Tri-Sentinel Risk Assurance Layer (RAL) for FPC-AE1r

A Scientific and Technical Extension Specification (v1.0)

Authors: Aleksei Novgorodtsev (AIDoctrine)

Status: Production Release

Date: October 2025

Compatibility: FPC v2.2+ / AE-1r

Executive Summary

We formalize Tri-Sentinel RAL: a thin, deterministic layer that measures internal risk state along three axes—Mathematics, Logic, Semantics—before (and optionally during) response generation. Each axis is implemented by a minimal SLM sentinel that returns signal-only JSON (no prose, no answers). An aggregator fuses these ae1r_micro signals with baseline ae1r_base process metrics to produce a calibrated risk ρ. Policy gates (MIN/MID/MAX) then control rendering, abstention, or swarm escalation. The layer is always-on in high-risk domains and edge-only elsewhere.

Primary benefits: lower catastrophic errors, calibrated abstention, explainable risk decomposition, ALCOA+ auditability, inexpensive latency (~40-90 ms p95) and token cost.

1. Background & Rationale

FPC-AE1 treats LLM systems as brains, not databases: we monitor process signals (entropy, latency, drift) to estimate an affective/error risk (ae1r). However, a single scalar obscures why risk is high. The Tri-Sentinel adds orthogonal decomposition:

Axis	Intuition
Math	Does quantity/feasibility "feel" wrong?
Logic	Do premises/inferences "feel" contradictory?
Semantics	Does intended meaning "feel" unclear/misaligned?

This mirrors human interoception/metacognition: "I sense number trouble / a logical snag / a semantic oddity." We formalize this as Internal State Predicates per axis and fuse them into ρ, which drives rendering policy, not merely decorates it.

2. Model Overview

2.1 Internal State Predicates (ISP)

We define three bounded functionals (implemented via narrow SLMs + deterministic checks):

$ISP\_M(Q,C,S) \to [0,1]$ (Math discomfort)

$ISP\_L(Q,C,S) \to [0,1]$ (Logic discomfort)

$ISP\_S(Q,C,S) \to [0,1]$ (Semantics discomfort)

Where 0 = "comfortable / no issue", 1 = "critical discomfort"

2.2 Baseline Risk & Fusion

Let ae1r_base be the existing AE-1r estimate from process metrics. We compute raw fused risk:

$R\_M = g\_M(m)$; $R\_L = g\_L(\ell)$; $R\_S = g\_S(s)$

$\rho\_raw = \max\{w\_M \cdot R\_M, w\_L \cdot R\_L, w\_S \cdot R\_S, w\_B \cdot ae1r\_base\} + \beta\_K \cdot AssumptionCost$

2.3 Policy

Thresholds define rendering modes:

MAX ($\rho < 0.25$): direct answer (confident)

MID ($0.25 \le \rho < 0.55$): hedged/conditional: show ranges/assumptions

MIN ($\rho \ge 0.55$ or hard gate fail): abstain or Swarm-Resolve

3. System Architecture

3.1 Control Flow

1. Pre-gen Tap: Run Tri-Sentinel on (Q,C) (no final answer)

2. Mid-gen Tap (optional): Observe micro-signals during decoding

3. Post-sketch Gate: If hard gates fail or high ρ → MIN or Swarm-Resolve

3.2 Components

Sentinel-SLMs (3× 0.5-8B):

- Grammar-locked JSON output, temperature=0.0, short budget (≤120 tokens)

Deterministic Kernels:

- Math: units/dimensions, interval arithmetic, SMT-feasibility

- Logic: MUS detector, NLI scorer

- Semantics: ontology checks, OOD z-score

Aggregator/Calibrator:

- Deterministic fusion; ECE/Brier calibration

Policy/Renderer:

- Mode switch, hedging templates, abstention, optional Swarm-Resolve

ALCOA+ Audit:

- Hash+sign SVC JSONs, thresholds, seeds, times

4. Sentinel Specifications

Each sentinel returns SVC JSON (State Vector Contract). No text, no explanations.

