

Để chứng minh tính khả thi của phương pháp luận đề xuất, chúng tôi đã xây dựng một hệ thống nguyên mẫu (prototype) mang tên "**Project SDEAL**". Kiến trúc của hệ thống này được thiết kế theo 4 tầng chức năng chính: (1) Thu thập Dữ liệu, (2) Xử lý & Phân tích Trí tuệ, (3.a) Đầu ra & Hành động, và (3.b) Vòng lặp Học hỏi & Cải tiến.

Lưu ý quan trọng về lựa chọn công nghệ: Để tối ưu hóa chi phí và đảm bảo khả năng tiếp cận cho mục đích nghiên cứu học thuật, phiên bản hiện tại của **Project SDEAL-IntelliSafe** được triển khai hoàn toàn trên các dịch vụ có Gói Miễn phí Vĩnh viễn (Perpetual Free Tier) của Google Cloud và các công cụ mã nguồn mở như Google Colaboratory. Theo đó, các tác vụ đòi hỏi tài nguyên tính toán cao như huấn luyện mô hình (Model Training) và điều phối luồng phân tích (Workflow Orchestration) được thực hiện trong môi trường Google Colab.

Tuy nhiên, trong một kịch bản triển khai thực tế ở quy mô sản xuất (production-scale), hệ thống này có thể được nâng cấp bằng cách tích hợp thêm các dịch vụ trả phí chuyên dụng trên Google Cloud Platform. Việc nâng cấp này sẽ mang lại những lợi ích vượt trội về **tự động hóa, khả năng mở rộng (scalability), và độ tin cậy**, cụ thể như sau:

- **Tăng cường Tự động hóa và Hiệu suất Real-time:** Thay vì xử lý dữ liệu theo lô (batch processing) trong Colab, kiến trúc nâng cao sẽ sử dụng **Cloud Dataflow** để xử lý luồng dữ liệu viễn trắc thời gian thực và **Vertex AI Pipelines** để tự động hóa toàn bộ chuỗi phân tích (từ lúc phát hiện bất thường đến lúc đưa ra khuyến nghị) mà không cần can thiệp thủ công.
- **Nâng cao Khả năng Phục vụ và Tích hợp:** Các mô hình AI cốt lõi như Anomaly Detector và GNN/GAN Reconstruct sẽ được triển khai dưới dạng các API có khả năng co giãn trên **Vertex AI Endpoints**. Điều này cho phép hệ thống **Project SDEAL** phục vụ hàng ngàn yêu cầu dự đoán đồng thời với độ trễ thấp và dễ dàng tích hợp vào các hệ thống điều hành bay (OCC) hiện hữu.
- **Hoàn thiện Vòng lặp MLOps:** Thay vì tái huấn luyện thủ công, kiến trúc nâng cao sẽ tận dụng **Cloud Scheduler** và **Vertex AI Training** để thiết lập một vòng lặp MLOps hoàn chỉnh. Hệ thống có thể tự động "học hỏi" từ phản hồi của các nhà phân tích hàng đêm (Tactical Learning), liên tục cải thiện độ chính xác và giảm thiểu cảnh báo sai một cách tự chủ.

Bằng cách này, kiến trúc của **Project SDEAL** không chỉ chứng minh hiệu quả của phương pháp luận cốt lõi trong môi trường nghiên cứu miễn phí, mà còn vạch ra một lộ trình rõ ràng để phát triển thành một giải pháp an toàn hàng không toàn diện, mạnh mẽ và sẵn sàng cho môi trường công nghiệp.

Tầng 1: Thu thập & Chuẩn bị Dữ liệu (Data Ingestion & Preparation)

1. Toàn bộ các nguồn dữ liệu (Telemetry, Logs, Weather, Reports):

- **Công nghệ:** Mô phỏng bằng Python trong Google Colab và lưu trữ dưới dạng file trên Google Drive hoặc GitHub.
- **Luồng:** Dữ liệu không được thu thập real-time. Thay vào đó, một module `simulator.py` chạy trong Colab sẽ tạo ra các bộ dữ liệu đa nguồn (dạng file CSV, JSON) cho từng kịch bản nghiên cứu. Dữ liệu thực tế (như file Excel tải nạp) cũng được tải trực tiếp lên Google Drive. Toàn bộ việc làm sạch, chuẩn hóa được thực hiện bằng thư viện Pandas trong Colab.
- **Lý do:** Chi phí bằng 0, đơn giản hóa tối đa, và hoàn toàn phù hợp với quy trình nghiên cứu khoa học nơi dữ liệu thường được xử lý theo lô (batch) thay vì luồng thời gian thực.

Tầng 2: Xử lý Trung tâm & Phân tích Trí tuệ (Core Processing in Google Colab)

1. Data Lake & Feature Store (Mô phỏng):

- **Công nghệ:** Google Drive (để lưu file) và các đối tượng Pandas DataFrame trong bộ nhớ của Colab.
- **Luồng:** Notebook Colab đọc dữ liệu thô từ Google Drive, xử lý và lưu trữ chúng dưới dạng các DataFrame. Các DataFrame này đóng vai trò là "nguồn chân lý duy nhất" trong suốt phiên làm việc, thay thế cho BigQuery và Feature Store.
- **Lý do:** Tận dụng bộ nhớ và khả năng xử lý của Colab, không cần đến các dịch vụ lưu trữ và truy vấn phức tạp, giữ chi phí bằng không.

