Prometheus Chains

Patient-Owned Medical Records & Real-Time Healthcare Payments on Ethereum
Subtitle: Weaving a New Fabric for the Human Experience
Version: 1.5 (Draft) — September 2025
Authors: Patrick Ayelle (and collaborators)
Contact: info@prometheuschains.org

Executive Summary

Healthcare today is defined by fragmentation: patient records scattered across proprietary EMRs, and payments that crawl through clearinghouses and denials before providers see a dime. Prometheus Chains proposes new neutral rails:

- **Rail 1 — Patient-Owned Record.** Patients anchor their medical snapshots on Ethereum (hash only) and store encrypted content on an L2. The record is portable, verifiable, and always theirs.
- **Rail 2 — Real-Time Claims.** Providers submit claims that adjudicate automatically against transparent rules and settle instantly in stablecoins (USDC).

Together, these rails form public infrastructure for healthcare data and finance—composable, neutral, and privacy-preserving.

Pilot invitation. We are seeking **collaborative pilot partners** to co-design a scoped evaluation that fits your environment (workflows, systems, staffing). Rather than prescribing fixed improvement targets, we will **jointly define objectives, baselines, and success criteria** during a short discovery phase. Our aim is to demonstrate practical value while validating security, privacy, and operational fit.

---

1. Introduction: Why Healthcare Needs New Rails

Healthcare is one of the most information- and payment-intensive systems on earth. Yet its foundations are brittle:

- **Data fragmentation.** A patient's history is siloed across dozens of EMRs that don't talk to each other.
- **Payment friction.** A single office visit may take weeks to clear across insurers, intermediaries, and bank rails.

The result: high administrative cost, poor transparency, and frustrated patients and clinicians.

Why previous attempts fell short

- **Interoperability without neutrality.** Contractual interfaces glued systems together, but lacked cryptographic guarantees or credible neutrality. Trust stayed concentrated in intermediaries.
- **Premature blockchain pilots.** Early efforts faced high gas costs, immature privacy for PHI, and clunky UX—pushing many into permissioned "private chains" that delivered databases, not

networks.

2. Vision: A New Fabric for Trust

Every major leap in human coordination followed new infrastructure: the printing press, the internet, and now blockchains. Ethereum extends trust into code—transparent rules that no single institution can change.

Prometheus Chains brings this fabric into healthcare by establishing:

- **Patient-owned records** anchored cryptographically, not institutionally.
- **Programmable payments** that flow in seconds, not months.

These rails unlock immediate wins—faster payments, portable data—while laying groundwork for future services: outcome-based incentives, transparent contracting, consented research feeds, and AI agents coordinating care.

3. How It Works (High-Level)

Rail 1 — Patient Record Lifecycle

1. **Anchor.** Hash of a patient's medical snapshot (FHIR JSON) is written to Ethereum L1. No PHI ever touches the chain.
2. **Store.** Encrypted snapshot is stored on an L2 vault, indexed by a pseudorandom tag.
3. **Restore.** Patient signs again on any device, derives keys/tags, decrypts locally, and verifies the plaintext against the L1 hash.
   (See Appendix A for derivation details and invariants.)

Rail 2 — Claim Lifecycle

1. **Submit.** A provider console or API call submits a claim (patient ID, code, year).
2. **Adjudicate.** The engine checks: provider active, patient covered, code enabled/within limits, vault funded.
3. **Settle.** If checks pass, the provider is paid instantly in USDC; if not, the claim is transparently rejected with reason.
   (See Appendix B for rules, events, and contract interfaces.)

Multi-Chain Model ("Prometheus Chains")

- **Ethereum Mainnet (neutral).** Immutable patient receipts, global audit spine.
- **L2 (local).** Encrypted storage + claims, governed per jurisdiction (HIPAA, GDPR, research).
- **Mobility.** If an L2 censors or deletes, patients can re-publish to another L2 with the same L1 continuity.

