

**ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**NGUYỄN THỊ THU PHƯƠNG**

**GIẢI PHÁP TÍCH HỢP DỊCH VỤ NGHIỆP VỤ  
NGÂN HÀNG THEO MÔ HÌNH SOA**

**LUẬN VĂN THẠC SỸ**

**Ngành: Hệ thống thông tin**

**Hà Nội - 2016**

**ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**NGUYỄN THỊ THU PHƯƠNG**

**GIẢI PHÁP TÍCH HỢP DỊCH VỤ NGHIỆP VỤ  
NGÂN HÀNG THEO MÔ HÌNH SOA**

Ngành: Hệ thống thông tin

Chuyên ngành: Hệ thống thông tin

Mã số: 60480104

**LUẬN VĂN THẠC SĨ HỆ THỐNG THÔNG TIN**

**NGƯỜI HƯỚNG DẪN KHOA HỌC: PGS.TS Nguyễn Ngọc Hóa**

**Hà Nội - 2016**

## LỜI CAM ĐOAN

Tôi xin cam đoan luận văn tốt nghiệp do tôi tự mình thực hiện dưới sự hướng dẫn của Thầy Nguyễn Ngọc Hóa, mọi thông tin tham khảo sử dụng trong luận văn đều được trích dẫn đầy đủ và hợp pháp.

Tôi xin hoàn toàn chịu trách nhiệm và chịu mọi hình thức kỷ luật theo quy định của nhà trường cho lời cam đoan của mình.

Hà Nội, ngày 26 tháng 10 năm 2016

Người cam đoan

Nguyễn Thị Thu Phương

## LỜI CẢM ƠN

Tôi xin chân thành cảm ơn PGS.TS Nguyễn Ngọc Hóa là giảng viên của Trường Đại học Công Nghệ đã tận tình giúp đỡ tôi về kiến thức, định hướng phát triển và cả về tinh thần cố gắng trong suốt quá trình làm luận văn tốt nghiệp.

Tôi cũng xin gửi lời cảm ơn đến các thầy cô của khoa Công Nghệ Thông Tin vì đã giảng dạy và hướng dẫn tôi trong suốt những năm theo học tại Trường Đại học Công Nghệ.

Cuối cùng, tôi xin gửi lời biết ơn sâu sắc đến với gia đình vì đã luôn ở bên cạnh ủng hộ tôi trên con đường học tập và nghiên cứu đầy khó khăn.

*Xin chân thành cảm ơn!*

Hà Nội, Tháng 10 Năm 2016

Nguyễn Thị Thu Phương

# TÓM TẮT NỘI DUNG

## *Tóm tắt:*

Nội dung luận văn sẽ tập trung trình bày một số phương pháp tích hợp hệ thống (tích hợp (i) mức dữ liệu, (ii) mức chức năng và (iii) mức quy trình/dịch vụ); chú trọng phương pháp tích hợp theo mô hình kiến trúc hướng dịch vụ SOA (Service Oriented Architecture) sử dụng trục dịch vụ tổng thể ESB (Enterprise Service Bus). Từ đó, ứng dụng để xây dựng giải pháp tích hợp một số dịch vụ nghiệp vụ tại NHNN theo mô hình SOA với công nghệ ESB và Webservice.

# MỤC LỤC

LỜI CAM ĐOAN.....	
LỜI CẢM ƠN.....	
TÓM TẮT NỘI DUNG.....	
MỤC LỤC.....	
DANH MỤC HÌNH.....	
DANH MỤC BẢNG.....	
CÁC TỪ VIẾT TẮT.....	
GIỚI THIỆU CHUNG.....	1
<b>CHƯƠNG 1. TỔNG QUAN VỀ TÍCH HỢP HỆ THỐNG.....</b>	<b>3</b>
1.1. TỔNG QUAN VỀ TÍCH HỢP HỆ THỐNG.....	3
1.1.1. Khái niệm.....	3
1.1.2. Mục tiêu của tích hợp hệ thống.....	3
1.1.3. Thách thức của tích hợp hệ thống.....	3
1.2. KIẾN TRÚC ĐA TẦNG TRONG TÍCH HỢP HỆ THỐNG.....	4
1.2.1. Kiến trúc 1-tier:.....	5
1.2.2. Kiến trúc 2-tier.....	6
1.2.3. Middleware.....	8
1.2.4. Kiến trúc 3-tier.....	9
1.3. MỘT SỐ PHƯƠNG PHÁP TÍCH HỢP.....	10
1.3.1. Tích hợp mức dữ liệu.....	10
1.3.2. Tích hợp mức chức năng.....	14
1.3.3. Tích hợp mức dịch vụ (quy trình).....	20
1.4. KẾT LUẬN.....	22
<b>CHƯƠNG 2. TÍCH HỢP DỊCH VỤ THEO MÔ HÌNH TRỰC DỊCH VỤ TỔNG THỂ ESB.....</b>	<b>23</b>
2.1. TỔNG QUAN VỀ TRỰC DỊCH VỤ TỔNG THỂ ESB.....	23
2.2. CHỨC NĂNG LỖI CỦA ESB:.....	23
2.2.1. Kết nối định tuyến.....	23
2.2.2. Chuyển đổi giao thức.....	25
2.2.3. Chuyển đổi dữ liệu/ thông điệp.....	26
2.2.4. Các nhóm ảo hóa.....	26

2.2.5. Các yêu cầu phi chức năng đối với ESB:.....	27
2.3. CÁC THÀNH PHẦN LOGIC .....	28
2.3.1. Bộ chuyển đổi - Adapter.....	28
2.3.2. Thành phần điều phối - Dispatcher .....	28
2.3.3. Thành phần quản lý yêu cầu - Request Handle .....	29
2.3.4. Công cụ định tuyến - Routing and Rule Engine.....	29
2.3.5. Dịch vụ đại diện - Service Delegates.....	29
2.3.6. Công cụ chuyển đổi - Transformation Engine.....	29
2.3.7. Enrichment Component.....	29
2.3.8. Ghi nhật ký - Logging Component.....	29
2.3.9. Xử lý ngoại lệ - Exception-Handling Component.....	29
2.4. PHÂN LOẠI ESB .....	30
2.4.1. ESB dựa trên máy chủ ứng dụng.....	30
2.4.2. ESB dựa trên hệ thống thông điệp .....	30
2.4.3. ESB dựa trên hạ tầng phần cứng.....	30
2.5. MỘT SỐ NỀN TẢNG HỖ TRỢ TÍCH HỢP DỊCH VỤ THEO ESB .....	31
2.5.1. IBM WebSphere ESB.....	31
2.5.2. Talend ESB.....	32
2.5.3. TIBCO .....	33
2.6. KẾT LUẬN .....	35
<b>CHƯƠNG 3. GIẢI PHÁP TÍCH HỢP MỘT SỐ HỆ THỐNG</b>	
<b>NGHIỆP VỤ NGÂN HÀNG .....</b>	<b>36</b>
3.1. BÀI TOÁN TÍCH HỢP HỆ THỐNG NGHIỆP VỤ NGÂN HÀNG .....	36
3.1.1. Hệ thống ngân hàng lõi.....	36
3.1.2. Hệ thống sổ sách kế toán và kế toán tài chính.....	37
3.1.3. Hệ thống thanh toán điện tử liên ngân hàng IBPS (Inter-Bank Payment System).....	38
3.1.4. Trung tâm lưu ký chứng khoán CSD (central securities depository).....	39
3.2. YÊU CẦU ĐẶT RA.....	39
3.2.1. Yêu cầu của hệ thống.....	39
3.2.2. Môi trường thực nghiệm.....	40
3.3. MÔ HÌNH GIẢI PHÁP TÍCH HỢP .....	41

3.3.1. Mô hình liên thông giữa các hệ thống nghiệp vụ .....	42
3.3.2. Kiến trúc tích hợp .....	43
3.4. XÂY DỰNG VÀ PHÁT TRIỂN HỆ THỐNG THỬ NGHIỆM .....	43
3.4.1. Cài đặt hệ thống thử nghiệm .....	43
3.4.2. Phương thức quản lý người dùng trên các hệ thống .....	44
3.4.3. Tích hợp dịch vụ qua Tibco ESB sử dụng WebService .....	46
3.4.4. Tích hợp dịch vụ qua Tibco ESB sử dụng Adapter .....	52
3.4.5. Giao diện quản trị của Tibco.....	54
3.5. KẾT QUẢ THỬ NGHIỆM VÀ ĐÁNH GIÁ.....	55
3.5.1. Giao dịch chuyển tiền từ T24 sang CITAD:.....	55
3.5.2. Giao dịch chi lương thực hiện trên phân hệ phải thu phải trả của ERP, tích hợp sang hệ thống T24 để chi tiền mặt cho nhân viên.....	59
3.6. KẾT LUẬN .....	61
<b>CHƯƠNG 4. KẾT LUẬN CHUNG .....</b>	<b>62</b>
4.1. CÁC KẾT QUẢ ĐẠT ĐƯỢC TRONG LUẬN VĂN .....	62
4.2. ĐỊNH HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI.....	62
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>.....</b>



