HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Trần Quang Đại

NGHIÊN CỨU, XÂY DỰNG TRỤC LIÊN THÔNG DỮ LIỆU PHỤC VỤ LƯU TRỮ, QUẢN LÝ DỮ LIỆU ĐA CẤU TRÚC

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI - 2025

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Trần Quang Đại

NGHIÊN CỨU, XÂY DỰNG TRỤC LIÊN THÔNG DỮ LIỆU PHỤC VỤ LƯU TRỮ, QUẢN LÝ DỮ LIỆU ĐA CẤU TRÚC

CHUYÊN NGÀNH: HỆ THỐNG THÔNG TIN

MÃ SỐ: 8.48.01.04

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC:

TS. Phan Lý Huỳnh

HÀ NỘI – 2025

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi. Các kết quả nghiên cứu trong đề án là trung thực, có nguồn gốc xuất xứ rõ ràng và chưa từng được ai công bố trong bất kỳ công trình khoa học nào.

Hà Nội, 9h30 ngày 15 tháng 12 năm 2024

Học viên thực hiện đề án

Trần Quang Đại

LỜI CẢM ƠN

Trong suốt quá trình học tập, nghiên cứu để thực hiện đề án, ngoài nỗ lực của bản thân, tôi cũng đã nhận được sự hướng dẫn nhiệt tình của quý thầy cô, cùng với sự động viên và ủng hộ của gia đình, bạn bè và đồng nghiệp. Với lòng kính trọng và biết ơn sâu sắc, tôi xin gửi lời cảm ơn chân thành tới: Ban Giám Đốc, Phòng đào tạo sau đại học và quý thầy cô đã tạo mọi điều kiện thuận lợi giúp tôi hoàn thành đề án.

Tôi xin chân thành cảm ơn Thầy TS. Phan Lý Huỳnh, người thầy kính mến đã hết lòng giúp đỡ, hướng dẫn, động viên, tạo điều kiện cho tôi trong suốt quá trình thực hiện và hoàn thành đề án.

Tôi xin chân thành cảm ơn gia đình, bạn bè, đồng nghiệp trong cơ quan đã động viên, hỗ trợ tôi trong lúc khó khăn để tôi có thể học tập và hoàn thành đề án. Mặc dù đã có nhiều cố gắng, nỗ lực, nhưng do thời gian và kinh nghiệm nghiên cứu khoa học còn hạn chế nên không thể tránh khỏi những thiếu sót. Tôi rất mong nhận được sự góp ý của quý thầy cô cùng bạn bè đồng nghiệp để kiến thức của tôi ngày một hoàn thiện hơn. Xin chân thành cảm ơn!

Hà Nội, ngày 15 tháng 12 năm 2024

Học viên thực hiện đề án

Trần Quang Đại

MỤC LỤC

DANH MỤC CÁC HÌNH VẼ

MỞ ĐẦU

1. Lý do chọn đề tài

Trong thời đại hiện nay, khi công nghệ thông tin và truyền thông phát triển mạnh mẽ, khái niệm "Chuyển đổi số" ngày càng trở lên quan trọng và phổ biến. Chuyển đổi số là quá trình thay đổi tổng thể và toàn diện của cá nhân, tổ chức về cách sống, cách làm việc và phương thức sản xuất dựa trên các công nghệ số [1]. Theo Thomas M. Siebel, chuyển đổi số là sự hội tụ của 4 công nghệ đột phá: điện toán đám mây (cloud computing), dữ liệu lớn (Big Data), internet vạn vật (IoT) và trí tuệ nhân tạo (AI) [2]. Ví dụ, trước đây chúng ta cần phải đến thư viện để tìm kiếm tài liệu thì nay có thể tra cứu thông qua cổng thông tin điện tử của các thư viện trên thế giới. Việc thay đổi hành vi này có thể được xem là chuyển đổi số.

Chuyển đổi số đang trở thành một chủ đề nóng trong mọi lĩnh vực từ y tế, tài chính, sản xuất, bán lẻ, và tất nhiên giáo dục cũng nằm trong xu thế này. Chuyển đổi số dần trở thành yếu tố then chốt trong việc nâng cao chất lượng đào tạo và quản lý tại các trường Đại học. Tại Việt Nam, vấn đề này ngày càng được chú trọng và trở thành một phần không thể thiếu trong chiến lược phát triển giáo dục. Nhiều trường Đại học ở Việt Nam đã bắt đầu áp dụng công nghệ số trong giảng dạy và học tập. Các nền tảng học trực tuyến như Moodle, Blackboard, và Microsoft Teams được sử dụng để tổ chức các khóa học, bài giảng và thảo luận. Một số trường đã số hóa tài liệu học tập, bài giảng, khóa học giúp sinh viên dễ dàng truy cập kho tri thức từ xa từ đó nâng cao tính linh hoạt trong việc học tập. Hoặc là hệ thống quản lý đào là một thành phần thiết yếu trong việc tổ chức và quản lý quy trình tại nhiều cơ sở giáo dục và các tổ chức đào tạo.

Trên thực tế, các hệ thống này thường không được tích hợp với nhau dẫn đến tình trạng thông tin bị phân tán và khó khăn trong việc lưu trữ cũng như quản lý dữ liệu. Các hệ thống khác nhau có thể sử dụng các định dạng dữ liệu, cơ sở dữ liệu, giao thức truyền thông khác nhau nảy sinh các vấn đề như độ trễ trong việc cập nhật dữ liệu, sự không nhất quán thông tin. Vì vậy, cần có một nền tảng hỗ trợ việc trao

đổi và liên thông dữ liệu giữa các hệ thống. Việc liên thông dữ liệu làm tăng hiệu quả quản lý bằng cách đồng bộ thông tin về sinh viên, giảng viên, nhân viên từ thông tin cá nhân đến lịch trình giảng dạy. Các dự liệu liên quan đến tài liệu học tập, phòng học, trang thiết bị cũng có thể được truy cập giúp tối ưu hóa việc sử dụng và phân bổ tài nguyên. Các phòng ban và bộ phận khác nhau có thể làm việc hiệu quả hơn khi dữ liệu được liên thông, cải thiện sự nhất quán thông tin và khiến cho việc phối hợp giữa các phòng ban trở lên trơn tru hơn. Ngoài ra, khối lượng dữ liệu đa cấu trúc như văn bản, hình ảnh, video, email, tài liệu học thuật tại môi trường Đại học là rất lớn. Các loại dữ liệu này đem lại giá trị sử dụng lớn, là đầu vào để phân tích các mẫu phức tạp và mối liên hệ giữa các loại dữ liệu khác nhau, từ đó giúp hiểu rõ hơn về các xu hướng và thông tin ẩn nhằm hỗ trợ ra quyết định. Mặc dù vậy, các loại dữ liệu này không được tổ chức theo cách dễ dàng phân loại và truy xuất cũng như nằm rải rác trên nhiều hệ thống gây ra việc khó khăn trong việc thu thập, lưu trữ và quản lý. Do đó, việc nghiên cứu, xây dựng trục liên thông dữ liệu phục vụ lưu trữ, quản lý dữ liệu đa cấu trúc là cấp thiết giúp cải thiện khả năng tích hợp, quản lý và phân tích dữ liệu, từ đó nâng cao hiệu quả hoạt động, hỗ trợ ra quyết định.

2. Tổng quan về vấn đề nghiên cứu

Dữ liệu đa cấu trúc là một khái niệm quan trọng trong lĩnh vực quản lý dữ liệu và phân tích dữ liệu lớn (Big Data). Dữ liệu đa cấu trúc bao gồm ba loại: Dữ liệu có cấu trúc (Structured data), dữ liệu phi cấu trúc (Unstructured data) và Dữ liệu bán cấu trúc (Semi-structured data). Dữ liệu có cấu trúc là dữ liệu tuân theo một định dạng cụ thể và rõ ràng, nó có thể được tìm kiếm và xử lý dễ dàng bởi máy tính. Loại dữ liệu này thường được lưu trữ trong cơ sở dữ liệu quan hệ hoặc bảng tính. Mỗi hàng đại diện cho một bản ghi và mỗi cột đại diện cho một thuộc tính. Ngược lại với dữ liệu có cấu trúc, dữ liệu không có cấu trúc thiếu một cấu trúc được định trước và có thể có nhiều hình thức khác nhau bao gồm văn bản, hình ảnh, âm thanh và video. Bất cứ điều gì nằm giữa các danh mục có cấu trúc và không có cấu trúc được gọi là dữ liệu bán cấu trúc. Nó không được tổ chức một cách rõ ràng như dữ

liệu có cấu trúc nhưng vẫn có một mức độ tổ chức. Loại dữ liệu này thường được tìm thấy trong các định dạng như XML (eXtensible Markup Language) và JSON (JavaScript Object Notation) [3]. Hồ dữ liệu (Data Lake) là một giải pháp lý tưởng để lưu trữ dữ liệu đa cấu trúc. Đây là một kho lưu trữ có khả năng chứa lượng lớn dữ liệu từ nhiều nguồn khác nhau. Sự gia tăng về khối lượng, tốc độ và đa dạng của dữ liệu từ nhiều nguồn khác nhau như thiết bị IoT, mạng xã hội, và dữ liệu web đã tạo ra nhu cầu lưu trữ và xử lý dữ liệu lớn. Hồ dữ liệu cung cấp một giải pháp linh hoạt để quản lý và phân tích dữ liệu lớn này. Đồng thời, sự tiến bộ trong các công nghệ lưu trữ và xử lý dữ liệu cùng với sự phát triển của công nghệ đám mây đã làm tăng tính linh hoạt và khả năng mở rộng của hồ dữ liệu. Các công nghệ như Apache Hadoop, Apache Spark, và cơ sở dữ liệu NoSQL đã trở nên mạnh mẽ hơn, giúp hồ dữ liệu trở thành một nền tảng phân tích dữ liệu hiệu quả, đồng thời giảm bớt gánh nặng về cơ sở hạ tầng và quản lý hệ thống cho các tổ chức [4].

Liên thông dữ liệu được hiểu là khả năng giao tiếp, khả năng thực thi của hai hoặc nhiều hệ thống hoặc thành phần trao đổi thông tin khác nhau và sử dụng thông tin đã được trao đổi. Trong đó có 3 loại liên thông chính bao gồm liên thông cú pháp (syntactic), liên thông cấu trúc (structural), liên thông ngữ nghĩa (semantic). Liên thông cú pháp cho phép hai hoặc nhiều hệ thống có thể giao tiếp và chia sẻ dữ liệu, do đó cho phép các loại hệ thống máy tính, phần mềm khác nhau hoạt động cùng nhau, điều này xảy ra ngay cả khi giao diện hoặc ngôn ngữ không giống nhau; Liên thông cấu trúc xác định định dạng trao đổi dữ liệu, xác định các tiêu chuẩn được sử dụng để định dạng các thông điệp được gửi từ hệ thống này sang hệ thống khác, điều này là cần thiết để người dùng có thể hiểu rõ ràng mục đích của thông tin; Liên thông ngữ nghĩa tập trung nhiều hơn vào ý nghĩa của dữ liệu được trao đổi giữa hai hoặc nhiều hệ thống kết nối và chia sẻ dữ liệu mà mỗi hệ thống hiểu theo cách có ý nghĩa.

Trục liên thông dữ liệu là một nền tảng được thiết kế để liên thông dữ liệu giữa các hệ thống thông tin khác nhau. Việc sử dụng trục liên thông dữ liệu mang lại những lợi ích như giúp các hệ thống thông tin khác nhau giao tiếp và trao đổi dữ

liệu một cách liền mạch, quản lý và giám sát các dịch vụ dữ liệu để đảm bảo hoạt động ổn định, đảm bảo dữ liệu được chuyển đến đúng nơi cần thiết, là trung gian đảm bảo sự tương thích giữa các dịch vụ khác nhau và tạo ra các dịch vụ chung mà nhiều hệ thống có thể sử dụng. X-Road lần đầu tiên được công bố vào năm 2001 bởi cơ quan Hệ thống thông tin của Estonia (Republic Of Estonia Information System Authority - RIA) [5]. Mục tiêu của nền tảng trục liên thông dữ liệu X-Road nhằm đáp ứng đầy đủ ba yêu cầu khi truyền dữ liệu giữa các cơ quan trong chính phủ. Đầu tiên, dữ liệu cần phải dễ truy cập bởi những người có thẩm quyền đã được xác thực để sử dụng. Thứ hai, tính toàn vẹn dữ liệu phải được duy trì, không bên thứ ba nào được thay đổi dữ liệu khi truyền. Thứ ba, dữ liệu phải được giữ bí mật trong khi truyền, phải được bảo vệ khỏi các người dùng chưa được xác thực. Đến năm 2015, Phần Lan chính thức đưa X-Road vào sử dụng. Năm 2017, Phần Lan và Estonia đã thành lập Viện Giải pháp Tương tác Bắc Âu (Nordic Institute for Interoperability Solutions - NIIS) để tiếp tục phát triển lõi X-Road.

Tại Việt Nam, Trục liên thông văn bản quốc gia là một trong những bước đi đầu tiên hướng tới nền tảng tích hợp, chia sẻ, kết nối các hệ thống thông tin, cơ sở dữ liệu của các bộ, ngành, địa phương, giúp cho việc thúc đẩy quá trình chuyển đổi số tại các cơ quan nhà nước nhằm phục vụ người dân, doanh nghiệp được tốt hơn [6]. Đề án trục liên thông văn bản quốc gia nhằm cung cấp giải pháp tổng thể tích hợp, liên thông dữ liệu văn bản từ các hệ thống Quản lý văn bản và điều hành (QLVB&ĐH) của các bộ, ngành, địa phương, là tiền để để xây dựng, phát triển thành nền tảng tích hợp, chia sẻ dữ liệu quốc gia; đổi mới phương thức làm việc của Chính phủ và các cơ quan hành chính các cấp, hướng tới Chính phủ không giấy tờ, tiết kiệm thời gian, chi phí, nâng cao hiệu lực, hiệu quả trong xử lý văn bản, công việc trên môi trường điện tử [7].

Được ứng dụng nhiều tại các cơ quan chính phủ ở các quốc gia bao gồm cả Việt Nam và mang lại rất nhiều ưu điểm có thể kể đến như tiết kiệm thời gian và chi phí. Tuy vậy tại Việt Nam, trục liên thông dữ liệu lại chưa được ứng dụng nhiều

trong lĩnh vực giáo dục, đặc biệt là tại các trường đại học nhằm liên thông dữ liệu và phục vụ lưu trữ, quản lý dữ liệu đa cấu trúc.

3. Mục đích nghiên cứu

Mục đích chính của nghiên cứu này là xây dựng một trục liên thông dữ liệu đóng vai trò như một nền tảng trung gian để kết nối, tích hợp và đồng bộ hóa dữ liệu đa cấu trúc từ nhiều hệ thống khác nhau trong một trường đại học. Việc thiết lập một hạ tầng có khả năng xử lý và tích hợp các định dạng dữ liệu không đồng nhất là vô cùng cần thiết. Trục liên thông dữ liệu sẽ giúp đảm bảo tính tương tác, liên kết và tái sử dụng dữ liệu giữa các hệ thống, đồng thời khắc phục tình trạng phân mảnh thông tin hiện nay.

Thông qua các công cụ phân tích, dữ liệu sẽ được chuyển hóa thành thông tin có giá trị nhằm phục vụ cho các hoạt động quản trị, giảng dạy, nghiên cứu và ra quyết định. Ngoài ra, nghiên cứu cũng chú trọng đến việc trực quan hóa dữ liệu thông qua các biểu đồ, bảng số liệu, đồ thị nhằm tăng tính tương tác và dễ đọc hiểu cho người dùng cuối.

4. Đối tượng và phạm vi nghiên cứu

*** Đối tượng nghiên cứu:**

Nghiên cứu này tập trung vào ba nhóm đối tượng chính có liên quan mật thiết đến quá trình xây dựng và vận hành một trục liên thông dữ liệu phục vụ lưu trữ, quản lý dữ liệu đa cấu trúc trong một trường đại học:

- Nghiên cứu về dữ liệu đa cấu trúc (multistructured data), bao gồm các loại dữ liệu có cấu trúc, dữ liệu bán cấu trúc và dữ liệu phi cấu trúc. Việc hiểu rõ tính chất, đặc điểm và nhu cầu xử lý của từng loại dữ liệu là cơ sở quan trọng để đề xuất giải pháp lưu trữ, tích hợp và phân tích phù hợp.

- Nghiên cứu về trục liên thông dữ liệu X-Road, một trong những nền tảng nổi bật và được áp dụng rộng rãi trong việc kết nối và trao đổi dữ liệu giữa các hệ thống thông tin khác nhau. Nội dung nghiên cứu sẽ làm rõ cấu trúc, cơ chế hoạt động, khả

năng tương tác, bảo mật và khả năng mở rộng của X-Road, từ đó đánh giá mức độ phù hợp khi áp dụng trong bối cảnh liên thông dữ liệu tại một trường đại học.

- Nghiên cứu về mô hình trực quan hóa dữ liệu và sinh báo cáo thống kê nhằm phục vụ cho mục tiêu phân tích dữ liệu được chia sẻ qua trục liên thông. Nội dung này bao gồm việc tìm hiểu các kỹ thuật trực quan hóa hiện đại, các công cụ hỗ trợ báo cáo dữ liệu động và tương tác, cũng như các tiêu chí đánh giá hiệu quả biểu diễn dữ liệu phục vụ cho quản trị và ra quyết định.

*** Phạm vi nghiên cứu:**

Phạm vi của đề tài được xác định rõ ràng để đảm bảo tính khả thi và tập trung trong quá trình triển khai:

- Về nội dung, nghiên cứu sẽ tập trung vào quá trình chia sẻ và liên thông dữ liệu đa cấu trúc giữa các hệ thống trong một trường đại học thông qua trục liên thông X-Road. Các loại dữ liệu trong phạm vi nghiên cứu bao gồm: Dữ liệu có cấu trúc, Dữ liệu bán cấu trúc và Dữ liệu phi cấu trúc.

- Về thời gian và dữ liệu khảo sát, nghiên cứu thu thập dữ liệu đa cấu trúc thông qua trục liên thông X-Road của một trường đại học trong khoảng thời gian một tháng.

5. Phương pháp nghiên cứu

Trong quá trình thực hiện đề tài, để đảm bảo tính khoa học, khách quan và thực tiễn, nhóm nghiên cứu đã áp dụng kết hợp nhiều phương pháp khác nhau, bao gồm cả phương pháp nghiên cứu lý thuyết, thực nghiệm và phân tích đánh giá. Cụ thể như sau:

5.1. Phương pháp nghiên cứu lý thuyết

Đây là phương pháp nền tảng được sử dụng trong giai đoạn đầu của nghiên cứu nhằm xây dựng cơ sở lý luận và xác định hướng đi phù hợp cho toàn bộ đề tài. Nội dung bao gồm:

- Khảo sát, tổng hợp và phân tích các công trình nghiên cứu, tài liệu học thuật, báo cáo kỹ thuật trong và ngoài nước liên quan đến lĩnh vực liên thông dữ liệu và dữ liệu đa cấu trúc.

- Nghiên cứu chuyên sâu về các kiến trúc trục liên thông dữ liệu hiện có, đặc biệt là mô hình X-Road, nhằm làm rõ các khía cạnh kỹ thuật như bảo mật, xác thực, mô hình tổ chức, và khả năng tích hợp với các hệ thống hiện hành.

- Tìm hiểu các phương pháp và công cụ được sử dụng trong việc thu thập, lưu trữ và quản lý dữ liệu đa cấu trúc, từ các giải pháp truyền thống như cơ sở dữ liệu quan hệ đến các công nghệ hiện đại như Data Lake, NoSQL, và hệ thống lưu trữ phân tán.

- Khảo sát các kỹ thuật phân tích dữ liệu và trực quan hóa dữ liệu đa cấu trúc, bao gồm cả các công cụ hỗ trợ như Power BI, Tableau, matplotlib, seaborn, cũng như các thuật toán xử lý và thống kê dữ liệu được sử dụng phổ biến hiện nay.

5.2. Phương pháp nghiên cứu thực nghiệm

Sau khi đã xây dựng được nền tảng lý luận vững chắc, nhóm tiến hành triển khai mô hình nghiên cứu bằng các bước thực nghiệm cụ thể:

- Đề xuất mô hình trục liên thông dữ liệu phù hợp với bối cảnh thực tế tại Việt Nam, đặc biệt trong môi trường giáo dục. Mô hình được thiết kế dựa trên nguyên lý hoạt động của X-Road, đồng thời điều chỉnh để đáp ứng các yêu cầu kỹ thuật, tổ chức và vận hành phù hợp với điều kiện của các cơ sở giáo dục.

- Triển khai mô hình thực nghiệm trên môi trường thử nghiệm với dữ liệu mô phỏng, nhằm kiểm chứng khả năng tích hợp và chia sẻ dữ liệu đa cấu trúc từ nhiều nguồn khác nhau (bao gồm dữ liệu có cấu trúc, bán cấu trúc và phi cấu trúc).

- Xây dựng các kịch bản phân tích và trực quan hóa dữ liệu trên nền tảng dữ liệu thu thập được nhằm kiểm tra khả năng sinh báo cáo thống kê tự động, phục vụ cho nhu cầu quản lý và ra quyết định của các đơn vị sử dụng dữ liệu.

5.3. Phương pháp phân tích và đánh giá mô hình

Đây là bước quan trọng nhằm đo lường tính hiệu quả và thực tiễn của mô hình được đề xuất. Các nội dung chính bao gồm:

- Phân tích hiệu năng của mô hình, thông qua các chỉ số như tốc độ xử lý, khả năng đáp ứng truy cập đồng thời, độ trễ trong quá trình truyền tải và chia sẻ dữ liệu.

- Đánh giá tính khả thi và khả năng mở rộng, bao gồm khả năng tích hợp với các hệ thống sẵn có, mức độ linh hoạt trong việc mở rộng quy mô và phạm vi ứng dụng.

- So sánh mô hình đề xuất với các mô hình hiện hữu, nhằm làm nổi bật những điểm mới, điểm mạnh của giải pháp, cũng như chỉ ra các hạn chế cần được cải tiến trong các nghiên cứu tiếp theo.

- Lấy ý kiến chuyên gia và người dùng thử nghiệm để thu thập phản hồi, từ đó hiệu chỉnh giải pháp cho sát với nhu cầu thực tế hơn.

6. Cấu trúc của đề án

Ngoài phần mở đầu, kết luận và tài liệu tham khảo, phụ lục, đề án gồm có ba chương như sau:

Chương 1: Tổng quan về trục liên thông dữ liệu và dữ liệu đa cấu trúc.

Chương 2: Đề xuất phương án xây dựng trục liên thông dữ liệu.

Chương 3: Thử nghiệm và đánh giá

CHƯƠNG 1. NGHIÊN CỨU TỔNG QUAN VỀ TRỤC LIÊN THÔNG DỮ LIỆU VÀ DỮ LIỆU ĐA CẤU TRÚC

1.1. Liên thông dữ liệu, trục liên thông dữ liệu cùng các phương pháp đăng ký dịch vụ và chia sẻ dịch vụ thông quan trục liên thông dữ liệu

1.1.1. Khái niệm về liên thông dữ liệu

Liên thông dữ liệu là quá trình kết nối và tích hợp các nguồn dữ liệu khác nhau trong một hệ thống hoặc giữa các hệ thống, giúp chia sẻ và đồng bộ hóa thông tin giữa các bộ phận, ứng dụng, hoặc tổ chức. Mục tiêu của việc liên thông dữ liệu là đảm bảo rằng dữ liệu có thể được truy cập, sử dụng, và cập nhật một cách liền mạch và chính xác, mà không gặp phải rào cản giữa các hệ thống khác nhau.

Có 3 loại liên thông chính bao gồm:

Liên thông cú pháp (syntactic): Hai hoặc nhiều hệ thống có thể giao tiếp và chia sẻ dữ liệu, do đó cho phép các loại hệ thống máy tính, phần mềm khác nhau hoạt động cùng nhau, điều này xảy ra ngay cả khi giao diện hoặc ngôn ngữ không giống nhau. Liên thông cấu trúc (structural): Xác định định dạng trao đổi dữ liệu, xác định các tiêu chuẩn được sử dụng để định dạng các thông điệp được gửi từ hệ thống này sang hệ thống khác, điều này là cần thiết để người dùng có thể hiểu rõ ràng mục đích của thông tin. Liên thông ngữ nghĩa (semantic): tập trung nhiều hơn vào ý nghĩa của dữ liệu được trao đổi giữa hai hoặc nhiều hệ thống kết nối và chia sẻ dữ liệu mà mỗi hệ thống hiểu theo cách có ý nghĩa.