4. Security & Privacy Posture

- **No plaintext PHI on-chain.** L1 stores hashes only; L2 stores ciphertext + random tags.
- **Device-first custody.** Patients hold keys; plaintext exists only in memory during operations.
- **Pseudonymous claims.** On-chain claims show only code/year, never patient identity.
- **Operational controls.** Claims engine can be paused; underfunded banks trigger soft rejects instead of reverts.
- **Compliance alignment.** Local rules (HIPAA, GDPR, 42 CFR Part 2) can be encoded at L2.

5. Interoperability

Prometheus Chains is built on the **SMART on FHIR** standard, already mandated in U.S. certified EMRs:

- **POC today:** paste in FHIR JSON to anchor/store/restore.
- **Pilot phase:** mobile OAuth2 + PKCE login; app fetches patient data bundle, anchors it, encrypts, and stores.
- **Outcome:** providers and patients can interact using **existing EMR capabilities**, minimizing custom integration.

6. Pilot Plan (Collaborative & Open-Scoped)

Purpose. Establish a **co-designed pilot** that validates utility, usability, and compliance in your environment.

Approach (indicative).

- **Discovery (2–4 weeks).** Jointly define objectives, scope, baseline measures, data flows, and governance.
- **Build & configure.** Align SMART scopes, provider console access, and L2 parameters in a sandbox.
- **Limited-scope trial.** Run with a small cohort to observe operational fit and value signals.
- **Review & path forward.** Assess outcomes together; decide on extensions or broader rollout.

What we'll define together.

- **Objectives & measures.** e.g., operational speed, staff effort, data availability, patient experience, auditability.
- **Scope & duration.** Number/types of providers, patient cohort size, and pilot length tuned to your constraints.

- **Guardrails.** Security, privacy, and change-management boundaries aligned with your policies.
- **Success criteria.** Mutually agreed indicators of value (qualitative and/or quantitative), set during discovery.

What we provide.

- Mobile app (patient), provider console (claims), admin tools, dashboards, and verifiable on-chain transaction links.
- Technical support for SMART on FHIR connectivity and pilot environment setup.

(If desired, we can share example KPI templates; final metrics are defined collaboratively.)

---

7. Governance & Business Model

- **Public-good core.** Open contracts (PatientRecord, EventVault, ClaimEngine). Stewardship of audits, docs, and governance.
- **For-profit layer.** Enterprise connectors, developer tooling, compliance wrappers, SLAs. Monetization via subscriptions, fees, and enterprise contracts.
- **Funding paths.** Traditional VC, compliant token sale, or hybrid DAO approach—aligned to sustain neutral infrastructure and enterprise-grade delivery.

---

8. What's Live Today

- **Web MVP:** anchor → store → restore flow.
- **Admin/Provider Console:** rules, enrollment, instant claims settlement.
- **Contracts deployed (testnets):** PatientRecord (L1), EventVault + Claims stack (L2).
- **DevOps:** client-side simulation, bytecode checks, clear error handling.

---

9. Roadmap

- **Near-term.** Finish SMART mobile client; run first provider pilots.
- **Mid-term.** ZK proofs for claim circuits; TEE-backed confidential evaluation.
- **Long-term.** Outcome-based payments, AI agents, on-chain contracting, research data streams.

---

10. References

Yue et al. FHIRChain (Vanderbilt, 2018)
ONC. 21st Century Cures Act & API certification rule

HHS HIPAA Privacy & Security Rules
EU GDPR
HL7 FHIR R4, SMART on FHIR Framework
NIST AES-256-GCM
Centre/Circle USDC Whitepaper
Vitalik Buterin, Ethereum White Paper

Appendix A — Cryptographic Model & Invariants (Summary)

- Session root derivation
- Deterministic tags/keys/nonces
- Integrity & privacy guarantees

Appendix B — Contract Interfaces & Data Flows (Summary)

- PatientRecord, EventVault, ClaimEngine ABIs
- Example function calls and event flows

Appendix C — Interoperability Flow (SMART on FHIR) (Summary)

- OAuth2 + PKCE login sequence
- Canonicalization recipe
- Example FHIR bundle anchoring

Access to full technical supplement:

- **GitHub Repository** (latest supplement PDFs and code samples)
- Contact **info@prometheuschains.org** for reviewer copies or private technical notes.