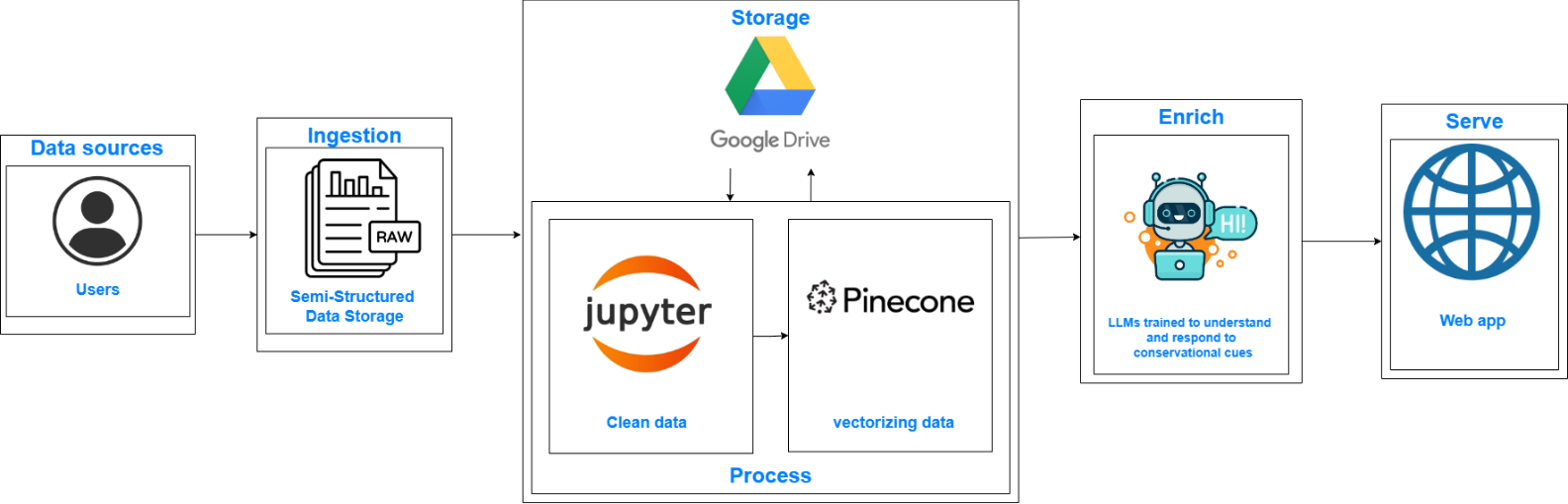
1. Data Sources: 2 file dữ liệu video.json và problem.json mà nhóm sử dụng trong đồ án này được thu thập tại phần “The course recources” của bộ dữ liệu MOOCCubeX [github.com/THU-KEG/MOOCCubeX/tree/main](https://github.com/THU-KEG/MOOCCubeX/tree/main) gồm theo dõi hành động của người dùng với video, bộ câu hỏi và đáp án của các khoá học trên XuetangX
2. Ingestion:

* Thu thập dữ liệu: Dữ liệu từ các file JSON được thu thập định kỳ. Việc thu thập có thể được thực hiện tự động hoặc theo lịch trình nhất định.
* Lưu trữ bán cấu trúc: Các file JSON được lưu trữ trong một hệ thống lưu trữ bán cấu trúc. Điều này cho phép lưu trữ dữ liệu ở định dạng linh hoạt, dễ dàng truy xuất và quản lý mà không cần tuân theo một cấu trúc cố định nghiêm ngặt như trong cơ sở dữ liệu quan hệ.

1. Storage: Nhóm sử dụng Google Drive làm công cụ lưu trữ, Google Drive là dịch vụ lưu trữ đám mây được phát triển bởi Google. Google Drive cho phép người dùng lưu trữ tệp tin, đồng bộ hóa tệp giữa các thiết bị và chia sẻ tệp với người khác:

* Lưu trữ trung tâm: Google Drive được sử dụng làm nơi lưu trữ trung tâm cho tất cả các file dữ liệu thu thập được, đảm bảo tính nhất quán và dễ dàng truy cập.
* Quản lý file: Các file JSON từ bước Ingestion được tải lên Google Drive, nơi chúng có thể được tổ chức theo thư mục, ngày tháng, hoặc sự kiện cụ thể.
* Tích hợp API: Sử dụng Google Drive API để tự động hóa quá trình tải lên và quản lý file.