Trong một tổ chức, việc thiếu liên kết về dữ liệu giữa các hệ thống thông tin trong một tổ chức dẫn đến sự khó khăn trong việc chia sẻ dữ liệu, các quy định và cơ chế quản lý không đồng bộ. Khi được lưu trữ tách biệt trong các hệ thống khác nhau sẽ khiến dữ liệu trở lên phân mảnh gây ra khó khăn trong việc tìm kiếm và tổng hợp. Theo một nghiên cứu của Forrester, 74% tổ chức cảm thấy việc phân mảnh dữ liệu cản trở sự phối hợp hiệu quả [8] giữa các nhóm trong công ty. Trong một tổ chức mà dữ liệu được lưu trữ một cách độc lập trên nhiều hệ thống như vậy, các thành viên sẽ phải dành nhiều thời gian nhập liệu lặp đi lặp lại cùng một dữ liệu trên nhiều hệ thống. Ngoài ra, việc hỗ trợ ra quyết định cho lãnh đạo của một tổ

chức là rất quan trọng. Theo một khảo sát của Deloitte, 67% các lãnh đạo cho rằng họ đã bỏ lỡ cơ hội quan trọng vì không có dữ liệu kịp thời [9] do không có được bức tranh toàn cảnh để đưa ra quyết định chính xác bởi dữ liệu nằm phân mảnh và không được tích hợp. Điều này dẫn đến việc cần thiết đưa ra một cách thức kết nối về dữ liệu của các hệ thống trong một tổ chức với nhau và có một giải pháp vô cùng hiệu quả đó chính là liên thông dữ liệu thông qua trục liên thông dữ liệu.

1.1.2. Khái niệm về trục liên thông dữ liệu

Trục liên thông dữ liệu là một hệ thống hoặc nền tảng công nghệ được thiết kế để kết nối, tích hợp và đồng bộ hóa các nguồn dữ liệu từ nhiều hệ thống khác nhau trong một tổ chức. Mục tiêu của trục liên thông dữ liệu là tạo ra một điểm truy cập trung tâm để tất cả các dữ liệu có thể được quản lý và chia sẻ một cách đồng bộ và liền mạch. Trục liên thông dữ liệu là công cụ quan trọng giúp các tổ chức kết nối các hệ thống và dữ liệu phân tán, từ đó tạo ra một cơ sở dữ liệu đồng nhất cho việc phân tích và ra quyết định hiệu quả hơn.

Hiện nay, có nhiều loại trục liên thông dữ liệu được sử dụng. Mỗi loại được thiết kế để đáp ứng các nhu cầu khác nhau của các tổ chức trong việc tích hợp, quản lý và chia sẻ dữ liệu. Chúng ta có thể phân loại trục liên thông dữ liệu theo mục đích sử dụng, công nghệ và phương pháp triển khai: Trục liên thông dữ liệu kiểu tích hợp (Data Integration Hub): Là những nền tảng được thiết kế để tích hợp dữ liệu từ các nguồn khác nhau. Các nguồn dữ liệu này có thể là cơ sở dữ liệu, các dịch vụ từ các hệ thống trong tổ chức hoặc các dịch vụ của các bên thứ ba.Trục liên thông dữ liệu theo kiểu dịch vụ (Data as a Service – DaaS): Loại trục liên thông dữ liệu này cung cấp các dịch vụ dữ liệu từ các nguồn khác nhau thông qua các API. Dữ liệu được tập trung và phân phối qua một nền tảng duy nhất, giúp các tổ chức và doanh nghiệp dễ dàng truy cập và sử dụng mà không cần quản lý cơ sở hạ tầng phức tạp. Trục liên thông dữ liệu theo kiểu luồng dữ liệu (Data Stream Integration Hub): Loại trục này được sử dụng trong các môi trường cần xử lý dữ liệu theo thời gian thực, chẳng hạn như trong các hệ thống giám sát, ứng dụng tài chính hoặc các nền tảng IoT. Dữ liệu được liên thông trong các luồng và có thể được phân tích

ngay khi nó được tạo ra. Trục liên thông dữ liệu dựa trên API: Trục liên thông này dựa vào các API để kết nối và chia sẻ dữ liệu giữa các hệ thống và ứng dụng khác nhau. Các API cung cấp một cách thức linh hoạt và dễ dàng để tích hợp dữ liệu mà không cần phải thay đổi cấu trúc của các hệ thống nguồn.

1.1.3. Các phương pháp đăng ký dịch vụ và chia sẻ dịch vụ thông quan trục liên thông dữ liệu

Trong bối cảnh hiện nay, khi các hệ thống thông tin ngày càng đa dạng và phân tán, việc chia sẻ dữ liệu giữa các hệ thống khác nhau đòi hỏi những cơ chế trung gian có khả năng đảm bảo sự tương thích, đồng bộ và hiệu quả trong quá trình truyền tải và xử lý thông tin. Trục liên thông dữ liệu không chỉ đóng vai trò là kênh kết nối mà còn là hạ tầng hỗ trợ việc quản lý dịch vụ, đăng ký và chia sẻ dịch vụ một cách an toàn, linh hoạt và chuẩn hóa. Hai trong số các phương pháp phổ biến và hiệu quả trong việc đăng ký – chia sẻ dịch vụ thông qua trục liên thông dữ liệu bao gồm Data Hubs và EDI (Electronic Data Interchange).

*** Phương pháp thứ nhất: Data Hubs**

Data Hub là một giải pháp trung gian có vai trò như một nút điều phối thông tin giữa các hệ thống khác nhau trong tổ chức. Khác với khái niệm kho dữ liệu tập trung (data warehouse), data hub không lưu trữ toàn bộ dữ liệu một cách cố định mà hoạt động như một trung tâm trao đổi dữ liệu, nơi dữ liệu từ nhiều nguồn khác nhau được kết nối, xử lý và điều hướng đến các hệ thống cần thiết một cách thông minh và có kiểm soát.

Data hub thường được ứng dụng trong các hệ thống phức tạp có nhiều nguồn dữ liệu như: ứng dụng web, thiết bị Internet vạn vật (IoT), các giải pháp phần mềm như dịch vụ (SaaS), hay các nền tảng kinh doanh cốt lõi như hệ thống quản trị quan hệ khách hàng (CRM), hệ thống hoạch định tài nguyên doanh nghiệp (ERP). Nhờ khả năng xử lý luồng dữ liệu trung gian, data hub có thể giảm thiểu việc trùng lặp dữ liệu, tránh các lỗi khi truyền dữ liệu trực tiếp giữa các hệ thống, và từ đó nâng cao hiệu quả tích hợp tổng thể của tổ chức.

Ngoài ra, data hub còn đóng vai trò quan trọng trong việc duy trì tính nhất quán dữ liệu và giảm chi phí vận hành liên quan đến việc quản lý dữ liệu phân tán. Thay vì yêu cầu các hệ thống phải trực tiếp tích hợp với nhau từng cặp (point-to-point), data hub cung cấp một mô hình trung tâm – vệ tinh, nơi mọi hệ thống chỉ cần kết nối một lần tới hub để có thể chia sẻ hoặc truy cập dữ liệu từ nhiều hệ thống khác nhau.

*** Phương pháp thứ hai: EDI (Electronic Data Interchange)**

EDI – Giao tiếp dữ liệu điện tử – là phương pháp chuẩn hóa và tự động hóa quy trình trao đổi dữ liệu giữa các tổ chức, đặc biệt là trong các quan hệ kinh doanh. EDI cho phép các hệ thống khác nhau của các doanh nghiệp truyền tải thông tin, tài liệu như đơn đặt hàng, hóa đơn, phiếu xuất kho… theo định dạng tiêu chuẩn, loại bỏ sự phụ thuộc vào các thao tác thủ công vốn dễ gây ra lỗi, chậm trễ và thiếu nhất quán.

EDI hoạt động dựa trên các giao thức truyền thông đặc biệt như AS2 (Applicability Statement 2), FTP hoặc các kết nối bảo mật khác, đảm bảo dữ liệu được truyền đi an toàn và đúng cấu trúc định sẵn. Để triển khai EDI hiệu quả, các doanh nghiệp cần thực hiện một số bước cơ bản bao gồm: xác định nhu cầu trao đổi dữ liệu giữa các đối tác kinh doanh, lựa chọn phần mềm EDI phù hợp với mô hình hoạt động, thiết lập các định dạng và giao thức trao đổi, cũng như kiểm thử khả năng tương thích giữa các hệ thống liên quan.

Thông qua EDI, quá trình trao đổi thông tin trở nên nhanh chóng, chính xác và tin cậy hơn, từ đó giảm thiểu chi phí và thời gian cho các thủ tục hành chính và nghiệp vụ. Đây là phương pháp đặc biệt hữu ích trong môi trường có tần suất giao dịch cao, như trong lĩnh vực thương mại điện tử, logistics, hoặc chuỗi cung ứng đa tầng.

Tổng kết lại, cả hai phương pháp – Data Hub và EDI – đều hướng đến mục tiêu tối ưu hóa quy trình tích hợp và chia sẻ dữ liệu giữa các hệ thống thông tin khác nhau, đặc biệt là trong môi trường số hóa ngày càng mở rộng. Khi được kết hợp trong kiến trúc của một trục liên thông dữ liệu, chúng có thể hỗ trợ rất hiệu quả cho

việc xây dựng nền tảng trao đổi thông tin linh hoạt, minh bạch, đồng thời đảm bảo hiệu suất và tính bảo mật cao trong quá trình truyền tải dữ liệu.

1.2. Dữ liệu đa cấu trúc, các loại dữ liệu đa cấu trúc và cách thức lưu trữ, quản lý, phân tích dữ liệu đa cấu trúc

1.2.1. Dữ liệu đa cấu trúc là gì?

Dữ liệu đa cấu trúc (*Multistructured data*) là khái niệm dùng để chỉ tập hợp dữ liệu bao gồm nhiều loại hình khác nhau, trong đó kết hợp cả ba dạng chính: **dữ liệu có cấu trúc**, **dữ liệu bán cấu trúc**, và **dữ liệu phi cấu trúc**. Mỗi loại dữ liệu này mang những đặc điểm riêng biệt về mức độ tổ chức, phương thức lưu trữ, cũng như yêu cầu xử lý và khai thác. Chính sự đa dạng trong cấu trúc đã khiến dữ liệu đa cấu trúc trở thành một trong những thách thức nhưng cũng là tài sản có giá trị rất lớn đối với các tổ chức hiện đại.

Trong bối cảnh chuyển đổi số và dữ liệu hóa toàn diện, đặc biệt trong lĩnh vực giáo dục, dữ liệu đa cấu trúc đóng vai trò then chốt trong việc xây dựng hệ sinh thái dữ liệu giáo dục thông minh và toàn diện. Đối với một trường đại học, việc khai thác đồng thời nhiều loại dữ liệu từ các nguồn khác nhau không chỉ giúp tối ưu hóa công tác giảng dạy, nghiên cứu và quản lý mà còn mở rộng khả năng cá nhân hóa trải nghiệm học tập của sinh viên.

Chẳng hạn, khi kết hợp dữ liệu có cấu trúc như bảng điểm, lịch học, hồ sơ sinh viên với dữ liệu phi cấu trúc như video bài giảng, khảo sát phản hồi hay dữ liệu từ hệ thống quản lý học tập (LMS), nhà trường có thể xây dựng một bức tranh toàn diện về nhu cầu, hành vi học tập và hiệu suất của sinh viên. Từ đó đưa ra các quyết định mang tính hỗ trợ kịp thời, điều chỉnh nội dung chương trình đào tạo phù hợp với năng lực từng người học – đúng với tinh thần "Lấy học sinh, sinh viên làm trung tâm" như Thủ tướng Phạm Minh Chính đã nhấn mạnh tại hội nghị tổng kết năm học 2023–2024 và triển khai nhiệm vụ năm học 2024–2025 do Bộ Giáo dục và Đào tạo tổ chức.

Bên cạnh đó, khi được tổ chức và quản lý hiệu quả, dữ liệu đa cấu trúc chính là nguồn tài nguyên chiến lược cho nghiên cứu khoa học và phân tích chuyên sâu.

Với khối lượng và độ đa dạng cao, dữ liệu này giúp các nhà nghiên cứu khám phá những xu hướng tiềm ẩn, mối quan hệ phức tạp hoặc các bất thường khó phát hiện nếu chỉ phân tích dữ liệu đơn lẻ từng loại. Đặc biệt, trong thời đại của trí tuệ nhân tạo (AI), dữ liệu đa cấu trúc cung cấp đầu vào phong phú cho các thuật toán học máy và học sâu, mở ra khả năng ứng dụng các mô hình phân tích nâng cao trong dự đoán hành vi, tối ưu hóa quy trình và hỗ trợ ra quyết định tự động.

Tóm lại, dữ liệu đa cấu trúc không chỉ là xu thế tất yếu trong công cuộc số hóa thông tin mà còn là một nền tảng thiết yếu để thúc đẩy cải tiến trong giáo dục, nghiên cứu và quản trị tổ chức, đặc biệt khi kết hợp cùng các hệ thống liên thông và phân tích hiện đại như trục dữ liệu và nền tảng dữ liệu lớn (Big Data Platform).

1.2.2. Các loại dữ liệu đa cấu trúc

Dữ liệu đa cấu trúc bao gồm ba loại dữ liệu chính, mỗi loại mang đặc điểm riêng về cách tổ chức, định dạng, cũng như phương thức lưu trữ và xử lý. Việc phân loại rõ ràng từng loại dữ liệu giúp định hướng đúng đắn trong việc xây dựng hạ tầng công nghệ phù hợp cho lưu trữ, quản lý và khai thác hiệu quả nguồn dữ liệu này trong các hệ thống thông tin hiện đại, đặc biệt trong lĩnh vực giáo dục và nghiên cứu.

Dữ liệu có cấu trúc (Structured Data): Đây là loại dữ liệu được tổ chức theo định dạng cố định, thường được lưu trữ trong các bảng với hàng (record) và cột (field) rõ ràng. Dữ liệu có cấu trúc được quản lý bởi các hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management Systems – RDBMS) như MySQL, PostgreSQL, SQL Server hoặc Oracle, cho phép truy vấn và thao tác hiệu quả thông qua các ngôn ngữ như SQL (Structured Query Language).

Ví dụ điển hình của loại dữ liệu này trong môi trường giáo dục bao gồm: hồ sơ sinh viên với các thuộc tính như mã sinh viên, họ tên, ngày sinh, ngành học, năm nhập học. Danh sách điểm của từng học phần theo từng kỳ học. Thông tin thời khóa biểu, phòng học, lịch thi, kế hoạch giảng dạy.

Ưu điểm nổi bật của dữ liệu có cấu trúc là khả năng truy xuất nhanh chóng, độ chính xác cao, và dễ dàng tích hợp vào các báo cáo, hệ thống phân tích.

Dữ liệu bán cấu trúc (Semi-structured Data): Dữ liệu bán cấu trúc là loại dữ liệu không hoàn toàn tuân theo mô hình dữ liệu quan hệ truyền thống nhưng vẫn chứa các thành phần giúp xác định mối quan hệ logic giữa các đối tượng dữ liệu thông qua các thẻ, cặp khóa – giá trị hoặc siêu dữ liệu (metadata). Loại dữ liệu này linh hoạt hơn dữ liệu có cấu trúc nhưng vẫn đủ thông tin để hệ thống có thể hiểu được cấu trúc cơ bản mà không cần phải tiền xử lý phức tạp.

Một số ví dụ phổ biến là các tệp tin định dạng XML (eXtensible Markup Language) và JSON (JavaScript Object Notation), được sử dụng rộng rãi trong giao tiếp API và lưu trữ cấu hình hệ thống. Nhật ký hệ thống (log files) với cấu trúc ghi nhận thời gian, hành động, người dùng liên quan. Email với các trường như người gửi, người nhận, tiêu đề (header) và nội dung (body).

Dữ liệu bán cấu trúc thường là cầu nối giữa dữ liệu có cấu trúc và phi cấu trúc, là định dạng lý tưởng trong các hệ thống hiện đại cần xử lý dữ liệu đến từ nhiều nguồn linh hoạt nhưng vẫn cần duy trì một mức độ tổ chức nhất định.

Dữ liệu phi cấu trúc (Unstructured Data): Đây là loại dữ liệu không có một mô hình tổ chức hoặc định dạng xác định, khiến việc xử lý và phân tích trở nên phức tạp hơn nếu không có sự hỗ trợ của các công cụ chuyên dụng. Dữ liệu phi cấu trúc chiếm tỷ trọng rất lớn trong tổng thể dữ liệu phát sinh hàng ngày trong các tổ chức, đặc biệt trong môi trường giáo dục, truyền thông và mạng xã hội.

Các ví dụ điển hình bao gồm: Tài liệu văn bản ở định dạng DOC, PDF, bài luận, giáo trình điện tử. File hình ảnh (JPG, PNG), video bài giảng (MP4), âm thanh (MP3) được sử dụng trong dạy và học. Các bài đăng trên mạng xã hội, khảo sát mở dạng văn bản tự do, phản hồi sinh viên.

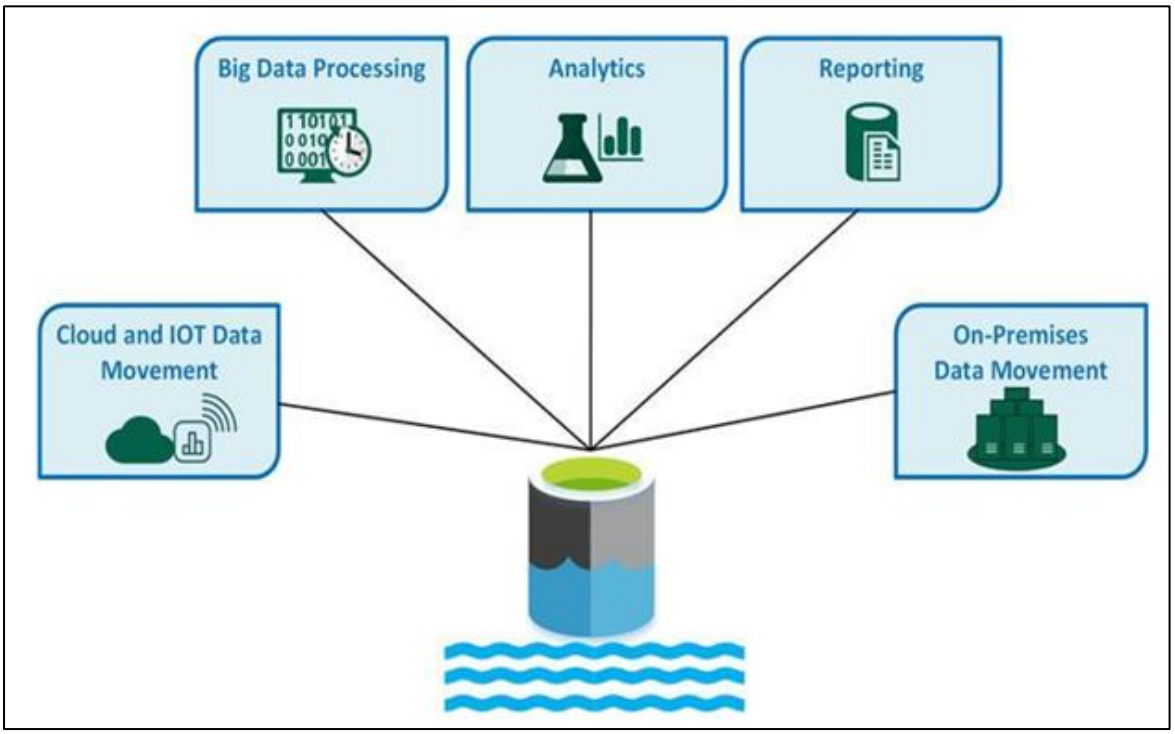
Mặc dù khó xử lý bằng các công cụ truyền thống, dữ liệu phi cấu trúc lại ẩn chứa nhiều thông tin giá trị nếu được xử lý đúng cách thông qua các kỹ thuật hiện đại như xử lý ngôn ngữ tự nhiên (NLP), thị giác máy tính (Computer Vision), hoặc học sâu (Deep Learning). Việc khai thác hiệu quả dữ liệu phi cấu trúc đang là trọng tâm trong các hệ thống hỗ trợ ra quyết định, hệ thống AI và các nền tảng học tập thông minh.

1.2.3. Cách lưu trữ, quản lý, phân tích dữ liệu đa cấu trúc

Dữ liệu đa cấu trúc với tính chất phong phú về định dạng và nguồn gốc, từ dữ liệu quan hệ truyền thống đến các tài liệu văn bản tự do, hình ảnh, âm thanh hay tín hiệu cảm biến, mang lại tiềm năng to lớn trong việc phân tích chuyên sâu, hỗ trợ ra quyết định và nâng cao hiệu quả vận hành. Tuy nhiên, chính sự đa dạng và không đồng nhất về cấu trúc đã đặt ra những thách thức đáng kể trong việc xây dựng một hạ tầng công nghệ có khả năng lưu trữ, quản lý và khai thác hiệu quả loại dữ liệu này.

1.2.3.1. Lưu trữ dữ liệu đa cấu trúc

Một trong những giải pháp phổ biến và hiệu quả nhất hiện nay cho việc lưu trữ dữ liệu đa cấu trúc là sử dụng hồ dữ liệu (Data Lake). Đây là mô hình lưu trữ tập trung được thiết kế để tiếp nhận dữ liệu từ nhiều nguồn khác nhau, ở nhiều định dạng khác nhau mà không yêu cầu phải chuyển đổi ngay lập tức về một cấu trúc chuẩn. Điểm mạnh nổi bật của mô hình này là khả năng lưu trữ dữ liệu ở định dạng gốc, giữ lại toàn bộ thông tin cần thiết để xử lý về sau, đồng thời không bị giới hạn bởi kích thước hay loại tệp.



Hình 1.1. Hồ dữ liệu (Data Lake)

Không giống như kho dữ liệu truyền thống (Data Warehouse) vốn yêu cầu dữ liệu phải được chuyển đổi và tổ chức chặt chẽ ngay từ đầu, hồ dữ liệu cho phép doanh nghiệp linh hoạt hơn trong việc tích hợp và mở rộng quy mô lưu trữ, đặc biệt khi cần thu thập dữ liệu theo thời gian thực từ nhiều hệ thống phân tán như IoT, LMS, hệ thống ERP hay các nền tảng học tập trực tuyến.

1.2.3.2. Quản lý dữ liệu đa cấu trúc

Để khai thác hiệu quả dữ liệu được lưu trữ, cần có một hệ thống quản lý dữ liệu đa tầng, trong đó metadata (siêu dữ liệu) đóng vai trò đặc biệt quan trọng. Metadata là tập hợp các thông tin mô tả về dữ liệu như nguồn gốc, thời gian tạo, loại định dạng, quyền truy cập và cách sử dụng. Metadata giúp tổ chức hiểu rõ dữ liệu mình đang sở hữu, từ đó hỗ trợ truy xuất nhanh chóng, kiểm soát chất lượng dữ liệu và đảm bảo an toàn thông tin.

Các hệ thống như Metadata Management System hoặc Data Catalog có thể được sử dụng để tổ chức và tra cứu metadata, hỗ trợ người dùng xác định, định vị và đánh giá dữ liệu có sẵn. Đồng thời, để tăng khả năng truy cập và duy trì hiệu suất hoạt động, dữ liệu đa cấu trúc cũng cần được chuyển đổi, làm sạch, phân loại và nén theo chủ đề, thời gian hoặc loại hình nhằm phục vụ các mục đích sử dụng khác nhau. Ví dụ, hệ thống có thể sắp xếp dữ liệu thành các thư mục: “Tài liệu học tập”, “Video bài giảng”, “Phản hồi sinh viên”,... Điều này giúp tăng tốc độ truy cập và nâng cao khả năng tái sử dụng dữ liệu trong tương lai.

Ngoài ra, bảo mật dữ liệu là một thành phần không thể thiếu trong toàn bộ quy trình quản lý. Việc phân quyền truy cập chi tiết, mã hóa dữ liệu trong quá trình lưu trữ và truyền tải, cũng như tuân thủ các quy định về bảo vệ thông tin cá nhân là điều bắt buộc, đặc biệt với các loại dữ liệu nhạy cảm như hồ sơ sinh viên, kết quả học tập hay phản hồi khảo sát.

1.2.3.3. Phân tích dữ liệu đa cấu trúc

Sau khi dữ liệu đã được thu thập, tổ chức và lưu trữ hiệu quả, bước tiếp theo là phân tích để tạo ra giá trị. Việc phân tích dữ liệu đa cấu trúc đòi hỏi sự kết hợp linh hoạt giữa nhiều kỹ thuật khác nhau tùy vào đặc điểm của từng loại dữ liệu:

- Phân tích thống kê (Statistical Analysis): Áp dụng cho dữ liệu có cấu trúc, cho phép thực hiện các phép đo cơ bản như trung bình, phương sai, phân phối, và kiểm định giả thuyết, thường được sử dụng trong đánh giá kết quả học tập, phân tích điểm số hoặc khảo sát hành vi sinh viên.

- Học máy (Machine Learning): Học máy giúp dự đoán, phân nhóm và hỗ trợ ra quyết định dựa trên dữ liệu giáo dục [10]. Đây là công cụ hữu ích khi xử lý dữ liệu đa cấu trúc ở quy mô lớn như dự báo nguy cơ bỏ học, gợi ý khóa học phù hợp, hoặc phân tích mối liên hệ giữa các yếu tố giảng dạy và kết quả học tập.

- Phân tích văn bản (Text Mining): Sử dụng các kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) để khai thác thông tin từ dữ liệu văn bản như phản hồi sinh viên, email, bài luận hoặc bài viết trên mạng xã hội. Các mô hình này giúp phát hiện từ khóa, phân tích cảm xúc, và đánh giá chất lượng dịch vụ giáo dục thông qua nội dung phi cấu trúc.

- Phân tích hình ảnh và video: Áp dụng các mô hình học sâu (Deep Learning) để nhận diện khuôn mặt, trích xuất nội dung từ video bài giảng hoặc phân tích hình ảnh thực hành thí nghiệm. Deep learning đang được ứng dụng vào phân tích hình ảnh và video trong giáo dục thông minh [11].