## DANH MỤC HÌNH

Hình 1.1 Kiến trúc tầng trong hệ thống thông tin.....	4
Hình 1.2 Mô hình kiến trúc đa tầng .....	5
Hình 1.3 Kiến trúc 1-tier .....	6
Hình 1.4 Kiến trúc 2-tier .....	7
Hình 1.5 Kiến trúc Middleware .....	9
Hình 1.6 Kiến trúc 3-tier .....	10
Hình 1.7 Hai ứng dụng A, B cùng được cài đặt trên một máy chủ.....	11
Hình 1.8 Hai ứng dụng A và B được cài trên hai máy chủ khác nhau .....	11
Hình 1.9 Các ứng dụng chia sẻ cơ sở dữ liệu.....	12
Hình 1.10 Các bước xây dựng socket .....	14
Hình 1.11 Local function call.....	15
Hình 1.12 Restricted RPC .....	15
Hình 1.13 Hai ứng dụng trên hai máy chủ khác nhau.....	15
Hình 1.14 Các bước cơ bản khi gọi hàm.....	16
Hình 1.15 Mô hình thông điệp không đồng bộ sử dụng hàng đợi .....	17
Hình 1.16 Các bước cơ bản để truyền thông điệp.....	18
Hình 1.17 Hàng đợi kiểu point – to – point .....	19
Hình 1.18 Hàng đợi kiểu publish – and – subscribe .....	19
Hình 1.19 Thành phần của SOA .....	21
Hình 2.1 Mô hình kết nối sử dụng phương pháp điểm – điểm .....	24
Hình 2.2 Mô hình kết nối sử dụng ESB .....	24
Hình 2.3 Các ứng dụng sử dụng các giao thức khác nhau kết nối qua ESB.....	26
Hình 2.4 Các ứng dụng có định dạng dữ liệu khác nhau kết nối qua ESB.....	26
Hình 2.5 Các thành phần logic của ESB.....	28
Hình 2.6 Mô hình tích hợp cho ứng dụng CICS mainframe.....	32
Hình 2.7 Kiến trúc logic của trục tích hợp Tibco ESB .....	33
Hình 3.1 Các phần mềm ứng dụng cài đặt .....	41
Hình 3.2 Mô hình tương tác giữa các hệ thống nghiệp vụ.....	42
Hình 3.3 Kiến trúc tích hợp.....	43
Hình 3.4 Mô hình hệ thống SSO.....	45
Hình 3.5 Mô hình tích hợp OAM với các ứng dụng.....	46
Hình 3.6 Cấu trúc thông điệp gửi đi.....	47

Hình 3.7 Cấu trúc thông điệp nhận về.....	51
Hình 3.8 Mô hình luồng nghiệp vụ hệ thống thanh toán liên hàng yêu cầu số dư tài khoản từ T24 .....	54
Hình 3.9 Các máy chủ ứng dụng.....	54
Hình 3.10 Các phần mềm cài đặt .....	54
Hình 3.11 Các dịch vụ cài đặt .....	55
Hình 3.12 Màn hình giao dịch chuyển nợ trên T24 .....	57
Hình 3.13 Màn hình giao dịch tương ứng chuyển sang CITAD.....	58
Hình 3.14 Màn hình báo cáo liệt kê giao dịch in trên hệ thống ERP .....	58
Hình 3.15 Hóa đơn trên phân hệ phải thu phải trả ERP.....	60
Hình 3.16 Màn hình thực hiện thanh toán hóa đơn và chuyển giao dịch sang T24 để chi tiền mặt .....	60
Hình 3.17 Màn hình giao dịch tiền mặt tương ứng nhận từ ERP .....	61

## DANH MỤC BẢNG

Bảng 3.1 Danh sách máy chủ cài đặt hệ thống thực nghiệm .....	44
Bảng 3.2 Mô tả thẻ trong cấu trúc thông điệp gửi đi .....	47
Bảng 3.3 Cấu trúc AppHdr.....	50
Bảng 3.4 Mô tả cấu trúc thông điệp nhận về .....	52
Bảng 3.5 Cấu hình tham số Adapter Tuxedo .....	53

## CÁC TỪ VIẾT TẮT

ESB	Enterprise Service Bus Trục dịch vụ tổng thể
NHNN	Ngân hàng Nhà nước Việt Nam
MOM	Message – Oriented Middleware Nền tảng trung gian hướng thông điệp
RPC	Remote Procedure Call Lời gọi hàm xa
SSO	Single Sign On Truy cập một lần
SOA	Service Oriented Architecture Kiến trúc hướng dịch vụ
TCTD	Tổ chức tín dụng
WAN	Wide Area Network Mạng diện rộng

# GIỚI THIỆU CHUNG

## 1. Động lực thực hiện luận văn

Ngày nay, việc phát triển nhanh chóng các hệ thống thông tin được xây dựng trên nền tảng các công nghệ khác nhau, sử dụng các hệ quản trị cơ sở dữ liệu đa dạng, triển khai trên nhiều nền tảng dẫn tới sự không đồng bộ trong các tổ chức. Lượng lớn thông tin được tạo ra nhưng không thể truy xuất, khai thác dẫn đến việc vừa thừa vừa thiếu dữ liệu hay tốn chi phí để phát triển lại những module đang hoạt động ổn định. Nhu cầu cấp thiết đặt ra cho các tổ chức nói chung và Ngân hàng Nhà nước nói riêng là tích hợp các hệ thống "không đồng bộ" này thành "hệ thống đồng nhất" nhằm tối ưu hóa về dữ liệu và chi phí.

Từ đó, tôi nhận thấy việc nghiên cứu các công nghệ tích hợp, đưa ra các giải pháp và xây dựng công cụ tích hợp các hệ thống rất có ý nghĩa và phù hợp thực tiễn.

## 2. Mục tiêu của luận văn

Luận văn cao học này có mục tiêu nghiên cứu, tìm hiểu các phương pháp tích hợp hệ thống; chú trọng mô hình tích hợp mức dịch vụ theo định hướng kiến trúc hướng dịch vụ SOA và ứng dụng trong việc tích hợp một số hệ thống thông tin nghiệp vụ cơ bản trong ngân hàng.

Mục tiêu trên sẽ được cụ thể hoá thông qua những nội dung thực hiện chính sau:

- Tìm hiểu, đánh giá một số phương pháp tích hợp hệ thống, chú trọng đến phương pháp tích hợp mức dịch vụ theo mô hình hướng dịch vụ SOA.
- Chú trọng nghiên cứu mô hình tích hợp hướng dịch vụ dựa trên trục dịch vụ tổng thể ESB và dịch vụ Web; từ đó xây dựng giải pháp tích hợp một số hệ thống thông tin nghiệp vụ trong ngân hàng.
- Xây dựng hệ thống thử nghiệm tích hợp 4 hệ thống nghiệp vụ lõi trong Ngân hàng nhà nước dựa trên nền tảng Tibco và tiến hành đánh giá thử nghiệm tại Cục Công nghệ tin học – Ngân hàng Nhà nước.

## 3. Tổ chức luận văn

Luận văn được thực hiện xuyên suốt trong quá trình từ khi hình thành các khái niệm, ý tưởng, phân tích thiết kế, trình bày cài đặt sản phẩm cho đến khi hoàn thành sản phẩm và kiểm tra kiểm thử đánh giá sản phẩm. Các kết quả chính của luận văn sẽ trình bày trong 4 chương có nội dung vắn tắt như sau:

- **Chương 1:** Tổng quan về tích hợp hệ thống. Chương này giới thiệu các khái niệm về tích hợp hệ thống, kiến trúc đa tầng trong tích hợp hệ thống và một số phương pháp tích hợp hệ thống như tích hợp mức dữ liệu, tích hợp mức chức năng, tích hợp mức dịch vụ.

- **Chương 2:** Tích hợp dịch vụ theo mô hình trực dịch vụ tổng thể ESB. Chương này giới thiệu sâu hơn về mô hình tích hợp sử dụng ESB: các chức năng, các thành phần logic của ESB, phân loại ESB và một số nền tảng hỗ trợ tích hợp ESB.

- **Chương 3:** Đề xuất giải pháp tích hợp các hệ thống nghiệp vụ ngân hàng. Chương này nêu ra bài toán tích hợp một số hệ thống nghiệp vụ tại NHNN, từ đó đưa ra giải pháp tích hợp các hệ thống nghiệp vụ này dựa trên tích hợp dịch vụ sử dụng ESB và WebService.

- **Chương 4:** Kết luận chung. Chương này nêu các kết quả đạt được trong luận văn và định hướng phát triển mô hình tích hợp trong tương lai tại NHNN.

## **CHƯƠNG 1. TỔNG QUAN VỀ TÍCH HỢP HỆ THỐNG**

### **1.1. Tổng quan về tích hợp hệ thống**

#### **1.1.1. Khái niệm**

Ngày nay nhu cầu về thông tin ngày càng lớn với yêu cầu chất lượng thông tin ngày càng cao như độ chính xác tin cậy của thông tin, tốc độ truy xuất nhanh, mức độ sẵn sàng cao. Các tổ chức, doanh nghiệp nói chung, NHNN nói riêng thường đã có sẵn các hệ thống nghiệp vụ riêng biệt, sử dụng nền tảng công nghệ khác nhau. Cùng với sự phát triển của tổ chức, nhu cầu cần có một hệ thống tổng thể phục vụ nhu cầu thông tin là cần thiết. Từ đó người ta nghiên cứu đưa ra những phương pháp, kỹ thuật, mẫu và công nghệ để có thể nối ghép, tương tác các hệ thống riêng biệt này với nhau.

