Heavenkey: A Protocol for Autonomous and Perpetual Digital Inheritance

October 2025

Abstract

Heavenkey is a decentralized, non-custodial protocol on Ethereum, designed to ensure continuity of digital assets and the owner's intent. The architecture follows the modular **EIP-2535 (Diamond)** standard and a **Proof-of-Activity** flow: when the owner becomes inactive, **sequential claim windows** open for heirs; if nobody claims, funds are routed to a **Legacy Fund** (ethical fallback) and the protocol **treasury**.

Privacy is native: heirs do not appear on-chain until they claim (a **commit–reveal** scheme with **domain separation** and **commitment freeze** at succession start). Per-heir metadata is encrypted with **AES-256-GCM** plus **AAD** and stored on **IPFS**; the key is delivered securely to the active heir (via **XMTP** E2EE or secure **OTP**).

In production (V1), on-chain automation is handled by **Chainlink Automation**. The mid-term vision (V2) introduces **Account Abstraction (ERC-4337)** for **gasless claims** and simplified UX.

1. Introduction: Continuity of Value and Intent

The growth of on-chain assets has introduced a systemic problem: **dormant capital** lost through inactivity or lost keys. Heavenkey exists to **reactivate** this capital **trustlessly**, preserving the owner's **intent**—to heirs, or, failing that, to a **transparent philanthropic path**. It is not just an inheritance tool; it is a **public mechanism for economic continuity**.

2. The Challenge of Digital Continuity

Existing approaches are insufficient:

- **Custodial services:** introduce counterparty risk and single points of failure.
- **Traditional legal instruments:** slow, costly, and technically misaligned with on-chain systems.
- **Manual agreements:** fragile and insecure.
- **Asset stagnation:** value exits the active economy.

3. Core Principles of the Protocol

1. **Absolute self-custody** — No third party ever accesses private keys; funds are governed by **on-chain logic**.
2. **Autonomous and perpetual operation** — A deterministic engine that lives on Ethereum, independent of centralized entities.
3. **Verifiable transparency** — Every state/transfer is attested by **public transactions**.
4. **Purposeful legacy** — If the primary path (heirs) fails, an **ethical path** re-injects capital into the economy (Legacy Fund).

4. System Architecture and Lifecycle

4.1 On-chain Core: Diamond EIP-2535

The Diamond pattern separates responsibilities into **facets** over shared **AppStorage**:

- **PlanManagementFacet** — Create/cancel plan; record **IPFS digest (bytes32)**; set **heir commitments**.
- **PlanMonitorFacet** — `ping()`, inactivity detection, **window rotation**; **freeze** commitments when succession starts.
- **InheritanceFacet** — **Commit–reveal** enforcement and payout within the active window.
- **FallbackFacet** — Route funds to **Legacy Fund** / ethical wallet when heirs fail.
- **TreasuryFacet** — Protocol fees and accounting.
- **ConfigFacet** — Global parameters (timers, fees, system wallets, fallback allowlist).
- **GasManagerFacet** — **Gas reserve** to underwrite automated operations.

Commit–reveal (domain-separated & frozen)
For each heir *i*:

```
commit[i] = keccak256(
  abi.encodePacked(
    DOMAIN,          // e.g. keccak256("HeavenkeyHeirCommit:v1")
    block.chainid,
    address(this),   // diamond
    owner,
    i,               // index
    heir,            // address
    salt             // bytes32 (claim code)
  )
);
```

- The **claim** is valid **only** during the active window and **only** from `msg.sender == heir`.
- Commitments are **frozen** when succession starts to prevent ex-post changes.

4.2 Privacy-First: Off-Chain Data, On-Chain Pointer

1. **Per-heir data collection** (JSON) in the backend.
2. **Encryption: AES-256-GCM** with **unique IVs** and **AAD** bound to context (`plan/window`).
3. **Storage:** upload to **IPFS**; on-chain we record **only** a **bytes32 digest**.
4. **Secret delivery:** the **salt** is delivered to the active heir via **XMTP** (E2EE) or **OTP** with rate-limit/expiry.