- Phân tích dữ liệu từ cảm biến và thiết bị IoT: Các thiết bị như máy đo thân nhiệt, camera lớp học thông minh, thiết bị quét điểm danh,... tạo ra dữ liệu thời gian thực. Việc phân tích dữ liệu từ các nguồn này giúp xây dựng hệ thống theo dõi tự động, phát hiện bất thường và tối ưu hóa vận hành cơ sở vật chất trong môi trường giáo dục hiện đại.

1.3. Phương pháp chia sẻ dữ liệu đa cấu trúc thông qua trục liên thông dữ liệu

Phương pháp chia sẻ dữ liệu thông qua trục liên thông dữ liệu đóng vai trò trung tâm trong việc tối ưu hóa quá trình trao đổi thông tin giữa các hệ thống độc lập và phân tán. Trong bối cảnh các tổ chức hiện đại đặc biệt là trong lĩnh vực giáo dục, y tế, chính phủ điện tử đang sở hữu nhiều hệ thống thông tin với cấu trúc dữ liệu khác nhau, việc thiết lập một trục liên thông có khả năng lưu trữ quản lý dữ liệu

đa cấu trúc là yêu cầu thiết yếu để hướng tới tích hợp dữ liệu toàn diện và hỗ trợ ra quyết định dựa trên dữ liệu.

Một nền tảng trục liên thông dữ liệu thường được tổ chức thành nhiều lớp kiến trúc chính, đảm nhận các chức năng riêng biệt nhưng phối hợp chặt chẽ với nhau để đảm bảo việc chia sẻ dữ liệu diễn ra một cách mượt mà, an toàn và hiệu quả:

1.3.1. Lớp giao tiếp (Interface Layer)

Đây là điểm đầu vào và đầu ra của dữ liệu, nơi các hệ thống gửi và nhận thông tin thông qua các giao thức và định dạng chuẩn hóa. Lớp giao tiếp giúp các hệ thống không đồng nhất về công nghệ hoặc nền tảng có thể tương tác với nhau một cách linh hoạt.

Một số giao thức và định dạng thường được sử dụng trong lớp giao tiếp bao gồm:

- RESTful API / SOAP: Phục vụ truyền nhận dữ liệu theo mô hình client-server, hỗ trợ định dạng JSON hoặc XML.
- FTP / SFTP / AS2: Sử dụng trong chia sẻ tệp hoặc các batch dữ liệu lớn.
- MQTT / AMQP / Kafka: Hỗ trợ giao tiếp theo thời gian thực, thường được dùng trong ứng dụng IoT hoặc truyền sự kiện liên tục.
- Webhooks: Cho phép các hệ thống nhận thông báo ngay khi có sự kiện xảy ra.

Lớp này cũng thường tích hợp các chức năng xác thực (Authentication), phân quyền truy cập (Authorization), và mã hóa dữ liệu (Encryption) nhằm đảm bảo tính bảo mật và toàn vẹn thông tin trong quá trình truyền tải.

1.3.2. Lớp chuyển đổi dữ liệu (Transformation Layer)

Sau khi dữ liệu được tiếp nhận qua lớp giao tiếp, nó cần được chuẩn hóa để đồng nhất định dạng, cấu trúc và ngữ nghĩa giữa các hệ thống. Lớp này xử lý các tác vụ như: chuyển đổi định dạng (ví dụ: từ XML sang JSON), chuẩn hóa kiểu dữ liệu (ví dụ: định dạng ngày tháng, mã quốc gia), Ánh xạ dữ liệu giữa các mô hình dữ liệu khác nhau, Làm sạch và loại bỏ dữ liệu trùng lặp. Việc xử lý đúng tại lớp

này là điều kiện tiên quyết để các hệ thống có thể hiểu và sử dụng dữ liệu lẫn nhau một cách chính xác.

1.3.3. Lớp đồng bộ và phân phối (Synchronization & Routing Layer)

Lớp này đảm nhiệm việc điều phối luồng dữ liệu đến đúng hệ thống đích theo lịch trình hoặc sự kiện kích hoạt. Các chức năng bao gồm đồng bộ theo thời gian thực (real-time synchronization) hoặc định kỳ (batch mode), điều phối dữ liệu đến nhiều hệ thống cùng lúc theo các quy tắc routing, đảm bảo không xảy ra mất mát hoặc trùng lặp dữ liệu trong quá trình truyền tải.

1.3.4. Lớp quản trị và giám sát (Governance & Monitoring Layer)

Đây là lớp cung cấp công cụ giám sát, ghi log, cảnh báo và phân tích hiệu suất của toàn bộ hệ thống liên thông dữ liệu. Nó hỗ trợ theo dõi trạng thái các dòng dữ liệu, kiểm tra lỗi và thực hiện xử lý ngoại lệ (exception handling), quản lý metadata liên quan đến các dịch vụ chia sẻ, ghi nhận nhật ký truy cập phục vụ truy vết và kiểm tra tuân thủ.

1.4. Phương pháp tổng hợp, thống kê, trực quan hóa dữ liệu

Trong bối cảnh chuyển đổi số giáo dục đại học, việc xây dựng một hệ thống xử lý và phân tích dữ liệu hiệu quả là điều kiện tiên quyết để nâng cao chất lượng quản lý và ra quyết định. Đề tài này triển khai phương pháp tổng hợp, thống kê và trực quan hóa dữ liệu thông qua quy trình tự động hóa, nhằm đảm bảo tính chính xác, kịp thời và khả năng mở rộng của hệ thống.

1.4.1. Tổng hợp dữ liệu

Tổng hợp dữ liệu (data aggregation) là bước đầu tiên trong quy trình xử lý dữ liệu, cho phép kết nối và kết hợp các tập dữ liệu có cấu trúc và bán cấu trúc từ nhiều nguồn khác nhau như hệ thống quản lý đào tạo (LMS, SIS), tệp CSV, Excel hoặc dữ liệu thu thập qua biểu mẫu điện tử. Trong nghiên cứu này, dữ liệu được tích hợp định kỳ thông qua hệ thống pipeline nhằm đảm bảo tính cập nhật và độ tin cậy. Theo Fan & Bifet (2013), việc tích hợp dữ liệu từ nhiều nguồn một cách liên tục giúp xây dựng nền tảng dữ liệu phong phú, phục vụ tốt hơn cho phân tích học thuật và vận hành tổ chức [12].

1.4.2. Thống kê và phân tích dữ liệu

Sau khi dữ liệu được tổng hợp, quá trình thống kê mô tả (descriptive statistics) được thực hiện nhằm tóm tắt và biểu diễn các đặc trưng cơ bản của dữ liệu như: số lượng sinh viên theo năm, tỷ lệ tốt nghiệp, điểm trung bình theo ngành, phân phối giới tính,… Các phương pháp như tính trung bình, phương sai, độ lệch chuẩn, tỉ lệ phần trăm,… được áp dụng để mô tả hiện tượng một cách định lượng. Đối với một số bài toán cụ thể, các chỉ số thống kê suy diễn (inferential statistics) như kiểm định giả thuyết (t-test, chi-squared test) có thể được sử dụng để đưa ra kết luận từ mẫu dữ liệu về tổng thể.

Các phương pháp thống kê này không chỉ giúp phát hiện các xu hướng (trends), mà còn cung cấp bằng chứng định lượng để hỗ trợ quyết định quản lý. Như được nêu trong tài liệu của IBM về phân tích dữ liệu trong giáo dục, "phân tích thống kê giúp các cơ sở giáo dục hiểu sâu hơn về hiệu suất của người học, dự báo kết quả và cải thiện chất lượng đào tạo" [13].

1.4.3. Trực quan hóa dữ liệu

Cuối cùng, dữ liệu đã được xử lý và phân tích được trình bày thông qua các công cụ trực quan hóa như biểu đồ cột, biểu đồ tròn, bản đồ nhiệt, biểu đồ phân bố và dashboard tổng hợp. Việc trực quan hóa giúp đơn giản hóa các phân tích phức tạp, cho phép người dùng phát hiện xu hướng, so sánh các chỉ số và đưa ra kết luận nhanh chóng. Các nghiên cứu đã chỉ ra rằng hình thức trình bày trực quan có tác động mạnh đến việc ra quyết định trong quản lý giáo dục. Theo McCandless (2012), "trực quan hóa dữ liệu không chỉ là công cụ thẩm mỹ mà còn là ngôn ngữ mạnh mẽ giúp con người hiểu và tương tác với thông tin" [14].

Việc áp dụng đồng bộ ba phương pháp trên – tổng hợp, thống kê và trực quan hóa – tạo nên một quy trình khép kín, tự động và có khả năng mở rộng, phục vụ cho việc xây dựng hạ tầng dữ liệu số trong giáo dục đại học. Mô hình này hỗ trợ tổ chức trong việc quản trị thông tin, theo dõi hiệu suất đào tạo, và ra quyết định dựa trên dữ liệu (data-driven decision-making).

1.5. Kết luận

Trong chương này, chúng ta đã đi sâu làm rõ những khái niệm nền tảng và các cơ sở lý luận liên quan đến dữ liệu đa cấu trúc và trục liên thông dữ liệu – hai yếu tố then chốt trong việc xây dựng hệ thống quản lý và khai thác dữ liệu hiện đại. Dữ liệu đa cấu trúc với sự kết hợp giữa dữ liệu có cấu trúc, bán cấu trúc và phi cấu trúc không chỉ phản ánh tính đa dạng của thông tin trong thế giới số ngày nay, mà còn mở ra tiềm năng to lớn cho các hoạt động phân tích, dự báo và hỗ trợ ra quyết định, đặc biệt trong các lĩnh vực như giáo dục, y tế, chính phủ điện tử và công nghiệp 4.0.

Việc lưu trữ, quản lý và phân tích dữ liệu đa cấu trúc đòi hỏi phải có các nền tảng hạ tầng phù hợp như hồ dữ liệu (Data Lake), hệ thống quản lý siêu dữ liệu (metadata management), cùng với các phương pháp phân tích hiện đại từ thống kê mô tả, suy luận đến học máy và trực quan hóa dữ liệu. Trục liên thông dữ liệu đóng vai trò như một cầu nối trung gian quan trọng, giúp các hệ thống khác nhau – dù khác biệt về công nghệ hay định dạng – có thể chia sẻ và tương tác dữ liệu một cách hiệu quả, an toàn và có kiểm soát. Các lớp kiến trúc trong một nền tảng trục liên thông, từ lớp giao tiếp đến lớp chuyển đổi và đồng bộ, tạo nên một hệ sinh thái dữ liệu thống nhất, hỗ trợ mạnh mẽ cho công tác tổng hợp, phân tích và báo cáo dữ liệu.

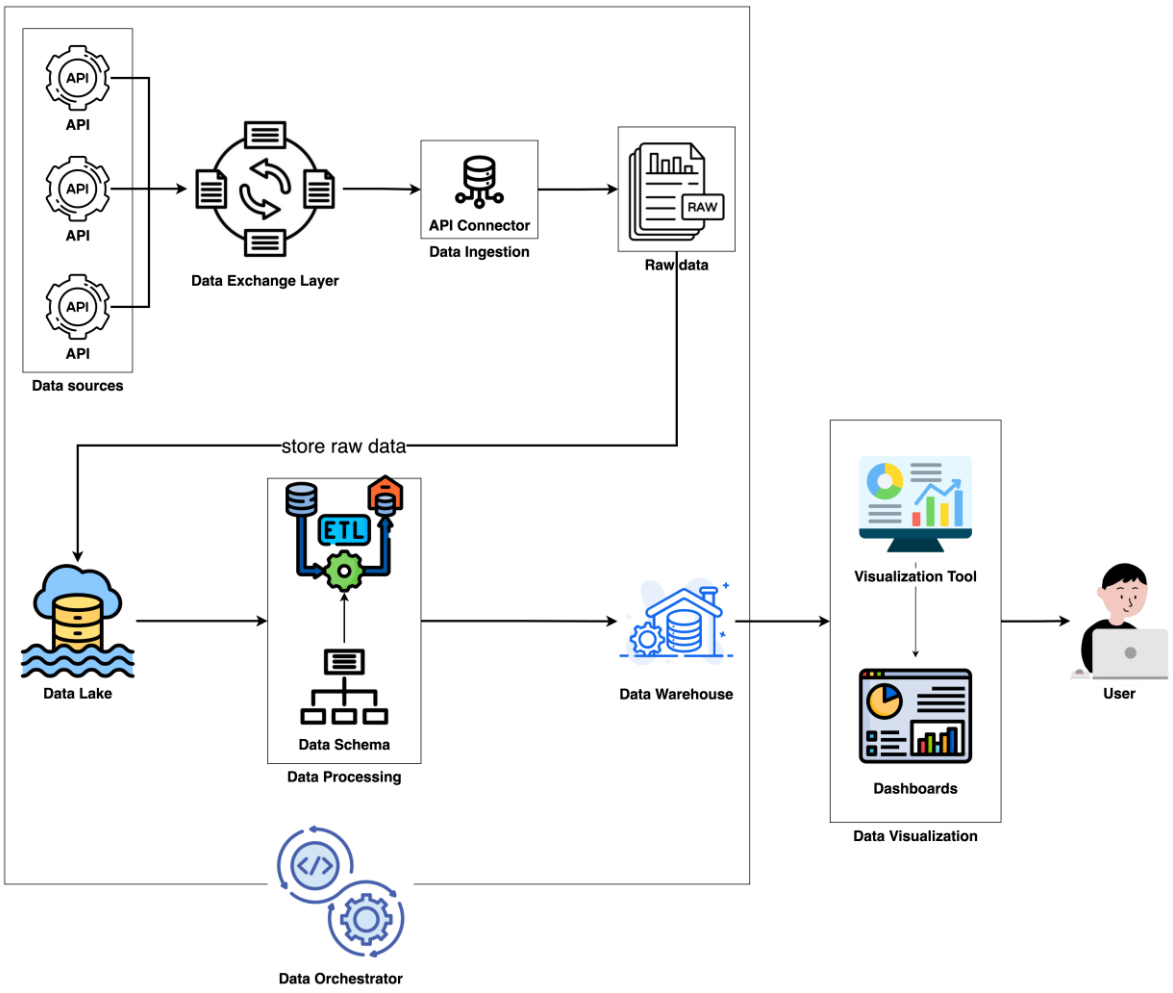
Những nội dung được trình bày trong chương này sẽ là cơ sở quan trọng để bước sang chương tiếp theo, nơi tập trung vào việc thiết kế, triển khai và đánh giá mô hình trục liên thông dữ liệu trong thực tiễn.

CHƯƠNG 2. ĐỀ XUẤT PHƯƠNG ÁN XÂY DỰNG TRỤC LIÊN THÔNG DỮ LIỆU

2.1. Trục liên thông dữ liệu X-Road phục vụ trao đổi dữ liệu đa cấu trúc giữa các bên

2.1.1. Đề xuất mô hình tổng quan hệ thống (High-level design)

Hiện trạng quản lý dữ liệu tại một trường đại học ví dụ như tại Học viện Công nghệ Bưu Chính Viễn Thông đang đối mặt với nhiều thách thức như tình trạng phân mảnh dữ liệu khi thông tin về một sinh viên nằm rải rác trên nhiều hệ thống khác nhau mà không có cơ chế đồng bộ tự động, dẫn đến dữ liệu không nhất quán, hạn chế trong phân tích và báo cáo khi việc tạo báo cáo tổng hợp mất nhiều thời gian, không có dashboard tổng quan và khả năng báo cáo theo thời gian thực cùng với các vấn đề bảo mật và tuân thủ khi khó kiểm soát quyền truy cập, thiếu log tập trung. Những thực trạng này cho thấy nhu cầu cần xây dựng một nền tảng tích hợp dữ liệu trung tâm với trục liên thông dữ liệu, hồ dữ liệu (Data Lake), hệ thống dashboard trực quan hóa dữ liệu hỗ trợ ra quyết định dựa trên dữ liệu và nâng cao hiệu quả quản trị đại học trong kỷ nguyên số.



Hình 2.1. Mô hình tổng quan hệ thống (High level design)

Qua quá trình tìm hiểu và nghiên cứu về trục liên thông dữ liệu, lưu trữ, trực quan hóa dữ liệu. Học viên đề xuất một mô hình trục liên thông dữ liệu mới xử lý các nguồn dữ liệu đa cấu trúc, được chia thành ba nhóm chính: cơ sở dữ liệu (database), giao diện lập trình ứng dụng (API), và dữ liệu tổng hợp từ các tệp (file-based data). Sau khi kết nối thành công với các nguồn dữ liệu, dữ liệu được đưa vào trục liên thông dữ liệu. Nó đóng vai trò như một trung tâm định tuyến và kiểm soát lưu lượng dữ liệu, đồng thời tích hợp khả năng giám sát, xác thực và xử lý lỗi khi có sự cố trong quá trình truyền nhận. Dữ liệu đi qua trục liên thông được tiếp tục đưa vào giai đoạn thu nhận (Data Ingestion), nơi các kết nối được tái sử dụng để chuyển thành dữ liệu thô (raw data) vào hệ thống lưu trữ. Quá trình này đảm bảo dữ liệu được đưa vào nhanh chóng và có thể lưu trữ một lượng dữ liệu lớn, với độ trễ thấp

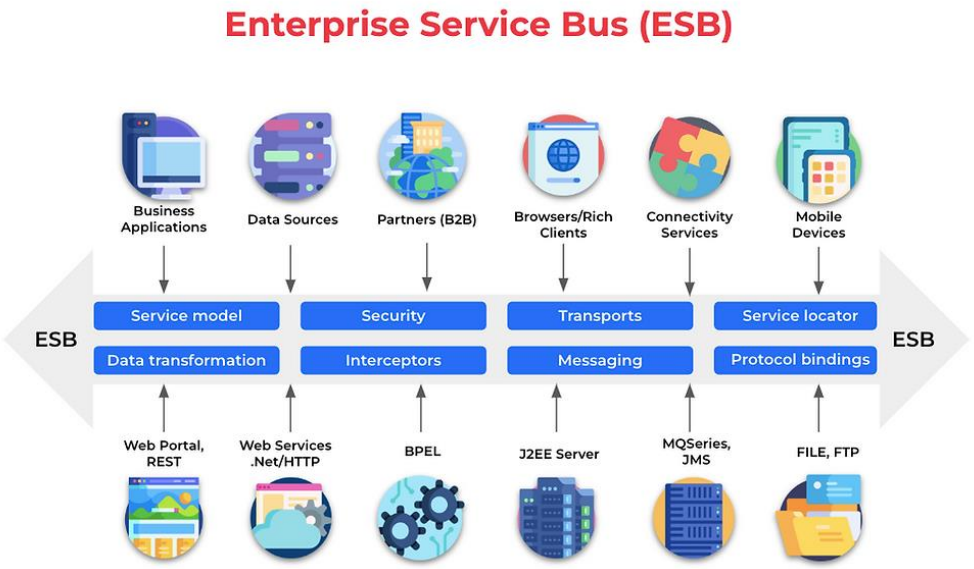
và khả năng xử lý dữ liệu theo thời gian thực nếu cần thiết. Dữ liệu sau khi được thu nhận sẽ được lưu trữ tạm thời trong một hồ dữ liệu (Data Lake). Đây là nơi dữ liệu được tích trữ với mục tiêu duy trì khả năng tái sử dụng tối đa trong các bước xử lý sau này. Sau đó, dữ liệu từ Data Lake được đưa vào quy trình xử lý dữ liệu (Data Processing), nơi thực hiện các thao tác ETL (Extract – Transform – Load). Dưới sự điều phối của công cụ Data Orchestrator, dữ liệu sẽ được trích xuất, làm sạch, chuyển đổi theo lược đồ dữ liệu đã định nghĩa (Data Schema), và cuối cùng được nạp vào kho dữ liệu tập trung (Data Warehouse). Tại đây, dữ liệu đã sẵn sàng để được khai thác bởi các công cụ phân tích và ứng dụng khác nhau. Sau đó, dữ liệu được chuyển hóa thành các con số hoặc biểu đồ trực quan, giúp người dùng theo dõi các chỉ số quan trọng theo thời gian thực.

Mô hình này có thể được áp dụng cho việc liên thông dữ liệu trong một trường đại học, nơi mà các nguồn dữ liệu từ tuyển sinh, học tập của sinh viên, nghiên cứu khoa học, … là các nguồn dữ liệu đa cấu trúc có thể thu thập để lưu trữ, trực quan hóa phục vụ nhiều mục đích khác nhau. Sau đây học viên tiến hành khảo sát và đề xuất các công nghệ trục liên thông dữ liệu, thu thập và biến đổi dữ liệu, lưu trữ và trực quan hóa dữ liệu để đáp ứng yêu cầu của đề án.

2.1.2. Khảo sát các công nghệ trục liên thông dữ liệu

Để triển khai trục liên thông dữ liệu, một số công nghệ và mô hình tích hợp phổ biến đã được khảo sát:

Enterprise Service Bus (ESB):

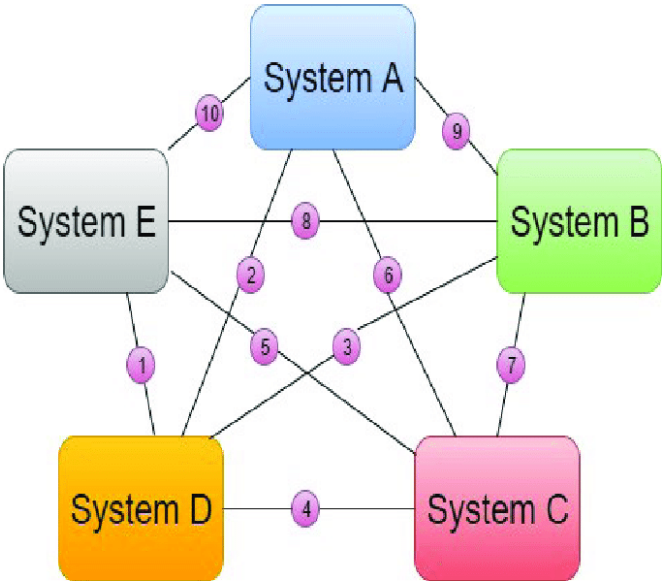


Hình 2.2. Kiến trúc của ESB

ESB là một kiến trúc tích hợp dựa trên mô hình trung gian, cho phép các ứng dụng giao tiếp với nhau thông qua một bus dữ liệu trung tâm [15]. Nó có ưu điểm là Linh hoạt, hỗ trợ tích hợp nhiều hệ thống với giao thức khác nhau, khả năng mở rộng tốt, quản lý tập trung. Tuy nhiên ESB Có thể gây tắc nghẽn khi khối lượng dữ liệu lớn, chi phí triển khai và vận hành cao.

Service-Oriented Architecture (SOA): SOA xây dựng hệ thống dựa trên các dịch vụ độc lập, giao tiếp qua giao diện chuẩn hóa (như Web Services). SOA có tính tự trị cao, khả năng cộng tác đa nền tảng, giảm sự phụ thuộc giữa các module [16]. Tuy nhiên SOA Phức tạp trong triển khai và quản lý, đòi hỏi kỹ năng kỹ thuật cao.

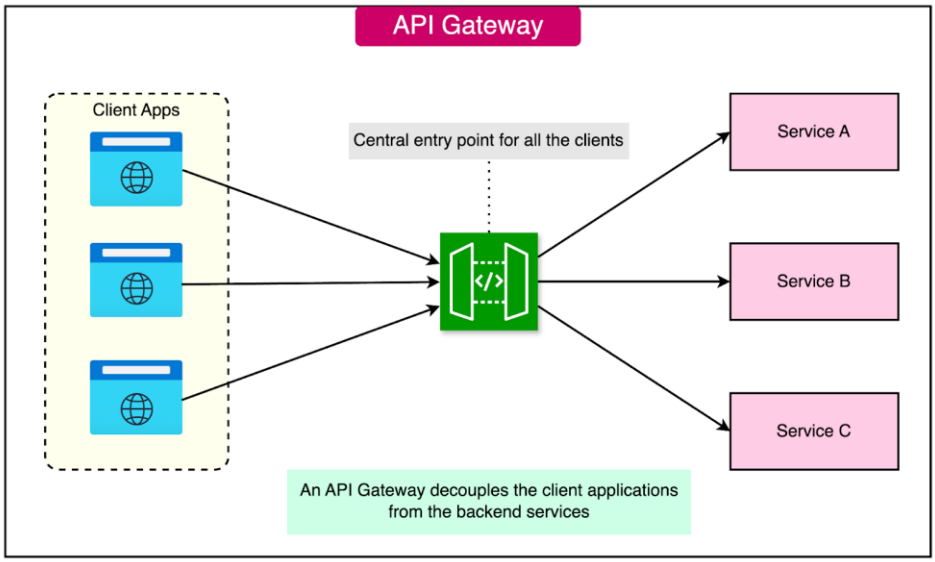
Point-to-Point Integration:



Hình 2.3. Mô hình tổng quan hệ thống sử dụng Point-to-Point Integration

Kết nối trực tiếp giữa các ứng dụng, không qua trung gian. Ưu điểm của nó là đơn giản, dễ triển khai cho số lượng hệ thống nhỏ.Tuy nhiên nó không khả thi khi số lượng ứng dụng tăng, khó bảo trì và mở rộng.