Tích hợp hệ thống là quá trình liên kết, kết nối các hệ thống thông tin cả về khía cạnh chức năng lẫn hạ tầng tính toán, để hoạt động như một hệ thống thống nhất [9]. Nói cách khác, hệ thống tích hợp là tập hợp các hệ thống rời rạc sử dụng một loạt các kỹ thuật như mạng máy tính, tích hợp ứng dụng doanh nghiệp, quy trình quản lý kinh doanh hoặc chương trình.

#### **1.1.2. Mục tiêu của tích hợp hệ thống**

Tích hợp hệ thống nhằm tạo ra hệ thống tổng thể mà từ đó người dùng có thể truy xuất được đúng thông tin, đúng thời điểm, đạt chất lượng với chi phí rẻ nhất.

#### **1.1.3. Thách thức của tích hợp hệ thống**

Khi một ứng dụng mới ra đời nó thường không được tính toán trước để tích hợp, thiết kế của nó thường độc lập, khó có thể dễ dàng kết hợp với những thành phần đã có hoặc những thành phần mới khác nhằm giải quyết các bài toán cụ thể. Điều này bắt nguồn từ thực tế các tổ chức chưa quan tâm đến vấn đề tích hợp một cách nghiêm túc, họ thường chỉ tập trung tạo ra sản phẩm mới để giải quyết ngay lập tức vấn đề đang tồn tại.

Bên cạnh đó các ứng dụng đôi khi được viết trên những nền tảng khác nhau như ứng dụng Web, ứng dụng cho hệ điều hành Windows, Linux...; với những ngôn ngữ khác nhau: C++, Java, dotNet,... cũng như phương thức quản lý dữ

liệu khác nhau: Tập lưu trữ, Dữ liệu quan hệ, Dữ liệu phi cấu trúc, dữ liệu có cấu trúc. Việc vượt qua những khác biệt này để tích hợp chúng là khó khăn.

Với những khó khăn, thách thức trên, các tổ chức, chuyên gia cần có kiến thức tổng thể lớn về hệ thống cùng với kinh phí rất cao để có thể thực hiện tốt việc tích hợp.

## 1.2. Kiến trúc đa tầng trong tích hợp hệ thống

Hệ thống thông tin được chia thành nhiều tầng (tier), mỗi tầng có thể là một thực thể quan niệm hoặc một thực thể thực. Việc phân biệt các tầng phụ thuộc vào tổ chức đơn vị, chức năng nghiệp vụ, công nghệ sử dụng,... Mô hình hóa hệ thống thông tin theo tầng cho phép trừu tượng hóa được những hệ thống phức tạp, hiện đại.

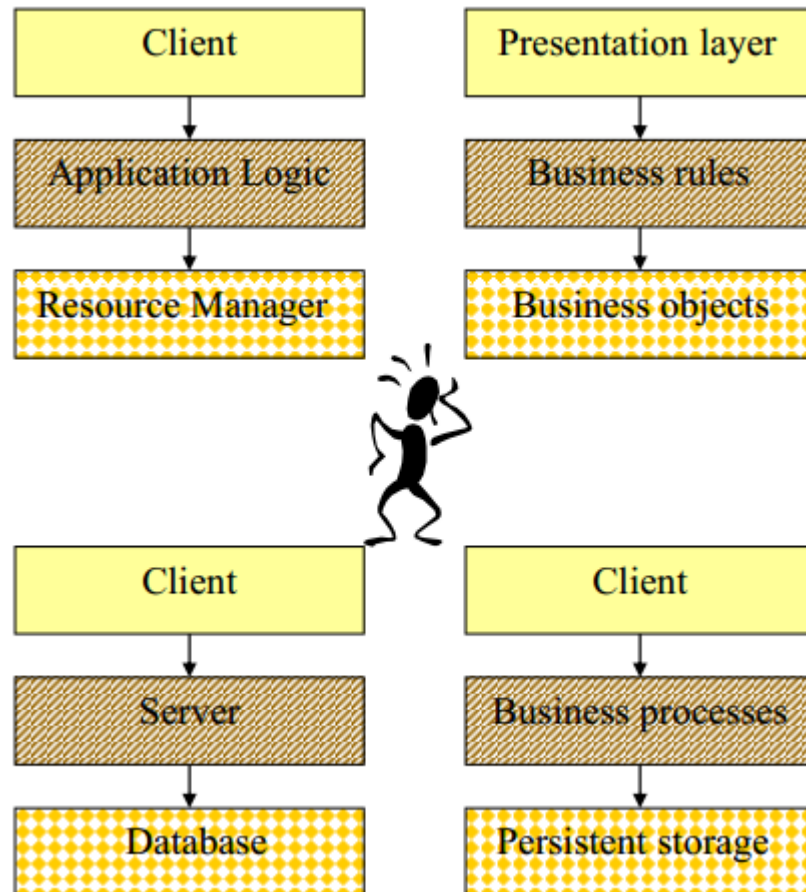


Hình 1.1 Kiến trúc tầng trong hệ thống thông tin

### ***Kiến trúc đa tầng bao gồm các tầng:***

- Client: người dùng hoặc chương trình thực hiện tác vụ trên hệ thống.
- Presentation layer: tầng giúp client gửi yêu cầu và nhận kết quả phản hồi.
- Application logic: tầng đảm bảo thực hiện các quy trình nghiệp vụ đồng thời xác lập những thao tác nào có thể được thực hiện bởi client.
- Resource manager: tầng tương tác mức thấp với tài nguyên dữ liệu. Tầng này có thể là hệ quản trị cơ sở dữ liệu hoặc hệ thống quản lý dữ liệu khác có khả năng bảo quản dữ liệu và xử lý truy vấn.





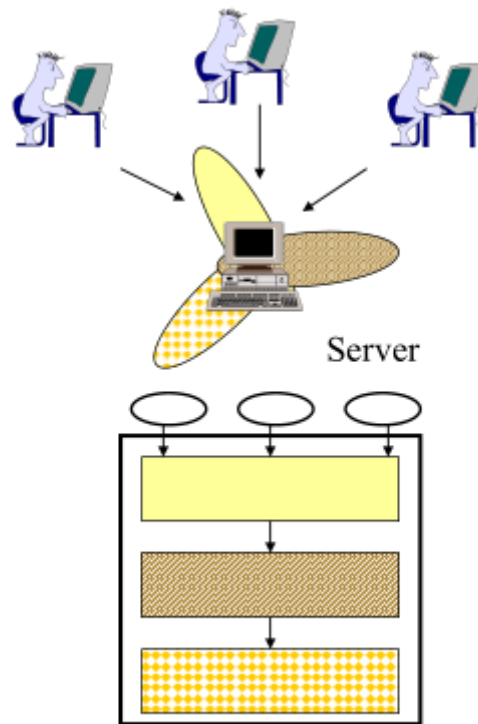
Hình 1.2 Mô hình kiến trúc đa tầng

### 1.2.1. Kiến trúc 1-tier:

Cả ba tầng presentation, application logic và resource manager được xây dựng trong cùng thực thể nguyên khối.

Người dùng/ chương trình truy cập hệ thống thông qua thiết bị cuối.

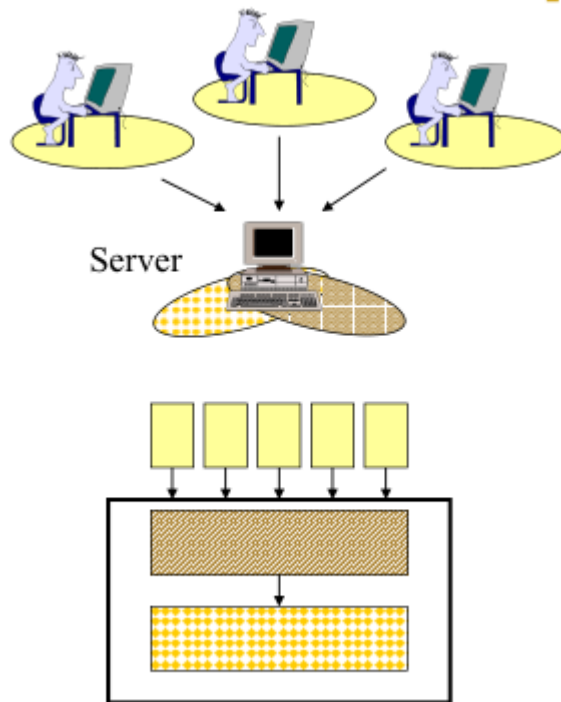
Phù hợp để triển khai với các ứng dụng trên mainframe.



Hình 1.3 Kiến trúc 1-tier

### 1.2.2. Kiến trúc 2-tier

Tầng Presentation được chuyển về phía client



Hình 1.4 Kiến trúc 2-tier

*Ưu điểm:*

- Clients độc lập với nhau: mỗi client có thể có nhiều tầng Presentation tùy theo yêu cầu.
- Có thể xây dựng tầng Presentations phức tạp hơn dựa vào năng lực tính toán của máy tính tại client, giảm bớt gánh nặng sử dụng tài nguyên tại máy chủ.
- Đưa ra khái niệm API (Application Program Interface) – giao diện tương tác với hệ thống từ bên ngoài. Từ đó cho phép tích hợp hệ thống phức tạp thông qua liên kết nhiều hệ thống khác nhau.
- Tầng Resource Manager chỉ quản lý duy nhất một application logic, giúp nâng cao hiệu năng quản lý kết nối trong nội tại máy chủ.

