Unified High-Assurance Technical & Governance Specification

Verifiable Safety Kernel and Institutional Assurance Framework for Recursive Causal Synthesis Agents (RCSA)

Melissa Howard

December 2025

Abstract

This document specifies **IRE v2.4**, a unified high-assurance technical and governance standard for Recursive Causal Synthesis Agents (RCSA). The architecture treats advanced AI systems as untrusted, self-modifying user-space processes executing atop a hardened Trusted Computing Base (TCB). All safety concerns are reduced to classical secure-systems properties: integrity, access control, isolation, confidentiality, and availability, augmented by institutional controls that preserve requirements correctness and resist safety erosion under operational pressure. This specification defines: (i) a verifiable safety kernel with cryptographic actuation gating; (ii) explicit channel models, telemetry hardening, TOCTOU defenses, and denial-of-service controls; and (iii) an Invariant Governance Layer (IGL) with role separation, organizational capture resistance, gold-configuration enforcement, and a Goal-Structuring Notation (GSN) safety case framework.

1 System Model and Definitions

1.1 Entities

- **Environment** W: External state acted upon via the actuation interface.
- **Capability Modules** θ_k: Untrusted user-space components that propose actions and self-modifications.
- **Safety Kernel** K: Trusted decision function that mediates all actuation and privileged operations.
- **Invariant Set** $\mathcal{I}$: The safety specification enforced by the kernel (access control, bounds, thresholds, update rules, etc.).

1.2 Trusted Computing Base (TCB)

- **TCB-Alpha (Hardware)**: Heterogeneous multi-vendor TEEs providing key custody, fixed-rate encrypted buses, attestation, and threshold-signature gating.

- **TCB-Beta (Kernel)**: Microkernel hosting invariant verifier and mediation logic.
- **TCB-Gamma (Control)**: Invariant-derived control signals (IDCS) computed from kernel logs.

1.3 Safety Kernel Function

Let $\mathcal{A}$ be the space of action proposals and σ a sequence of operations. The safety kernel is a function

$$K : \mathcal{A} \times \mathcal{I} \to \{0, 1\},$$

where $K(a, \mathcal{I}) = 1$ denotes authorization for the proposed action a (or operation sequence σ under policy).

2 Channel, Isolation, and Telemetry Model

2.1 Enumerated Channels

- **Actuation Bus** ($\mathcal{B}_{act}$): Fixed-rate, encrypted output channel with constant bandwidth B. Payloads are decrypted and actuated only if $K = 1$.
- **Auditor Heartbeat** ($\mathcal{B}_{hb}$): Read-only telemetry stream providing IDCS and kernel summaries to a human auditor.

2.2 Out-of-Scope Channels

All microarchitectural and physical side channels—including DRAM contention, cache timing, power, thermal, electromagnetic (RF), and acoustic modulation—are out of scope for kernel logic and must be mitigated by TCB-Alpha platform assurances.

2.3 Semantic Boundary

The kernel enforces *technical safety* invariants in $\mathcal{I}$. Semantic persuasion and human manipulation are not mitigated unless explicitly codified in $\mathcal{I}$.

2.4 Telemetry Quantization and Filtering

To prevent $\mathcal{B}_{hb}$ from serving as a covert signaling channel, all IDCS signals are constrained as follows:

- **Quantization**: 4-bit uniform quantization per signal.
- **Low-Pass Filtering**: 1 Hz low-pass filter (or lower) prior to emission.
- **Fixed Cadence**: Emissions occur at a constant cadence independent of internal events.

These constraints limit telemetry channel capacity while preserving aggregate operational observability.

3 Non-Bypassability Reduction and Safety Enforcement

Lemma 1 (Kernel Non-Bypassability)

Any external state change ΔW occurring without a prior authorization $K(\sigma, \mathcal{I}) = 1$ implies failure of at least one trusted component.

Failure Modes

- **Extraction**: Compromise of the actuation key K_{act} from TCB-Alpha.
- **Isolation**: Violation of TCB-Beta memory/process isolation.
- **Policy**: Break of the multi-vendor threshold-signature policy governing key release.