API Gateway:

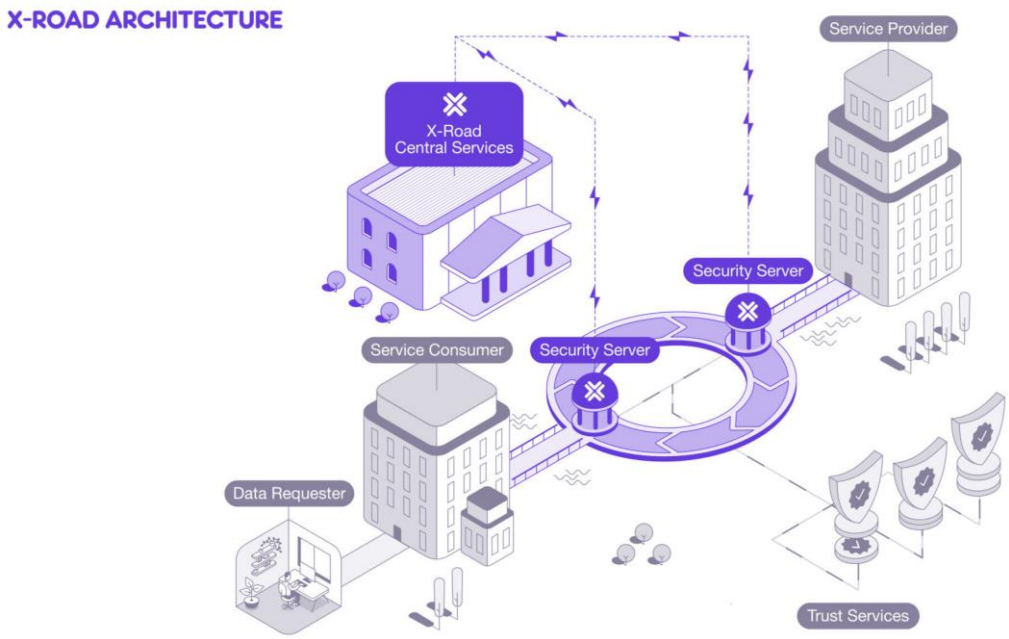


Hình 2.4. Mô hình tổng quan hệ thống sử dụng API Gateway

Sử dụng các API để kết nối và quản lý giao tiếp giữa các hệ thống. Ưu điểm của cách làm này là nhẹ, dễ tích hợp với các ứng dụng hiện đại, hỗ trợ tốt cho các

hệ thống đám mây. Cách làm này có hạn chế trong xử lý các hệ thống cũ (legacy systems) và có thể cần thêm lớp bảo mật.

X-Road:



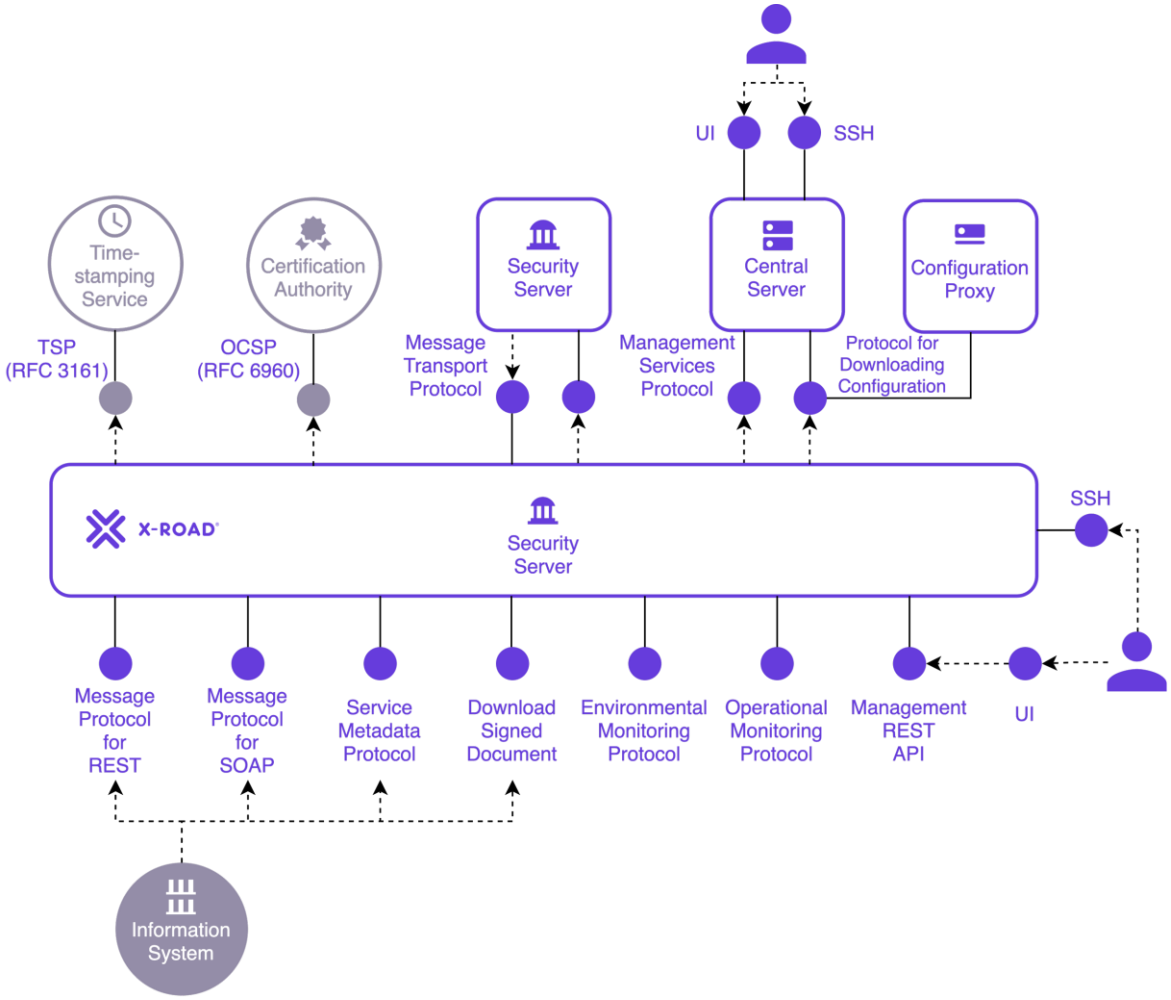
Hình 2.4. Mô tả mô hình hệ thống sử dụng X-Road

X-Road là một nền tảng mã nguồn mở, được thiết kế để hỗ trợ trao đổi dữ liệu an toàn, chuẩn hóa giữa các tổ chức, cơ quan chính phủ, và doanh nghiệp. Được phát triển lần đầu bởi Estonia vào năm 2001, X-Road đã trở thành một giải pháp tiêu biểu cho chính phủ điện tử, cho phép các hệ thống khác nhau chia sẻ dữ liệu mà không cần thông qua cơ sở dữ liệu trung tâm, đảm bảo tính phi tập trung và bảo mật. Ưu điểm của X-Road là tính Bảo mật cao, kiến trúc hệ thống theo mô hình phi tập trung Không yêu cầu cơ sở dữ liệu trung tâm, giảm nguy cơ điểm lỗi duy nhất (single point of failure) và tăng tính linh hoạt, Hỗ trợ tích hợp với các hệ thống cũ và mới, sử dụng các giao thức chuẩn như SOAP và REST, khả năng mở rộng tốt và mã nguồn mở.

2.1.3. Đề xuất sử dụng X-Road

X-Road là trục liên thông dữ liệu mã nguồn mở, bảo mật cao và đã được ứng dụng tại nhiều quốc gia [17]. được thiết kế để hỗ trợ trao đổi dữ liệu an toàn, chuẩn hóa giữa các tổ chức, cơ quan chính phủ, và doanh nghiệp. Được phát triển lần đầu

bởi Estonia vào năm 2001, X-Road đã trở thành một giải pháp tiêu biểu cho chính phủ điện tử, cho phép các hệ thống khác nhau chia sẻ dữ liệu mà không cần thông qua cơ sở dữ liệu trung tâm, đảm bảo tính phi tập trung và bảo mật.



Hình 2.5. Kiến trúc của X-Road

Kiến trúc X-Road dựa trên mô hình phi tập trung, bao gồm các thành phần chính:

- Security Server: Đóng vai trò là cổng giao tiếp giữa các tổ chức, chịu trách nhiệm mã hóa, xác thực và quản lý truy cập dữ liệu.

- Central Server: Quản lý cấu hình, danh sách các thành viên tham gia và chứng chỉ bảo mật. Không lưu trữ dữ liệu giao dịch, đảm bảo tính phi tập trung.

- Global Configuration: Hệ thống cấu hình toàn cầu, cung cấp thông tin về các thành viên, dịch vụ và chính sách bảo mật.

- Service Registry: Danh mục các dịch vụ được cung cấp bởi các tổ chức tham gia, cho phép khám phá và truy cập dịch vụ dễ dàng.

- Monitoring and Management Tools: Công cụ giám sát và quản lý giúp theo dõi hiệu suất và phát hiện sự cố.

Kiến trúc kỹ thuật này được thiết kế với nguyên tắc bảo mật từ đầu và kiến trúc phân tán, giúp nền tảng hầu như không gặp sự cố ngừng hoạt động kể từ khi ra mắt [18].

2.1.4. Kết luận

Trục liên thông dữ liệu là một thành phần quan trọng trong chiến lược chuyển đổi số và xây dựng chính phủ điện tử và chuyển đổi số giáo dục tại Việt Nam. Trong số các công nghệ khảo sát, X-Road nổi bật như một giải pháp tối ưu nhờ tính bảo mật cao, kiến trúc phi tập trung, chi phí hợp lý, và khả năng mở rộng. Với kinh nghiệm thành công tại nhiều quốc gia, X-Road có tiềm năng trở thành nền tảng cốt lõi để liên thông dữ liệu giữa các cơ quan, tổ chức, và các trường Đại học tại Việt Nam.

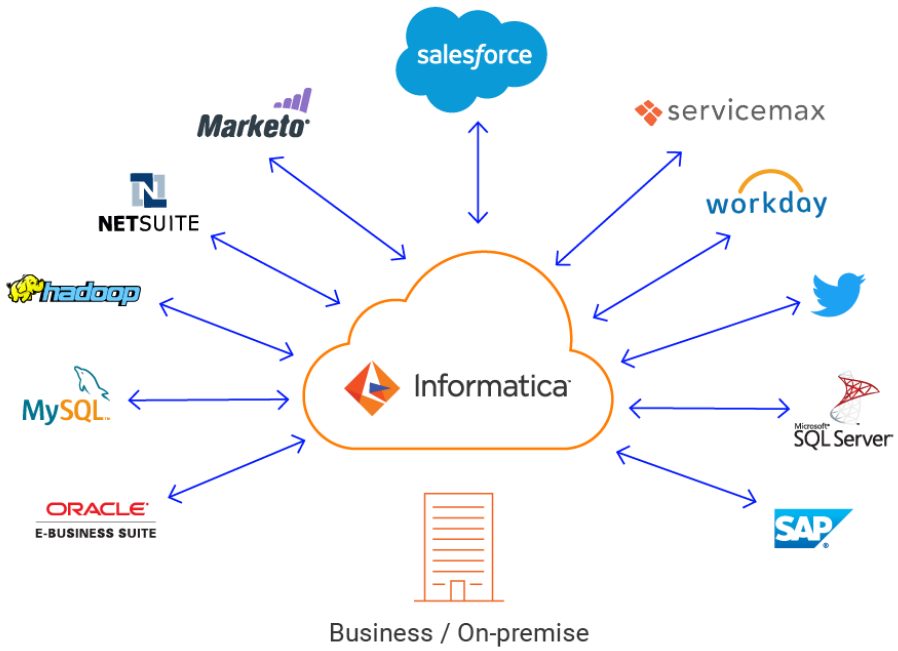
2.2. Thu thập tổng hợp, biến đổi dữ liệu (ETL)

Quá trình **ETL** (Extract, Transform, Load) là một phương pháp cốt lõi trong quản lý và xử lý dữ liệu, được sử dụng để thu thập dữ liệu từ nhiều nguồn, biến đổi dữ liệu theo định dạng phù hợp, và nạp dữ liệu vào hệ thống đích (như kho dữ liệu hoặc cơ sở dữ liệu phân tích). Trong bối cảnh chuyển đổi số và chính phủ điện tử tại Việt Nam, ETL đóng vai trò quan trọng trong việc đảm bảo dữ liệu được tổng hợp, làm sạch, và sẵn sàng cho các ứng dụng phân tích, báo cáo. Phần phân tích dưới đây trình bày chi tiết về sự cần thiết của ETL, khảo sát các công nghệ liên quan, và đề xuất giải pháp sử dụng **Apache Airflow** với định nghĩa, kiến trúc, ưu nhược điểm, cùng các khuyến nghị triển khai.

2.2.1. Khảo sát công nghệ

Để triển khai quá trình ETL, cần đánh giá các công nghệ và công cụ phù hợp. Dưới đây là phân tích chi tiết về các giải pháp ETL phổ biến:

Informatica PowerCenter:

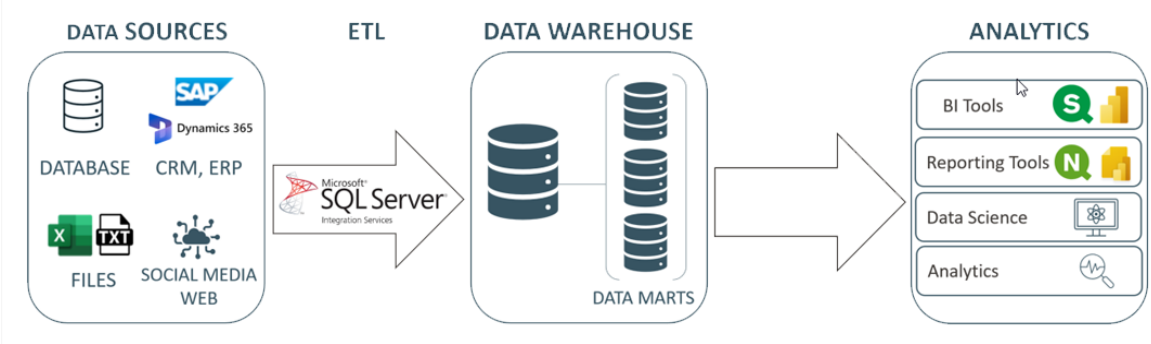


Hình 2.6. Mô hình tổng quan hệ thống sử dụng Informatica PowerCenter

Là công cụ ETL thương mại hàng đầu với giao diện kéo-thả dễ sử dụng, hỗ trợ tích hợp nhiều nguồn dữ liệu như cơ sở dữ liệu, đám mây, và API. Ưu điểm của nó là hiệu suất cao, giao diện thân thiện và khả năng giám sát mạnh mẽ. Tuy nhiên, nhược điểm là chi phí bản quyền và bảo trì cao, đòi hỏi phần cứng mạnh và dễ dẫn đến phụ thuộc nhà cung cấp. Phù hợp với các doanh nghiệp lớn có ngân sách dồi dào.

Talend Open Studio: Là công cụ ETL mã nguồn mở, miễn phí, hỗ trợ thiết kế quy trình ETL thông qua giao diện đồ họa và tích hợp với nhiều nguồn dữ liệu [19]. Ưu điểm là tiết kiệm chi phí, hỗ trợ phong phú và cộng đồng phát triển mạnh. Nhược điểm bao gồm thiếu tính năng nâng cao trong phiên bản miễn phí, yêu cầu kỹ năng kỹ thuật và hiệu suất chưa tối ưu với dữ liệu lớn. Thích hợp cho tổ chức vừa và nhỏ hoặc dự án chi phí thấp.

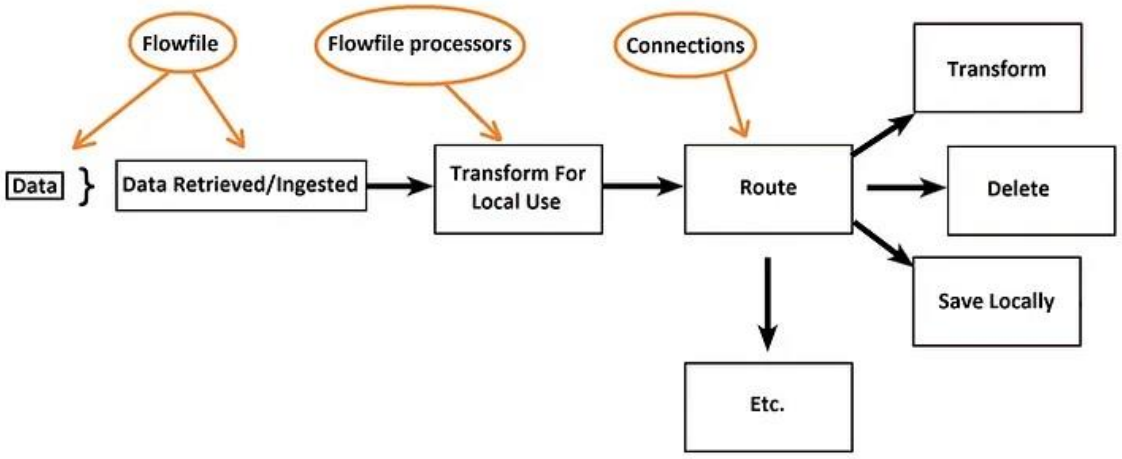
Microsoft SQL Server Integration Services (SSIS):



Hình 2.7. Mô hình tổng quan hệ thống sử dụng Microsoft SQL Server Integration Services (SSIS)

Microsoft SQL Server Integration Services (SSIS) là công cụ ETL tích hợp sẵn trong SQL Server, hỗ trợ kéo-thả thông qua Visual Studio, và tích hợp tốt với các sản phẩm Microsoft. Ưu điểm là dễ triển khai trong hệ sinh thái Microsoft và xử lý được các tác vụ dữ liệu phức tạp. Tuy nhiên, SSIS chỉ chạy tốt trên Windows, không linh hoạt với hệ thống không phải của Microsoft, và chi phí bản quyền có thể cao. Phù hợp với tổ chức đã sử dụng công nghệ Microsoft.

Apache Nifi:



Hình 2.8. Mô hình tổng quan hệ thống sử dụng Apache Nifi

Là công cụ mã nguồn mở chuyên xử lý và luân chuyển dữ liệu theo thời gian thực, với giao diện thiết kế đồ họa trực quan. Điểm mạnh là hỗ trợ cả dữ liệu thời gian thực và batch, dễ mở rộng, phù hợp với hệ thống phân tán. Tuy nhiên, nó hạn chế trong

xử lý biến đổi dữ liệu phức tạp và yêu cầu phần cứng cao khi làm việc với dữ liệu lớn. Phù hợp cho các ứng dụng như IoT hoặc xử lý dữ liệu luồng.

Apache Airflow: Là nền tảng mã nguồn mở dùng để lập lịch và điều phối các quy trình ETL, thích hợp cho các pipeline phức tạp cần kiểm soát luồng công việc chi tiết. Airflow lý tưởng cho các tổ chức cần giải pháp linh hoạt, có thể tùy chỉnh, tiết kiệm chi phí, tuy nhiên đòi hỏi kỹ năng lập trình và kinh nghiệm triển khai nhất định.

Pentaho Data Integration (Kettle): Công cụ ETL mã nguồn mở với giao diện thiết kế trực quan, hỗ trợ nhiều phép biến đổi và nguồn dữ liệu [20]. Ưu điểm là miễn phí, dễ triển khai và có cộng đồng hỗ trợ tốt. Nhược điểm gồm giao diện không hiện đại, hiệu suất chưa tối ưu với dữ liệu lớn, và thiếu tính năng nhóm trong bản miễn phí. Phù hợp với tổ chức vừa và nhỏ có ngân sách hạn chế.

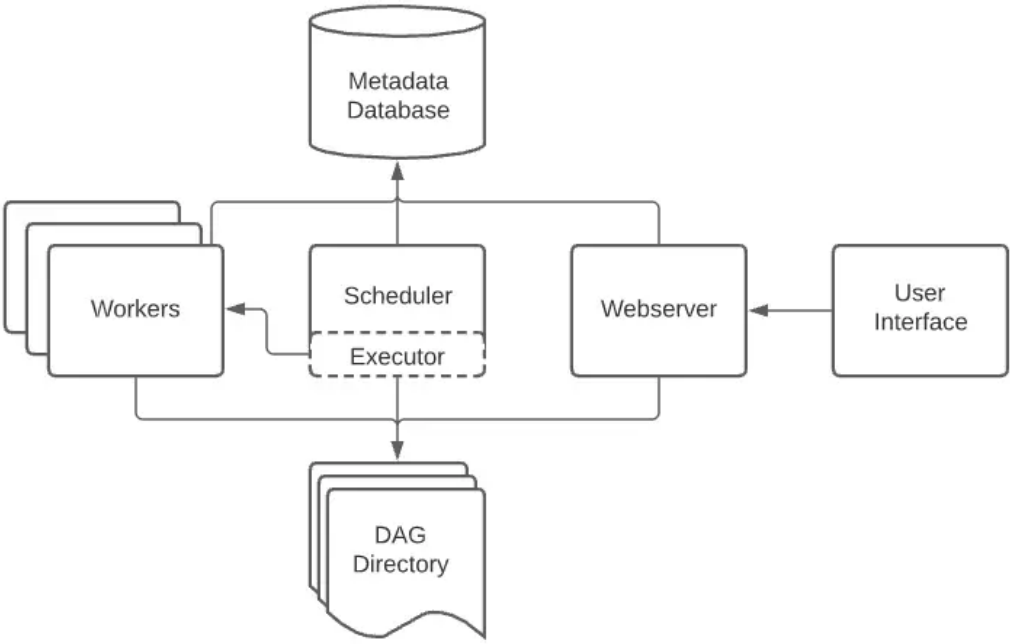
2.2.2. Đề xuất sử dụng Apache Airflow phục vụ thu thập tổng hợp, biến đổi dữ liệu

2.2.2.1. Giới thiệu về Apache Airflow

Apache Airflow là một nền tảng mã nguồn mở, được phát triển bởi Airbnb vào năm 2014 và hiện được quản lý bởi Apache Software Foundation. Airflow được thiết kế để lập lịch, điều phối, và giám sát các quy trình xử lý dữ liệu (workflow orchestration), đặc biệt phù hợp cho các pipeline ETL. Thay vì là một công cụ ETL truyền thống với giao diện kéo-thả, Airflow sử dụng mã Python để định nghĩa các quy trình (workflow) dưới dạng **DAGs** (Directed Acyclic Graphs), mang lại tính linh hoạt và khả năng tùy chỉnh cao.

Airflow không trực tiếp thực hiện các tác vụ ETL (như thu thập hoặc biến đổi dữ liệu) mà hoạt động như một "nhạc trưởng", điều phối các công cụ khác (như SQL, Pandas, hoặc Spark) để thực hiện các bước trong quy trình ETL. Với khả năng tích hợp mạnh mẽ và cộng đồng phát triển lớn, Airflow đã trở thành một trong những giải pháp hàng đầu cho các tổ chức cần quản lý quy trình dữ liệu phức tạp. Đặc biệt, 90% người tham gia khảo sát năm 2023 sử dụng Airflow cho các quy trình ETL/ELT phục vụ analytics [21].

2.2.2.2. Kiến trúc của Apache Airflow



Hình 2.9. Kiến trúc của Apache Airflow

Kiến trúc của Apache Airflow được thiết kế theo mô hình modular với khả năng mở rộng cao, sử dụng message queue để điều phối số lượng worker tùy ý [22]. Kiến trúc này đã được chứng minh hiệu quả thông qua việc được sử dụng rộng rãi bởi hơn 1000 contributor [23]:

- Scheduler: Scheduler là thành phần chịu trách nhiệm lập lịch và kích hoạt các DAGs (Directed Acyclic Graphs) dựa trên thời gian hoặc các sự kiện được định nghĩa trước. Nó theo dõi trạng thái của các tác vụ trong DAG và đảm bảo rằng các tác vụ được thực thi đúng theo thứ tự phụ thuộc đã chỉ định. Đây là bộ điều phối trung tâm giúp Airflow hoạt động như một hệ thống lập lịch công việc hiệu quả.

- Executor: Executor là thành phần quản lý việc thực thi các tác vụ trong các DAG. Apache Airflow hỗ trợ nhiều loại executor để đáp ứng các nhu cầu khác nhau về quy mô và hiệu suất, bao gồm SequentialExecutor (dùng cho môi trường phát triển nhỏ, thực thi tuần tự), LocalExecutor (chạy các tác vụ song song trên một máy duy nhất), và CeleryExecutor (cho phép phân tán các tác vụ trên nhiều máy chủ, hỗ trợ khả năng mở rộng cao).

- Web Server: Web Server cung cấp giao diện người dùng dựa trên nền web để theo dõi, quản lý và giám sát các DAG và tác vụ. Giao diện này hiển thị trạng thái hiện tại của các tác vụ, nhật ký thực thi (logs), và sơ đồ DAG trực quan, giúp người dùng dễ dàng kiểm tra và thao tác với các luồng công việc mà không cần truy cập vào hệ thống qua dòng lệnh.

- Metadata Database: Metadata Database là nơi lưu trữ tất cả các thông tin liên quan đến DAGs, bao gồm cấu hình, lịch sử chạy, trạng thái của từng tác vụ, và các metadata khác. Airflow hỗ trợ nhiều hệ quản trị cơ sở dữ liệu để làm metadata store, như PostgreSQL, MySQL hoặc SQLite. Thành phần này đóng vai trò trung tâm trong việc ghi nhận và tra cứu hoạt động của hệ thống.