*Nhược điểm:*

- Hệ thống phải xử lý tất cả các kết nối, số lượng client tối đa phụ thuộc vào số kết nối được hỗ trợ ở phía máy chủ.
- Client gắn chặt với hệ thống do không có tầng Presentation chuẩn. Nếu client muốn kết nối đến hai hệ thống, client phải có hai tầng Presentation khác nhau.
- Không có đóng gói tải và lỗi. Nếu hệ thống lỗi, không một client nào có thể hoạt động. Bên cạnh đó việc thi hành thao tác của một client sẽ ảnh hưởng

trực tiếp đến các client khác do tất cả các thao tác được thi hành trên cùng tài nguyên máy chủ.

- Việc thiết kế tầng Application Logic và tầng Resource Manager gắn liền với nhau một cách chặt chẽ, gây khó khăn khi thay đổi hay tách biệt chúng để cải thiện hiệu năng.

- Thiết kế theo mô hình này phức tạp và khó để chuyển sang môi trường khác.

- Khi client muốn truy cập đến hai hay nhiều hệ thống, kiến trúc này gây ra nhiều vấn đề:

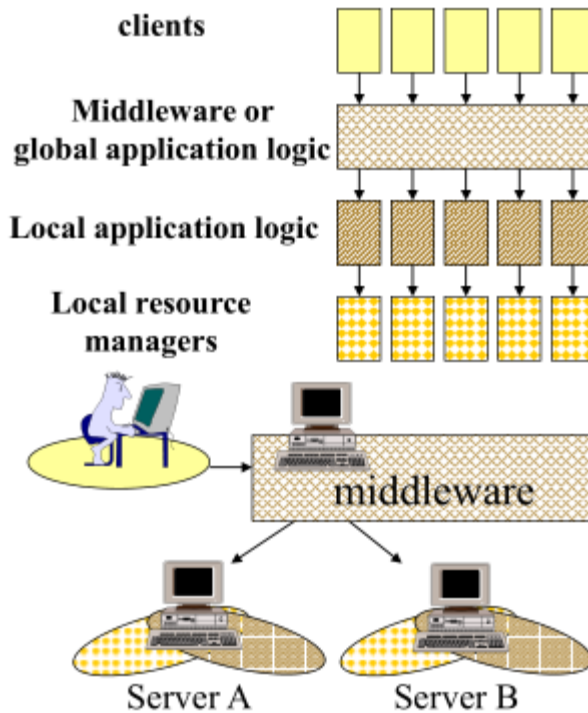
- + Các hệ thống cơ bản không biết về nhau, không có logic nghiệp vụ chung nên business logic đôi khi phải đặt ở phía client.

- + Các hệ thống cơ bản khác nhau. Sự phức tạp đối phó với hai hệ thống không đồng nhất cần phải được giải quyết ở client.

- + Client chịu trách nhiệm phải biết mọi thứ ở đâu, làm thế nào để có được chúng và làm thế nào để đảm bảo tính nhất quán.

### **1.2.3. Middleware**

Các nhược điểm về phía client khó có thể giải quyết được ở mô hình 2-tier. Khi đó cần thêm mức gián tiếp Middleware giữa clients và các tầng khác trong hệ thống.



Hình 1.5 Kiến trúc Middleware

*Ưu điểm:*

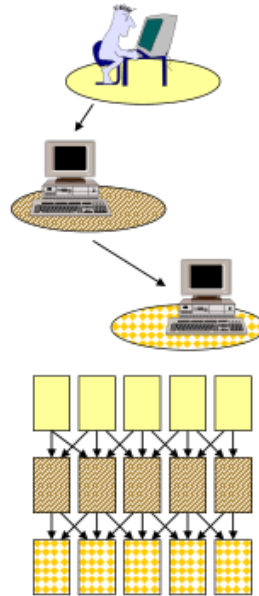
- Đơn giản hóa thiết kế client thông qua việc giảm số giao diện.
- Cho phép truy cập trong suốt đối với các hệ thống cơ bản.
- Đóng vai trò nền tảng cho các chức năng và tầng logic ứng dụng mức cao.
- Chú trọng hơn đến việc phân bổ tài nguyên, truy cập và thu nhận kết quả.

#### 1.2.4. Kiến trúc 3-tier

Trong kiến trúc 3-tier, ba tầng Presentation, Application và Resource Manager được tách biệt rõ ràng.

Với một số nhà nghiên cứu, kiến trúc Middleware cũng được coi là kiến trúc 3-tier. Điều này đúng về khái niệm do các hệ thống có thể xem là các hộp đen. Thực tế kiến trúc 3-tier cũng chỉ có ý nghĩa trong hệ thống Middleware.

Kiến trúc 3-tier có cùng ưu điểm với kiến trúc Middleware [1].



Hình 1.6 Kiến trúc 3-tier

### 1.3. Một số phương pháp tích hợp

Tích hợp hệ thống có thể được thực hiện với hai hướng tiếp cận: từ dưới lên và từ trên xuống.

- Từ dưới lên (Bottom – Up):
  - + Đánh giá hiện trạng và phân loại các hệ thống đã có.
  - + Phân tích những vấn đề cụ thể phát sinh do thiếu sự tích hợp giữa các hệ thống.
  - + Giải quyết những vấn đề đó thông qua những dự án tích hợp không có điều phối, không cần xây dựng kiến trúc tích hợp tổng thể.
- Từ trên xuống (Top – Down):
  - + Đánh giá hiện trạng và phân loại các hệ thống đã có.
  - + Xây dựng kiến trúc tích hợp tổng thể sớm nhất có thể, đảm bảo cả những khía cạnh về quy trình nghiệp vụ lẫn công nghệ.

Hiện nay người ta thường sử dụng phương pháp từ trên xuống kết hợp với từ dưới lên[1].

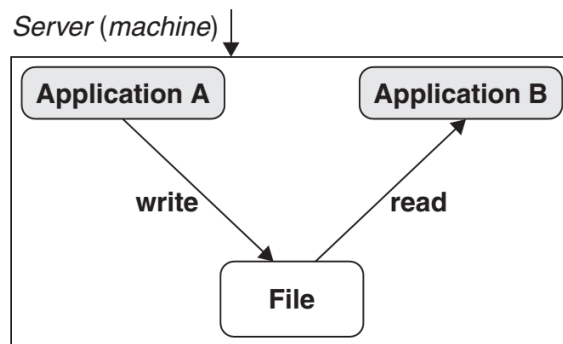
#### 1.3.1. Tích hợp mức dữ liệu

Đây là kiểu tích hợp ở mức thấp, các ứng dụng/hệ thống tham gia vào hệ tích hợp chia sẻ dữ liệu chung với nhau. Một số phương pháp chia sẻ dữ liệu

điển hình: Chia sẻ dữ liệu dạng tệp (File-based data sharing), Chia sẻ cơ sở dữ liệu (Shared Database), Đồng bộ tệp (Socket).

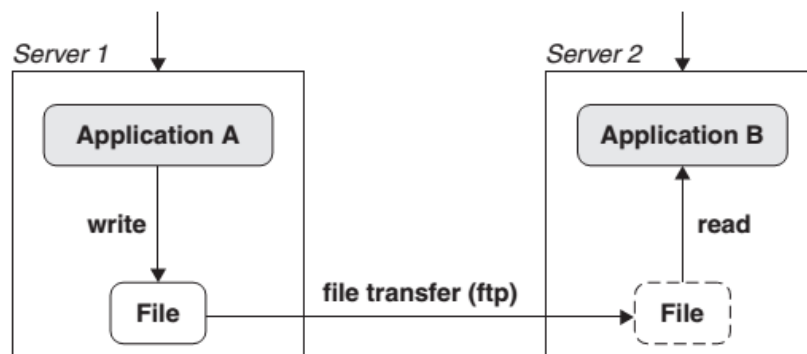
### ***Chia sẻ dữ liệu dạng tệp***

Đây là phương pháp phổ biến nhất trong chia sẻ dữ liệu. Phương pháp này phụ thuộc vào hạ tầng phần cứng và hệ điều hành. Với kiểu tích hợp này, một ứng dụng ghi dữ liệu vào tệp trong khi ứng dụng khác đọc dữ liệu từ những tệp tương tự. Nếu hai ứng dụng đều chạy trên một máy chủ, chúng sẽ sử dụng đĩa vật lý để ghi và đọc như hình minh họa 1.7.



Hình 1.7 Hai ứng dụng A, B cùng được cài đặt trên một máy chủ

Nếu hai ứng dụng được cài đặt trên hai máy chủ khác nhau thì sẽ sử dụng các kỹ thuật truyền tệp để truyền tệp dữ liệu giữa hai đĩa vật lý điển hình là sử dụng giao thức ftp (file transfer protocol).



Hình 1.8 Hai ứng dụng A và B được cài trên hai máy chủ khác nhau

Tệp dữ liệu sử dụng chủ yếu với kiểu tích hợp này là text vì một ký tự được biểu diễn một byte trong phần lớn hệ điều hành, ngôn ngữ.

