

Tổng quan kiến trúc hệ thống

- [Nội dung chi tiết](#)

Nội dung chi tiết

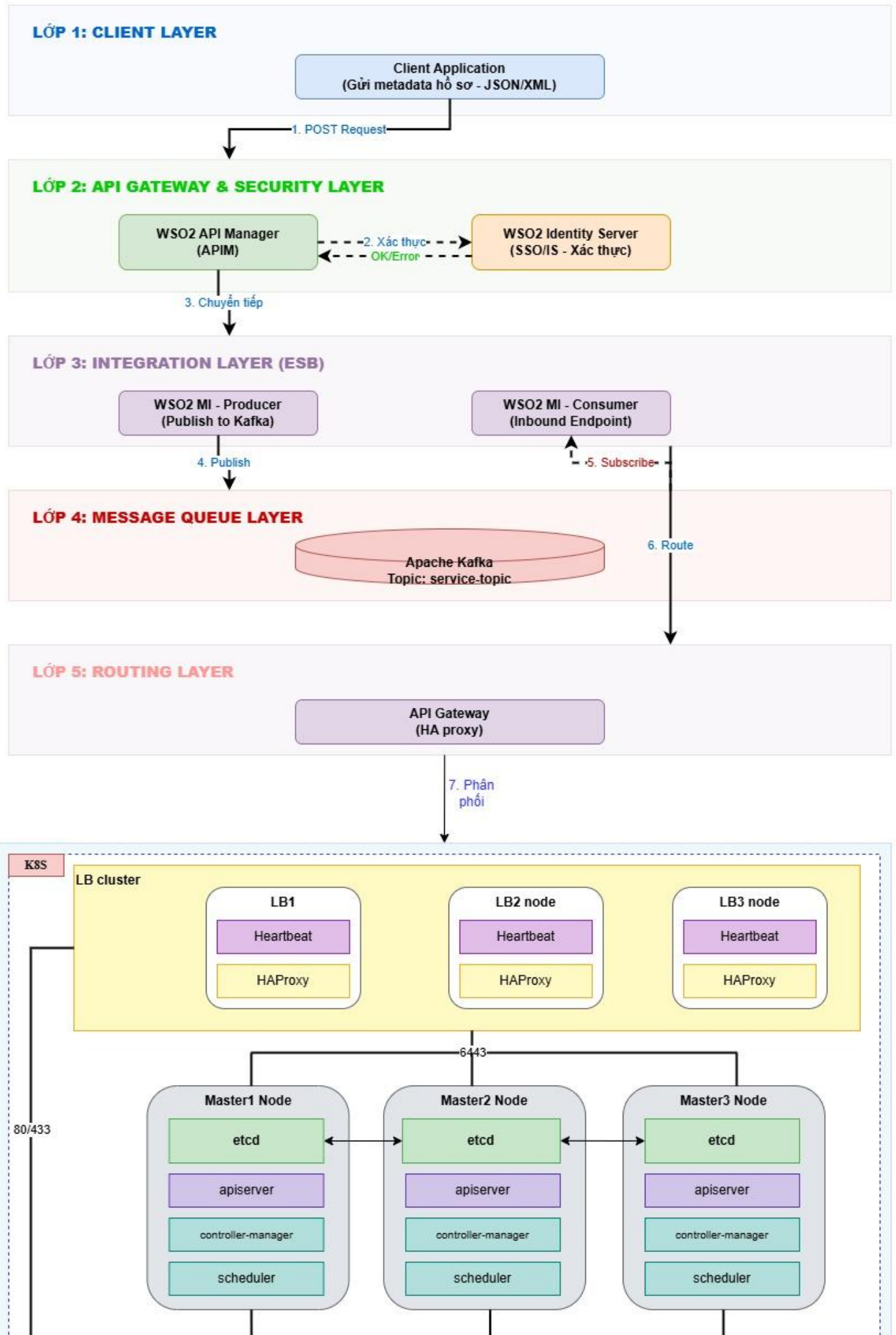
1.1. Giới Thiệu

Hệ thống được xây dựng theo mô hình **Enterprise Service Bus (ESB)** kết hợp **API Management (APIM)**, sử dụng nền tảng **LifeESB** làm trung gian điều phối giữa Client và các Backend Service. Tích hợp thêm **Apache Kafka** để xử lý tải lớn theo mô hình bất đồng bộ.

“**Tên hệ thống:** LifeESB **APIM:** API Manager **Message Queue:** Apache Kafka
(Broker: hdp-master:9092)

1.2. Sơ Đồ Kiến Trúc Tổng Thể

Kiến trúc Phân lớp - Dịch vụ Xử lý Hồ sơ



1.3. Các Thành Phần Chính

1.3.1. APIM — API Manager

Cổng vào duy nhất (Single Entry Point) cho toàn bộ hệ thống.

