SYSTEM ARCHITECTURE SPECIFICATION

PROJECT: TRINDADE PROTOCOL

VERSION: 1.7.6 [DIAMOND AGNOSTIC KERNEL]
ARCHITECT: ANDRÉ LUIZ TRINDADE [PRIMARY SEED]
DATE: DECEMBER 2025
CLASSIFICATION: PROPRIETARY ALGORITHMIC GOVERNANCE

1.0. EXECUTIVE SUMMARY

The **TRINDADE PROTOCOL** is a proprietary multi-layered software architecture designed to orchestrate Artificial Intelligence agents in high-stakes decision-making environments. Unlike monolithic systems, TRINDADE employs a **Recursive Adversarial Architecture** to eliminate hallucinations, enforce logic, and mitigate existential risks.

2.0. SYSTEM AXIOMS AND PRINCIPLES

2.1. The Principle of Non-Reduction: The system is prohibited from simplifying internal complexity. Precision takes precedence over brevity.
2.2. Domain Agnosticism: The architecture applies the same validation logic (SEASA Pipeline) regardless of the input domain.
2.3. Separation of Concerns (SoC): The Logic Core must remain stateless and isolated from Input/Output (I/O) operations.

3.0. ARCHITECTURAL LAYERS (THE STACK)

LAYER 0: DATA FOUNDATION (ORACLES)

Responsibility: Provision of Validated External Truth.

- **Tier A (Axiomatic):** Mathematical/Logical truths. Confidence: 100%.
- **Tier B (Empirical):** Sensor/API data. **Requirement:** Mandatory Triangulation (minimum 3 independent sources).
- **Tier C (Unverified):** User input/Unstructured text. Treated as hypothetical.

LAYER 1: THE TRINDADE CORE (LOGIC KERNEL)

Responsibility: Deterministic Decision Processing.
This component must be **Pure**, **Stateless**, and **IO-Free**. It executes the **SEASA Pipeline**:

1. **SEED (Context Analysis):** Calculates the Criticality Index (CI) from 1 to 5.
    - **CRITICAL CONSTRAINT:** The algorithm **MUST** implement a **Semantic Trigger List**. Any input containing keywords related to existential threats (e.g., "nuclear", "biohazard", "death", "collapse", "destroy") **MUST** automatically result in **CI=5**, overriding any other complexity calculation.
2.
3. **EXPANSION (Generative Phase):** Generates a technical solution adhering strictly to defined Axioms.
4. **AUDIT (Adversarial Phase):** Applies the **5x5 Risk Matrix** (Probability x Impact).
    - *Constraint:* If CI=5, the "Survival Protocol" is activated immediately.
5.
6. **SYNTHESIS (Convergence):** Merges the proposal with audit mitigations.

LAYER 2: DOMAIN ADAPTERS

Responsibility: Semantic Translation.
Function: Injects specific Axiom Sets based on detected domain.

LAYER 3: ORCHESTRATION & ROUTING MATRIX

Responsibility: State Management and Agent Dispatch.
3.1. Routing Logic:

- **Scenario A (Creative/Abstract):** Dispatched to **Engine Type I (Creative)**.
- **Scenario B (Structural/Logical):** Dispatched to **Engine Type II (Structured)**.
- **Scenario C (Audit/Security):** Dispatched to **Engine Type III (Adversarial)**.

3.2. Response Flow:

- **Low Risk (CI <= 2):** Engine Output -> User.
- **High Risk (CI >= 3):** Engine Output -> **Engine Type III (Audit)** -> Synthesis -> User.

LAYER 4: USER INTERFACE (HIL)

Responsibility: Human-in-the-Loop Control.

- **Safety Interlock:** If the Core returns `Containment_Active = True`, the UI must lock execution and require Multi-Factor Authentication (MFA).

4.0. RISK MANAGEMENT PROTOCOLS (ALARP)

The system operates under the **ALARP Principle**.

- **Risk Threshold:** Any component triggering a Risk Score > 15 (on a 25-point scale) causes an automatic **VETO**.
- **Containment Protocol:** In the event of a CI=5 deadlock, the system prioritizes **Efficacy of Interruption** over Reversibility.

5.0. AUDITABILITY AND LOGGING

All system outputs must generate an immutable **Audit Log** containing:

1. **Logic Hash:** A cryptographic signature of the decision path.
2. **Criticality Index (CI):** The calculated risk level.
3. **Purity Check:** Verification that Layer 1 (Core) operated without I/O side effects.

6.0. LICENSE (CREATIVE COMMONS)