*Ưu điểm:*

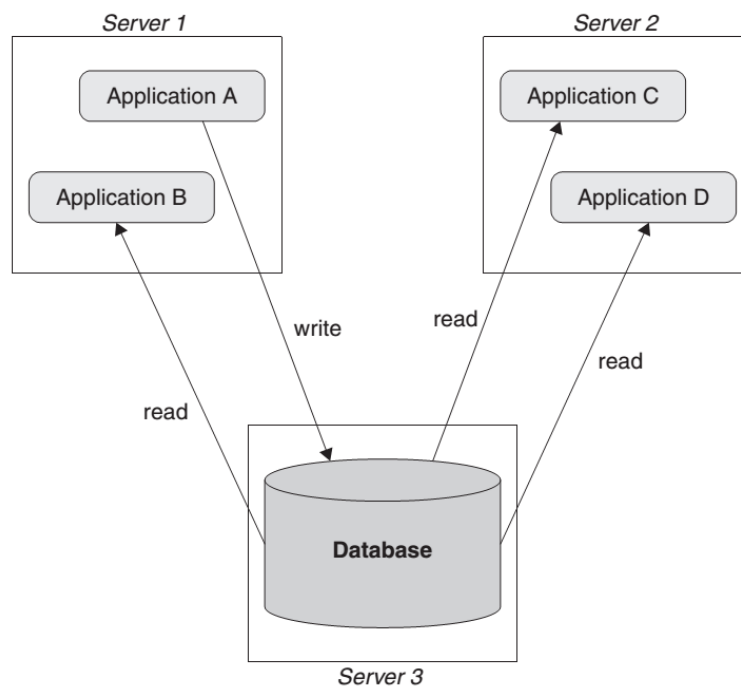
- Đơn giản, dễ xây dựng.

Nhược điểm:

- Dữ liệu không được chia sẻ trong thời gian thực. Tồn tại độ trễ đáng kể giữa thời điểm một ứng dụng ghi tệp và ứng dụng khác đọc tệp. Độ trễ này có thể được tính bằng giờ, ngày thậm chí cả tuần.
- Phương pháp này không đáng tin cậy nếu có số lượng lớn tệp dữ liệu được chia sẻ.
- Việc truyền tệp dữ liệu yêu cầu về cơ chế quản lý việc đọc ghi tệp dữ liệu tại một thời điểm. Nếu đọc ghi đồng thời lên tệp dữ liệu có thể gây ra tranh chấp, dữ liệu không nhất quán.
- Không phù hợp khi tích hợp nhiều ứng dụng. Với  $n$  ứng dụng cần tích hợp sẽ cần  $n*(n-1)/2$  phương thức chia sẻ tệp dữ liệu.

### ***Chia sẻ cơ sở dữ liệu***

Phương pháp này gần giống với phương pháp chia sẻ dữ liệu dạng tệp, tuy nhiên ở phương pháp này một ứng dụng ghi dữ liệu vào cơ sở dữ liệu, ứng dụng khác đọc dữ liệu từ cơ sở dữ liệu. Điểm khác biệt với phương pháp chia sẻ dữ liệu dạng tệp là cơ sở dữ liệu luôn được đặt trên một máy chủ riêng. Điều đó có nghĩa việc chia sẻ dữ liệu luôn xảy ra trên môi trường mạng mặc dù các ứng dụng có thể được cài trên cùng máy chủ. Vì thế phương pháp này thường xử lý lâu hơn so với phương pháp chia sẻ dữ liệu dạng tệp.



Hình 1.9 Các ứng dụng chia sẻ cơ sở dữ liệu



*Ưu điểm:*

- Hệ quản trị cơ sở dữ liệu có trách nhiệm bảo đảm tính nhất quán của dữ liệu được chia sẻ giữa các ứng dụng.
- Có thể có nhiều ứng dụng cùng chia sẻ dữ liệu.

*Nhược điểm:*

- Dữ liệu không được chia sẻ trong thời gian thực vì không có cơ chế thông báo dữ liệu cập nhật từ một ứng dụng tới các ứng dụng liên quan khác.
- Các ứng dụng phải sử dụng chung mô hình dữ liệu gây khó khăn cho các lập trình viên.
- Khi có nhiều ứng dụng tích hợp có thể gây quá tải cho hệ quản trị cơ sở dữ liệu.
- Các máy chủ ứng dụng không nên đặt tại nhiều địa điểm vì việc truy cập cơ sở dữ liệu qua mạng máy tính WAN có thể có độ trễ.

***Đồng bộ tập (Socket)***

Phương pháp này giải quyết vấn đề thời gian thực của hai phương pháp chia sẻ dữ liệu dạng tệp và chia sẻ cơ sở dữ liệu. Phương pháp này sử dụng kết nối trực tiếp để chia sẻ dữ liệu. Phương pháp này cho phép một ứng dụng lắng nghe trên một cổng nhất định trong khi các ứng dụng khác ghi vào cùng socket của địa chỉ và cổng của ứng dụng đầu tiên. Ứng dụng đầu tiên có thể đọc dữ liệu ngay khi ứng dụng thứ hai thực hiện ghi xong dữ liệu.

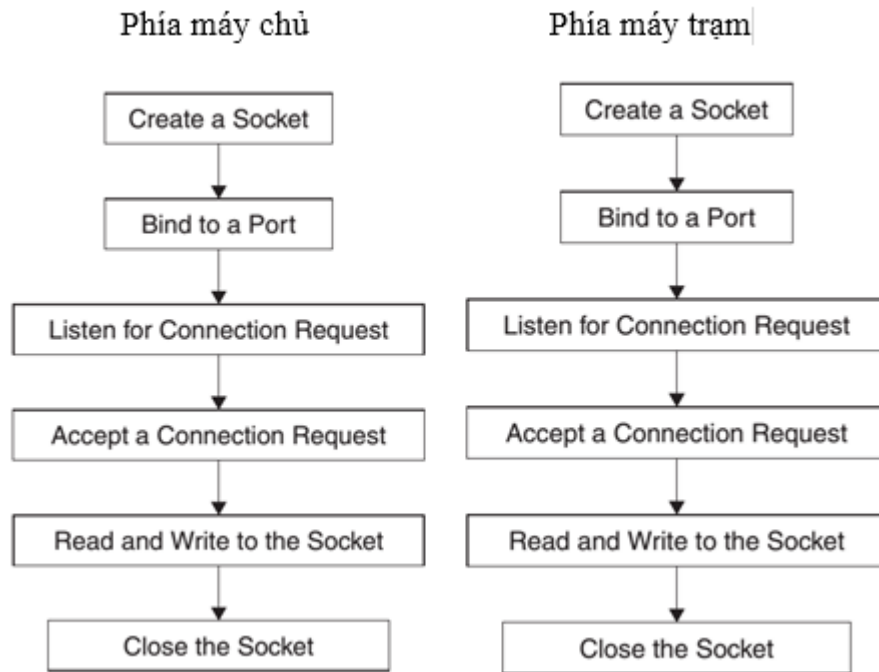
*Ưu điểm:*

- Không phải chia sẻ toàn bộ dữ liệu.
- Dữ liệu có thể được cập nhật nhanh hơn.
- Có thể tạo mô hình kết nối 1-n.

*Nhược điểm:*

- Cần xây dựng ứng dụng ở mức thấp, sử dụng các system-call về mạng.
- Cần kết nối chặt chẽ giữa các ứng dụng.

*Các bước xây dựng socket:* phía máy chủ và phía máy trạm



Hình 1.10 Các bước xây dựng socket

### 1.3.2. Tích hợp mức chức năng

Là phương pháp cho phép các ứng dụng chia sẻ các chức năng lẫn nhau.

Một số phương thức điển hình của tích hợp mức chức năng:

- Gọi thủ tục từ xa (Remote Procedure Call)
- Đối tượng phân tán (Distributed Object)
- Thông điệp (Message)

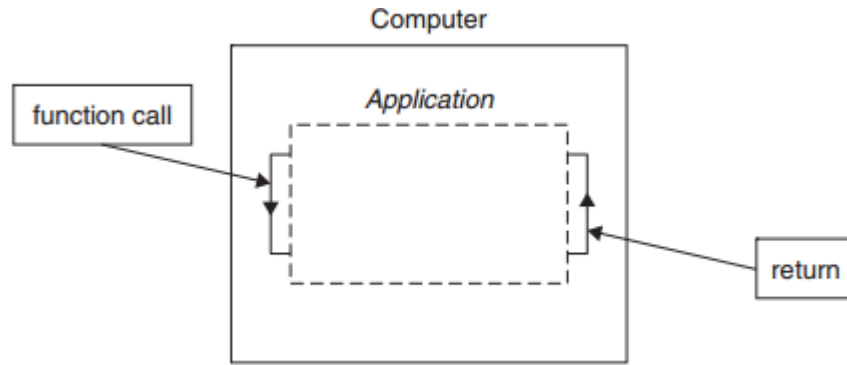
#### ***Gọi thủ tục từ xa – RPC***

RPC là một bước quan trọng trong quá trình hướng tới tích hợp vì nó giới thiệu một số nội dung và chức năng quan trọng, đặc biệt là một bước cơ bản trong chia sẻ chức năng.

RPC được thực hiện theo kiểu đồng bộ chức năng: ứng dụng gọi đến hàm phải chờ đến khi nhận được kết quả trả về mới tiếp tục công việc khác.