4.3 Autonomous Engine: Chainlink + Notifier

- **Chainlink Automation (on-chain):** monitors timeouts and reliably triggers transitions (`performUpkeep`/equivalents).
- **Notifier Service (off-chain):** listens to events (e.g., `HeirAdvanced`), fetches/decrypts the IPFS payload, and **notifies the heir** with the **claim code**.

4.4 Plan Lifecycle

1. **Creation:** deposit **ETH** (MVP/V1 **ETH-only**), define heirs + fallback, set **commitments**, save the **IPFS digest**.
2. **Active:** `ping()` maintains the **ACTIVE** state.
3. **Inactivity:** upon timeout, open the window for `heir0`, then `heir1`, …
4. **Claim:** the active heir reveals the **salt** and receives the payout.
5. **Fallback:** if all windows expire without claims, route funds to the **Legacy Fund** / ethical wallet.

4.5 Data: What's On-Chain vs Off-Chain

- **On-chain:** `commit[i]` (hash), IPFS **bytes32** digest, parameters, plan balance, indices/phases.
- **Off-chain:** encrypted per-heir metadata; keys/IVs; XMTP/OTP channel.

5. Security Model (Summary)

- **Replay / cross-plan / cross-chain:** mitigated via **domain separation** in the commitment.
- **Front-running of claims:** `msg.sender == heir`, secret **salt**, commitment **freeze**.
- **Reentrancy / CEI: Checks-Effects-Interactions**, pull patterns where appropriate, OpenZeppelin primitives.
- **Griefing / timing DoS:** strict windows + **gas reserve**.
- **Frontend/backend loss:** critical operations are **on-chain**; UI is replaceable; direct contract interaction is possible.

V1 hardening: Chainlink Automation, **Tier-1 audit** + **bug bounty**, secret management (vault, rotation), automated analysis (Slither/Echidna), **signed UI/API payloads**.

6. Economic Model

- **One-time fee** on deposits/top-ups (e.g., **2.5%**) + **annual subscription** per active plan (e.g., **$125**).
- **Per-plan gas reserve** to cover automated operations.
- **Treasury** for sustainability (development, audits, maintenance).

7. Legacy Fund: From Intent to Impact

If heirs do not claim, capital is **not** lost—it is **re-deployed** per the owner's will (**Legacy Statement**) toward **public goods** and ethical initiatives (e.g., education, open-source, infrastructure). The fallback becomes an **economic-social engine** for the ecosystem.

8. Roadmap & Future Vision

- **Phase 1 — MVP/Testnet (current):**
  Contracts + dApp on **local/Sepolia**; off-chain privacy + on-chain pointer; local keeper; **ETH-only**; multi-owner tests and **salt export**; end-to-end claim validated.
- **Phase 2 — V1 Mainnet Hardening:**
  Chainlink Automation, **Tier-1 audit** + **Immunefi**, formalized **AES-GCM + AAD**, **ETH-only**, initial deposit caps (gated launch).
- **Phase 3 — V2 Adoption:**
  ERC-4337 (smart wallets, **gasless claim** via Paymaster, **EIP-1271**), base **ERC-20** support (e.g., USDT) and **NFT** visibility (no yield), **Legacy Statement** signed (EIP-712/1271) on IPFS.
- **Phase 4 — Public Governance:**
  Guardian Guild (DAO), progressive transfer of parameters (fees/fallback/treasury), **open standard** for on-chain succession.

9. Compatibility & Implementation

- **Networks:** Sepolia (MVP), Mainnet in V1.
- **Accounts:** EOA today; contracts are **AA-compatible**; AA flows in V2.
- **Wallets:** MetaMask (desktop) + hardware wallets.
- **Frontend:** React + ethers v6; **XMTP** for secure delivery of claim codes.

10. Conclusion

Heavenkey provides a **secure**, **privacy-preserving**, and **ethical** infrastructure for **on-chain estate continuity**. It reactivates dormant capital, preserves the owner's intent, and creates a responsible path (**heirs → Legacy Fund**) without relying on custodians or yield mechanisms. It is a blueprint for a **more resilient, perpetual, and fair** digital economy.

Disclaimer

This document is for informational purposes only and does not constitute investment, financial, legal, or tax advice. Heavenkey is experimental software: users must conduct their own due diligence and are solely responsible for actions taken when interacting with the protocol.