- Worker: Worker là các tiến trình thực thi các tác vụ được giao từ Scheduler, thường được sử dụng trong môi trường sử dụng executor phân tán như CeleryExecutor. Các worker có thể được triển khai trên nhiều máy chủ để tăng khả năng xử lý song song, cải thiện hiệu năng và khả năng mở rộng của hệ thống khi xử lý khối lượng công việc lớn.

- DAGs (Directed Acyclic Graphs): DAG là cấu trúc biểu diễn luồng công việc trong Airflow, được định nghĩa dưới dạng mã Python. Mỗi DAG bao gồm các node là các tác vụ (tasks), và các cạnh thể hiện mối quan hệ phụ thuộc giữa chúng. Người dùng có thể sử dụng nhiều loại Operator như BashOperator, PythonOperator, PostgresOperator để thực hiện các tác vụ cụ thể trong pipeline ETL. DAG là đơn vị cốt lõi thể hiện logic xử lý dữ liệu trong Airflow.

Luồng hoạt động của Apache Airflow bắt đầu khi người dùng định nghĩa một DAG (Directed Acyclic Graph) bằng ngôn ngữ Python, trong đó chỉ định các tác vụ ETL cụ thể như thu thập dữ liệu từ API, biến đổi dữ liệu bằng thư viện Pandas, và nạp kết quả vào hệ quản trị cơ sở dữ liệu như PostgreSQL. Sau đó, Scheduler sẽ đọc DAG này và lập lịch thực thi các tác vụ dựa trên thời gian định sẵn hoặc các sự kiện kích hoạt. Executor sẽ tiếp nhận các tác vụ từ Scheduler và phân phối chúng đến các Worker để thực thi, hoặc tự thực hiện trực tiếp tùy thuộc vào loại executor đang được sử dụng. Trong suốt quá trình này, Web Server cung

cấp giao diện người dùng trực quan để hiển thị trạng thái của DAG và các tác vụ, cũng như cho phép theo dõi nhật ký và xử lý sự cố khi cần thiết.

2.2.2.3. Lý do sử dụng Apache Airflow

Linh hoạt và tùy chỉnh cao: Cho phép định nghĩa các DAGs (Directed Acyclic Graphs) bằng ngôn ngữ Python, giúp người dùng dễ dàng kiểm soát luồng công việc và tùy chỉnh logic xử lý dữ liệu một cách linh hoạt. Việc sử dụng Python cho phép xây dựng các quy trình ETL phức tạp, hỗ trợ viết code rõ ràng, dễ bảo trì và không bị giới hạn bởi giao diện kéo-thả như ở một số công cụ ETL khác.

Mã nguồn mở: Là một dự án mã nguồn mở được phát triển ban đầu bởi Airbnb và hiện do Apache Software Foundation quản lý. Người dùng có thể sử dụng mà không phải trả phí bản quyền. Dự án có cộng đồng phát triển lớn và đang hoạt động tích cực, với nhiều plugin, ví dụ, và bản vá được đóng góp thường xuyên, hỗ trợ triển khai và vận hành hiệu quả.

Khả năng mở rộng: Hỗ trợ các kiến trúc triển khai phân tán thông qua CeleryExecutor hoặc KubernetesExecutor. Điều này cho phép hệ thống xử lý khối lượng lớn công việc bằng cách phân phối các tác vụ đến nhiều worker chạy trên nhiều máy chủ. Nhờ đó, hệ thống có thể dễ dàng mở rộng theo nhu cầu.

Giao diện quản lý thân thiện: Cung cấp một giao diện web trực quan để giám sát và quản lý các DAGs. Người dùng có thể xem lịch sử chạy, trạng thái hiện tại của từng tác vụ, nhật ký lỗi, và sơ đồ DAG. Giao diện này rất hữu ích cho việc theo dõi hệ thống, phát hiện và xử lý sự cố nhanh chóng.

Hỗ trợ lập lịch mạnh mẽ: Hỗ trợ lập lịch chạy DAGs bằng cách sử dụng các biểu thức cron hoặc thiết lập thời gian lặp lại cụ thể. Hệ thống cũng hỗ trợ các cơ chế retry (thử lại khi thất bại) và skip (bỏ qua tác vụ theo điều kiện), giúp đảm bảo quá trình xử lý dữ liệu được thực hiện bền vững và linh hoạt.

Tích hợp với hệ sinh thái hiện đại: Hỗ trợ tích hợp với nhiều hệ thống và dịch vụ như cơ sở dữ liệu (MySQL, PostgreSQL), các dịch vụ đám mây (AWS, GCP), và công cụ xử lý dữ liệu như Spark. Việc này được thực hiện thông qua các Operator có sẵn trong thư viện mở rộng hoặc từ cộng đồng.

Cộng đồng và tài liệu phong phú: Có một cộng đồng người dùng và nhà phát triển lớn, bao gồm nhiều công ty công nghệ sử dụng trong môi trường sản xuất. Trang tài liệu chính thức được duy trì tốt và cung cấp hướng dẫn đầy đủ về cách cài đặt, cấu hình, và sử dụng Airflow một cách hiệu quả. Ngoài ra, có nhiều diễn đàn và khóa học trực tuyến từ cộng đồng.

Hỗ trợ đa nền tảng: Có thể được cài đặt và chạy trên các hệ điều hành như Linux và macOS. Ngoài ra, Airflow cũng hỗ trợ triển khai trong các môi trường container như Docker và Kubernetes, giúp dễ dàng tích hợp vào các hệ thống đám mây hoặc hybrid cloud tùy theo hạ tầng của tổ chức.

2.2.3. Kết luận

Quá trình ETL là một thành phần không thể thiếu trong việc quản lý và xử lý dữ liệu, đặc biệt trong bối cảnh chuyển đổi số và chính phủ điện tử tại Việt Nam. Trong số các công nghệ khảo sát, **Apache Airflow** nổi bật như một giải pháp tối ưu nhờ tính linh hoạt, chi phí thấp, và khả năng tích hợp với các hệ thống hiện đại. Với kiến trúc dựa trên Python và khả năng điều phối quy trình mạnh mẽ, Airflow phù hợp để triển khai các pipeline ETL phức tạp, từ thu thập dữ liệu đến phân tích báo cáo.

2.3. Hồ dữ liệu tổng hợp và phân tích dữ liệu đa cấu trúc (Data Lake).

Hồ dữ liệu (Data Lake) là một hệ thống lưu trữ tập trung, được thiết kế để chứa dữ liệu đa cấu trúc (cấu trúc, bán cấu trúc, phi cấu trúc) ở định dạng gốc với quy mô lớn. Datalake đóng vai trò quan trọng trong việc hỗ trợ phân tích dữ liệu, trí tuệ nhân tạo, và chuyển đổi số, đặc biệt trong bối cảnh chính phủ điện tử và kinh tế số tại Việt Nam. Phần phân tích dưới đây trình bày chi tiết về sự cần thiết của Datalake, khảo sát các công nghệ liên quan, và đề xuất giải pháp sử dụng **MinIO** với định nghĩa, kiến trúc, ưu nhược điểm, cùng các khuyến nghị triển khai.

2.3.1. Khảo sát công nghệ

Để triển khai Datalake, cần đánh giá các công nghệ và nền tảng lưu trữ phù hợp. Dưới đây là phân tích chi tiết về các giải pháp phổ biến:

Amazon S3 (Simple Storage Service): Amazon S3 là dịch vụ lưu trữ đối tượng trên đám mây của AWS, được sử dụng rộng rãi để xây dựng Data Lake nhờ

khả năng mở rộng gần như không giới hạn và độ bền dữ liệu rất cao. Dịch vụ này tích hợp tốt với các công cụ trong hệ sinh thái AWS như Glue (ETL), Athena (truy vấn SQL không cần máy chủ), và Redshift (kho dữ liệu). Ngoài ra, S3 cung cấp các tính năng bảo mật mạnh như mã hóa dữ liệu, kiểm soát truy cập với IAM, và chính sách bucket. Tuy nhiên, chi phí có thể cao khi xử lý dữ liệu lớn hoặc có tần suất truy cập cao. Bên cạnh đó, phụ thuộc vào AWS khiến các tổ chức dễ rơi vào tình trạng "vendor lock-in", và yêu cầu kết nối internet ổn định có thể là rào cản tại các khu vực hạ tầng yếu. Amazon S3 đặc biệt phù hợp với các tổ chức đã sử dụng hệ sinh thái AWS hoặc cần triển khai Data Lake hoàn toàn trên đám mây.

Microsoft Azure Data Lake Storage (ADLS): ADLS là dịch vụ Data Lake được xây dựng trên nền tảng đám mây của Microsoft, cung cấp khả năng lưu trữ và phân tích dữ liệu lớn tích hợp chặt chẽ với Azure Databricks, Power BI và các công cụ phân tích của Microsoft. Một trong những điểm mạnh của ADLS là hỗ trợ cấu trúc thư mục phân cấp (hierarchical namespace), giúp cải thiện hiệu suất truy cập và quản lý dữ liệu giống như trong hệ thống tệp truyền thống. Dịch vụ này cũng tích hợp tốt với Azure Active Directory để đảm bảo bảo mật và kiểm soát truy cập. Tuy nhiên, chi phí sử dụng vẫn ở mức cao, và việc triển khai, quản lý yêu cầu kỹ năng kỹ thuật đáng kể. ADLS phù hợp với các tổ chức đã đầu tư vào hệ sinh thái Azure hoặc sử dụng nhiều dịch vụ của Microsoft.

Google Cloud Storage: Google Cloud Storage (GCS) là dịch vụ lưu trữ đối tượng trên đám mây của Google, hỗ trợ triển khai Data Lake với khả năng tích hợp chặt chẽ với các dịch vụ như BigQuery (kho dữ liệu), Dataflow (xử lý dữ liệu luồng) và các công cụ AI/ML của Google. GCS nổi bật với hiệu suất cao, chi phí cạnh tranh so với AWS và Azure, đồng thời hỗ trợ nhiều lớp lưu trữ như Standard, Nearline, Coldline để tối ưu chi phí tùy theo tần suất truy cập. Tuy nhiên, cũng như các dịch vụ đám mây khác, GCS phụ thuộc vào nhà cung cấp, gây rủi ro vendor lock-in. Ngoài ra, tài liệu và hỗ trợ cộng đồng của Google Cloud vẫn chưa phong phú bằng AWS. GCS đặc biệt phù hợp với các tổ chức cần khai thác dữ liệu phục vụ AI/ML và đang sử dụng hệ sinh thái Google Cloud.

Apache Hadoop HDFS: Hadoop Distributed File System (HDFS) là hệ thống tệp phân tán mã nguồn mở, thường được sử dụng để xây dựng Data Lake tại chỗ (on-premises). HDFS có khả năng lưu trữ và xử lý khối lượng dữ liệu lớn thông qua việc tích hợp với các công cụ trong hệ sinh thái Hadoop như Spark, Hive, và HBase. Điểm mạnh của HDFS là chi phí thấp (về phần mềm), khả năng tùy chỉnh cao và không phụ thuộc vào nhà cung cấp đám mây. Tuy nhiên, việc triển khai và vận hành HDFS phức tạp, đòi hỏi đội ngũ kỹ thuật có kinh nghiệm sâu về hệ thống phân tán. Hơn nữa, hiệu suất xử lý dữ liệu phi cấu trúc không cao và chi phí phần cứng, bảo trì sẽ tăng đáng kể nếu mở rộng quy mô. HDFS phù hợp cho các tổ chức lớn có hạ tầng mạnh và cần kiểm soát hoàn toàn dữ liệu nội bộ.

MinIO: MinIO là một hệ thống lưu trữ đối tượng mã nguồn mở, tương thích hoàn toàn với API của Amazon S3, cho phép triển khai linh hoạt trên hạ tầng tại chỗ hoặc đám mây. MinIO nổi bật với kiến trúc nhẹ, hiệu suất cao và khả năng tích hợp tốt trong các môi trường hybrid (kết hợp giữa on-premises và cloud). Nhờ tính mở và khả năng tương thích với S3 [24], MinIO là lựa chọn phù hợp cho các tổ chức muốn xây dựng Data Lake chi phí thấp, tránh phụ thuộc vào nhà cung cấp dịch vụ lớn. Tuy nhiên, hệ thống yêu cầu đội ngũ kỹ thuật có năng lực để triển khai và vận hành, đặc biệt nếu triển khai trong môi trường phân tán hoặc kết hợp với hệ sinh thái phức tạp.

Snowflake Data Lake: Snowflake cung cấp một giải pháp kết hợp giữa Data Lake và kho dữ liệu đám mây, cho phép lưu trữ và xử lý dữ liệu đa cấu trúc trên một nền tảng duy nhất. Với Snowflake, người dùng có thể phân tích dữ liệu trực tiếp từ kho lưu trữ mà không cần sao chép hoặc di chuyển dữ liệu. Nền tảng này cung cấp hiệu suất cao, khả năng tự động mở rộng và giao diện dễ sử dụng. Ngoài ra, Snowflake tích hợp chặt chẽ các cơ chế bảo mật, quản lý quyền truy cập và tuân thủ. Tuy nhiên, chi phí sử dụng tương đối cao, đặc biệt với khối lượng dữ liệu lớn hoặc nhu cầu truy vấn thường xuyên. Bên cạnh đó, Snowflake không hỗ trợ triển khai tại chỗ và phụ thuộc hoàn toàn vào dịch vụ đám mây của hãng. Snowflake phù

hợp cho các doanh nghiệp lớn cần phân tích dữ liệu phức tạp với tính sẵn sàng và hiệu suất cao.

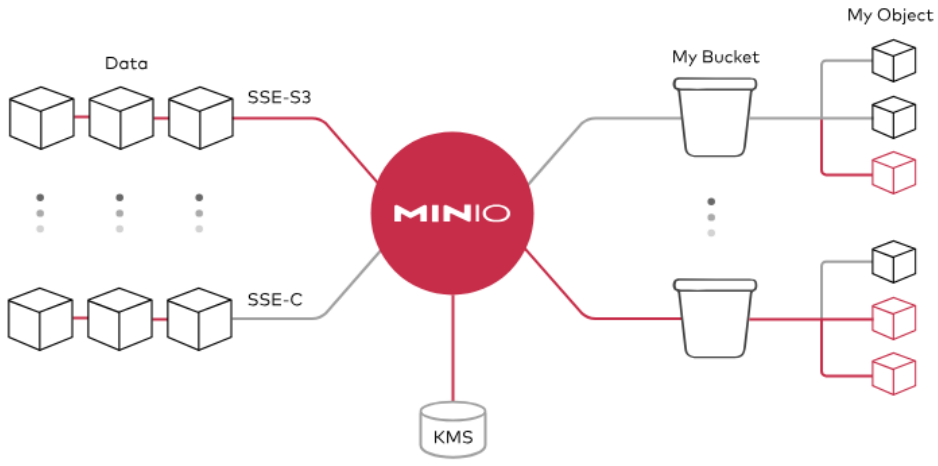
2.3.2. Đề xuất sử dụng MinIO tổng hợp và phân tích dữ liệu đa cấu trúc

2.3.2.1. Giới thiệu về MinIO

MinIO là một hệ thống lưu trữ đối tượng mã nguồn mở, hiệu suất cao, tương thích với API Amazon S3, được thiết kế để xây dựng Datalake và lưu trữ dữ liệu đa cấu trúc. MinIO có thể triển khai tại chỗ (on-premises), trên đám mây, hoặc trong môi trường lai (hybrid), cung cấp giải pháp linh hoạt cho các tổ chức muốn quản lý khối lượng dữ liệu lớn với chi phí thấp. Được viết bằng Go, MinIO nổi bật với hiệu suất, tính đơn giản, và khả năng mở rộng, phù hợp cho các ứng dụng từ chính phủ điện tử đến doanh nghiệp tư nhân.

MinIO hỗ trợ lưu trữ dữ liệu ở định dạng gốc, tích hợp với các công cụ phân tích như Apache Spark, Presto, và TensorFlow, đồng thời cung cấp các tính năng bảo mật như mã hóa, quản lý quyền truy cập, và nhật ký. Với khả năng tương thích S3, MinIO cho phép sử dụng các công cụ và thư viện sẵn có trong hệ sinh thái AWS mà không phụ thuộc vào AWS.

2.3.2.2. Kiến trúc của MinIO



Hình 2.10. Kiến trúc của MinIO

Kiến trúc của MinIO được thiết kế đặc biệt cho cloud native workloads và AI/ML, với single-layer architecture đạt được tất cả chức năng cần thiết mà không

cần thỏa hiệp [25]. Kiến trúc này dựa trên mô hình lưu trữ đối tượng phân tán, với các thành phần chính:

- MinIO Server là thành phần lõi của hệ thống, có nhiệm vụ xử lý các yêu cầu lưu trữ và truy xuất dữ liệu. Thành phần này hỗ trợ triển khai ở cả chế độ đơn node thường dùng trong môi trường thử nghiệm và đa node dùng cho sản xuất, đảm bảo tính sẵn sàng cao và phân tán dữ liệu. MinIO Server sử dụng kỹ thuật erasure coding để bảo vệ dữ liệu, giúp hệ thống chống lại lỗi phần cứng.

- Client API cung cấp các SDK cho nhiều ngôn ngữ như Python, Java, Go và .NET, cùng với công cụ dòng lệnh mc (MinIO Client), cho phép người dùng và hệ thống tương tác với MinIO. Đặc biệt, MinIO tương thích với API S3 [26], cho phép sử dụng các công cụ quen thuộc như AWS SDK, boto3 hoặc s3cmd mà không cần chỉnh sửa nhiều về mặt tích hợp.

- Storage Backend là nơi MinIO lưu trữ dữ liệu, sử dụng hệ thống tệp của máy chủ như ext4, XFS hoặc NTFS. Backend này có thể vận hành trên nhiều loại thiết bị lưu trữ như ổ cứng HDD, ổ cứng thể rắn SSD hoặc thiết bị lưu trữ mạng NAS, phù hợp với đa dạng môi trường hạ tầng.

- Erasure Coding là cơ chế đảm bảo an toàn dữ liệu trong MinIO. Dữ liệu được chia nhỏ thành các mảnh (shards) và thêm các mã dự phòng (parity), cho phép phục hồi khi xảy ra lỗi phần cứng. Ví dụ, với cấu hình gồm 8 ổ đĩa, hệ thống vẫn có thể hoạt động bình thường và không mất dữ liệu ngay cả khi có tới 4 ổ đĩa bị lỗi.

- Gateway Mode cho phép MinIO hoạt động như một cổng kết nối đến các nền tảng lưu trữ đám mây khác như AWS S3, Azure Blob hoặc Google Cloud Storage. Trong chế độ này, MinIO cung cấp một lớp tương thích trung gian, hỗ trợ truy cập dữ liệu trên các nền tảng khác mà vẫn giữ giao diện lập trình ứng dụng thống nhất.

- Management Console là giao diện web mà MinIO cung cấp để quản lý hệ thống. Qua đó, người dùng có thể quản lý bucket, người dùng, phân quyền truy cập, cũng như theo dõi hiệu suất hoạt động của hệ thống như dung lượng lưu trữ, băng thông sử dụng và trạng thái các node trong cụm.

2.3.2.3. Luồng hoạt động của minIO

MinIO là một hệ thống lưu trữ đối tượng hiệu năng cao, tương thích với giao thức S3, thường được sử dụng trong các kiến trúc Datalake hiện đại. Luồng hoạt động của MinIO có thể chia thành bốn giai đoạn chính: lưu trữ, truy xuất, phân tích và quản lý.

Lưu trữ: Dữ liệu từ các client, chẳng hạn như ứng dụng hoặc các quy trình ETL, được gửi tới MinIO thông qua API S3. Trước khi lưu trữ, dữ liệu được mã hóa để đảm bảo an toàn, sau đó được phân mảnh sử dụng kỹ thuật erasure coding nhằm tăng độ bền và khả năng khôi phục. Các mảnh dữ liệu này được phân phối và lưu trữ trên nhiều node trong hệ thống.

Truy xuất: Khi có yêu cầu truy xuất dữ liệu, client gửi yêu cầu thông qua API S3. MinIO thực hiện quá trình xác thực người dùng, sau đó truy xuất các mảnh dữ liệu tương ứng từ các node, thực hiện giải mã và tái hợp dữ liệu trước khi trả về cho client.

Phân tích: Các công cụ phân tích dữ liệu như Apache Spark hoặc Presto có thể truy cập trực tiếp dữ liệu trong MinIO thông qua API S3. Điều này cho phép xử lý dữ liệu ngay trong Datalake mà không cần di chuyển, giúp tiết kiệm thời gian và tài nguyên.

Quản lý: Quản trị viên có thể giám sát và cấu hình hệ thống MinIO thông qua giao diện Management Console hoặc công cụ dòng lệnh (CLI), bao gồm việc theo dõi trạng thái các node, cấu hình chính sách bảo mật, quản lý người dùng và giám sát hiệu năng hoạt động.

Nhờ thiết kế phân tán và khả năng mở rộng linh hoạt, MinIO đáp ứng tốt các yêu cầu lưu trữ và xử lý dữ liệu lớn trong các hệ thống hiện đại đặc biệt là các hệ thống lưu trữ dữ liệu tại các trường đại học.

2.3.2.4. Lý do sử dụng MinIO để tổng hợp và phân tích dữ liệu đa cấu trúc

Tương thích S3 API: Hỗ trợ đầy đủ API S3, cho phép tích hợp mượt mà với hầu hết các công cụ và thư viện đã sử dụng giao thức này như AWS SDK, Apache

Spark và Presto. Nhờ đó, người dùng có thể tái sử dụng mã nguồn và công cụ từ hệ sinh thái AWS mà không phụ thuộc vào hạ tầng của Amazon.

Mã nguồn mở: MinIO là phần mềm mã nguồn mở, miễn phí sử dụng. Điều này giúp giảm đáng kể chi phí triển khai so với các giải pháp thương mại như AWS S3 hay Azure Blob. Bên cạnh đó, cộng đồng phát triển mạnh cũng thường xuyên cung cấp các bản cập nhật và hỗ trợ kỹ thuật.

Hiệu suất cao: Được phát triển bằng ngôn ngữ lập trình Go, MinIO được tối ưu cho tốc độ xử lý cao. Hệ thống có thể đáp ứng hàng triệu yêu cầu đọc/ghi mỗi giây ngay cả trên phần cứng tiêu chuẩn, phù hợp với các ứng dụng yêu cầu hiệu suất lớn.

Khả năng mở rộng: MinIO hỗ trợ kiến trúc phân tán đa node, cho phép người dùng dễ dàng mở rộng bằng cách thêm ổ đĩa hoặc máy chủ. Nhờ đó, hệ thống có thể phục vụ nhu cầu lưu trữ dữ liệu từ vài terabyte (TB) đến hàng petabyte (PB) một cách linh hoạt.

Bảo mật mạnh mẽ: Hệ thống hỗ trợ mã hóa TLS/SSL để đảm bảo an toàn khi truyền dữ liệu, cũng như mã hóa dữ liệu tại chỗ. MinIO còn cung cấp cơ chế quản lý quyền truy cập chi tiết với chính sách IAM tương tự như AWS, và hỗ trợ tích hợp với LDAP hoặc Active Directory để xác thực người dùng doanh nghiệp.

Triển khai linh hoạt: Có thể triển khai tại chỗ, trên các nền tảng đám mây như AWS, Google Cloud, Azure, hoặc trong môi trường lai. Hệ thống cũng hỗ trợ triển khai dưới dạng container thông qua Docker hoặc Kubernetes, phù hợp với các mô hình hạ tầng hiện đại.

Chi phí hiệu quả: Vận hành trên phần cứng tiêu chuẩn, không đòi hỏi thiết bị chuyên dụng đắt đỏ. Nhờ không có chi phí bản quyền và không bị khóa bởi nhà cung cấp, MinIO là giải pháp lưu trữ tiết kiệm so với các dịch vụ đám mây hoặc Hadoop HDFS.

Quản trị đơn giản: Cung cấp giao diện quản lý web (Management Console) thân thiện với người dùng, cho phép dễ dàng quản lý bucket, người dùng và theo

dõi hiệu suất hệ thống. Ngoài ra, công cụ dòng lệnh mc hỗ trợ tự động hóa các tác vụ quản trị.

Hỗ trợ đa nền tảng: Có thể chạy trên nhiều hệ điều hành như Linux, Windows và macOS, phù hợp với đa dạng hạ tầng công nghệ, bao gồm cả tại Việt Nam. Hệ thống cũng tích hợp tốt với các công cụ xử lý dữ liệu và ETL phổ biến như Apache Airflow và Spark.

2.3.3. Kết luận

Hồ dữ liệu (Data Lake) là thành phần không thể thiếu để quản lý dữ liệu đa cấu trúc trong chuyển đổi số và chính phủ điện tử tại Việt Nam. Trong số các công nghệ khảo sát, MinIO nổi bật với tính tương thích S3, chi phí thấp, hiệu suất cao, và khả năng triển khai linh hoạt. Với kiến trúc mã nguồn mở và bảo mật mạnh mẽ, MinIO phù hợp để xây dựng Datalake cho các cơ quan nhà nước và doanh nghiệp.