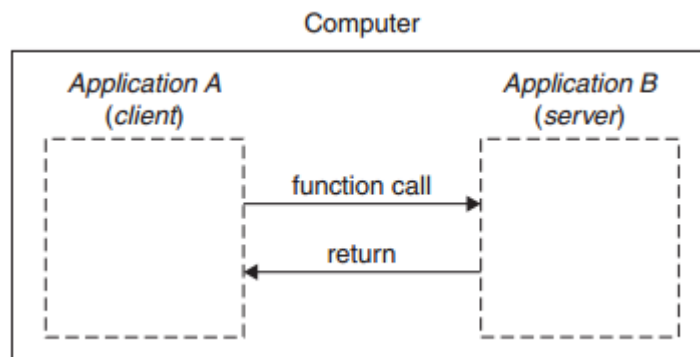
Đồng bộ chức năng có 3 loại hàm gọi:

- Local function call: hàm gọi và chức năng được gọi trên cùng một ứng dụng.



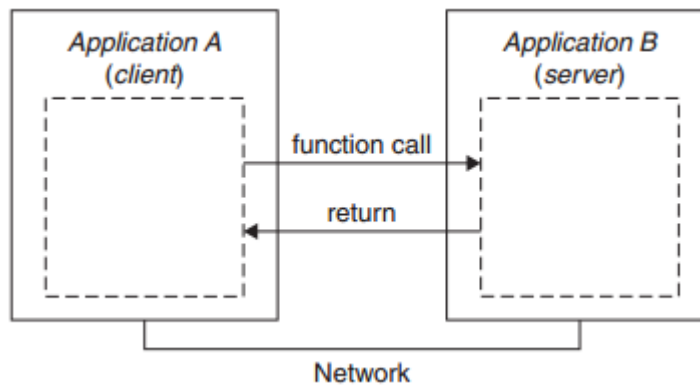
Hình 1.11 Local function call

- Restricted RPC: Hàm gọi và chức năng được gọi trên các ứng dụng khác nhau cài trên cùng một máy chủ



Hình 1.12 Restricted RPC

- Loại 3: Ứng dụng client trên một máy chủ gọi hàm trên một máy chủ ứng dụng khác, hai máy chủ này kết nối với nhau qua mạng máy tính.



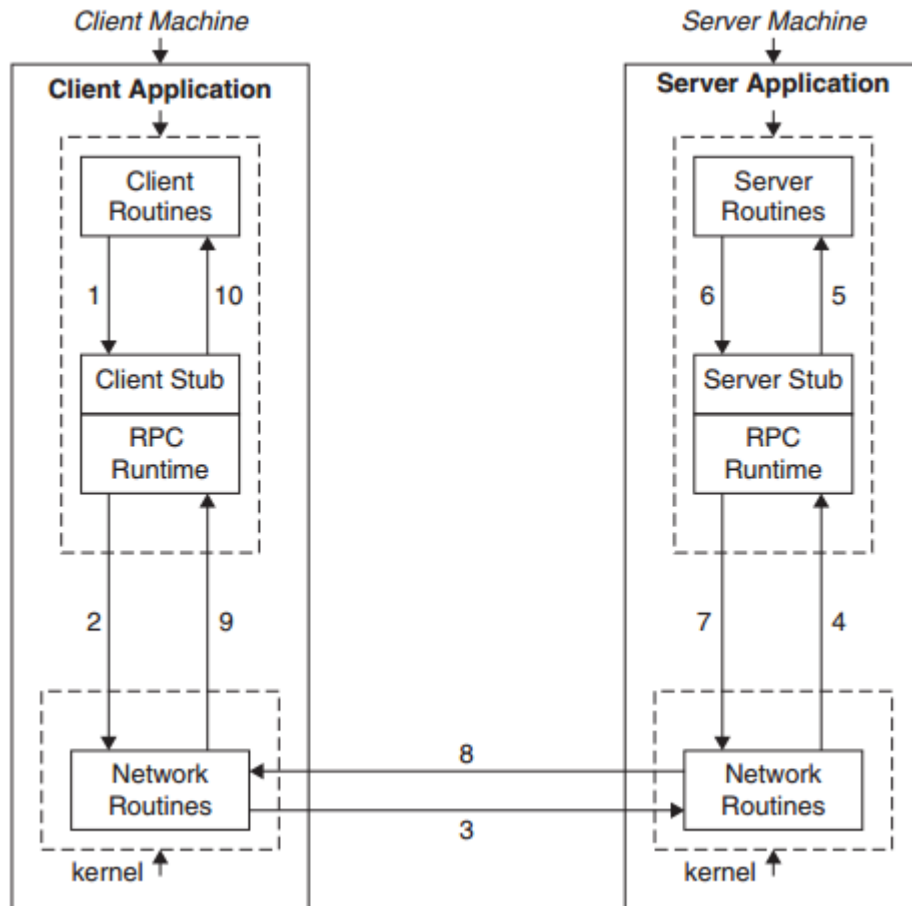
Hình 1.13 Hai ứng dụng trên hai máy chủ khác nhau

RPC cho phép:

- Ẩn chi tiết truyền thông trong các lời gọi hàm.
- Trở thành cầu nối giữa các môi trường nền tảng khác nhau.

RPC là một trong những chuẩn điển hình trong tính toán phân tán.

*Các bước cơ bản khi gọi hàm:*



Hình 1.14 Các bước cơ bản khi gọi hàm

*Ưu điểm:*

- Là phương thức đầu tiên cho phép chia sẻ hàm (chức năng) giữa các ứng dụng.

- Có thể triển khai với nhiều nền tảng khác nhau.

*Nhược điểm:*

- Các ứng dụng phải sử dụng cùng ngôn ngữ lập trình.

- Kết nối chặt chẽ do sử dụng cơ chế hàm đồng bộ.

- Vẫn dựa trên mô hình tích hợp point – to – point.

- Rất phức tạp khi có nhiều lời gọi hàm.

### ***Đối tượng phân tán – Distributed Object***

Trong khi RPC chưa cho phép tích hợp ứng dụng với nhiều ngôn ngữ lập trình trên nhiều hệ điều hành khác nhau thì phương thức này giải quyết vấn đề đó.

*Ưu điểm:*

- Độc lập với ngôn ngữ lập trình và nền tảng môi trường.
- Làm mờ vai trò của máy khách và máy chủ.

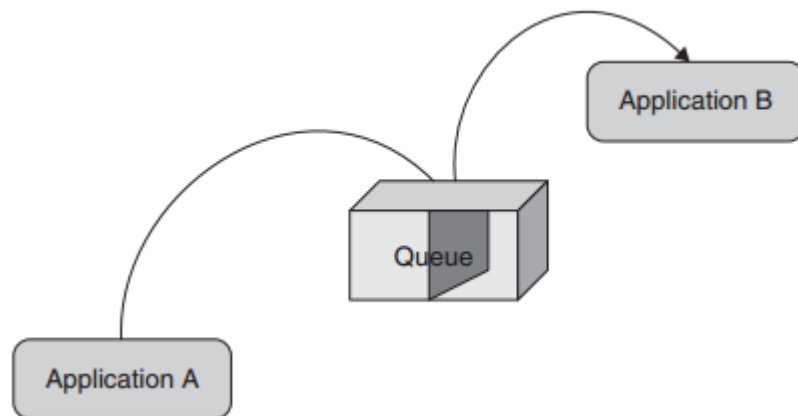
*Nhược điểm:*

- Vẫn cần cơ chế gọi hàm đồng bộ: máy khách phải bị khóa đến khi nhận được kết quả từ máy chủ.
- Dựa trên phương thức truyền thông không đảm bảo tin cậy: có thể yêu cầu và kết quả trả về không đến được đích mong muốn.

### ***Thông điệp – Message***

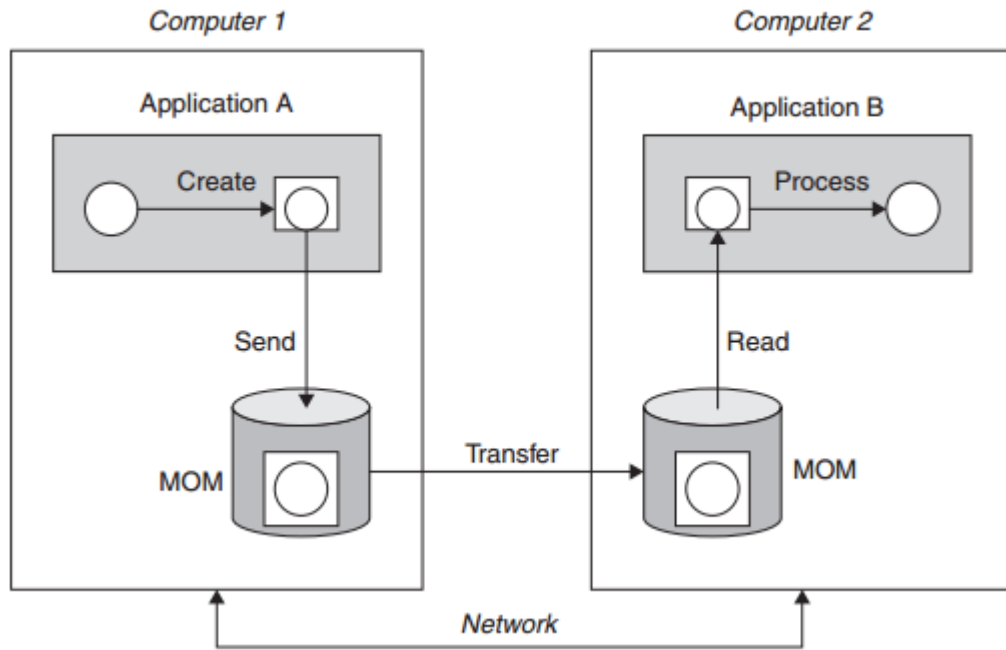
Phương thức này cho phép giải quyết các vấn đề tồn tại của hai phương thức trên. Phương thức này dựa trên cơ chế tương tác thông điệp không đồng bộ máy khách gửi yêu cầu tới máy chủ mà không cần chờ phản hồi từ máy chủ. Điều đó cho phép máy khách thực hiện các công việc khác trong khi chờ máy chủ hoàn thành yêu cầu từ máy khách.

