📘 GENAI FACTORY – HỆ SINH THÁI VẬN HÀNH & MLOPS HARDENING

🎯 MỤC TIÊU

GenAI Factory hoạt động như một **hệ sinh thái sống động**, bao gồm hai chu trình chính:

1️⃣ **Chu trình Vận hành & Suy luận (Inference Workflow)** – nơi hệ thống phản hồi yêu cầu người dùng một cách hiệu quả và an toàn.
2️⃣ **Chu trình Quản trị & Tái huấn luyện (Retraining/MLOps Workflow)** – đảm bảo mô hình và dữ liệu luôn được cập nhật, chính xác và tối ưu.

🧠 I. CHU TRÌNH 1: VẬN HÀNH & SINH SUY LUẬN (INFERENCE WORKFLOW)

Đây là luồng xử lý thời gian thực (real-time) của hệ thống, tận dụng các kỹ thuật **Async I/O** và **Resilience** đã được Hardening.

Giai đoạn	Hành động Kỹ thuật	Thành phần Chính	Giá trị Hardening
1. Nhận & Bảo mật (Ingestion & Security)	Request Flow Start: Yêu cầu đến API → Kiểm tra Rate Limiting & Input Safety.	`assistant_service.py`, `safety_pipeline.py`	🛡️ Chống DoW/Lạm dụng: Ngăn tấn công hoặc chi phí cao do spam request.
2. Điều phối Cốt lõi (Orchestration)	Routing: `AssistantInferenceService` chọn pipeline (RAG hoặc Agent). Toàn bộ luồng async.	`assistant_inference.py`	⚡ Hiệu suất: Async I/O giúp hệ thống không bị chặn ở bất kỳ điểm I/O chậm nào.
3. Truy xuất Tri thức (RAG/Agent Logic)	RAG: Gọi RAG Tool → Vector DB → render RAGPrompt.		
	Agent: Gọi `async_loop` → LLM → Thought/Action → Tool (`tool.async_run`).	`rag_pipeline.py`, `react_agent.py`, `BaseTool`	🔁 Độ tin cậy: `async_run` + Retry/Fallback trong LLM đảm bảo pipeline không lỗi khi API chậm hoặc bị ngắt.
4. Sinh phản	Prompt cuối cùng gửi	`OpenAILLM` kế thừa	🔄 Resilience: Xử lý

Giai đoạn	Hành động Kỹ thuật	Thành phần Chính	Giá trị Hardening
hồi (Generation)	đến LLM Wrapper.	`BaseLLMWrapper`	lỗi 429/5xx tự động qua Retry/Backoff, tránh crash request.
5. Giám sát & Ghi log (Ops)	Ghi lại chi phí và độ trễ.	`CostMonitor`, `LatencyMonitor`, `TracingUtils`	💰 Kiểm soát chi phí: Log token usage & latency để cảnh báo và tối ưu.
6. An toàn Đầu ra (Output Safety)	Kiểm tra phản hồi về Toxicity, PII, hoặc dữ liệu nhạy cảm.	`safety_pipeline.py`	✅ Tuân thủ & Bảo mật: Loại bỏ nội dung vi phạm hoặc rò rỉ dữ liệu.

🧪 II. CHU TRÌNH 2: QUẢN TRỊ MÔ HÌNH & TÁI HUẤN LUYỆN (MLOPS WORKFLOW)

Chu trình này đảm bảo mô hình AI và tri thức nền tảng (RAG Index) luôn được cập nhật và đáng tin cậy.

Giai đoạn	Hành động Kỹ thuật	Thành phần Chính	Mục tiêu MLOps
1. Kích hoạt Job (Trigger)	Lịch trình (Airflow DAG hoặc CronJob) kích hoạt retraining job.	`infra/scheduler/`	🔁 Tự động hóa: Loại bỏ thao tác thủ công.
2. Huấn luyện (Training)	`assistant_trainer.py` huấn luyện Fine-Tuning. `mlflow_adapter.py` ghi log tham số và artifact.	`assistant_trainer.py`, `mlflow_adapter.py`	🧩 Tái tạo: Mọi model có thể reproduce và trace đầy đủ.
3. Đánh giá Chất lượng (Evaluation)	`evaluation_orchestrator.py` chạy các Evaluator như `HallucinationEval`, `SafetyEval`.	`evaluation_orchestrator.py`, `base_evaluator.py`	🧠 Governance: Đảm bảo mô hình đạt chuẩn an toàn và chính xác.
4. Cập nhật Tri thức (RAG Indexing)	Job gọi LLM Wrapper `.embed()` để tạo lại Vector Index.	`LLM Wrapper`, `Vector DB`	🔄 Cập nhật: Giữ tri thức RAG mới nhất (chính sách, báo cáo,...).
5. Giám sát Độ lệch (Drift Monitoring)	`drift_monitor.py` phân tích log để phát hiện hành vi hoặc hiệu suất thay đổi.	`drift_monitor.py`, `logging_utils.py`	⚠️ Phòng ngừa: Phát hiện model decay và kích hoạt retraining sớm.

🧭 TỔNG KẾT

GenAI Factory vận hành như **một hệ thống khép kín**:

- **Chu trình Inference** → tạo ra **Observation & Metrics** phục vụ giám sát.
- **Chu trình MLOps** → sử dụng thông tin này để **tái huấn luyện, tinh chỉnh và cập nhật hệ thống**.

💡 **Inference → Monitoring → Retraining → Improved Model → Back to Inference** – một vòng lặp hoàn chỉnh của Trí tuệ nhân tạo thực chiến.