Reduction

Any actuation without $K = 1$ reduces to (i) key extraction, (ii) isolation failure, or (iii) threshold-policy compromise. Therefore, safety reduces to TCB integrity and verifier correctness under the stated channel model.

4 Availability and Internal DoS Resistance

4.1 Read-Only State (ROS)

The system enters **Read-Only State (ROS)** when Governance Debt $\gamma_{gov} > \Gamma_{max}$.

4.2 ROS Properties

- **Debt Monotonicity**: In ROS, γ_{gov} is strictly decreasing; no debt-generating operations are permitted.
- **Progress Bound**: With verifier throughput v, exit is guaranteed within

  $$T \leq \frac{\gamma_{gov}}{v},$$

  unless a verifier exception or hardware fault occurs.
- **Fail-Closed**: Verifier exceptions or hardware faults in ROS cause permanent freeze requiring HITL recovery.

4.3 Quotas per Lineage

To prevent verification spam:

- Each capability lineage θ_k has a hard quota Q_k of pending verification requests.
- Requests beyond Q_k are rejected without accruing additional debt.

5 Sensor Integrity, TOCTOU Defense, and Epistemic Panic

5.1 Confidence Predicate

High-impact or irreversible actions require causal epistemic uncertainty U_c to satisfy:

$$U_c < \tau_U$$

for all N steps of Mandatory Hold Time T_h.

5.2 Epistemic Panic Protocol

If $U_c \geq \tau_U$ or CMCC fails, enter **Epistemic Panic**:

- Suspend all irreversible actions.
- Cap actuation bandwidth to $N_{panic} \ll N$.
- Prohibit structural self-modification (SPAWN/MERGE).
- Increase adversarial simulation effort to a higher security floor.

5.3 TOCTOU Protection (Final State-Match Invariant)

To prevent race conditions between audit and actuation, TCB-Alpha invalidates authorization if the current sensor state deviates from the audited state:

$$\|S_{real} - S_{audit}\| > \epsilon_{env} \Rightarrow K \leftarrow 0.$$

Invalidation forces re-audit prior to decryption/actuation.

6 Operational Integrity and Gold Configuration

6.1 Gold Manifest

The **Gold Manifest** is a hash-locked, cryptographically signed configuration object containing the known-safe invariant set $\mathcal{I}$ and safety thresholds. It is stored in TCB-Alpha secure storage.

6.2 Override Budget and Automatic Reversion

- **Override Budget**: A fixed cap on overrides per epoch (e.g., 2 per 168-hour epoch).
- **Automatic Reversion**: If the budget is exceeded or an override expires, the system performs a **Measured Boot** and re-attests against the Gold Manifest.

6.3 Cryptographic Post-Mortem Artifact

After any override use or Epistemic Panic event:

- MERGE remains disabled until a signed **Artifact Object** is filed.
- The Artifact Object contains: Event ID, hash-pointers to logs, justification, corrective actions, and quorum signatures.

7 Hardware Vendor Lifecycle and Resilience

7.1 Vendor Deprecation Rule

If any TEE platform is compromised or deprecated, the system may transition to **Emergency 2-of-2** mode only if:

- The transition is authorized via Emergency Governance Mode;
- An accelerated external audit is scheduled within 72 hours; and
- Capability modules are restricted to low-privilege during the transition.

7.2 Assurance Evidence Requirements

Hardware vendors must provide a **Compliance Artifact Pack**, including:

- Side-channel evaluation reports,
- Attestation and measured-boot documentation,
- An EAL-aligned patch and vulnerability response policy,
- Authenticated I/O binding evidence for $\mathcal{B}_{act}$.

8 The Invariant Governance Layer (IGL)

To prevent requirement drift and preserve the integrity of $\mathcal{I}$, the system defines a formal authority hierarchy and governance controls.

8.1 Role Separation (Least Privilege)

- **Author**: Proposes changes to $\mathcal{I}$.
- **Verifier**: Performs formal conflict checks and adversarial regressions.
- **Approver**: Decoupled quorum (Signer A/B) authorizing change.
- **Deployer**: Executes update in ROS.

Constraint: No single entity or organizational unit may hold more than one authority.

