

CHƯƠNG II: KIẾN TRÚC VÀ CÁC LOẠI HÌNH DỊCH VỤ

Cấu trúc dự án và loại hình hiện có

- [TRANG 1: Dịch vụ API REST \(Gateway Layer\)](#)
- [TRANG 2: Tầng Dịch vụ Dữ liệu \(Data Services - DSS\)](#)
- [TRANG 3: Hệ thống Proxy và Cầu nối Hệ thống cũ \(Legacy\)](#)
- [TRANG 4: Trục Tin nhắn RabbitMQ & Kafka \(Messaging Layer\)](#)

TRANG 1: Dịch vụ API REST (Gateway Layer)

1.1. Cấu trúc và Vị trí

Các định nghĩa API nằm tại: `src/main/wso2mi/artifacts/apis`. Mỗi API trong dự án LGSP đóng vai trò là một Gateway, thực hiện bảo mật, ghi log và điều phối yêu cầu.

1.2. Luồng xử lý tiêu biểu (Ví dụ: GuiNhanVanBan API)

Hệ thống sử dụng các Mediator mạnh mẽ để xử lý nghiệp vụ:

- **Xác thực Header:** Lấy các thông tin như `activityid`, `servicetype`, `messagetype` từ Transport Header.
- **Xác thực Đơn vị (CheckUnitCodeDBS):** Sử dụng `<dataServiceCall>` để kiểm tra mã hệ thống (`SystemCode`) của đơn vị gửi. Nếu không tồn tại, API sẽ trả về lỗi ngay lập tức.
- **Lưu trữ tin nhắn (Message Store):** Sử dụng lệnh `<store messageStore="EDXMLMessageStore"/>` để đưa payload vào hàng đợi RabbitMQ thay vì gửi trực tiếp, giúp tăng khả năng chịu tải và tin cậy.

1.3. Cơ chế Logging Tập trung

Mọi API đều gọi đến `DatabaseLogSequence` và `ResultLogSequence` để:

- Lưu vết vào bảng `api_logs`.
- Ghi lại thông tin người dùng (Client IP), payload request và response code.

TRANG 2: Tầng Dịch vụ Dữ liệu (Data Services - DSS)

2.1. Khái niệm và Vai trò

Dự án sử dụng các file cấu hình `.dbs` (tại `src/main/wso2mi/artifacts/data-services`) để trực tiếp thao tác với cơ sở dữ liệu mà không cần viết code Java/Spring.

2.2. Các dịch vụ trọng yếu

- **SyncTaskService.dbs:** Quản lý toàn bộ vòng đời của các Task đồng bộ. Cung cấp các thao tác `getTasks`, `insertTask`, `updateLastRun`.
- **syncLockDataService.dbs:** Chứa các logic nghiệp vụ về Khóa (Lock), giúp đảm bảo tính toàn vẹn khi có nhiều tác vụ chạy song song.
- **danhMucTheoMa.dbs:** Một dịch vụ quy mô lớn (91KB) cung cấp hàng trăm API tra cứu danh mục dựa trên các bảng dữ liệu chuyên ngành.

2.3. Ưu điểm triển khai

- **Cấu hình động:** Thay đổi câu lệnh SQL dễ dàng mà không cần biên dịch lại ứng dụng.
- **Xác thực tập trung:** Sử dụng xác thực ở mức API Gateway sau đó mới gọi vào DSS.

TRANG 3: Hệ thống Proxy và Cầu nối Hệ thống cũ (Legacy)

3.1. Proxy Services

Dành cho các hệ thống backend cũ sử dụng giao thức SOAP/XML. Các Proxy này nằm tại `src/main/wso2mi/artifacts/proxy-services`.

- Thực hiện chuyển đổi Payload từ JSON sang XML (qua Payload Factory) để gọi vào các hệ thống cũ.
- Trải qua các bộ lọc (Filter) để loại bỏ các thẻ XML thừa trước khi trả về cho Client.

3.2. Điều hướng Động (Dynamic Routing)

Sử dụng `switch` mediator dựa trên thuộc tính `REST_METHOD` (GET, POST, PUT, DELETE) để quyết định gọi URL tương ứng, giúp giảm thiểu số lượng API cần quản lý.

TRANG 4: Trục Tin nhắn RabbitMQ & Kafka (Messaging Layer)

4.1. Hệ thống RabbitMQ (Reliable Messaging)

Được cấu hình thông qua `MessageStore` và `MessageProcessor`:

- **Message Store:** `EDXMLMessageStore` kết nối trực tiếp tới Queue trên RabbitMQ.
- **Message Processor:** `SendEdoc2` đóng vai trò là "Worker" thức dậy định kỳ để đẩy tin nhắn đi.
- **Cơ chế Retry:** Nếu hệ thống đích (NGSP/Bộ ngành) lỗi 503, tin nhắn sẽ quay lại Queue và thử lại sau N giây.

4.2. Hệ thống Kafka (Event Streaming)

Sử dụng `KafkaProducerAPI` để đẩy các sự kiện dữ liệu lớn:

- Đẩy dữ liệu nhật ký hội thoại hoặc log hệ thống.
- Cung cấp dữ liệu cho hệ thống giám sát thời gian thực của tỉnh.