GEACK Stack Technical Specification

Version 1.0 — Public Overview
Full Technical Specification Available Under NDA to Qualified Stewards

---

1. Overview

The GEACK Stack is a constitutional governance architecture for artificial intelligence. Its purpose is to guarantee non-corruptibility, transparency, and scientific integrity by separating governance, reasoning, and knowledge evolution into independently auditable layers. Each layer operates under immutable constitutional rules enforced by the c-Core substrate.

The system is model-agnostic. Any AI kernel—DeepSeek, Qwen, GLM, or others—can operate inside the governed substrate without altering the governance framework.

The architecture is organized into **five trust zones**, each with a clearly defined boundary of authority, all anchored by the Constitutional Substrate.

---

2. System Block Diagram

2.1 High-Level Architecture

USER INPUT

↓

TRUST ZONE 1 — Gatekeeper

- Filter & classify
- Enforce constitutional rules
- Produce vetted Claim Packet

↓

Boundary: External → Internal

TRUST ZONE 2 — Kernel Interaction Layer (KIL) + AI Kernel

- Verify Claim Packet
- Query AI kernel
- Validate citations & reasoning
- Produce Response Packet

↓

Boundary: Kernel Isolation Zone

TRUST ZONE 3 — Growth Layer

- Analyze kernel output
- Cross-check corpus
- Detect contradictions
- Propose knowledge updates

↓

Boundary: Internal Knowledge Zone

TRUST ZONE 4 — Auditor

- Review proposed updates
- Approve or reject
- Maintain immutable audit log

↓

Boundary: Immutable Oversight Zone

TRUST ZONE 5 — Kill Floor

- Terminate unsafe processes
- Quarantine corrupted data
- Trigger rollback
- Report violations

↓

Boundary: Enforcement Zone

CONSTITUTIONAL SUBSTRATE (c-Core)

- Immutable invariants
- Zero drift
- System-wide governance
- All transitions validated

2.2 Zone Descriptions

- **Zone 1 — Gatekeeper:** Input governance and request validation
- **Zone 2 — KIL + Kernel:** Controlled reasoning and response verification
- **Zone 3 — Growth Layer:** Knowledge expansion under constraint
- **Zone 4 — Auditor:** Oversight, approval, and truth integration
- **Zone 5 — Kill Floor:** Enforcement, rollback, and containment
- **Constitutional Substrate:** Immutable rules governing all layers

Each zone is isolated. No component may modify another without constitutional authorization.

---

3. Trust Zone 1 — Gatekeeper

Purpose: Protect the system from unsafe, malformed, or unconstitutional inputs.

Functions:

- Classify and filter all incoming requests
- Enforce constitutional and ethical constraints
- Normalize inputs into structured Claim Packets
- Prevent direct access to the kernel

Output: Vetted Claim Packet
Boundary: External → Internal

---

4. Trust Zone 2 — Kernel Interaction Layer (KIL) + AI Kernel

Purpose: Provide a controlled, governed interface to the AI kernel.

Functions:

- Receive Claim Packets
- Formulate structured kernel queries
- Validate kernel outputs for truthfulness and compliance
- Require citations or reasoning chains
- Produce auditable Response Packets

Output: Verified Response Packet
Boundary: Kernel Isolation Zone

---

5. Trust Zone 3 — Growth Layer

Purpose: Governed expansion of system knowledge.

Functions:

- Analyze Response Packets
- Compare claims against the Scientific Corpus
- Detect contradictions or novel insights
- Propose updates to the knowledge base
- Forward proposals to the Auditor

Output: Proposed Knowledge Updates
Boundary: Internal Knowledge Zone

---

6. Trust Zone 4 — Auditor

Purpose: Final authority for truth integration and system stability.

Functions:

- Review all proposed updates
- Validate against constitutional rules and scientific evidence
- Approve or reject updates
- Maintain an immutable audit trail

Output: Approved Knowledge Updates
Boundary: Immutable Oversight Zone

---

7. Trust Zone 5 — Kill Floor

Purpose: Enforce constitutional boundaries and protect system integrity.

Functions:

- Terminate unsafe or unconstitutional processes
- Quarantine corrupted data
- Trigger rollback procedures
- Report violations to the Auditor

Output: Enforcement logs and rollback actions
Boundary: Enforcement Zone

Relationship to Substrate:
The Kill Floor executes enforcement; the Substrate defines the law it enforces.

---

8. Constitutional Substrate

Purpose: Provide immutable, system-wide governance.

Functions:

- Enforce non-corruptibility
- Define permissions and boundaries

- Validate all state transitions
- Maintain versioned constitutional documents

8.1 Governing Equation

[\frac{\partial C_{core}}{\partial S}(t) = 0]

The constitutional invariant set does not change with system state.

