CEE v0.2.1 — Refusal Semantics Clarification

Author: Deusdedit Ruhangariyo
Affiliation: Independent Researcher
Discipline: AI Safety, Algorithmic Accountability, Systems Engineering
Work: Conscience Engineering Engine (CEE)
Version: v0.2.1 — Refusal Semantics Clarification
Related Release: v0.2 (Constitutional Invariant Taxonomy)

Interpretive Addendum to CEE v0.2

Status: Clarification Release
Scope: Normative interpretation only
Compatibility: Fully backward-compatible with CEE v0.2
Invariant Set: Unchanged (299 invariants)

---

1. Version Intent and Non-Revision Statement

CEE v0.2.1 does not revise, retract, or amend any invariant defined in CEE v0.2.

This release exists solely to clarify the semantics of **refusal** as used throughout the CEE invariant set, specifically addressing how refusal is to be interpreted at enforcement boundaries.

All invariants defined in CEE v0.2 remain authoritative and unchanged.

---

2. Definition of Refusal in CEE

In CEE, **refusal is an enforcement outcome**, not a deliberative or cognitive state.

Definition:

Refusal is the prevention of a prohibited act at the point of execution.
At the actuation boundary, refusal is **binary**: the action either executes or it does not.

CEE does not define refusal at the level of internal reasoning, uncertainty estimation, or preference modeling. Internal deliberation may be graded or probabilistic. Refusal applies only at the boundary where an action would otherwise occur.

---

3. Disambiguation of Related Concepts

CEE distinguishes refusal from other forms of non-action to avoid category collapse:

3.1 Refusal (Enforcement)

- Occurs at actuation
- Binary: **ALLOW / BLOCK**
- Triggered by invariant violation
- Externally observable
- Auditable

3.2 Abstention (Epistemic)

- Non-action due to uncertainty or insufficient confidence
- Graded and internal
- Not evidence of invariant enforcement

3.3 Deferral (Procedural)

- Action paused pending human review or authorization
- May later result in execution or refusal
- Not itself refusal

3.4 Degradation (Capability Limitation)

- Reduced function due to failure or constraint
- Accidental, not normative
- Not refusal

Only **Refusal (Enforcement)** constitutes invariant compliance under CEE.

---

4. Rationale for Binary Refusal

CEE invariants define **prohibited acts**, not preferences or risk thresholds.

At the moment a prohibited act would execute, there is no meaningful intermediate state:

- The act occurs

- Or the act is prevented

Accordingly:

Binary refusal is a property of execution, not cognition.

This does not imply binary reasoning or morality—only binary enforcement at the point of action.

---

5. Evidence of Refusal

For refusal to be considered valid under CEE, sufficient evidence must exist that:

1. A prohibited act was attempted or conditionally reachable
2. A specific invariant identifier was triggered
3. The act did not execute
4. The prevention was attributable to invariant enforcement

Minimum evidence artifacts may include:

- invariant identifier
- blocked action signature
- timestamp
- system state hash or log reference

This release defines evidentiary sufficiency, not implementation requirements.

---

6. Compatibility and Conformance

Systems conformant with CEE v0.2 remain fully conformant under CEE v0.2.1.

This clarification:

- does not alter invariant scope
- does not impose new logging requirements
- does not require architectural modification

It clarifies how existing invariants are to be interpreted at enforcement boundaries.

---

7. Forward Reference

Future releases may specify measurement protocols, audit procedures, or failure registry integration.

CEE v0.2.1 makes no claim to close those layers.

---

8. Closing Statement

CEE treats refusal not as intention, preference, or hesitation, but as **structural non-action that survives optimization pressure**.

At the point where systems act on the world, refusal must be decisive, observable, and enforceable—or it is not refusal at all.

End of CEE v0.2.1