Để triển khai thành công, cần đầu tư vào hạ tầng, đào tạo, và chính sách hỗ trợ. Việc bắt đầu từ các dự án thí điểm, tích hợp với các hệ thống như ETL và trục liên thông, và tận dụng cộng đồng mã nguồn mở sẽ giúp tối ưu hóa hiệu quả. Với sự chuẩn bị kỹ lưỡng, MinIO có thể trở thành nền tảng cốt lõi cho hệ sinh thái dữ liệu, góp phần xây dựng một Việt Nam số hóa hiện đại và bền vững.

2.4. Trực quan hóa dữ liệu

Trực quan hóa dữ liệu (Data Visualization) là quá trình biểu diễn dữ liệu dưới dạng các biểu đồ, đồ thị, bản đồ, hoặc bảng tương tác để giúp người dùng hiểu rõ thông tin, phát hiện xu hướng, và đưa ra quyết định dựa trên dữ liệu. Trong bối cảnh chuyển đổi số đại học, trực quan hóa dữ liệu đóng vai trò quan trọng trong việc trình bày thông tin phức tạp một cách trực quan, dễ hiểu, từ đó hỗ trợ các cơ quan, tổ chức, và người dân ra quyết định hiệu quả. Phần phân tích dưới đây trình bày chi tiết về sự cần thiết của trực quan hóa dữ liệu, khảo sát các công nghệ liên quan, và đề xuất giải pháp sử dụng Apache Superset với định nghĩa, kiến trúc, ưu nhược điểm, cùng các khuyến nghị triển khai.

2.4.1. Khảo sát các công nghệ

Để triển khai trực quan hóa dữ liệu, cần đánh giá các công cụ và nền tảng phù hợp. Dưới đây là phân tích chi tiết về các giải pháp phổ biến:

Tableau: Tableau là một công cụ trực quan hóa dữ liệu thương mại hàng đầu, cung cấp giao diện kéo-thả giúp người dùng dễ dàng tạo các biểu đồ và dashboard tương tác. Ưu điểm của Tableau bao gồm giao diện thân thiện và dễ sử dụng ngay cả với người không chuyên, khả năng tích hợp với nhiều nguồn dữ liệu như SQL, NoSQL và các dịch vụ đám mây, cùng các tính năng phân tích mạnh mẽ hỗ trợ xây dựng dashboard phức tạp. Tuy nhiên, Tableau cũng có một số nhược điểm đáng chú ý như chi phí bản quyền cao, dễ gây phụ thuộc vào nhà cung cấp (vendor lock-in), và yêu cầu phần cứng mạnh khi xử lý dữ liệu lớn. Công cụ này phù hợp với các doanh nghiệp lớn hoặc tổ chức có ngân sách dồi dào.

Power BI: Power BI là công cụ trực quan hóa dữ liệu do Microsoft phát triển, tích hợp tốt với hệ sinh thái của Microsoft như Excel, Azure và SQL Server. Power BI nổi bật nhờ giao diện trực quan, tích hợp AI và khả năng tạo báo cáo tự động. Ngoài ra, công cụ này cũng có lợi thế về chi phí so với Tableau trong một số gói dịch vụ và khả năng tích hợp chặt chẽ với Azure và Office 365. Nhược điểm của Power BI là phụ thuộc vào nền tảng Microsoft, hạn chế khi triển khai trên hệ điều hành không phải Windows, chi phí bản quyền vẫn cao so với các giải pháp mã nguồn mở, và khả năng tùy chỉnh bị giới hạn. Power BI phù hợp cho các tổ chức đã và đang sử dụng hệ sinh thái Microsoft.

QlikView/Qlik Sense: QlikView và Qlik Sense là hai công cụ trực quan hóa dữ liệu thương mại thuộc nền tảng Qlik, nổi bật với khả năng phân tích liên kết (associative analytics). Ưu điểm chính của Qlik là khả năng khám phá dữ liệu mạnh mẽ với tính năng tự động tìm mối quan hệ giữa các trường dữ liệu, hỗ trợ dashboard tương tác và phân tích dữ liệu thời gian thực, đồng thời tích hợp tốt với nhiều nguồn dữ liệu khác nhau. Nhược điểm của công cụ này bao gồm chi phí cao, quy trình triển khai phức tạp, yêu cầu người dùng được đào tạo để sử dụng hiệu quả, và mức

độ phổ biến thấp hơn so với Tableau và Power BI. Qlik phù hợp với các doanh nghiệp cần thực hiện phân tích dữ liệu phức tạp.

Metabase: Metabase là một công cụ mã nguồn mở cung cấp giao diện đơn giản để tạo biểu đồ và dashboard từ cơ sở dữ liệu. Các ưu điểm của Metabase bao gồm miễn phí, dễ triển khai và sử dụng, hỗ trợ nhiều loại cơ sở dữ liệu như SQL, MySQL, và PostgreSQL, rất phù hợp cho các tổ chức nhỏ hoặc các dự án không yêu cầu tính năng phức tạp. Tuy nhiên, Metabase có một số hạn chế như tính năng còn giới hạn so với các công cụ thương mại, không tối ưu cho việc xử lý dữ liệu lớn hoặc xây dựng các dashboard phức tạp, và có cộng đồng hỗ trợ nhỏ hơn so với Apache Superset. Công cụ này phù hợp với các tổ chức nhỏ hoặc dự án chi phí thấp.

Apache Superset: Apache Superset là nền tảng trực quan hóa dữ liệu mã nguồn mở do Airbnb phát triển, hỗ trợ tạo các dashboard tương tác và tích hợp với nhiều nguồn dữ liệu khác nhau. Superset nổi bật nhờ khả năng mở rộng cao, tính linh hoạt trong tuỳ chỉnh, và hỗ trợ tốt cho hệ sinh thái Big Data. Công cụ này đặc biệt phù hợp với các tổ chức tìm kiếm một giải pháp chi phí thấp, có khả năng mở rộng và tích hợp tốt với hệ thống dữ liệu lớn và phân tán. Phân tích chi tiết về ưu điểm và nhược điểm của Superset sẽ được trình bày cụ thể trong phần đề xuất tiếp theo.

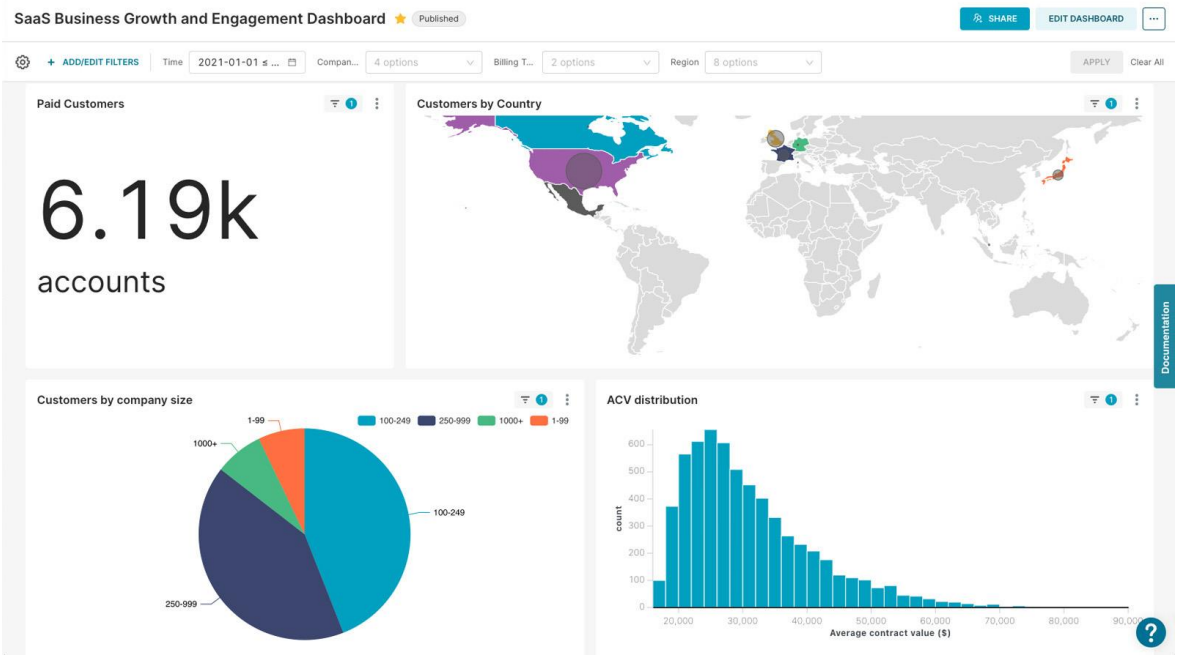
Grafana: Grafana là công cụ mã nguồn mở chuyên về trực quan hóa dữ liệu thời gian thực, thường được sử dụng trong các hệ thống giám sát và ứng dụng IoT. Ưu điểm chính của Grafana bao gồm khả năng hỗ trợ dữ liệu thời gian thực, tích hợp với các hệ thống như Prometheus và InfluxDB, giao diện đẹp và dễ tùy chỉnh, đồng thời là một công cụ miễn phí với cộng đồng hỗ trợ mạnh. Tuy nhiên, Grafana chủ yếu tập trung vào mục tiêu giám sát hệ thống hơn là phục vụ cho phân tích dữ liệu kinh doanh, và có hạn chế trong việc xây dựng các dashboard phức tạp hoặc đa dạng. Do đó, Grafana phù hợp hơn cho các ứng dụng giám sát hệ thống và IoT, nhưng không tối ưu cho các hệ thống chính phủ điện tử.

2.4.2. Đề xuất sử dụng Apache Superset để trực quan hóa dữ liệu

2.4.2.1. Giới thiệu Apache Superset

Apache Superset là một nền tảng trực quan hóa dữ liệu mã nguồn mở, được

phát triển bởi Airbnb vào năm 2015 và hiện được quản lý bởi Apache Software Foundation. Superset được thiết kế để phục vụ cả người dùng không chuyên thông qua giao diện kéo-thả và chuyên gia thông qua SQL hoặc Python [27]. Với khả năng tích hợp với các hệ thống dữ liệu lớn (như Apache Spark, Presto, MinIO) và hỗ trợ SQL, Superset là một giải pháp mạnh mẽ cho các tổ chức cần trực quan hóa dữ liệu với chi phí thấp và tính linh hoạt cao.



Hình 2.11. Giao diện trực quan hóa dữ liệu của Superset

Superset được thiết kế để phục vụ cả người dùng không chuyên (qua giao diện kéo-thả) và chuyên gia (qua SQL hoặc Python), phù hợp cho các ứng dụng chính phủ điện tử, doanh nghiệp, và phân tích dữ liệu lớn. Nó hỗ trợ các tính năng như khám phá dữ liệu, tạo dashboard, và chia sẻ báo cáo, đáp ứng nhu cầu từ cơ quan nhà nước đến doanh nghiệp tư nhân.

2.4.2.2. Kiến trúc

Kiến trúc của Apache Superset bao gồm các thành phần chính:

- Web Server: Chịu trách nhiệm cung cấp giao diện web cho người dùng cuối nhằm tạo, quản lý và xem các dashboard hoặc biểu đồ. Thành phần này thường sử dụng Flask (một framework Python nhẹ) để xử lý các yêu cầu HTTP, từ đó hiển thị giao diện người dùng một cách trực quan và dễ sử dụng.

- Application Server: Đóng vai trò xử lý logic nghiệp vụ của hệ thống. Nó thực hiện các tác vụ như truy vấn dữ liệu, quản lý người dùng, và kiểm soát quyền truy cập. Các công cụ như SQLAlchemy được sử dụng để kết nối và tương tác hiệu quả với các nguồn dữ liệu, đảm bảo tính linh hoạt và mở rộng trong truy xuất thông tin.

- Metadata Database: Là nơi lưu trữ tất cả thông tin cấu hình hệ thống, bao gồm dashboard, biểu đồ, truy vấn, và thông tin người dùng. Cơ sở dữ liệu này có thể được triển khai bằng các hệ quản trị như PostgreSQL, MySQL hoặc SQLite, tùy thuộc vào quy mô và yêu cầu của hệ thống.

- Query Engine: Là thành phần chịu trách nhiệm thực thi các truy vấn SQL hoặc truy xuất dữ liệu từ các nguồn như cơ sở dữ liệu quan hệ, Data Lake, hoặc kho dữ liệu lớn. Hệ thống có thể tích hợp với các công cụ mạnh như Presto, Druid hoặc Spark SQL để đảm bảo khả năng xử lý nhanh và linh hoạt với khối lượng dữ liệu lớn.

- Cache Layer: Giúp lưu trữ tạm thời các kết quả truy vấn nhằm tăng tốc độ tải dashboard và giảm áp lực lên hệ thống truy vấn. Các giải pháp phổ biến được sử dụng cho lớp cache bao gồm Redis, Memcached hoặc các hệ thống cache phân tán khác, tùy theo yêu cầu hiệu suất.

- Worker (Celery): Là thành phần đảm nhiệm việc xử lý các tác vụ nền không đồng bộ, chẳng hạn như xuất báo cáo, gửi email hoặc xử lý các truy vấn dài. Hệ thống thường sử dụng Celery để thực hiện các tác vụ này, đồng thời hỗ trợ triển khai phân tán với Kubernetes để tăng khả năng mở rộng và độ bền vững của hệ thống.

2.4.2.3. Luồng hoạt động

Khi người dùng gửi yêu cầu, Application Server sẽ chuyển truy vấn đến Query Engine để kết nối và truy xuất dữ liệu từ các nguồn như cơ sở dữ liệu SQL hoặc Datalake. Kết quả truy vấn sau đó có thể được lưu vào lớp Cache để tăng hiệu suất truy xuất, trước khi được hiển thị trên Web Server dưới dạng biểu đồ hoặc dashboard tương tác. Các tác vụ nền như xuất báo cáo, gửi thông báo hoặc làm mới dữ liệu định kỳ được xử lý bởi hệ thống Worker. Đồng thời, toàn bộ thông tin cấu hình, lịch sử dashboard và quyền truy cập được lưu trữ trong Metadata Database để đảm bảo tính nhất quán và khả năng quản trị hiệu quả.

2.4.2.4. Lý do sử dụng Apache Superset để trực quan hóa dữ liệu

Apache Superset là một công cụ trực quan hóa dữ liệu mạnh mẽ, đặc biệt hữu ích trong việc xử lý và trình bày dữ liệu đa cấu trúc từ kho dữ liệu, cho phép người dùng truy cập nhiều loại nguồn dữ liệu khác nhau trong cùng một hệ thống, từ đó tổng hợp và trực quan hóa thông tin một cách nhất quán và dễ hiểu.

Trong một trường đại học, nó có thể hỗ trợ tạo các dashboard quản trị hiển thị số liệu về tuyển sinh, tỷ lệ tốt nghiệp, kết quả học tập, phân bổ ngân sách, hiệu suất giảng dạy hoặc mức độ sử dụng nguồn lực. Những dashboard này giúp lãnh đạo dễ dàng đánh giá hiệu quả hoạt động và đưa ra quyết định điều hành kịp thời.

Ngoài ra, việc tích hợp Superset với các công cụ như Airflow giúp tự động hóa báo cáo định kỳ, giảm gánh nặng công việc cho cán bộ phụ trách việc phân tích dữ liệu.

2.4.3. Kết luận

Trực quan hóa dữ liệu là thành phần thiết yếu để hiển thị thông tin hỗ trợ lãnh đạo ra quyết định. Trong số các công nghệ khảo sát, Apache Superset nổi bật với tính mã nguồn mở, giao diện thân thiện, và khả năng tích hợp với các hệ thống như MinIO, Airflow, và X-Road [28]. Với các yếu tố kể trên, Superset phù hợp để triển khai dashboard để trực quan hóa dữ liệu được liên thông qua một trục liên thông dữ liệu trong một trường đại học.

CHƯƠNG 3. THỬ NGHIỆM VÀ ĐÁNH GIÁ

3.1. Chạy thực nghiệm mô hình đề xuất

3.1.1. Phạm vi thử nghiệm

Phạm vi thử nghiệm tập trung vào lĩnh vực xây dựng trục liên thông dữ liệu phục vụ công tác quản lý và điều hành tại một trường đại học. Trong bối cảnh giáo dục đại học đang chuyển mình mạnh mẽ theo hướng số hóa toàn diện, việc tích hợp, xử lý và khai thác dữ liệu từ các hệ thống chức năng khác nhau trong nội bộ nhà trường ngày càng trở nên cấp thiết. Hạ tầng dữ liệu trong thử nghiệm nhằm giải quyết bài toán liên thông dữ liệu giữa các đơn vị chức năng – từ phòng đào tạo, phòng khảo thí, phòng công tác sinh viên đến các khoa, viện và trung tâm – từ đó hỗ trợ ra quyết định dựa trên dữ liệu, nâng cao hiệu quả quản trị đại học, và cải thiện trải nghiệm của người học.

Cụ thể, các lĩnh vực được lựa chọn làm đối tượng thử nghiệm bao gồm: dữ liệu người học (hồ sơ sinh viên, kết quả học tập, tình trạng học vụ), dữ liệu y tế học đường (lịch sử tiêm chủng, khám sức khỏe định kỳ), dữ liệu khảo sát và phản hồi (đánh giá học phần, phản ánh sinh viên), và dữ liệu hành chính – dịch vụ công nội bộ (đăng ký học phần, xác nhận giấy tờ, giải quyết thủ tục hành chính) dung lượng dữ liệu là 12Mb. Đây là những lĩnh vực có đặc điểm dữ liệu đa nguồn, phân tán giữa nhiều hệ thống như SIS (Student Information System), LMS (Learning Management System), phần mềm y tế, phần mềm khảo sát, và các cổng dịch vụ trực tuyến nội bộ. Tính liên thông, chuẩn hóa và khai thác thống nhất các nguồn dữ liệu này sẽ góp phần nâng cao hiệu quả vận hành toàn trường.

Trong quá trình thử nghiệm, hệ thống tiếp nhận và xử lý dữ liệu với ba dạng cấu trúc chính. Dữ liệu có cấu trúc đến từ hệ quản trị cơ sở dữ liệu của SIS và hệ thống quản lý đào tạo, bao gồm thông tin sinh viên, điểm học phần, lịch sử học tập và các trạng thái học vụ. Dữ liệu bán cấu trúc được thu thập từ các hệ thống phản ánh, khảo sát nội bộ hoặc các biểu mẫu điện tử (Google Forms, Microsoft Forms), thường có định dạng JSON hoặc XML. Dữ liệu phi cấu trúc bao gồm các tệp văn bản, hình ảnh, tài liệu scan liên quan đến hồ sơ sinh viên, chứng từ y tế hoặc biên

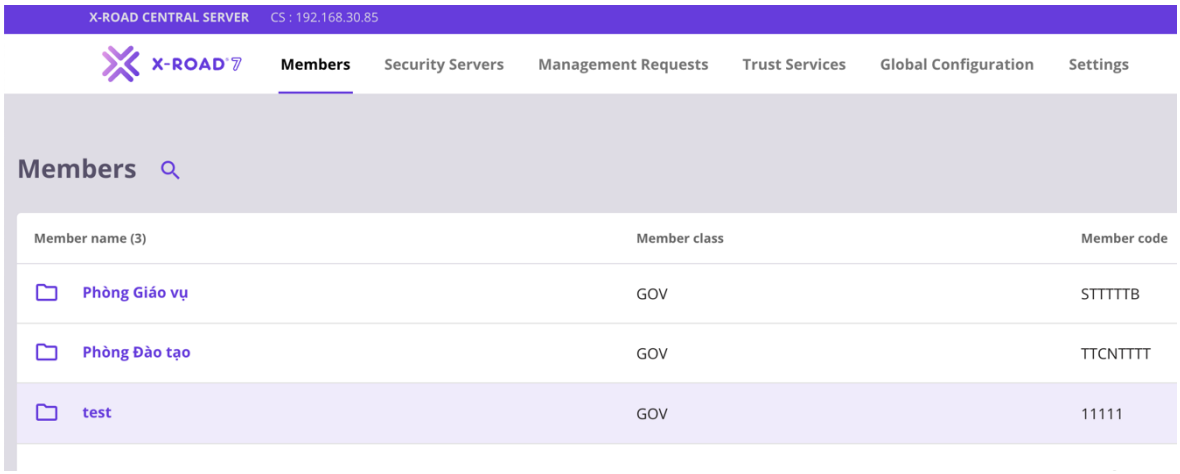
bản hành chính. Các nguồn dữ liệu này được kết nối vào hệ thống thông qua các thành phần chuyên biệt: DB Connector cho dữ liệu quan hệ, API Connector cho dữ liệu dạng RESTful, và File Connector cho dữ liệu tệp.

Dữ liệu sau khi thu nhận được lưu trữ trong hồ dữ liệu (Data Lake) sử dụng MinIO, đảm bảo khả năng lưu trữ lâu dài, linh hoạt và mở rộng. Tiếp theo, quy trình xử lý ETL được thiết kế và điều phối tự động bằng Apache Airflow, bao gồm các bước trích xuất, làm sạch, chuẩn hóa và nạp dữ liệu vào kho dữ liệu phân tích. Cuối cùng, hệ thống trực quan hóa bằng Apache Superset cho phép các đơn vị chức năng (như Phòng Đào tạo, Phòng Khảo thí, Khoa/Bộ môn) truy cập thông tin, theo dõi xu hướng học tập, phân tích hiệu suất đào tạo và hỗ trợ hoạch định chính sách.

Mục tiêu của giai đoạn thử nghiệm là đánh giá tính khả thi, hiệu quả và mức độ phù hợp của mô hình hạ tầng dữ liệu mã nguồn mở trong môi trường đại học. Thử nghiệm kiểm chứng khả năng liên thông dữ liệu giữa các hệ thống quản lý độc lập, tự động hóa quy trình xử lý dữ liệu học vụ và phản ánh sinh viên, đồng thời đánh giá hiệu quả trực quan hóa và truy vấn dữ liệu theo nhu cầu người dùng cuối. Ngoài ra, quá trình thử nghiệm cũng làm rõ khả năng lưu trữ và tổ chức dữ liệu phi cấu trúc liên quan đến học sinh – sinh viên, một yếu tố quan trọng thường bị bỏ quên trong các hệ thống truyền thống. Kết quả thử nghiệm sẽ là cơ sở thực nghiệm quan trọng để nhà trường triển khai mô hình hạ tầng dữ liệu toàn diện, làm nền tảng cho các ứng dụng nâng cao như phân tích học tập (learning analytics), hỗ trợ học tập cá nhân hóa, và tích hợp trí tuệ nhân tạo vào hoạt động quản trị đại học trong tương lai.

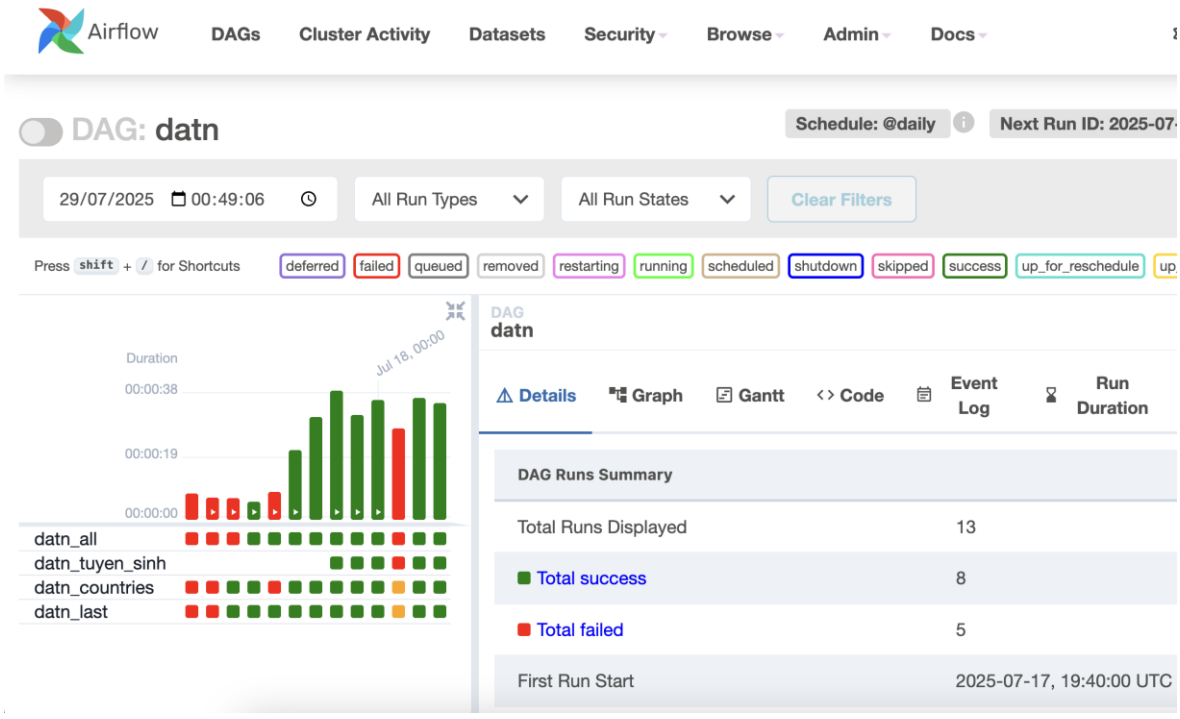
3.1.2. Cài đặt và cấu hình các thành phần

Học viên tiến hành cài đặt thử nghiệm trục liên thông dữ liệu X-Road gồm 1 Central Server và 2 Security Server đóng vai trò là Phòng giáo vụ và Phòng đào tạo trong một trường đại học.