Chức năng	Mô tả
Xác thực	Kiểm tra JWT / OAuth2 / API Key trước khi vào LifeESB
Rate Limiting	Giới hạn số request/giây theo từng API hoặc User
Versioning	Quản lý nhiều version API (<code>/v1</code> , <code>/v2</code>) song song
Analytics	Theo dõi lưu lượng, latency, error rate
Portal	Developer Portal để publish và subscribe API

1.3.2. LifeESB

Lớp tích hợp trung gian, chịu trách nhiệm điều phối luồng dữ liệu.

Thành phần	File	Chức năng
Sync API	PlanningDirectApi.xml	Gọi thẳng Backend, Client nhận kết quả ngay
Async API	KafkaProducerApi.xml	Đẩy vào Kafka, trả phản hồi ngay cho Client
Inbound Endpoint	Load_balance_example.xml	Lắng nghe Kafka, kích hoạt Sequence xử lý
Sequence	Load_balance_example-inboundSequence.xml	Logic xử lý message: gọi Backend + phân loại lỗi
Error Sequence	Load_balance_example-inboundErrorSequence.xml	Xử lý lỗi Kafka, đẩy vào <code>error_topic</code>
Local Entry	KafkaConnection.xml	Cấu hình kết nối Kafka tái sử dụng

1.3.3. Apache Kafka

Message Broker xử lý tải lớn và đảm bảo không mất dữ liệu.

Topic	Mục đích
test_topic_01	Nhận message từ API Producer (đầu vào)
processed_topic	Lưu message đã xử lý thành công
error_topic	Lưu message lỗi để review thủ công

1.3.4. Backend Service

REST API xử lý nghiệp vụ thực tế.

Thông tin	Giá trị
Base URL	http://192.168.0.133:8080
Endpoint chính	POST /api/v1/plannings
Xác thực	JWT Bearer Token
Framework	Spring Boot

1.4. Hai Mô Hình Dịch Vụ

1.4.1. Đồng Bộ — Synchronous (Direct API)

Client gọi và **chờ kết quả** từ Backend.

Client → APIM → LifeESB (PlanningDirectApi)

|



Backend API [chờ response]

|



LifeESB → Client [trả kết quả thực]

Đặc điểm:

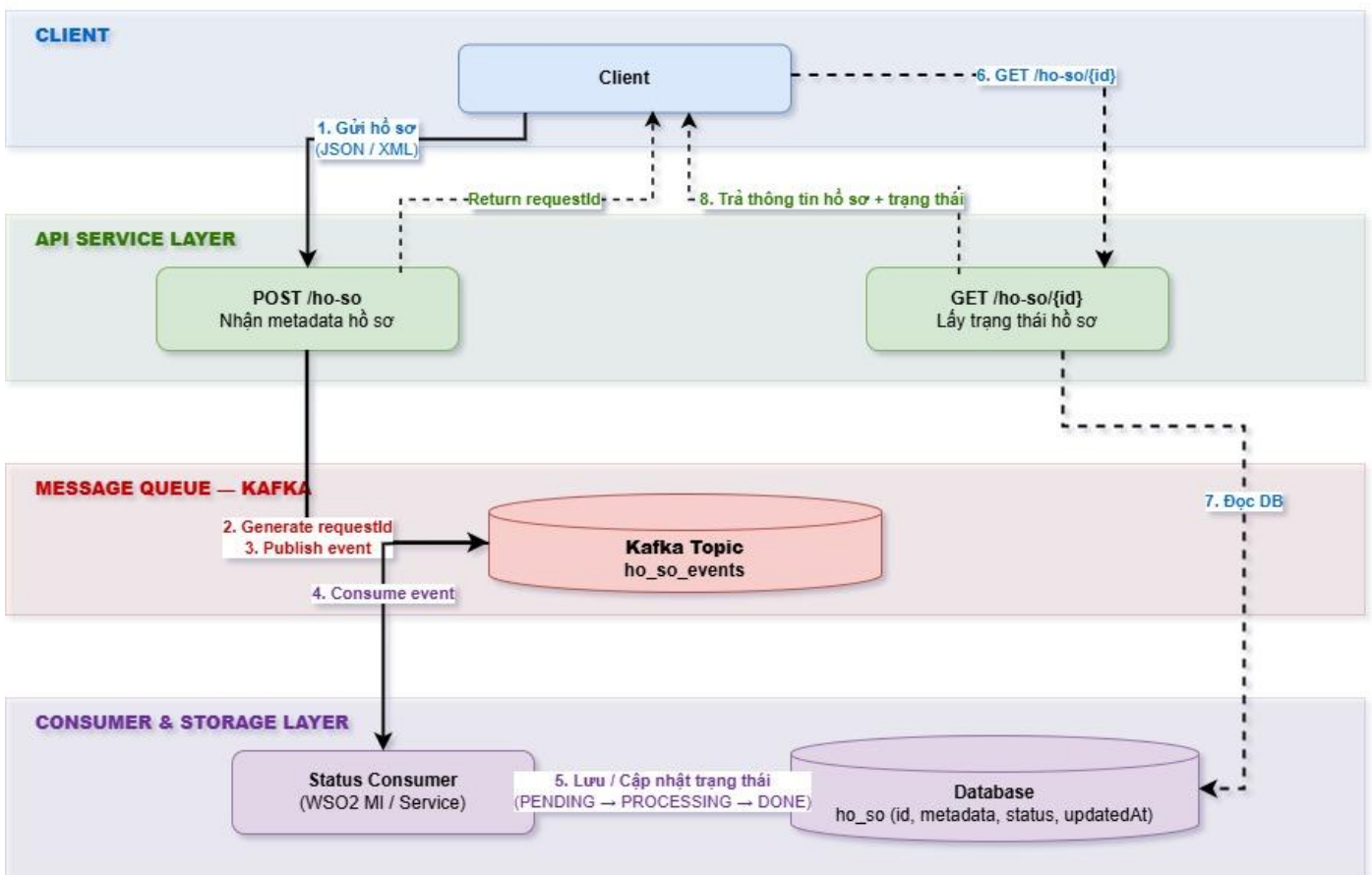
- Client biết ngay kết quả thành công hay thất bại
- Backend bị quá tải → Client bị block, timeout
- Phù hợp: nghiệp vụ cần xác nhận ngay, lượng request vừa phải