2. GNN/GAN Reconstruct & Anomaly Detector (Huấn luyện và Dự đoán):

- **Công nghệ:** Google Colab (với tài nguyên GPU/TPU miễn phí) và các thư viện như PyTorch/TensorFlow.
- **Luồng:**
 - **Huấn luyện:** Toàn bộ code huấn luyện các mô hình AI (GNN, GAN, LSTM, TCN...) được viết và thực thi trực tiếp trong notebook Colab, tận dụng GPU miễn phí để tăng tốc. Mô hình đã huấn luyện (`.pk1`, `.h5`) được lưu vào Google Drive.
 - **Dự đoán:** Khi cần chạy phân tích, notebook sẽ tải file mô hình từ Google Drive vào bộ nhớ và thực hiện dự đoán trực tiếp trên dữ liệu DataFrame. **Không có API Endpoint.**
- **Lý do:** Đây là sự thay thế cốt lõi. Nó loại bỏ hoàn toàn chi phí của Vertex AI Training và Endpoint, đồng thời vẫn cung cấp đủ sức mạnh tính toán cho quy mô dự án.

3. Luồng Phân tích (Orchestrated by Python Functions):

- **Điều phối:** Thay vì Vertex AI Pipelines, luồng phân tích được điều phối bởi một chuỗi các lệnh gọi hàm Python tuần tự bên trong notebook Colab.
- **Luồng chi tiết:**
 - **Khi Anomaly Detector phát hiện bất thường:** một hàm `run_analysis_flow()` được gọi.
 - **Giải thích Kỹ thuật:** Hàm này gọi một thư viện Python (như SHAP) để tính toán và trả về các feature quan trọng.
 - **Phân tích Con người:** Hàm này lấy các báo cáo tường thuật, xây dựng prompt, và gọi **Gemini Pro API** (sử dụng API key miễn phí từ Google AI Studio) để phân loại HFACS.

- **Lý do:** Đơn giản, dễ debug, và chứng minh được logic của luồng phân tích mà không cần đến một công cụ điều phối phức tạp và tốn kém.

4. Risk Triage Engine (Bộ máy Phân loại Rủi ro):

- **Công nghệ:** Một hàm Python cuối cùng trong luồng phân tích.
- **Luồng:** Hàm này nhận kết quả từ SHAP và Gemini, áp dụng một logic đơn giản để hợp nhất, chấm điểm rủi ro, và quyết định hành động tiếp theo.
- **Lý do:** Giữ toàn bộ logic trong một môi trường duy nhất, dễ kiểm soát và không tốn chi phí.

Tầng 3: Đầu ra & Hành động (Mô phỏng trên Console)

- **Công nghệ:** Lệnh `print()` trong Python.
- **Luồng:** Thay vì thực sự tích hợp và gọi các API bên ngoài, module `RiskTriageEngine` sẽ chỉ mô phỏng các hành động bằng cách in các thông điệp rõ ràng ra console của Colab. Ví dụ:
 - `print("[ACTION] Recommendation: Generate Maintenance Work Order for 'Flap System'.")`
 - `print("[ACTION] Recommendation: Push 'EFB Pilot Advisory' regarding flap asymmetry.")`
- **Lý do:** Tập trung vào việc chứng minh logic ra quyết định của AI, không tốn thời gian và công sức cho việc tích hợp không cần thiết trong giai đoạn nghiên cứu.

Tầng 4: Vòng lặp Học hỏi & Cải tiến (Mô phỏng Thủ công)

1. Analyst Disposition & Feedback/Label:

- **Công nghệ:** Lệnh `input()` của Python.
- **Luồng:** Sau khi hệ thống đưa ra giải thích, chương trình sẽ dừng lại và hỏi người dùng (đóng vai nhà phân tích) bằng một câu lệnh như: `feedback = input("Was this analysis correct? (y/n/partial): ")`. Phản hồi này sẽ được ghi lại vào một file log trên Google Drive.
- **Lý do:** Một cách đơn giản và hiệu quả để mô phỏng việc thu thập phản hồi từ chuyên gia.

2. Dual-Tier MLOps Loop (Mô phỏng):

- **Công nghệ:** Chạy lại notebook thủ công.
- **Luồng:**
 - **Tactical Learning:** Việc "tái huấn luyện hàng đêm" được mô phỏng bằng cách Hiếu sẽ chạy lại notebook huấn luyện mô hình với bộ dữ liệu đã được bổ sung thêm các phản hồi mới từ file log.
 - **Strategic Learning:** Việc "phân tích chiến lược" được mô phỏng bằng cách Hường sẽ mở một notebook khác để phân tích file log phản hồi và các kết quả thử nghiệm, từ đó rút ra các xu hướng và insight.
- **Lý do:** Chứng minh được khái niệm về một vòng lặp MLOps mà không cần xây dựng một hệ thống tự động hóa tốn kém, hoàn toàn phù hợp với mục tiêu của một bài báo khoa học.