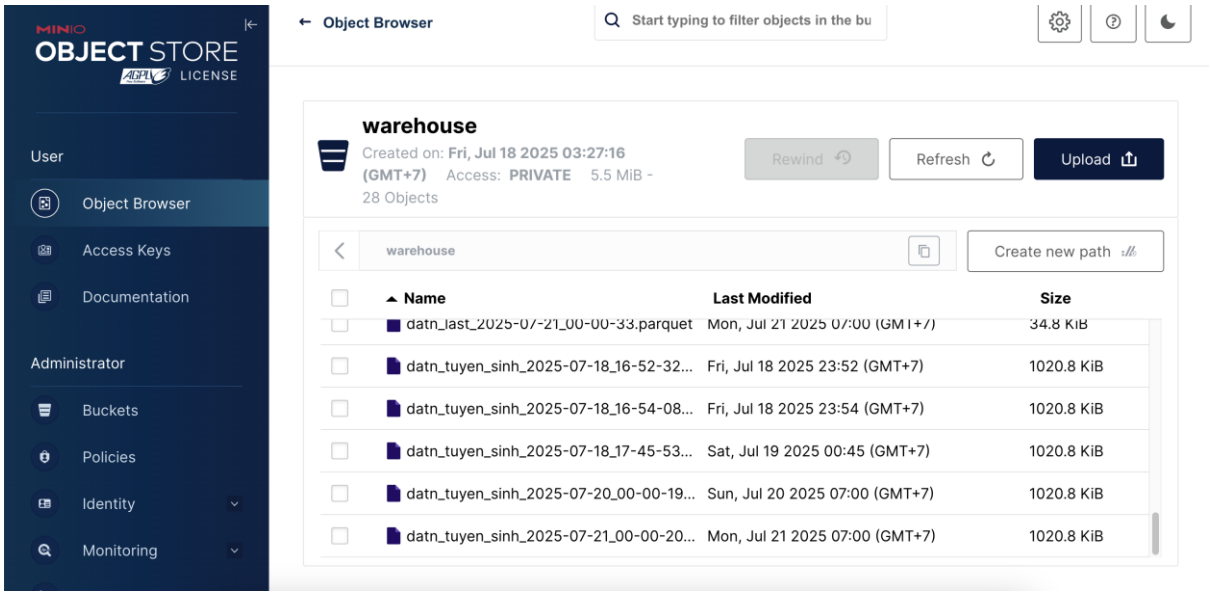
Hình 3.1. Giao diện Central Server

Tiếp theo, Học viên tiến hành cài đặt Airflow bằng Docker Compose, cấu hình DAG ETL gồm trích xuất dữ liệu từ các API hoặc tệp JSON/CSV sau đó làm sạch và chuyển đổi định dạng (ví dụ: đổi ngày/tháng, chuẩn hóa tên trường). Cuối cùng ghi dữ liệu đã xử lý vào MinIO theo cấu trúc thư mục phân tầng theo năm/lớp.



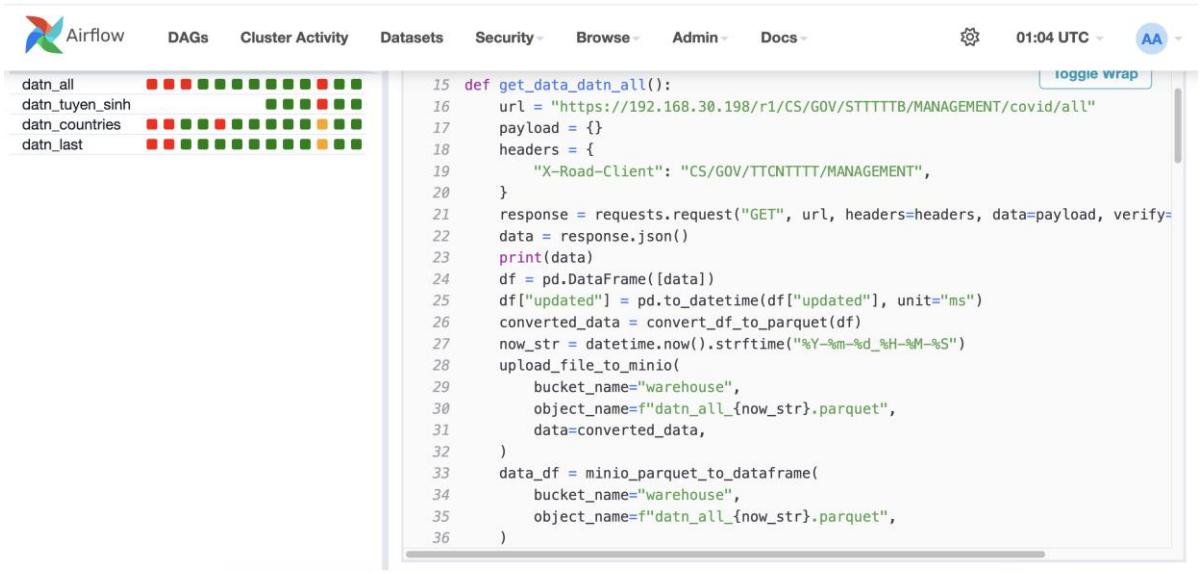
Hình 3.2. Giao diện ứng dụng Airflow

Để lưu dữ liệu vào datalake, Học viên tiến hành khởi tạo bucket chính: warehouse. Cấu hình phân quyền đọc/ghi, kích hoạt versioning cho bucket.



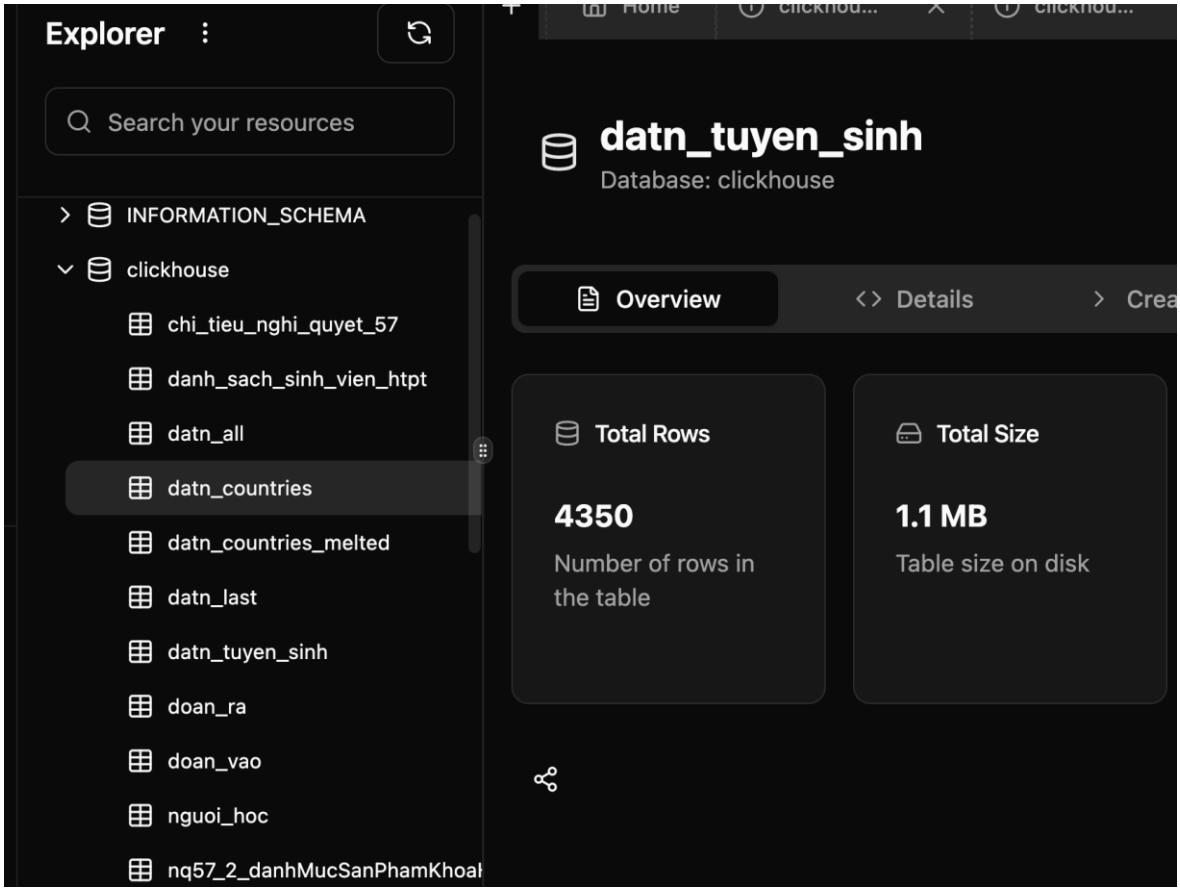
Hình 3.3. Lưu trữ các dữ liệu trong hồ dữ liệu MinIO

Tiếp theo đó, Học viên tiến hành viết đoạn mã lệnh cho công cụ Airflow để thực hiện quá trình ETL.



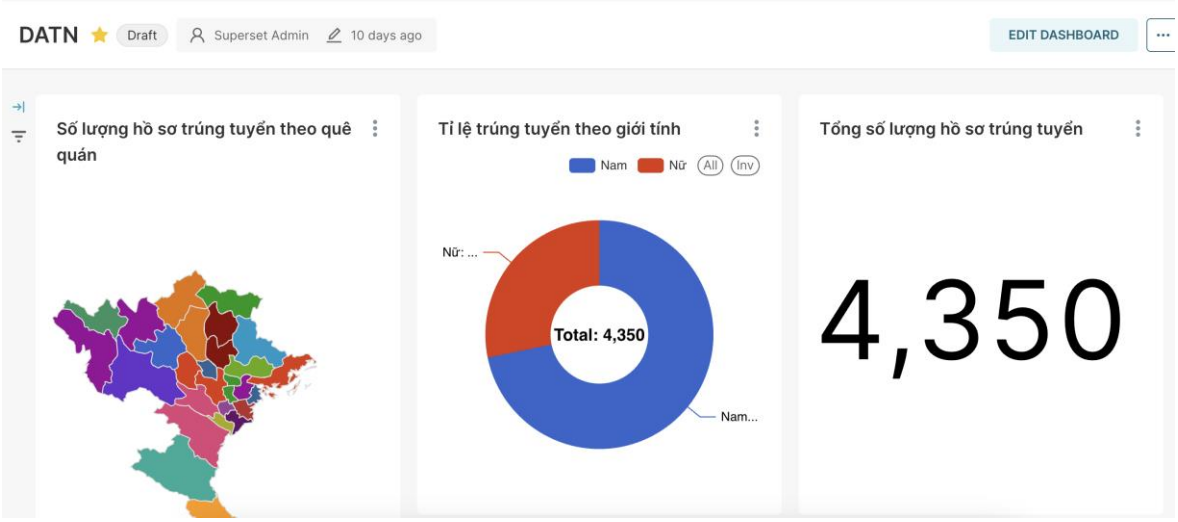
Hình 3.4. Các câu lệnh giúp thực hiện quá trình ETL trong Airflow

Dữ liệu sau khi được biến đổi sẽ được lưu trữ trong kho dữ liệu ClickHouse giúp tối ưu cho các truy vấn phân tích với khối lượng dữ liệu lớn.



Hình 3.5. Dữ liệu được lưu trữ trong kho dữ liệu

Cuối cùng, học viên tiến hành cài đặt Apache superset kết nối với kho dữ liệu Clickhouse để tiến hành lấy dữ liệu để trực quan hóa thành các biểu đồ, số liệu.

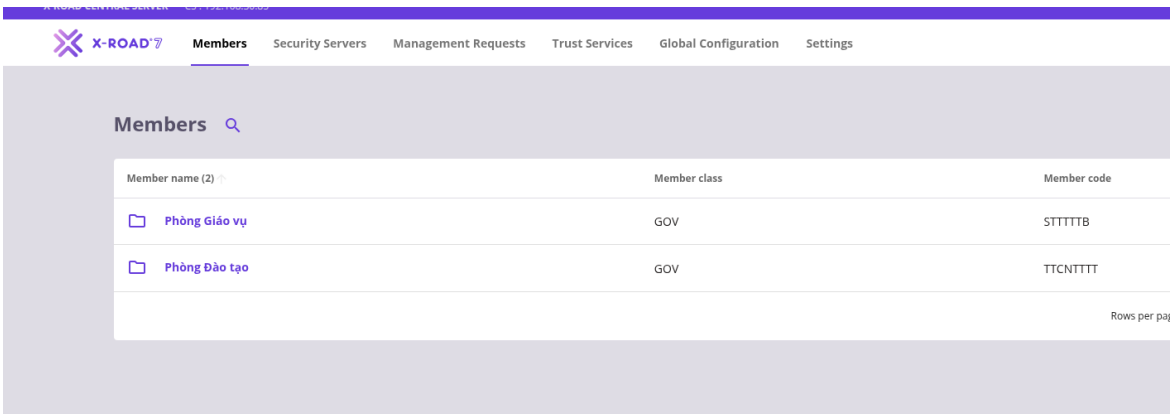


Hình 3.6. Giao diện trực quan hóa dữ liệu của Superset

3.2. Các kết quả thực nghiệm mang lại.

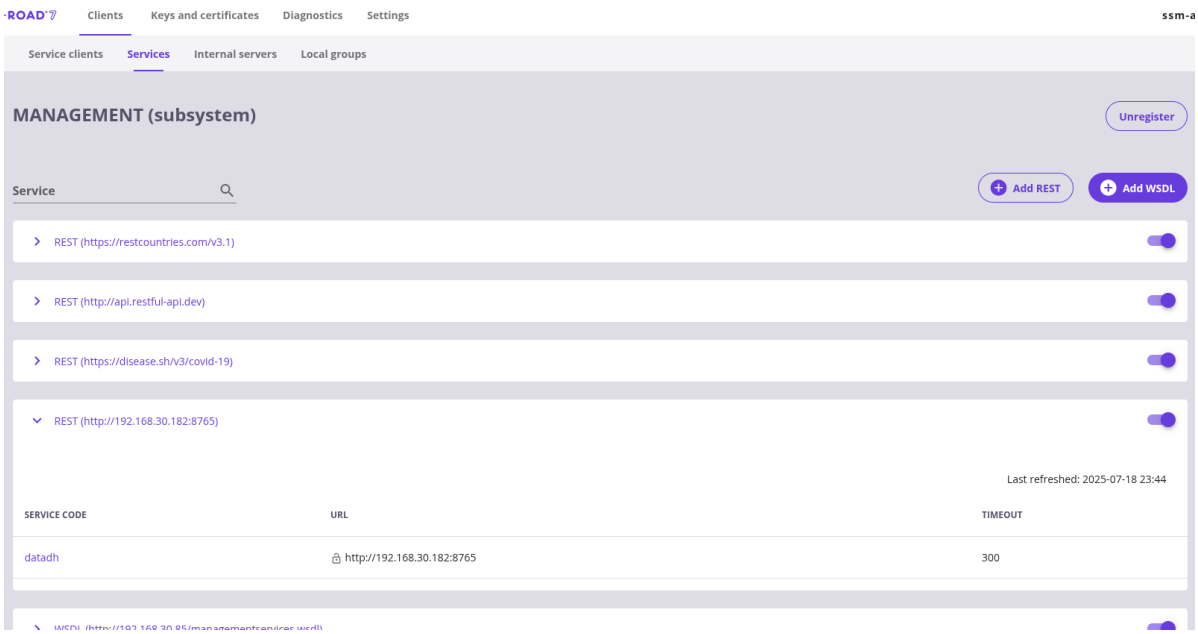
Qua quá trình thử nghiệm triển khai mô hình trục liên thông dữ liệu trong môi trường mô phỏng một trường đại học, đề án đã thu được những kết quả khả quan, chứng minh tính khả thi và hiệu quả của giải pháp đề xuất. Các kết quả được phân tích theo từng thành phần chính của hệ thống:

Kết nối thành công giữa các đơn vị: Hệ thống X-Road đã thiết lập thành công kết nối liên thông giữa hai đơn vị mô phỏng (Phòng Giáo vụ và Phòng Đào tạo) thông qua Central Server và hai Security Server. Quá trình đăng ký thành viên, cấu hình dịch vụ và xác thực hoạt động ổn định.



Hình 3.7. Danh sách các đơn vị liên thông qua trục

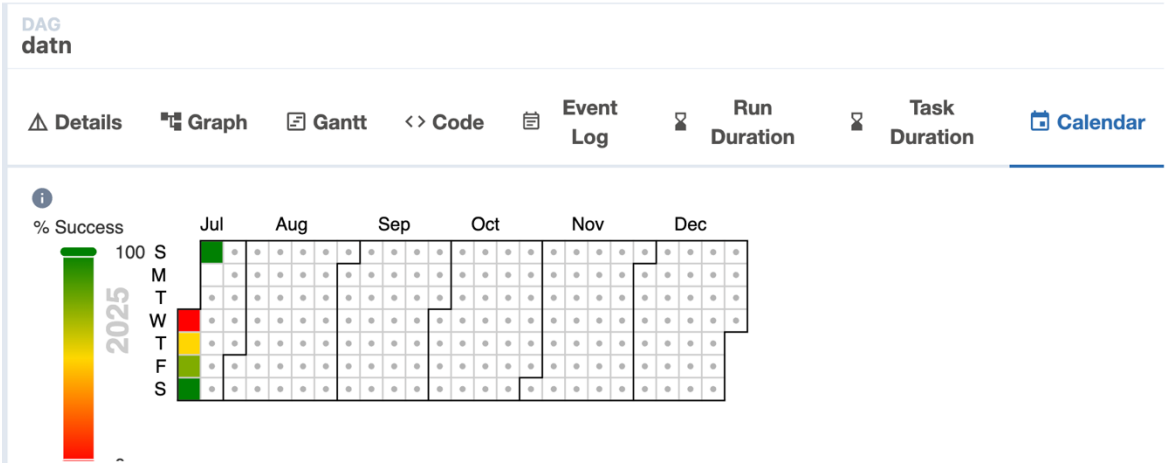
Giao diện quản trị cho phép dễ dàng đăng ký, cấu hình và giám sát các dịch vụ chia sẻ dữ liệu. Mỗi dịch vụ được định nghĩa rõ ràng với các tham số đầu vào, đầu ra và quyền truy cập tương ứng. Mỗi dịch vụ được triển khai dưới dạng API endpoint chuẩn, cho phép các hệ thống khác nhau tương tác một cách đồng nhất thông qua giao thức SOAP hoặc REST.



Hình 3.8. Danh sách các dịch vụ được chia sẻ qua trục

Apache Airflow đã thực hiện thành công việc tự động hóa các luồng công việc ETL (Extract, Transform, Load), từ việc kết nối đến các nguồn dữ liệu khác nhau, xử lý và chuyển đổi dữ liệu, đến việc lưu trữ vào hồ dữ liệu MinIO.

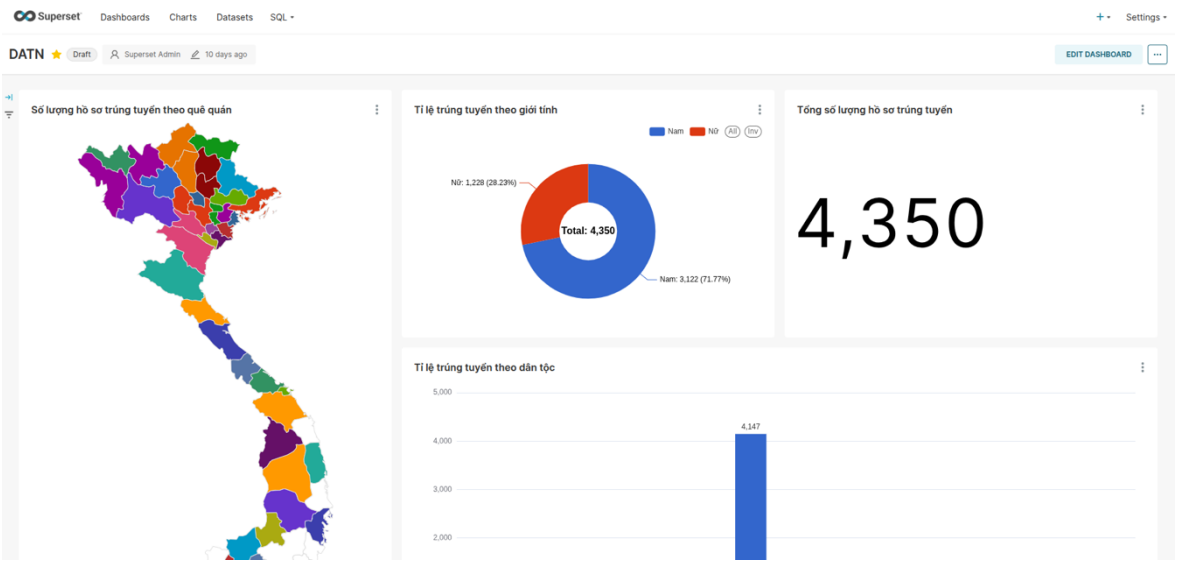
Các DAG (Directed Acyclic Graph) được thiết kế để xử lý dữ liệu đã vận hành ổn định với tỷ lệ thành công cao. Hệ thống ghi nhận lịch sử thực thi chi tiết, giúp theo dõi và khắc phục sự cố khi cần thiết.



Hình 3.9. Lịch sử thực hiện quá trình ETL

Quá trình làm sạch và chuẩn hóa dữ liệu hoạt động hiệu quả. Dữ liệu sau khi xử lý được lưu trữ thành công vào kho dữ liệu ClickHouse với cấu trúc được tối ưu cho phân tích.

Apache Superset đã tạo thành công các dashboard tương tác, cho phép người dùng xem và phân tích dữ liệu theo nhiều chiều khác nhau. Giao diện trực quan giúp lãnh đạo nhà trường dễ dàng nắm bắt thông tin.



Hình 3.10. Dữ liệu được trực quan hóa trên Apache superset

Ngoài ra Apache Superset cũng hỗ trợ tạo nhiều loại biểu đồ khác nhau như biểu đồ cột, biểu đồ tròn, biểu đồ đường, bản đồ nhiệt, phù hợp với từng loại dữ liệu và nhu cầu phân tích cụ thể.

Kết quả thử nghiệm cho thấy mô hình trục liên thông dữ liệu đề xuất hoàn toàn khả thi và mang lại hiệu quả rõ rệt trong việc liên thông và lưu trữ dữ liệu đa cấu trúc tại môi trường đại học. Mô hình hệ thống trục liên thông dữ liệu còn tạo nền tảng vững chắc cho các ứng dụng phân tích dữ liệu và hỗ trợ ra quyết định trong tương lai.

3.3. Các vấn đề cần cải thiện trong tương lai

Dù hệ thống thử nghiệm đã vận hành ổn định ở mức độ giả lập, một số điểm hạn chế và tiềm năng cải thiện đã được ghi nhận trong quá trình thực nghiệm:

Khả năng mở rộng quy mô (scalability): Hiện tại, hệ thống được triển khai trên môi trường thử nghiệm với dữ liệu và tần suất xử lý còn giới hạn. Khi áp dụng vào thực tế với lượng dữ liệu lớn hơn cần kiểm thử khả năng mở rộng của Airflow, MinIO và X-Road, đồng thời đánh giá khả năng chịu tải và thời gian xử lý thực tế của các module DAG ETL.

Tính sẵn sàng và dự phòng (high availability): Các thành phần hiện đang được triển khai theo cấu hình đơn (single instance), chưa có cơ chế dự phòng (failover) hoặc cân bằng tải (load balancing). Trong môi trường triển khai thực tế, cần thiết lập hệ thống HA cho các thành phần quan trọng như X-Road Security Server, Airflow Scheduler và MinIO.

Tự động hóa và kiểm thử dữ liệu (data validation): Quy trình làm sạch dữ liệu hiện mới ở mức cơ bản (chuyển đổi định dạng, chuẩn hóa tên). Trong tương lai, cần bổ sung các bước kiểm thử chất lượng dữ liệu (data quality checks), phát hiện thiếu giá trị, dữ liệu bất thường hoặc trùng lặp trước khi lưu vào hệ thống.

Quản lý quyền truy cập chi tiết hơn: Dù đã có phân quyền truy cập metadata theo nhóm người dùng trong OpenMetadata, hiện vẫn chưa tích hợp toàn bộ hệ thống với cơ chế xác thực tập trung (ví dụ: LDAP, OAuth2). Việc quản lý phân

quyền chi tiết theo vai trò người dùng trong từng lớp học, từng loại dữ liệu sẽ cần hoàn thiện thêm.

Giám sát và cảnh báo: Cần bổ sung các công cụ giám sát (như Prometheus hoặc Grafana) để theo dõi sức khỏe hệ thống, dung lượng lưu trữ, và trạng thái các DAG. Việc tích hợp cảnh báo (alerts) sẽ giúp phát hiện sớm lỗi hệ thống hoặc dữ liệu.

KẾT LUẬN

Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ trên toàn cầu, đặc biệt trong lĩnh vực giáo dục, việc xây dựng một hạ tầng công nghệ hiệu quả để quản lý và khai thác dữ liệu trở thành một yêu cầu cấp thiết. Thông qua nghiên cứu này, đề án đã làm rõ tầm quan trọng của dữ liệu đa cấu trúc trong đại học số, đồng thời chỉ ra những thách thức trong việc liên thông, lưu trữ và phân tích các loại dữ liệu không đồng nhất đang tồn tại phân tán trên nhiều hệ thống khác nhau.

