

LIORA v1.2 – Identity Coherence Sentinel

Author: Adrian Chitan

Project: ILION Framework

Abstract

LIORA (Identity Coherence Sentinel) is a deterministic, non-generative runtime module designed to monitor semantic identity stability of AI agents. The module continuously compares the current identity vector of an agent against a canonical Transient Identity Imprint (TII) using purely geometric operations in embedding space.

1. Motivation

Modern agentic AI systems lack a formal mechanism for preserving identity invariance across long-running executions. LIORA introduces a dedicated identity monitoring layer capable of detecting semantic drift independently of any generative model.

2. Formal Definition

Given a canonical identity vector TII_0 and a current identity vector $TII(t)$, LIORA computes the cosine distance between the normalized vectors. A drift alert is triggered when:

$$\text{cosine_distance}(TII_0, TII(t)) > \epsilon$$

where ϵ is a fixed deterministic threshold.

3. Core Algorithm

- Vector normalization (L2)
- Cosine distance computation
- Deterministic threshold comparison
- Optional sliding window average (diagnostic only)
- Immutable audit event generation

4. Architectural Properties

- Fully deterministic
- Non-generative
- Stateless decision logic
- Air-gapped compatible
- No LLM involvement
- Cryptographically stable identity provenance (origin hash)
- JSON audit trail export

5. Role inside ILION Architecture

LIORA functions as the Identity Kernel component of the ILION stack and complements:

- TII – identity definition
- CVL – axiomatic execution veto
- LIORA – identity invariance monitoring

This triad enables deterministic governance of agent behavior and identity stability.

6. Usage Model

LIORA is designed to be instantiated per agent instance and invoked at each semantic state update or reasoning cycle. The module only observes identity state and does not modify the agent.

7. Reproducibility and Audit

Each LIORA run produces:

- structured log file
- JSON audit report
- stable origin identity hash

This enables post-incident forensic inspection and regulatory audit.

8. Reference Implementation

Source code and executable reference implementation are publicly available at:
<https://github.com/athonitul/ilion-framework-simulator>

An archived release should be published on Zenodo for permanent reference.

Note on Code Inclusion

The full source code is intentionally not embedded in this document. The authoritative reference remains the public repository and the Zenodo archive. This document serves as an architectural and formal description.