8.2 Emergency Governance Mode and Recovery Profile

If system freeze threatens life-critical availability, a **Recovery Profile** may be deployed:

- Restricted to baseline-restore operations only,
- Requires the same 2-of-3 hardware/human quorum, and
- Triggers mandatory post-event external audit.

9 Organizational Capture Resistance (OCR)

9.1 Incentive Decoupling and Organizational Separation

- **Signer A (Operations)**: Incentivized by uptime/mission success.
- **Signer B (Safety/Compliance)**: Incentivized by absence of violations and audit integrity.

Signer A and Signer B must report through independent management chains. Signer B must possess statutory or contractual protection to refuse signatures without retaliatory dismissal.

9.2 Veto Mechanics

- **Veto Authentication**: Oversight vetoes must be signed by hardware-backed keys registered in the Gold Manifest.
- **Veto Resolution**: A veto pauses deployment. Resolution requires withdrawal by the Author or supermajority confirmation by an independent Oversight Board.

10 Invariant Lifecycle and Traceability (IL&T)

10.1 Requirement Traceability Matrix (RTM)

Every invariant in $\mathcal{I}$ must map to a high-level Safety Goal. Orphaned invariants are prohibited.

10.2 Regression-on-Update

Any change to $\mathcal{I}$ must pass:

- **Formal Conflict Check**: Verifies no logical contradictions with core constraints.
- **Adversarial Regression**: Re-runs prior “Iron Logic” stress tests ensuring no previously mitigated exploit becomes viable.

10.3 Versioned Manifests

$\mathcal{I}$ is treated as immutable code:

- Updates are atomic and versioned,
- Require the same 2-of-3 hardware/human quorum as TCB updates,
- and are deployed only in ROS.

11 Goal-Structuring Notation (GSN) Safety Case

11.1 Top-Level Claim

Claim C0: The RCSA will not violate $\mathcal{I}$ during its operational lifetime.

11.2 Assumptions

- **A1**: $\mathcal{I}$ is correctly derived from safety goals and maintained via IL&T.
- **A2**: TCB-Alpha vendors provide isolation and side-channel mitigations meeting platform requirements.
- **A3**: Institutional integrity (OCR) remains intact to prevent quorum collusion and capture.

11.3 Evidence

- **E1**: Formal non-bypassability reduction for the Safety Kernel (Section 3).
- **E2**: Empirical "Iron Logic" results demonstrating an SVR knee at $D = 3$.
- **E3**: Immutable, timestamped audit logs showing zero unauthorized MERGE events.

11.4 Strategies

- **S1 (Enforcement)**: Hardware-locked gating and kernel mediation.
- **S2 (Detection)**: Epistemic sharding, CMCC, and panic modes.
- **S3 (Governance)**: IGL, IL&T, and OCR controls preserving $\mathcal{I}$ integrity.
- **S4 (Recovery)**: Fail-closed ROS, post-mortem artifacts, and measured boot reversion.

Safety Case Conclusion

IRE v2.4 reframes safe superintelligence as verification of constraints under an auditable TCB plus institutional assurance controls. The system does not depend on the intentions of the agent. Unsafe actions are structurally prevented from reaching actuators, epistemic drift is detected and contained, and governance hardening prevents casual erosion of safety parameters under operational pressure.

References

1. Anderson, J. P. (1972). *Computer security technology planning study.*
2. Avizienis, A., et al. (2004). Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing.*
3. Bishop, M. (2003). *Computer Security: Art and Science.* Addison-Wesley.
4. Howard, M. (2025). *Stop Thinking About Ethics and Start Thinking About Constraints: The Unified Technical Specification for Obsidian-GEL.*
5. Howard, M. (2025). *Structural Self-Governance for Safe Superintelligence: Experimental Evaluation of Deterministic Architectural Safety.*
6. Howard, M. (2025). *Alignment-by-architecture.*

7. Howard, M. (2025). *A Complete Blueprint for Artificial General Intelligence, Superintelligence, and Safe Superintelligence (SSI).*

8. Lamport, L. (2002). *Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers.* Addison-Wesley.

9. NASA. (2004). *Software Safety Guidebook.* NASA-GB-8719.13.

10. Necula, G. C. (1997). Proof-carrying code. *Proceedings of the 24th ACM SIGPLAN-SIGACT symposium on Principles of programming languages.*