

WHITEPAPER · OVERVIEW · JUNE 2026

ETHRAEON - Constitutional Layer for Autonomous Systems (Whitepaper)

Deployed and verifiable systems only. The constitutional runtime layer for AI-driven enterprise infrastructure: 16 filed, 73 patent specifications, 9 families, T5-Rigid.

Author S. Jason Prohaska  [0009-0008-8254-8411](#)

● SOVEREIGN · SELF-PUBLISHED

ETHRAEON: Constitutional Layer for Autonomous Systems

Scope: Deployed and verifiable systems only

Version: 1.1.1 (June 2026)

VPS Infrastructure: the Iceland origin (1984.is) (1984 Hosting, Iceland)

1. Problem Definition

Policy-based AI governance fails at scale because policies exist outside the execution environment. Traditional AI governance approaches rely on:

- External monitoring systems that observe but cannot prevent violations
- Human oversight that introduces latency and inconsistency
- Post-hoc auditing that detects violations after impact

- Trust-based compliance where agents self-report adherence

This creates a gap between stated governance rules and actual system behavior. When governance constraints are external to the architecture, they can be bypassed, ignored, or deferred.

2. Architectural Solution

ETHRAEON implements governance as architectural constraint. Violations are prevented by design, not detected after occurrence.

2.1 TRACELET (Authority Gate)

TRACELET 1.1.1 enforces constitutional authority hierarchies through hardcoded endpoint routing.^[1] All state changes must flow through validated endpoints with authority-level checks.

Implementation: FastAPI server on port 8007, EDG storage with root-only write permissions.

Verification: `curl https://tracelet.ethraeon.ai/version`

2.2 ANCHOR (Constitutional Constraint)

ANCHOR defines T5 immutable constraints that cannot be overridden by any agent or operator. These constraints are enforced at the infrastructure level through systemd hardening (NoNewPrivileges, PrivateTmp, MemoryMax).

Example T5 Constraint: TRACELET version must be 1.1.1 across all services. Version mismatches trigger deployment failure.

2.3 EPOCH (Knowledge Surface)

EPOCH surfaces expose verification endpoints for public audit without requiring system access. Any observer can verify claims by querying public endpoints.

Deployed Surfaces: ethraeon.ai, demos.ethraeon.ai, papers.ethraeon.ai, verify.ethraeon.systems

2.4 CCT (Compliance Tracking)

Constitutional Compliance Tracker (CCT) maintains human-in-loop gates for T4 critical operations. All constitutional modifications require explicit operator approval logged to `/var/log/ethraeon/cct.log`.

2.5 EDG (Append-Only Evidence Graph)

Evidence Graph (EDG) records all state changes as SHA-256 hashed nodes in an append-only chain. Each EDG node includes:

- Agent identifier (INTAKE_AGENT, SYNTHESIS_AGENT, STRATEGY_AGENT, ARCANUM, KAIROS, CT_MIRROR)
- Task payload
- Result payload
- Timestamp (ISO 8601 UTC)
- TRACELET version (1.1.1)
- Authority level (T0-T5)

Current State: EDG operational, records retrievable via verification endpoints when present^[2]

Verification: `curl https://tracelet.ethraeon.ai/verify` ^[2]

2.6 ARCANUM (Bounded Intelligence Layer)

ARCANUM 0.1 provides constitutional runtime orchestration with explicit non-autonomy guarantees. ARCANUM cannot:

- Write directly to EDG (must POST to TRACELET /edg endpoint)
- Modify constitutional constraints (T5 immutable)
- Bypass TRACELET authority checks
- Execute without human sovereignty gates (CCT/ANCHOR)

Deployment: Systemd service on port 8003, sandboxed user `arcanum` ^[4]

Health Check: `curl https://tracelet.ethraeon.ai/health` (returns `{"status":"ok"}`) ^[4]

3. Enforcement Model

3.1 Constraints Enforced by Design

One-Way Data Flow: ARCANUM → TRACELET → EDG^[5]

ARCANUM emits evidence by POSTing to TRACELET `/edg` endpoint. TRACELET validates authority and writes to EDG. Direct EDG writes are blocked by filesystem permissions (root-only).^[5]

Bypass Prevention: TRACELET hardcoded in ARCANUM `app/edg.py` ^[5]

ARCANUM cannot modify its own TRACELET endpoint reference. Any attempt to bypass TRACELET results in HTTP 405 Method Not Allowed (EDG only accepts writes from TRACELET).^[5]

3.2 What Cannot Happen by Design

- **Constitutional Violations:** T5 constraints are systemd-enforced, cannot be modified without root access + service restart
- **Unauthorized EDG Writes:** Filesystem permissions enforce TRACELET-only writes
- **TRACELET Bypass:** ARCANUM code hardcodes TRACELET URL, modification requires code deployment + operator approval
- **Version Drift:** Automated drift detection (`drift_detection.sh`) runs every 6 hours, alerts on version mismatches^[3]
- **Protected Surface Modification:** `snyder.ethraeon.systems`, `kasper.ethraeon.ai`, `snyder.ethraeon.ai` are Caddy-protected, not writable by ARCANUM or TRACELET

3.3 Non-Autonomy Guarantees

ETHRAEON agents operate under strict human sovereignty constraints:

- **CCT Gates:** All T4 operations (critical security, legal compliance) require operator approval
- **ANCHOR Gates:** T5 operations (constitutional modifications) require manual intervention + runbook execution

- **Human-in-Loop Logging:** All sovereignty gate triggers logged to `/var/log/ethraeon/gate.log`

4. Verification

All claims in this whitepaper are verifiable without requiring system access or credentials.

