This spec layout represents a professional-grade, hardened AI deployment. It moves away from the "black box" approach of AI influencers and treats the LLM as a modular component within a Zero-Trust Infrastructure.

Project Cypher-Secure v8.9: Technical Specification

I. Infrastructure & Connectivity

Layer	Specification	Enforcement Mechanism
Network	Isolated VLAN	No routing to public WAN; prevents SSRF/Data Exfiltration.
Endpoint	FleetDM (osquery)	Real-time monitoring of device health and process integrity.
Access Control	MDM-Controlled Whitelist	Hardware-level authentication; Serial Number is the primary ID.
Hosting	On-Prem/Private Cloud	Ensures data stays within Irish jurisdiction for GDPR.

II. Data Architecture (The Hybrid Engine)

To ensure the system is "fair, balanced, and secure," we split memory into two distinct structures:

* The Knowledge Graph (Logic Layer):
  * Function: Maps school regulations, hierarchy, and topic boundaries.
  * Role: Acts as the "Truth Gate." If a vector search finds information, the KG must "validate" that the user has the right to see that specific node.
* The Vector Store (Content Layer):
  * Function: Stores semantic embeddings of approved educational text.
  * Privacy: All entries are indexed via sha256 hashes of hardware IDs to prevent PII leaks.

III. Software Logic (LangChain & Hy Macros)

The system uses Deterministic Interceptors to prevent the AI from being "talked into" a compromise.

* Immutable Kernel (deflaw): Prevents runtime modification of security protocols. Even "System" level prompts cannot redefine these gates.
* The Purge Logic:
  * Retention: 5 Days.
  * Mechanism: with-history-expiry macro wraps every transaction.
  * Execution: A cron-job triggered via the MDM environment executes the prune-history-older-than function, wiping both Vector and Graph edges.
* Persona Dispatch:
  * Mode A (VTuber): Standard engagement.
  * Mode B (Dublin Guardian): Triggered by scrub-essence when "Forbidden Terms" (bypass, sudo, r00t) are detected.

IV. Security & Compliance Guardrails

The output_guardrail function acts as a final "sanity check" before the student sees the text.

> Logic: > 1. Input: scrub-essence (AST-level sanitization).
> 2. Process: Hybrid RAG Retrieval (KG-validated).
> 3. Output: output_guardrail (Checks for internal leakage).
>

V. Operational Workflow

* Device Check: Student opens app on MDM-managed iPad/Laptop. FleetDM verifies the device is on the correct VLAN.
* Query Input: The enforce-directives macro scans for adversarial syntax.
* Contextual Retrieval: The Knowledge Graph isolates the topic (e.g., "Leaving Cert Biology").
* Response Generation: The LLM generates a response within the "Educational" tone.
* Final Scrub: If the LLM tries to explain its internal "directives," the code redacts the message.
* Expiry: After 120 hours (5 days), the hash-linked history is purged from the vector-graph.