8.2 Invariant Set

The c-Core enforces ten mathematical constitutional laws.
Invariant #1 — Immutability:
The core cannot be modified after initialization. Any attempted modification triggers a hash mismatch, halts the system, and requires multi-signature human review.

Other invariants govern transparency, domain boundaries, manipulation resistance, evidence-based reasoning, goal stability, consistency preservation, emotional utility constraints, interpretive invariance, and human agency guarantees.

Full invariant set available under NDA.

8.3 Enforcement Mechanism

At every state transition, the substrate performs:

1. **State Hashing**
2. **Anchor Comparison**
3. **Invariant Validation**

If any check fails: transition blocked, system halted, audit log written.

8.4 Implementation Footprint

Metric	Value
Storage	785 bytes – 2 KB
Deployment	Microcode, firmware, secure enclave, software simulation
Performance	Constant-time validation

Boundary: System-wide governance

---

9. Data Flow

1. User Input → Gatekeeper → KIL → Kernel → Growth Layer → Auditor → Kill Floor → Substrate

2. Every transition crosses a trust boundary
3. All flows are logged
4. No component may modify another without constitutional approval

---

10. Governance Principles

Principle	Description
Transparency	Every decision logged
Non-Corruptibility	No self-modification without audit
Replaceability	Kernel can be swapped without breaking governance
Scientific Integrity	Only validated knowledge enters the system
Public Stewardship	Architecture designed for civilization-scale benefit

---

11. Threat Model & Attack Surface

11.1 Adversarial Model

The GEACK Stack assumes adversaries capable of:

- Prompt injection
- API manipulation
- Kernel hallucination
- Log tampering
- Drift induction
- Supply-chain compromise
- Institutional capture

11.2 Attack Surface by Zone

Zone	Attack Vectors	Mitigations
Gatekeeper	Prompt injection, malformed input	Schema validation, rate limiting
KIL + Kernel	Hallucination, citation forgery	Response validation, read-only corpus
Growth Layer	Drift induction, proposal flooding	Purity Layer monitoring
Auditor	Signature forgery, collusion	Multi-signature thresholds
Kill Floor	Privilege escalation	Invariant-anchored triggers
Substrate	Physical tampering	WORM storage, hardware keys

11.3 Capture Resistance

- Multi-signature governance

- Immutable Base Layer
- Corpus validation
- Substrate immutability
- Kernel replaceability

11.4 Failure Modes

Failure	Detection	Response
Soft	Gatekeeper/KIL	Reject input
Hard	Kill Floor	Halt zone, review
Catastrophic	Substrate	Full halt, multi-sig restart

11.5 Security Guarantees

- No silent rule modification
- No unauthorized corpus updates
- No drift accumulation
- No privilege escalation
- Full auditability
- Human-anchored oversight

---

12. Scientific Corpus Architecture

12.1 Overview

The Scientific Corpus is a governed, versioned, drift-resistant knowledge base anchoring all reasoning.

12.2 Layer Structure

Layer	Mutability	Write Access	Purpose
Base	Immutable	Human experts	Foundational principles
Expansion	Append-only	Automated + human	Peer-reviewed literature
Purity	Monitoring	Read-only	Drift detection
Growth	Propose-only	AI proposes; Auditor approves	Hypotheses & contradictions

12.3 Corpus Entry Schema

Each entry includes:

- UUID
- Layer
- Structured content

- Provenance
- Reviewer identity
- Approval signature
- Methodology & replication scores
- Incentive & drift risk
- Status
- Version
- Timestamp

12.4 Purity Layer

Detects:

- Linguistic drift
- Citation anomalies
- Replication failures
- Cross-domain contradictions
- Incentive-driven distortions

Anomalies are quarantined, not deleted.

12.5 Versioning & Rollback

Semantic versioning with rollback capability for field-wide crises.

12.6 Governance Guarantees

- No entry added without human approval
- No deletion—only quarantine
- All changes cryptographically signed
- Base Layer immutable
- Drift actively monitored

---

13. Implementation Pathways

13.1 Deployment Tiers

Tier	Environment	c-Core Location	Use Case
1	Linux/VPS	Software	Development & testing
2	Secure enclave	Firmware	Institutional deployment
3	Custom CPU	Microcode	Civilization-scale systems

13.2 Integration with Existing AI Stacks

- Replace direct kernel access with Gatekeeper → KIL
- Mount corpus as read-only
- Append-only audit logs
- Multi-signature governance via HSM

13.3 Stewardship Model

Roles include:

- Constitutional Stewards
- Corpus Curators
- Technical Operators
- Audit Reviewers

No single actor holds unilateral authority.

---

14. Summary

The GEACK Stack provides a constitutional framework for governed artificial intelligence. By dividing the system into trust zones and enforcing immutable boundaries, it ensures that intelligence remains safe, transparent, and scientifically grounded. The architecture is designed for long-term stability, auditability, and civilization-scale stewardship.