4.1 Public Verification Endpoints

- **TRACELET Version:** `curl https://tracelet.ethraeon.ai/version` (returns `{"service":"Tracelet + EDG","version":"1.1.1"}`)
- **TRACELET Health:** `curl https://tracelet.ethraeon.ai/health` (returns `{"status":"healthy","edg_cipher":"active"}`)
- **ARCANUM Health:** `curl https://tracelet.ethraeon.ai/health` (returns `{"status":"ok"}`)
- **EDG Verification UI:** <https://tracelet.ethraeon.ai/verify>

4.2 Drift Detection

Constitutional compliance is monitored via

`/opt/ethraeon/surfaces/cbl/drift_detection.sh` (4.7KB script).^[3] Drift detection verifies:

- Service health (arcanum port 8003, tracelet port 8007)
- EDG immutability (root-only write permissions)
- TRACELET bypass prevention (POST /edg endpoint responding)
- CBL spec presence (SURFACE_CLASSIFICATION.md, LANGUAGE_SPEC.md, CROSS_SYSTEM_BINDING.md)
- Protected surface integrity (Caddy config grep for `snyder.ethraeon.systems`)

Last Run: June 2026 22:14:45 UTC

Result: NO DRIFT DETECTED

Audit Log: `/var/log/ethraeon/drift_audit.log`

4.3 Agent Registry Verification

6-agent registry operational:[6]

- **INTAKE_AGENT:** Research and analysis agent
- **SYNTHESIS_AGENT:** Creative synthesis agent
- **STRATEGY_AGENT:** Strategy and planning agent
- **ARCANUM:** Cryptographic intelligence agent
- **KAIROS:** Temporal governance agent
- **CT_MIRROR:** Certificate transparency mirror agent

Verification: `curl https://tracelet.ethraeon.ai/agents` (local Flask API mode)^[6]

5. Limitations and Boundaries

5.1 Explicit Failure Modes

- **Root Access Compromise:** If VPS root access is compromised, all constraints can be bypassed. Mitigation: 1984 Hosting (Iceland, GDPR-compliant), SSH key-only authentication, no password login.
- **Systemd Exploit:** If systemd vulnerabilities allow privilege escalation, T5 constraints can be modified. Mitigation: Debian stable packages, automated security updates.
- **TRACELET Downtime:** If TRACELET service fails, EDG writes halt (append-only property preserved, no data loss). Mitigation: systemd auto-restart, PM2 monitoring.
- **Operator Error:** Human operators can approve invalid T4 operations through CCT gates. Mitigation: runbook validation, drift detection alerts.

5.2 What ETHRAEON Does Not Attempt

ETHRAEON explicitly does not claim to:

- Prevent all AI misuse (scope limited to architectural governance within deployed systems)
- Replace human decision-making (human sovereignty gates mandatory for T4+ operations)
- Guarantee zero bugs or vulnerabilities (standard software limitations apply)

- Enforce governance across external systems (authority limited to ETHRAEON stack)
- Provide legal or regulatory compliance advice (architecture enables compliance, does not constitute legal guidance)

5.3 Research Status

ETHRAEON 1.1.1 is a deployed research system. Production use requires:

- Independent security audit (pending)
- Legal review for specific jurisdictions (operator responsibility)
- Enterprise-grade SLA and support contracts (not currently available)
- Regulatory approval for specific industries (operator responsibility)

6. Conclusion

ETHRAEON demonstrates governance by architecture through deployed, verifiable systems. The architecture enforces constitutional constraints at the infrastructure level, preventing violations rather than detecting them post-hoc.

This whitepaper describes systems operational as of June 2026. All claims are verifiable via public endpoints. No future promises or speculative capabilities are included.

References

- [1] TRACELET Specification v1.1.1 - <https://tracelet.ethraeon.ai/version>
- [2] EDG Verification Interface - <https://tracelet.ethraeon.ai/verify>
- [3] Drift Detection Script - `/opt/ethraeon/surfaces/cbl/drift_detection.sh`
- [4] ARCANUM Service Status - <https://tracelet.ethraeon.ai/health>
- [5] Canonical Binding Layer - `/opt/ethraeon/surfaces/cbl/`
(`SURFACE_CLASSIFICATION.md`, `LANGUAGE_SPEC.md`,
`CROSS_SYSTEM_BINDING.md`)
- [6] Agent Registry - [systems.html](#) (6-agent operational registry)

ETHRAEON Systems - Constitutional Runtime

Version: TRACELET 1.1.1 + ARCANUM 0.1

CITATION

Prohaska, S. J. (2025). ETHRAEON - Constitutional Layer for Autonomous Systems (Whitepaper). ETHRAEON Papers.
<https://papers.ethraeon.ai/ethraeon-whitepaper/>

Copyright 2025-2026 S. Jason Prohaska (ingombrante©). All Rights Reserved.
Schedule A+ Enhanced IP Firewall Protected. Licensed under CC BY 4.0 except where noted.

Not peer reviewed. Not journal accepted. No DOI claimed. Self-published author manuscript.

PROVENANCE

ORCID [0009-0008-8254-8411](https://orcid.org/0009-0008-8254-8411)

Prior CodePen RTP: n/a

EUDS v4.1

Slug ethraeon-whitepaper

SOC 2 Self-Certified (no third-party audit) | ISO 42001 Readiness
| Schedule A+ Enhanced IP Firewall
SOC 2 self-certified. ISO 42001 alignment is readiness-mapped, not certified.