1. Process: Xử lý dữ liệu

* Clean Data (Jupyter)
* Jupyter Notebook: Sử dụng Jupyter Notebook để làm sạch dữ liệu. Các thao tác làm sạch bao gồm loại bỏ các bản ghi lỗi, dữ liệu trùng lặp, chuẩn hóa định dạng dữ liệu và xử lý các giá trị thiếu.
* Quy trình làm sạch:
  + Đọc dữ liệu: Sử dụng Pandas để đọc dữ liệu từ các file JSON.
  + Loại bỏ dữ liệu lỗi: Xác định và loại bỏ các bản ghi lỗi hoặc không hợp lệ.
  + Chuẩn hóa dữ liệu: Đảm bảo các giá trị có định dạng thống nhất (ví dụ: định dạng thời gian, định dạng ID).
  + Xử lý dữ liệu thiếu: Điền vào các giá trị thiếu hoặc loại bỏ các bản ghi thiếu dữ liệu quan trọng.
* Normalized, Vectorizing Data (Pinecone):
* Pinecone: Sử dụng Pinecone để chuẩn hóa và vector hóa dữ liệu. Dữ liệu sau khi làm sạch sẽ được chuyển đổi thành các vector để phục vụ cho quá trình huấn luyện mô hình học máy.
* Quy trình chuẩn hóa và vector hóa:
  + Chuẩn hóa dữ liệu: Các giá trị trong dữ liệu được đưa về các phạm vi phù hợp. Ví dụ, các giá trị thời gian có thể được chuẩn hóa về cùng đơn vị (giây).
  + Vector hóa dữ liệu: Chuyển đổi dữ liệu thành các vector số học. Ví dụ, mỗi hành động của người dùng có thể được biểu diễn bằng một vector với các giá trị như thời gian, loại hành động, và ID video.

1. Enrich: Sử dụng mô hình ngôn ngữ lớn LLMs:

* LLMs (Large Language Models): Sử dụng các mô hình ngôn ngữ lớn như GPT-3 để hiểu và phản hồi lại các ngữ cảnh hội thoại từ người dùng. Mô hình này được huấn luyện để hiểu các ngữ cảnh phức tạp và cung cấp phản hồi phù hợp.
* Quy trình huấn luyện:
  + Chuẩn bị dữ liệu huấn luyện: Sử dụng dữ liệu đã được vector hóa từ bước trước để huấn luyện mô hình ngôn ngữ.
  + Huấn luyện mô hình: Sử dụng TensorFlow hoặc PyTorch để huấn luyện mô hình ngôn ngữ. Quá trình này bao gồm việc tối ưu hóa các tham số của mô hình để nó có thể hiểu và phản hồi lại các tín hiệu từ người dùng.

1. Serve:

* Ứng dụng web: Phát triển một ứng dụng web để người dùng có thể tương tác với hệ thống. Ứng dụng này cung cấp giao diện cho người dùng thực hiện các hành động và nhận phản hồi từ hệ thống.
* Tích hợp mô hình ngôn ngữ lớn: Mô hình ngôn ngữ lớn được tích hợp vào ứng dụng web để xử lý các yêu cầu từ người dùng và cung cấp các phản hồi phù hợp theo ngữ cảnh.
* Quy trình phát triển ứng dụng web:
  + Front-end: Xây dựng giao diện người dùng sử dụng các công nghệ như HTML, CSS, JavaScript, và các framework như React hoặc Angular.
  + Back-end: Xây dựng server để xử lý các yêu cầu từ front-end, sử dụng các framework như Flask, Django hoặc Node.js.
  + Tích hợp mô hình: Kết nối mô hình ngôn ngữ lớn với back-end để xử lý các câu hỏi và yêu cầu từ người dùng