Trong phương thức thông điệp, các ứng dụng không tương tác với nhau trực tiếp và không có một kênh truyền thông chuyên biệt được tạo ra giữa chúng. Thay vào đó, chúng tương tác gián tiếp qua hàng đợi. Một hàng đợi (đôi khi còn được gọi là một kênh) là tập các thông điệp có thể được chia sẻ với nhiều máy tính.



Hình 1.15 Mô hình thông điệp không đồng bộ sử dụng hàng đợi

Những đoạn mã để kết nối được xây dựng riêng biệt như một thành phần riêng biệt của phần mềm. Những đoạn mã này được gọi là messaging system hay Message-oriented middleware (MOM).



Hình 1.16 Các bước cơ bản để truyền thông điệp

Một tính năng quan trọng khác của Message system là khi thông điệp chưa được gửi tới đích mong muốn, MOM sẽ gửi lại thông điệp một lần nữa cho đến khi thông điệp được chuyển.

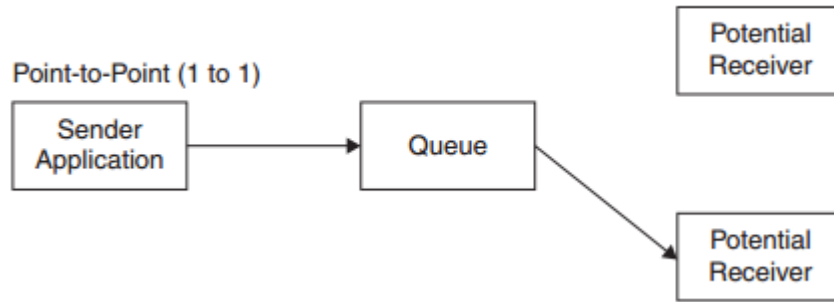
Ứng dụng có thể được thiết kế để chạy mà không cần kết nối mạng. Ví dụ ứng dụng trên máy tính cá nhân được đồng bộ vào hàng đợi và chờ cho đến khi máy tính có kết nối tới máy chủ.

Máy chủ có thể tránh được tình trạng quá tải thông qua cơ chế điều phối thông điệp ở MOM trong khi các máy khách không bị ảnh hưởng bởi điều này vì thông tin liên lạc là không đồng bộ.

Ba thành phần cơ bản của MOM: Hàng đợi, thông điệp, điểm kết thúc

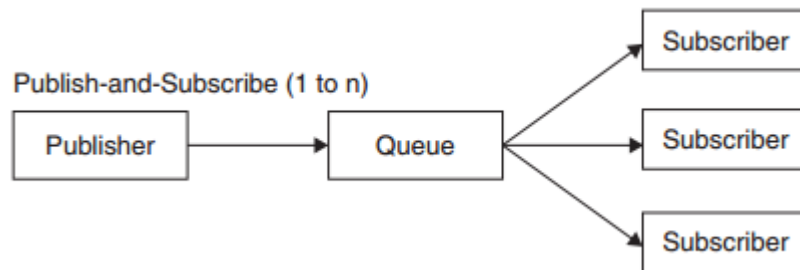
- Hàng đợi: được sử dụng để truyền dữ liệu. Mỗi hàng đợi như một ống ảo kết nối giữa bên gửi và bên nhận. Có hai loại hàng đợi:

+ point – to – point: có thể có nhiều đầu nhận nhưng chỉ có một đầu nhận có thể nhận đối với mỗi thông điệp.



Hình 1.17 Hàng đợi kiểu point – to – point

+ Push and Subscribe: thông điệp được gửi tới tất cả các subscriber. Mỗi subscriber lưu một bản sao của thông điệp. Publisher không quan tâm tới ai đang lắng nghe.



Hình 1.18 Hàng đợi kiểu publish – and – subscribe

- Thông điệp: bao gồm phần header và phần body.
  - + Phần header chứa định nghĩa thông điệp, thông tin điều khiển. Một số thuộc tính thông thường của phần header: ID, địa chỉ trả lại, mức độ quan trọng, gói tin, thời gian, vòng đời của thông điệp, phiên bản.
  - + Phần body chứa thông tin sẽ được xử lý trong ứng dụng nhận
- Điểm kết thúc: chứa tập các mã được sử dụng để kết nối tới MOM và để gửi hay nhận một thông điệp.

*Ưu điểm:*

- Nâng cao tính khả mở của hệ thống tích hợp.
- Đảm bảo được độ tin cậy của quá trình truyền thông.

*Nhược điểm:*

- Không đồng nhất:
  - + Middleware không đồng nhất: do sử dụng nhiều MOM
  - + Giao thức không đồng nhất: sử dụng nhiều giao thức như HTTP, HTTPS.

- + Ứng dụng hỗ trợ phương thức kết nối không đồng nhất: có ứng dụng sử dụng cách kết nối đồng bộ, có ứng dụng sử dụng cách kết nối không đồng bộ.
- + Định dạng thông điệp không đồng nhất.
- Có những ứng dụng cần cả phương thức gọi hàm đồng bộ và không đồng bộ[7].

### **1.3.3. Tích hợp mức dịch vụ (quy trình)**

Tích hợp mức dịch vụ là kiểu tích hợp mức cao, cho phép khắc phục những nhược điểm của phương pháp thông điệp.

Phương pháp này có 2 loại:

- Tích hợp hệ thống dựa vào tích hợp quy trình nghiệp vụ
- Tích hợp hệ thống dựa vào kiến trúc hướng dịch vụ

#### ***Tích hợp quy trình***

Tích hợp mức quy trình đảm bảo mục tiêu tạo mô hình nghiệp vụ chung giữa các hệ thống liên kết qua dịch vụ và quy trình.

Kiểu tích hợp này thường được sử dụng trong các hệ thống:

- Dịch vụ khách hàng
- Quản trị nguồn nhân lực
- Giao dịch tài chính

Mô hình quy trình chung thường phải đủ bao quát hết các quy trình trong hệ thống tích hợp[1].

#### ***Tích hợp hướng dịch vụ - SOA (Service Oriented Architecture)***

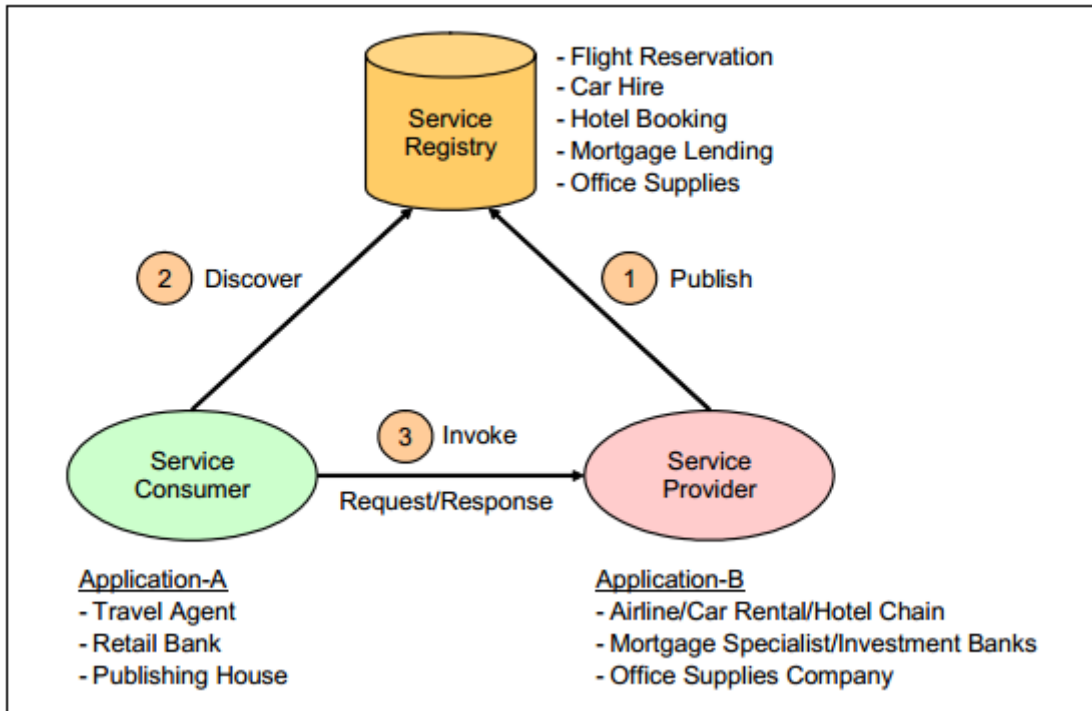
Kiến trúc hướng dịch vụ (SOA) là mô hình xây dựng ứng dụng dựa trên các dịch vụ đã có trên mạng chuyên biệt chẳng hạn như Web. SOA cho phép xác lập những mề đay giữa các thành phần, nâng cao hiệu quả tái sử dụng.