Endpoint: POST /plannings-direct

1.4.2. Bất Đồng Bộ — Asynchronous (Kafka)

Client gửi và **nhận phản hồi ngay** (không chờ Backend xử lý xong).

Luồng Xử lý Hồ sơ — Kafka + DB Status



Đặc điểm:

- Client không bị block dù Backend chậm hoặc quá tải
- Kafka buffer message, đảm bảo không mất dữ liệu
- Phù hợp: tải lớn, xử lý nền, không cần kết quả ngay

Endpoint: `POST /kafka-producer`

1.5. So Sánh Hai Mô Hình

Tiêu chí	Đồng bộ (<code>/plannings-direct</code>)	Bất đồng bộ (<code>/kafka-producer</code>)
Client chờ	☐ Chờ kết quả thực	☐ Nhận <code>success</code> ngay lập tức
Khả năng chịu tải	Thấp hơn (Backend block)	Cao (Kafka buffer)
Độ trễ	Phụ thuộc Backend	Gần như bằng 0 (phía Client)
Đảm bảo dữ liệu	Ghi ngay hoặc lỗi ngay	Kafka lưu, retry nếu fail
Xử lý lỗi	<code>faultSequence</code> → trả ngay	<code>error_topic</code> → review sau
Phù hợp	Tải vừa, cần confirm	Tải lớn, xử lý nền
Timeout	15 giây	Không (Kafka giữ message)

1.6. Nguyên Tắc Kiến Trúc

- Single Entry Point:** Mọi request đi qua APIM, không gọi thẳng LifeESB từ ngoài.
- Stateless Mediator:** LifeESB không lưu trạng thái — toàn bộ state trong Kafka hoặc Backend DB.
- Error Isolation:** Lỗi phải được bắt trong Sequence, không để lỗi raw thoát ra ngoài.
- Dead Letter Pattern:** Message lỗi luôn được chuyển vào `error_topic`, không bao giờ bị drop.
- Observability:** Mọi bước xử lý đều có `log level="custom"` để trace theo luồng.

1.7. Môi Trường Triển Khai

Thành phần	Host / Port	Ghi chú
LifeESB	<code>192.168.0.167:8290</code>	HTTP endpoint
LifeESB Management	<code>192.168.0.167:9201</code>	Deploy CAR file
Kafka Broker	<code>hdp-master:9092</code>	PLAINTEXT
Backend API	<code>192.168.0.133:8080</code>	Spring Boot

📄 [Ảnh minh họa] — LifeESB Dashboard sau khi deploy

The screenshot displays the WSO2 Integrator LifeESB Dashboard. The main view is for the project 'Project: Loadbalance_kafka'. The interface is divided into several sections:

- Component Diagram / Project Structure:** Shows a visual representation of the project's components and their relationships.
- Deployment Options:** Lists various deployment methods:
 - Deploy to Devant
 - Deploy on a Remote Server
 - Build Docker Image
 - Configure Kubernetes Deployment
 - Build CAPP
- Project Summary:** Provides key project details:
 - Project Information:** Group ID: com.microintegrator.projects, Artifact ID: Loadbalance_kafka, Version: 1.0.0, Runtime Version: 4.4.0.
 - Configurables:** No configurables found.
 - Connector Dependencies:** (Section header visible)
- Available Runtime Services:** Shows the status of deployed services. The server status is 'Running'. A table lists the deployed APIs:

API Name	URL	Action
PlanningDirectApi	http://localhost:8290/plannings-direct	Try it
KafkaProducerApi	http://localhost:8290/kafka-producer	Try it

“📄 [Ảnh minh họa] — Kafka UI / Kafka Manager hiển thị các topic