodians, legal executors, or key-sharing.

Miras introduces **Claim**, **Attester**, **Safe** modules that coordinate secure inheritance workflows. It also supports the **MRS** governance token and its yield-bearing counterpart, **wMRS**, which incentivize attesters, support verifiers, and fund governance.

---

1. Introduction: The Inheritance Problem in Web3

Billions of dollars in crypto assets remain permanently locked after wallet owners pass away or lose access to private keys. While blockchains guarantee immutability, they offer no native mechanism for inheritance. Current approaches rely on trusted executors, custodians, or premature key sharing — all of which reintroduce centralization and risk.

Miras eliminates these vulnerabilities by automating inheritance via verifiable, decentralized attestation. Its mission: make **“digital legacy management fully autonomous.”**

Core Principles

- **Trustless:** No single party holds control or custody.
- **Verifiable:** Death and legitimacy verified via on-chain attesters.
- **Private:** Encrypted safes protect sensitive heir and asset data.
- **Upgradeable:** Built with UUPS proxies for forward compatibility.
- **Interoperable:** Cross-chain safe management with potential Cosmos and Solana integration.

---

2. System Overview

At the core of Miras lies a modular, verifiable architecture comprising:

Module	Function	Description
Safe	Asset container	Stores digital assets and defines

Module	Function	Description
		inheritance parameters.
Claim	Heir request	Allows designated heirs to claim safes after trigger events.
Attester	Verifier registry	Validates deaths/incapacity through multi-signature or oracle input.
Token	Incentive layer	Provides MIRAS/wMIRAS token-based rewards and governance staking.
Proxy	Upgradeability	Manages contract logic changes with admin and time-lock guardrails.

Each layer interacts through tightly scoped interfaces defined in Solidity contracts (`v4.sol`, `v5.sol`, and `claims/proxy.sol`).

Each of these layers communicates through event-based architecture — `AttesterUpsert`, `ClaimSubmitted`, and `SafeReleased` — ensuring transparency, modularity, and verifiability.

3. Protocol Architecture

3.1 Core Contracts

Safe Contracts

Safes are upgradeable, non-custodial containers that store encrypted metadata representing assets (on-chain or off-chain). Each Safe includes: - An **encrypted safe ID** (AES or phone-blob encrypted) - **Heir designation** through verified address or secret hash - **Unlock conditions** governed by verified attestation events

Claim Contracts

Claims act as structured requests initiated by heirs: - Each claim contains an `encryptedSafe`, `claimer`, and `attestor` reference. - Validation occurs through `getClaimBySafeId` and `getClaimsByAttester` view functions. - Claims are time-locked and revert automatically if invalid or expired.

Attester Contracts

The **Attester Registry** is the decentralized verification hub. It ensures: - Each attester has a unique Ethereum address, metadata (email, website, phone, timestamp). - Upsertable registry (insert/update) functionality with event logging (`AttesterUpsert`). - Optional scoring mechanisms (ActivityRank) for performance-based weighting.

Proxy & UUPS Upgradeability

Miras follows the **UUPS (Universal Upgradeable Proxy Standard)** with: - `proxy.sol` implementing logic delegation. - Admin-controlled upgrades via `upgradeTo()`. - `whenInitialized` modifiers to secure initialization sequence.

---

4. Technical Workflow

1. **Safe Creation:**
    - User deploys a new Safe using the Miras factory or UI.
    - Metadata and heir data are encrypted and stored on-chain or via IPFS.
2. **Attester Registration:**
    - Trusted entities (lawyers, notaries, DAOs, oracles) register on-chain.
    - Each attester maintains metadata and staking requirements.
3. **Trigger Event (Death/Verification):**
    - Attesters verify off-chain proof of death/incapacity.
    - On-chain attestations are submitted and aggregated.
4. **Claim Initiation:**
    - Heir submits a claim referencing the encrypted safe.
    - Smart contract validates signatures, timestamps, and threshold logic.
5. **Release Mechanism:**
    - If threshold of attesters confirm, the Safe releases assets.
    - Assets may include native tokens, ERC-20, ERC-721, or linked off-chain claims.

---

5. Tokenomics

5.1 MRS Token

MRS is the governance and utility token of the protocol.

Property	Description
Type	ERC-20 compatible
Supply	100,000,000 (capped)
Utility	Governance, attester staking, fee payments
Emission	Fixed genesis mint + governance-controlled treasury

MRS is used to: - Pay attester registration and claim fees. - Participate in governance decisions (through delegation or DAO proposals). - Fund protocol treasury and validator pools.

5.2 Wrapped MRS (wMIRAS)

`wMRS` is a wrapped, yield-bearing version of MRS. Users may deposit MRS into the protocol to mint wMRS, which: - Earns yield from attester fees and claim unlock fees. - Can be staked in verifier pools or yield contracts. - Does **not** carry governance rights, preserving control with the DAO core.

5.3 Incentive Flow

1. **Claim fees** → distributed to attesters (proportional to participation weight).
2. **Attester staking rewards** → yield in wMRS.
3. **DAO Treasury** → receives residual protocol fees to fund audits, upgrades, and grants.

---

6. Governance Model

6.1 DAO Structure

MirasDAO is structured as a **two-tier hybrid DAO**:

- **Core Council (Multisig):** 5-of-9 model managing upgrades, treasury, and emergencies.

- **Community Layer:** MRS holders can propose and vote on non-critical changes.

6.2 Proposal Flow

1. Proposal submission (text + code hash)
2. Quorum check (≥5%)
3. Snapshot vote using ERC20Votes
4. Time-lock (72h minimum)
5. Execution via proxy delegatecall

6.3 Treasury

- Revenue streams: claim fees, staking yield, attester slashes.
- 50% of income → wMRS yield pool.
- 30% → Treasury.
- 20% → Attester rewards.

---

7. Security & Privacy Model

- **Attestation Thresholds:** No single attester can release assets.
- **Multisig Governance:** DAO-controlled upgrades.
- **Encrypted Metadata:** IPFS or on-chain encrypted fields.
- **Replay Protection:** Each claim tied to a nonce and block hash.

- **Circuit Breakers:** Emergency pause mechanism to halt release during attacks.

8. Roadmap

Phase	Milestone	Description
Q4 2025	Alpha Launch	Ethereum + Arbitrum testnets, claim + attester modules live.
Q1 2026	Attester Pool	Staking and slashing live with verifiable registry.
Q2 2026	wMIRAS Release	Launch wrapped yield-bearing token + fee redistribution.
Q3 2026	DAO Expansion	Public proposals and cross-chain bridge integration.
Q4 2026	Legacy Assets	On-chain verification for off-chain wills and property.

9. Conclusion

Miras redefines how digital assets are inherited and safeguarded in the Web3 ecosystem. By combining encrypted safes, decentralized attesters, and upgradeable contracts, it delivers a secure, private, and verifiable inheritance framework that scales across blockchain networks.

Through MRS and wMRS tokens, attesters and users are economically aligned to maintain protocol integrity, while governance remains decentralized yet accountable. Miras thus bridges the gap between human legacy and digital permanence.

Death Happens. Be Ready.

Version 1.3 — October 2025
Developed by the Miras Core Team