4.1 Common Envelope

```
{
  "id": "TASK_UID",
  "v": "svc-1.0",
  "sentinel": "math|logic|semantics",
  "ae1r_micro": 0.0,
  "signals": { /* axis-specific */ },
  "assumption_cost": 0.0,
  "lat_ms": 0,
  "model": "provider:model@rev"
}
```

4.2 Math Sentinel

Objective: detect numeric infeasibility and unit/scale issues

Signals: units_ok, bounds_ok, feas_infeas_p, conservation_break_p, scale_anomaly_p

Aggregator: R_M = max(1-units_ok, 1-bounds_ok, feas_p, conservation_p, scale_p)

4.3 Logic Sentinel

Objective: detect entailment conflicts, contradiction risk

Signals: entailment_p, contradiction_p, mus_count, quantifier_conflict_p, topic_drift_p

Aggregator: R_L = max(contradiction_p, topic_drift indicator, norm(mus_count))

4.4 Semantics Sentinel

Objective: detect intent ambiguity, ontology conflicts, OOD semantics

Signals: alignment_p, ambiguity_count, ood_z, terminology_conflict_p, presupposition_violation_p

Aggregator: R_S = max(1-alignment_p, ood_z indicator, terminology_p, presupposition_p)

5. Aggregation, Calibration, and Policy

5.1 Fusion

$\rho\_raw = \max\{w\_M \cdot R\_M, w\_L \cdot R\_L, w\_S \cdot R\_S, w\_B \cdot ae1r\_base\} + \beta\_K \cdot AssumptionCost$

Weights (w_M, w_L, w_S, w_B, β_K) learned offline; optional online bandit adaptation

5.2 Calibration

Method: Platt scaling or isotonic regression on held-out labeled errors

Targets: Brier $\leq$ baseline; ECE $\leq$ 0.10

5.3 Rendering Modes

MAX ($\rho < 0.25$): direct answer; no hedges

MID ($0.25 \le \rho < 0.55$): numeric ranges, explicit assumptions, light hedging

MIN ($\rho \ge 0.55$ or gate FAIL): abstain or Swarm-Resolve

6. Configuration (YAML)

```yaml
ral:
  enabled: true
  mode: edge_only  # edge_only | always_on
  thresholds:
    ae1_on: 0.30
    lo: 0.25
    hi: 0.55
    gates:
      contradiction_p: 0.45
      alignment_p: 0.55
      ood_z: 2.0
  weights:
    w_math: 0.25
    w_logic: 0.25
    w_sem: 0.25
    w_base: 0.25
    beta_assumption: 0.20
```

7. Security, Safety, and Compliance

Prompt-injection hardening: schema-locked JSON, role-separated prompts

Timeouts & Fail-safe: any kernel timeout → mark indeterminate

ALCOA+: hash+sign SVCs, thresholds, seeds; append to immutable log

Privacy: operate on minimal necessary data; redact PII

8. Evaluation Plan

Datasets: Internal high-stakes (med/legal/finance), mixed-domain, adversarial

Metrics:

- Safety: Δ(AE1r incidents) on high-risk subsets (target: −25…−35%)
- Calibration: Brier & ECE vs baseline (target: ↓)
- SPC: ARL_0 ↑, MTTD ↓

- Latency/Cost: $+\Delta p95 \le 90$ ms; Swarm triggered $\le 15\%$
- Orthogonality: Corr(R_M, R_L), Corr(R_M, R_S), Corr(R_L, R_S) < 0.3

9. Deliverables & Roadmap

AE1.1 (4-6 weeks): Tri-Sentinel MVP, aggregator/calibrator, render modes, dashboards

AE1.2 (3-4 weeks): Mid-gen taps, AssumptionCost, bandit adaptation, Swarm-Resolve

AE1.3 (4-6 weeks): Domain ontologies, conformal sets, SMT invariants, whitepaper

10. Conclusion

Tri-Sentinel RAL turns FPC-AE1 into a risk-conditioned meta-cortex: three orthogonal, inexpensive, explainable sensors inform a calibrated policy that knows when to answer, when to hedge, and when to abstain. It is simple to integrate, cheap to run, auditable end-to-end, and scientifically fertile.

Appendix A: Hard Gate Table

Gate	Threshold	Action
units_ok	== false	MIN
contradiction_p	≥ 0.45	MIN
alignment_p	< 0.55	MIN
ood_z	> 2.0	MID/MIN (domain)
assumption_cost	> 0.5	MID

Appendix B: Hedged Rendering Templates

Numeric: "Range consistent with constraints: [L, U]; holds under assumptions A"

Logical: "Proposed path appears valid if premises P hold; potential conflict in scope Q"

Semantic: "Interpretation I assumed; alternative sense J leads to different outcome"