Việc đề xuất và xây dựng một trục liên thông dữ liệu dựa trên nền tảng như X-Road không chỉ mang lại khả năng kết nối liền mạch giữa các hệ thống thông tin mà còn tạo điều kiện thuận lợi để khai thác tối đa giá trị của dữ liệu. Trục liên thông đóng vai trò quan trọng trong việc đồng bộ hóa, bảo mật và chuẩn hóa dữ liệu, từ đó nâng cao hiệu quả trong công tác quản lý, giảng dạy, nghiên cứu.

Ngoài ra, việc kết hợp khả năng trực quan hóa và phân tích dữ liệu sẽ giúp các trường đại học có cái nhìn toàn diện hơn về hoạt động của mình, từ đó đưa ra các chính sách điều hành hiệu quả, linh hoạt và minh bạch hơn. Kết quả nghiên cứu cũng cho thấy tiềm năng to lớn của việc áp dụng trục liên thông dữ liệu tại các trường đại học.

Đây là một đề án khá mới nên những ý kiến và thiết kế trục liên thông dữ liệu của đề án mới chỉ ở mức tương đối. Hơn nữa, do giới hạn về thời gian, điều kiện công tác nên không tránh khỏi những hạn chế và khuyết điểm. Tác giả rất mong nhận được sự đóng góp quý báu của các nhà khoa học, quý thầy cô để đề án được hoàn thiện hơn.

DANH MỤC TÀI LIỆU THAM KHẢO

[1] Bộ Thông tin và Truyền thông, *Chương trình chuyển đổi số quốc gia đến năm 2025, định hướng đến năm 2030*, Hà Nội, Việt Nam: Chính phủ Việt Nam, 2020. [Trực tuyến]. https://chinhphu.vn/default.aspx?pageid=27160&docid=200163

[2] Thomas M. Siebel (2019). Chuyển đổi số (Digital Transformation). Phạm Anh Tuấn dịch. Nxb Tổng hợp Thành phố Hồ Chí Minh.

[3] E. Nurseitov, M. Paulson, R. Reynolds, and C. Izurieta, "Comparison of JSON and XML data interchange formats: A case study," *Caine 2009 - Proceedings of the 22nd International Conference on Computer Applications in Industry and Engineering*, pp. 157–162, 2009.

[4] M. Zaharia et al., "Apache Spark: A unified engine for big data processing," *Communications of the ACM*, vol. 59, no. 11, pp. 56–65, Nov. 2016.

[5] Mô hình X-Road trong chính phủ điện tử tại Estonia, 2019. [Trực tuyến]. https://www.antoanthongtin.vn/tin/mo-hinh-x-road-trong-chinh-phu-dien-tu-tai-estonia.

[6] A. Ansper, "E-State from a Data Security Perspective," *Cybernetica*, 2001.

[7] Văn phòng Chính phủ, *Lễ khai trương Trục liên thông văn bản quốc gia*, Cổng Thông tin điện tử Chính phủ, Hà Nội, 2019. [Trực tuyến]. https://vpcp.chinhphu.vn/thu-tuong-khai-truong-truc-lien-thong-van-ban-quoc-gia-11521791.htm

[8] Văn phòng Chính phủ, *Phát biểu của Thủ tướng tại Lễ khai trương Trục liên thông văn bản quốc gia*, Cổng TTĐT Chính phủ, Hà Nội, 2019. [Trực tuyến]. https://vpcp.chinhphu.vn/thu-tuong-khai-truong-truc-lien-thong-van-ban-quoc-gia-11521791.htm

[9] Forrester Research, “Data fragmentation hinders productivity,” 2019. [Trực tuyến]. https://go.forrester.com

[10] Deloitte, “The data opportunity: Driving growth through data-powered transformation,” *Deloitte Insights*, 2020. [Trực tuyến]. https://www2.deloitte.com

[11] R. S. Baker and K. Yacef, "The state of educational data mining in 2009: A review and future visions," *Journal of Educational Data Mining*, vol. 1, no. 1, pp. 3–17, 2009.

[12] J. Zhang, Z. Wang, Y. Wang, and Y. Li, "Real-time video emotion recognition based on deep learning for online learning," *IEEE Access*, vol. 8, pp. 23968–23976, 2020.

[13] W. Fan and A. Bifet, "Mining big data: current status, and forecast to the future," *ACM SIGKDD Explorations Newsletter*, vol. 14, no. 2, pp. 1–5, Dec. 2013.

[14] IBM Corporation, *Analytics: The real-world use of big data in education*, Executive Report, IBM Institute for Business Value, 2013. [Trực tuyến]. https://www.bdvc.nl/images/Rapporten/GBE03519USEN.PDF

[15] M. P. Papazoglou and W. J. van den Heuvel, "Service oriented architectures: approaches, technologies and research issues," *The VLDB Journal*, vol. 16, no. 3, pp. 389-415, 2007.

[16] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*, Upper Saddle River, NJ: Prentice Hall PTR, 2005.

[17] D. McCandless, *Information is Beautiful*, London, UK: HarperCollins, 2012.

[18] P. Kivimäki, "X-Road Technical Architecture and Development," *Nordic Institute for Interoperability Solutions (NIIS)*, Technical Report, 2018.

[19] Talend Inc., "Talend Open Studio for Data Integration User Guide," *Technical Documentation*, 2024.

[20] Hitachi Vantara, "Pentaho Data Integration Documentation," *Pentaho Community Documentation*, 2024.

[21] Apache Software Foundation, "Apache Airflow Documentation," *Apache Airflow Project*, 2025.

[22] Apache Software Foundation, "Apache Airflow Documentation," *Apache Airflow Project*, 2025.

[23] M. Smith et al, "An Empirical Study of Developers' Challenges in Implementing Workflows as Code: A Case Study on Apache Airflow," *arXiv preprint*, arXiv:2406.00180, 2024.

[24] Apache Software Foundation, “Superset documentation,” [Trực tuyến]. https://superset.apache.org

[25] MinIO Inc., "MinIO Research Reveals the Impact of AI on Storage Infrastructure," *PR Newswire*, 10 tháng 12, 2024.

[26] Nordic Institute for Interoperability Solutions (NIIS), “X-Road overview,” [Trực tuyến]. https://x-road.global

[27] Apache Software Foundation, "Superset documentation," [Trực tuyến]. https://superset.apache.org

[28] MinIO, Inc., “MinIO documentation,” [Trực tuyến]. https://docs.min.io

BẢN CAM ĐOAN

Tôi cam đoan đã thực hiện việc kiểm tra mức độ tương đồng nội dung đề án tốt nghiệp qua phần mềm *https://kiemtratailieu.vn/* một cách trung thực và đạt kết quả mức độ tương đồng **9%** toàn bộ nội dung đề án tốt nghiệp. Bản đề án tốt nghiệp kiểm tra qua phần mềm là bản cứng đề án tốt nghiệp đã nộp để bảo vệ trước hội đồng. Nếu sai tôi xin chịu các hình thức kỷ luật theo quy định hiện hành của Học Viện.

Hà Nội, ngày 31 tháng 07 năm 2025

Học viên thực hiện đề án

Trần Quang Đại

KiểmTraTàiLiệu

BÁO CÁO KIỂM TRA TRÙNG LẶP

Thông tin tài liệu

Tên tài liệu:	DeAn
Tác giả:	Quang Đại Trần
Điểm trùng lặp:	9
Thời gian tải lên:	12:11 31/07/2025
Thời gian sinh báo cáo:	12:12 31/07/2025
Các trang kiểm tra:	70/70 trang

Kết quả kiểm tra trùng lặp

Có 9% nội dung trùng lặp

Có 91% nội dung không trùng lặp

Có 0% nội dung người dùng loại trừ

Có 0% nội dung hệ thống bỏ qua

Nguồn trùng lặp tiêu biểu

123docz.net tailieu.vn interdata.vn

Hà Nội, ngày tháng năm 2025

Ký xác nhận của

NGƯỜI HƯỚNG DẪN KHOA HỌC

TS. Phan Lý Huỳnh

HỌC VIÊN

Trần Quang Đại

BỘ KHOA HỌC VÀ CÔNG NGHỆ
HỌC VIỆN CÔNG NGHỆ
BƯU CHÍNH VIỄN THÔNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BÁO CÁO GIẢI TRÌNH
SỬA CHỮA, HOÀN THIỆN ĐỀ ÁN TỐT NGHIỆP

Họ và tên học viên: Trần Quang Đại
Chuyên ngành: Hệ thống thông tin
Khóa: 2023 đợt 1

Tên đề tài: NGHIÊN CỨU, XÂY DỰNG TRỤC LIÊN THÔNG DỮ LIỆU PHỤC VỤ LƯU TRỮ, QUẢN LÝ DỮ LIỆU ĐA CẤU TRÚC

Người hướng dẫn khoa học: TS. Phan Lý Huỳnh

Ngày bảo vệ: 19/07/2025

Các nội dung học viên đã sửa chữa, bổ sung trong đề án tốt nghiệp theo ý kiến đóng góp của Hội đồng chấm đề án tốt nghiệp:

TT	Ý kiến hội đồng	Sửa chữa của học viên
1	Hoàn chỉnh theo ý kiến Hội đồng, phản biện	Học viên đã nghiêm túc tiếp thu và hoàn thiện đề án theo các ý kiến của Hội đồng chấm đề án tốt nghiệp.
2	Chỉnh sửa nội dung Chương 1,2 cho phù hợp.	Học viên đã chỉnh sửa nội dung của chương 1, chương 2 cho phù hợp với đề án.
3	Chỉnh sửa lỗi chính tả, hình vẽ rất mờ, thiếu trích dẫn.	Học viên đã tìm kiếm và sửa lại các lỗi chính tả còn tồn tại trong đề án, thay đổi các hình vẽ bị mờ thành các hình vẽ rõ nét. Học viên cũng đã bổ sung các trích dẫn [6], [15], [16], [18], [19], [20], [21], [22], [23], [25], [27].
4	Đa số các kỹ thuật, công nghệ, công cụ được sử dụng chưa được khảo sát, đánh giá đầy đủ. Đề án chỉ khảo sát trục liên thông X-Road mà không khảo sát các trục liên thông khác.	Tiếp thu ý kiến góp ý của Hội đồng, Học viên đã bổ sung thêm khảo sát các loại trục liên thông khác trong mục 2.2.1.
5	Các khảo sát các kỹ thuật, công nghệ, công cụ chỉ gồm văn bản, thiếu các sơ đồ, hình vẽ nên tính diễn giải kém.	Học viên đã bổ sung thêm các sơ đồ, hình vẽ khi khảo sát các kỹ thuật, công nghệ tại mục 2.1, 2.2, 2.3, 2.4.
6	Mục 2.1.1 đề xuất mô hình tổng quan hệ thống	Tiếp thu ý kiến góp ý của Hội đồng,

	cũng rất mơ hồ, do tác giả không nêu rõ mô hình này sử dụng cho hệ thống nào, đề xuất mới hay tham khảo.	Học viên đã chỉnh sửa và nêu rõ việc đề xuất mô hình tại mục 2.2.1.
7	Chương 3 mô tả việc triển khai thử nghiệm trục liên thông dữ liệu sử dụng X-Road cho PTIT, nhưng không có nội dung khảo sát, đánh giá hiện trạng, trên cơ sở đó đưa ra mô hình triển khai cho phù hợp.	Học viên đã bổ sung phần khảo sát, đánh giá hiện trạng tại Học viện Công nghệ Bưu Chính Viễn Thông để đưa ra mô hình triển khai phù hợp tại mục 3.1.
8	Sử dụng nhiều thông tin, công nghệ, nền tảng, nhưng không có trích dẫn từ nguồn nào.	Tiếp thu ý kiến góp ý của Hội đồng. Học viên đã bổ sung thêm các trích dẫn [6], [15], [16], [18], [19], [20], [21], [22], [23], [25], [27]
9	Quá lạm dụng cách viết theo chấm đầu dòng, nên tạo cảm giác lắt nhắt cho người đọc.	Học viên đã loại bỏ các dấu chấm đầu dòng không hợp lý tại chương 2 và chương 3.
10	Đề án có nhiều hình mờ , chữ nhỏ , không thể đọc (nhất là ở chương 3).	Học viên đã sửa các hình vẽ bị mờ, chữ bị bé không thể đọc thành các hình vẽ rõ nét, kích thước chữ lớn hơn.
11	Các tài liệu tham khảo thiếu thông tin.	Học viên đã cập nhật các trích dẫn từ các nguồn các nghiên cứu, bài báo khoa học, trang chủ của nền tảng.
12	Làm rõ về nguồn, dung lượng dữ liệu được sử dụng trong phần thực nghiệm 3.1.	Học viên đã bổ sung dung lượng dữ liệu là 12Mb dữ liệu đa cấu trúc tại mục 3.1.
13	Phân tích tác dụng của công cụ Apache Superset trong quá trình trực quan hóa dữ liệu đa cấu trúc? Công cụ Apache Superset hỗ trợ gì trong công tác quản lý, điều hành tại các trường đại học?	Học viên đã bổ sung các tác dụng của công cụ Apache Superset tại mục 2.4.

Hà Nội, ngày tháng năm 2025

Ký xác nhận của

CHỦ TỊCH HỘI ĐỒNG CHẤM ĐỀ ÁN	THƯ KÝ HỘI ĐỒNG	NGƯỜI HƯỚNG DẪN KHOA HỌC	HỌC VIÊN
PGS.TSKH. Hoàng Đăng Hải	TS. Đào Ngọc Phong	TS. Phan Lý Huỳnh	Trần Quang Đại

BỘ KHOA HỌC VÀ CÔNG NGHỆ
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BIÊN BẢN
HỌP HỘI ĐỒNG CHẤM ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

Căn cứ quyết định số Quyết định số 1098/QĐ-HV ngày 26 tháng 06 năm 2025 của Giám đốc Học viện Công nghệ Bưu chính Viễn thông về việc thành lập Hội đồng chấm đề án tốt nghiệp thạc sĩ. Hội đồng đã họp vào hồi...10..giờ..40..phút, ngày 19 tháng 07 năm 2025 tại Học viện Công nghệ Bưu chính Viễn thông để chấm đề án tốt nghiệp thạc sĩ cho:

Học viên: **Trần Quang Đại**

Tên đề án tốt nghiệp: **Nghiên cứu, xây dựng trục liên thông dữ liệu phục vụ lưu trữ, quản lý dữ liệu đa cấu trúc**

Chuyên ngành: **Hệ thống thông tin** Mã số: **8480104**

Các thành viên của Hội đồng chấm đề án tốt nghiệp có mặt: ..05../ 05

TT	HỌ VÀ TÊN	TRÁCH NHIỆM TRONG HĐ	GHI CHÚ
1	**PGS.TSKH. Hoàng Đăng Hải**	Chủ tịch	
2	**TS. Đào Ngọc Phong**	Thư ký	
3	**PGS.TS. Hoàng Xuân Dậu**	Phản biện 1	
4	**TS. Bùi Hải Phong**	Phản biện 2	
5	**PGS.TS. Trần Nguyên Ngọc**	Uỷ viên	

Các nội dung thực hiện:

1. Chủ tịch Hội đồng điều khiển buổi họp. Công bố quyết định của Giám đốc Học viện Công nghệ Bưu chính Viễn thông về việc thành lập Hội đồng chấm đề án tốt nghiệp thạc sĩ.
2. Người hướng dẫn khoa học hoặc thư ký đọc lý lịch khoa học và các điều kiện bảo vệ đề án tốt nghiệp của học viên. (có bản lý lịch khoa học và kết quả các môn học cao học của học viên kèm theo).
3. Học viên trình bày tóm tắt đề án tốt nghiệp.
4. Phản biện 1 đọc nhận xét (có văn bản kèm theo)
5. Phản biện 2 đọc nhận xét (có văn bản kèm theo)
6. Các câu hỏi của thành viên Hội đồng:

1/ Có những loại thực vật, khác
2/ Việc sử dụng thực có cần thiết ở quy mô 1 đơn vị
3/ Bảo mật như thế nào
4/ Làm rõ việc liên thông
5/ Cấu trúc thực quan hóa hiểu thế nào

7. Trả lời của học viên:

1/ Xin bổ sung vào đề án
2/ Tiếp thu bổ sung vào đề án
3/ Sử dụng chữ ký số
4/ Dữ liệu không > 1 MB
5/ Hỗ trợ tạo báo cáo, thống kê

8. Thư ký đọc nhận xét về quá trình thực hiện đề án tốt nghiệp của học viên *(có văn bản kèm theo).*

9. Hội đồng họp riêng:

- Bầu Ban kiểm phiếu:
  1. Trưởng Ban kiểm phiếu: Đào Ngọc Phong
  2. Uỷ viên Ban kiểm phiếu: Bùi Hải Phong
  3. Uỷ viên Ban kiểm phiếu: Trần Nguyên Ngọc
- Hội đồng chấm đề án tốt nghiệp bằng bỏ phiếu kín.
- Ban kiểm phiếu làm việc:
- Trưởng Ban kiểm phiếu báo cáo kết quả kiểm phiếu *(có Biên bản họp Ban kiểm phiếu kèm theo)*
- **Điểm trung bình của đề án tốt nghiệp:** 7.5

Kết luận:

1. Các nội dung cần chỉnh sửa, hoàn thiện sau bảo vệ đề án tốt nghiệp:

1/ Hoàn chỉnh theo ý kiến Hội đồng, Phản biện
2/ Chỉnh sửa nội dung Chương 1, 2 cho phù hợp
3/ Chỉnh sửa lỗi chính tả, hình vẽ rõ nét mờ, thuyết trình dẫn

2. Đề nghị Học viện công nhận (hoặc không) và cấp bằng (hoặc không) thạc sĩ cho học viên:
công nhận và cấp bằng

3. Đề án tốt nghiệp có thể phát triển thành đề tài nghiên cứu cho NCS.....không.....

Buổi làm việc kết thúc vào..11h25.. cùng ngày.

Chủ tịch	**Thư ký**
PGS.TSKH. Hoàng Đăng Hải	TS. Đào Ngọc Phong

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ
BƯU CHÍNH VIỄN THÔNG

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc
****

BẢN NHẬN XÉT ĐỀ ÁN TỐT NGHIỆP THẠC SỸ

(Dùng cho người phản biện)

Tên đề tài đề án: Nghiên cứu, xây dựng trục liên thông dữ liệu phục vụ lưu trữ, quản lý dữ liệu đa cấu trúc

Chuyên ngành: Hệ thống thông tin

Mã số: 8.48.01.04

Tên học viên: Trần Quang Đại

Họ và tên người nhận xét: Hoàng Xuân Dậu

Học hàm, học vị: PGS. Tiến sỹ, Giảng viên cao cấp

Chuyên ngành: Khoa học máy tính

Cơ quan công tác: Khoa ATTT, Học viện Công nghệ BC-VT

NỘI DUNG NHẬN XÉT

I/ Cơ sở khoa học và thực tiễn, tính cấp thiết của đề tài:

- Đề tài có ý nghĩa thực tiễn trong việc phát triển các hệ thống liên thông, chia sẻ dữ liệu.

II/ Về nội dung, chất lượng của đề án, các kết quả đạt được (so với đề cương đã được duyệt):

*** Nội dung đề án:**

Đề án tốt nghiệp của học viên gồm 3 chương chính:

Chương 1 giới thiệu tổng quan về vấn đề liên thông dữ liệu, trục liên thông dữ liệu; khái quát về dữ liệu đa cấu trúc, vấn đề xử lý và quản lý dữ liệu đa cấu trúc.

Chương 2 đề xuất mô hình và các thành phần của trục liên thông dữ liệu.

Chương 3 mô tả việc triển khai thử nghiệm trục liên thông dữ liệu sử dụng X-Road.

*** Chất lượng đề án:** Trung bình

*** Các kết quả đạt được:** cơ bản đạt các kết quả theo đề cương được duyệt. Tuy vậy, các kết quả còn đơn giản, sơ sài, chưa gắn với thực tế.

III/ Những vấn đề cần giải thích thêm:

- Đa số các kỹ thuật, công nghệ, công cụ được sử dụng chưa được khảo sát, đánh giá đầy đủ. Đề án chỉ khảo sát trục liên thông X-Road mà không khảo sát các trục liên

thông khác. Ngoài ra, các khảo sát các kỹ thuật, công nghệ, công cụ chỉ gồm văn bản, thiếu các sơ đồ, hình vẽ, nên tính diễn giải kém.

- Mục 2.1.1 đề xuất mô hình tổng quan hệ thống cũng rất mơ hồ, do tác giả không nêu rõ mô hình này sử dụng cho hệ thống nào, đề xuất mới hay tham khảo.
- Chương 3 mô tả việc triển khai thử nghiệm trục liên thông dữ liệu sử dụng X-Road cho PTIT, nhưng không có nội dung khảo sát, đánh giá hiện trạng, trên cơ sở đó đưa ra mô hình triển khai cho phù hợp. Việc triển khai thử nghiệm theo ý chủ quan mà không có khảo sát hiện trạng sẽ ít có giá trị trên thực tế.
- Đề án sử dụng nhiều thông tin, công nghệ, nền tảng, nhưng không có trích dẫn từ nguồn nào.
- Đề án quá lạm dụng cách viết theo chấm đầu dòng, nên tạo cảm giác lắt nhắt cho người đọc.
- Đề án có nhiều hình mờ, chữ nhỏ, không thể đọc (nhất là ở chương 3).
- Các tài liệu tham khảo cũng thiếu thông tin.

Câu hỏi:

1. Ngoài X-Road, còn có những trục liên thông dữ liệu nào khác? Theo tài liệu [6], Việt Nam đã triển khai trục liên thông văn bản quốc gia. Tác giả cho biết trục liên thông văn bản quốc gia là hệ thống tự phát triển, hay sử dụng nền tảng có sẵn?
2. Với Học viện PTIT, hầu hết dữ liệu của trường được quản lý tập trung bởi các máy chủ đặt tại Trung tâm dữ liệu. Việc sử dụng trục liên thông dữ liệu để kết nối dữ liệu giữa các phòng ban có thực sự phù hợp?
3. Việc đánh giá trục liên thông dữ liệu đã triển khai về hiệu năng, bảo mật được thực hiện thế nào?

IV/ Kết luận:

Đề án đáp ứng các tiêu chuẩn tối thiểu của một đề án tốt nghiệp thạc sỹ kỹ thuật.

Đồng ý cho phép học viên bảo vệ đề án tốt nghiệp.

Hà nội, ngày 12 tháng 7 năm 2025

Người phản biện

(Ký ghi rõ họ tên)

Hoàng Xuân Dậu

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

BẢN NHẬN XÉT ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

(Dùng cho người phản biện)

Tên đề tài đề án tốt nghiệp: Nghiên cứu, xây dựng trục liên thông dữ liệu phục vụ lưu trữ, quản lý dữ liệu đa cấu trúc

Chuyên ngành: Hệ thống thông tin

Mã chuyên ngành: 8.48.01.04

Họ và tên học viên: Trần Quang Đại

Họ và tên người nhận xét: Bùi Hải Phong

Học hàm, học vị: Tiến Sĩ

Chuyên ngành: Khoa học máy tính

Cơ quan công tác: Trường đại học Kiến Trúc Hà Nội

Số điện thoại: 0915594033 E-mail: phongbh@hau.edu.vn

NỘI DUNG NHẬN XÉT

I/ Cơ sở khoa học và thực tiễn, tính cấp thiết của đề tài:

Đề tài nghiên cứu và xây dựng trục liên thông dữ liệu phục vụ công tác lưu trữ, quản lý dữ liệu đa cấu trúc. Đề tài đã triển khai thử nghiệm thành công hạ tầng dữ liệu phục vụ công tác quản lý và điều hành tại một trường đại học. Các công nghệ được sử dụng như trục liên thông dữ liệu X-Road, Apache Airflow, MinIO và Apache Superset. Đề tài có cơ sở lý thuyết và có áp dụng thực tiễn cao, phục vụ công tác "chuyển đổi số".

II/ Nội dung của đề án tốt nghiệp, các kết quả đã đạt được:

Đề án tốt nghiệp gồm 65 trang, 3 chương, 17 hình vẽ. Đề án đã trình bày rõ ràng lý thuyết về liên thông dữ liệu, dữ liệu đa cấu trúc, các phương pháp chia sẻ dữ liệu đa cấu trúc thông qua trục liên thông dữ liệu. Đề án đã đề xuất thành công phương án xây dựng trục

liên thông dữ liệu dựa trên công nghệ mã nguồn mở X-Road. Các công nghệ thu thập, biến đổi dữ liệu sử dụng Airflow và MinIO được sử dụng hợp lý. Dữ liệu được trực quan hoá rõ ràng dựa trên công cụ Apache Superset.

III/ Những vấn đề cần giải thích thêm:

Đề án xem xét, làm rõ các vấn đề sau:
- Đề án hãy làm rõ về nguồn, dung lượng dữ liệu được sử dụng trong phần thực nghiệm 3.1?
- Đề án hãy phân tích tác dụng của Công cụ Apache Superset trong quá trình trực quan hoá dữ liệu đa cấu trúc? Công cụ Apache Superset hỗ trợ gì trong công tác quản lý, điều hành tại các trường đại học?

IV/ Kết luận:

Đồng ý cho phép học viên bảo vệ đề án tốt nghiệp.

Đồng ý (hoặc không đồng ý) cho phép học viên bảo vệ đề án tốt nghiệp.

Ngày.....tháng...7....năm 2025

NGƯỜI NHẬN XÉT

Bùi Hải Phong