Các thành phần cơ bản của SOA:

- Service Provider: tạo ra dịch vụ và cung cấp thông tin về giao diện, truy cập cho service registry. Mỗi nhà cung cấp dịch vụ phải quyết định dịch vụ sẽ cung cấp, đánh giá giữa vấn đề an ninh và tính sẵn sàng, xác định làm sao để bán dịch vụ hoặc làm sao để khai thác dịch vụ miễn phí.



- Service Consumer: xác định thông tin của service registry, sau đó liên kết với service provider để gọi dịch vụ.
- Service Registry: tạo ra giao diện dịch vụ và cung cấp khả năng truy cập thông tin có sẵn tới service consumer[6].



Hình 1.19 Thành phần của SOA

Nguyên lý cơ bản của SOA:

- Liên kết lỏng lẻo: đảm bảo tính mềm dẻo của SOA, các dịch vụ không cần ràng buộc chặt chẽ với nhau.
- Tính tự trị: các dịch vụ có quyền kiểm soát dựa vào cấu trúc logic bên trong của nó.
- Tính chia sẻ hợp đồng: các dịch vụ trong hệ thống hoạt động tuân theo một hợp đồng chính thức.
  - Sử dụng lại
  - Đóng gói: các dịch vụ che giấu logic bên trong của mình.
  - Phi trạng thái: các dịch vụ hoạt động phi trạng thái.
  - Có thể tìm thấy: người dùng có thể tìm kiếm dịch vụ cần sử dụng và đăng ký sử dụng dịch vụ đó[1].

SOA có hai mô hình ứng dụng chính là Web service và ESB.

- Web service: hỗ trợ tích hợp dịch vụ, trong đó các dịch vụ được tích hợp phải kết nối trực tiếp với nhau. Dịch vụ web dựa trên các tiêu chuẩn XML, SOAP, WSDL, UDDI.
  - + XML (Extensible Markup Language): cung cấp một định dạng trung gian để trao đổi dữ liệu và tài liệu
  - + SOAP (Simple Object Access Protocol): cung cấp định dạng thông điệp để kết hợp.
  - + WSDL (Web Services Description Language): cung cấp một ngôn ngữ và nền tảng độc lập để xác định giao diện được cung cấp bởi một dịch vụ. Một tài liệu WSDL bao gồm hai phần: phần đầu mô tả tóm tắt các toán tử, tham số đầu vào, đầu ra và loại dữ liệu; phần hai bao gồm các thông tin về giao diện, quy định giao thức thực hiện, định dạng thông điệp và địa chỉ mạng.
  - + UDDI (Universal Description, Discovery and Integration): giao thức hỗ trợ quá trình tích hợp, phát hiện cũng như mô tả tổng thể các dịch vụ.
- Trục dịch vụ tổng thể ESB: cung cấp giải pháp kết nối nhiều ứng dụng mà không cần mỗi cặp ứng dụng phải kết nối trực tiếp với nhau mà kết nối thông qua một trục tích hợp [1, 7].

Hiện tại NHNN với các nghiệp vụ riêng biệt sử dụng các công nghệ khác nhau nên để tích hợp các hệ thống, chúng tôi hướng đến sử dụng ESB. Các khái niệm về ESB sẽ được trình bày rõ hơn trong chương sau.

#### **1.4. Kết luận**

Chương này trình bày tổng quan về tích hợp hệ thống bao gồm: các khái niệm tích hợp hệ thống, kiến trúc đa tầng trong tích hợp hệ thống và một số phương pháp tích hợp hệ thống (tích hợp mức dữ liệu, mức chức năng và mức dịch vụ)

## **CHƯƠNG 2. TÍCH HỢP DỊCH VỤ THEO MÔ HÌNH TRỰC DỊCH VỤ TỔNG THỂ ESB**

### **2.1. Tổng quan về trực dịch vụ tổng thể ESB**

Trực dịch vụ tổng thể ESB (Enterprise Service Bus) cung cấp một cách toàn diện, mở rộng việc kết nối nhiều ứng dụng mà không cần mỗi cặp ứng dụng phải kết nối trực tiếp với nhau. ESB là một trong những kỹ thuật chủ yếu trong phương thức tích hợp mức dịch vụ theo mô hình kiến trúc hướng dịch vụ SOA.

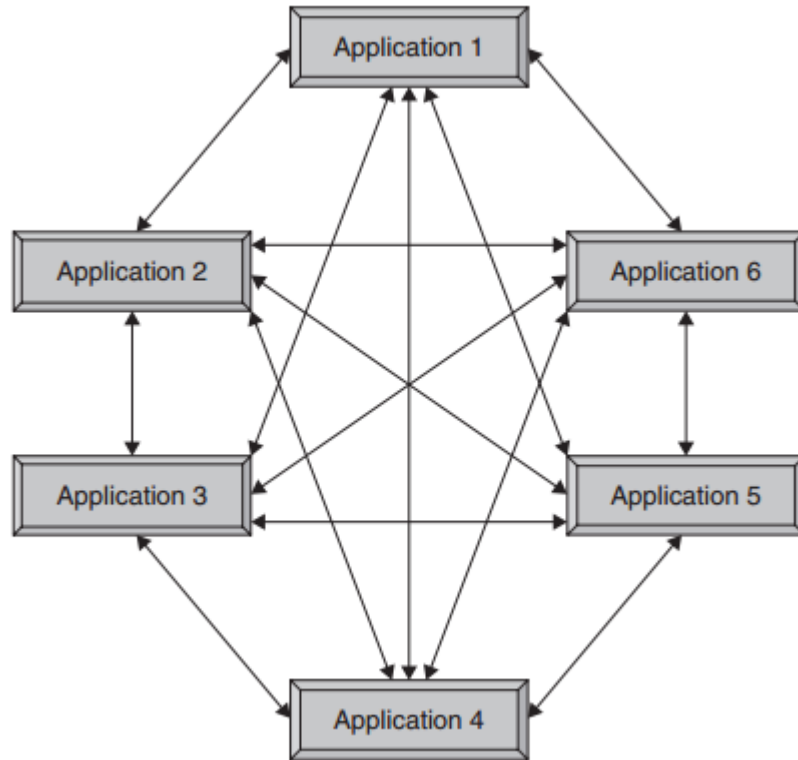
Nhìn chung, các công cụ xây dựng theo mô hình ESB sẽ cung cấp một mô hình chung để triển khai, quản lý và quản trị các dịch vụ, cho phép tích hợp hệ thống mức dịch vụ.

### **2.2. Chức năng lõi của ESB:**

- Kết nối định tuyến dựa trên nội dung và bối cảnh
- Chuyển đổi giao thức
- Chuyển đổi dữ liệu/thông điệp

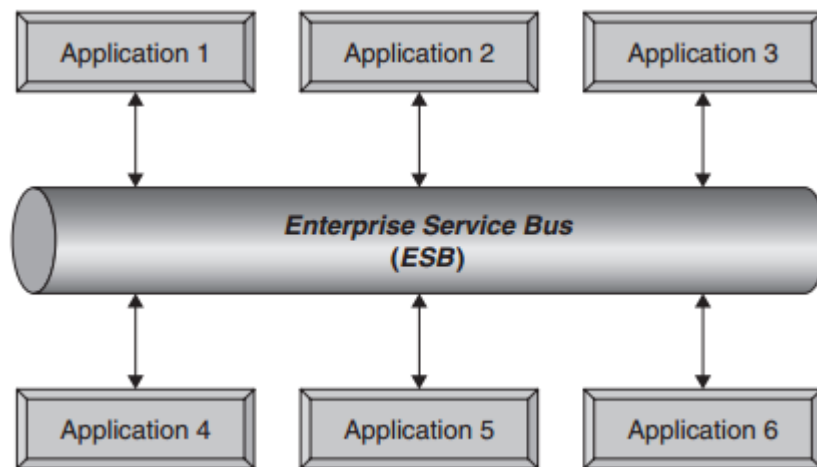
#### **2.2.1. Kết nối định tuyến**

Mặc dù dịch vụ Web đã giải quyết được các vấn đề về đồng bộ trong hệ thống lớn, tuy nhiên chúng chỉ cung cấp một phần của giải pháp vì tích hợp dịch vụ Web vẫn sử dụng phương pháp điểm – điểm. Với N ứng dụng, phương pháp này sẽ cần  $N*(N-1)/2$  kết nối giữa các ứng dụng, do đó phương pháp này chỉ phù hợp với những tổ chức nhỏ với một vài ứng dụng không phù hợp với các tổ chức và hệ thống lớn. Ví dụ với 6 ứng dụng sẽ cần 15 kết nối giữa các ứng dụng



Hình 2.1 Mô hình kết nối sử dụng phương pháp điểm – điểm

Giải quyết vấn đề của phương pháp điểm – điểm, ESB cung cấp một giải pháp phù hợp với các tổ chức lớn cần tích hợp số lượng lớn ứng dụng. Trong ESB, các ứng dụng không tương tác trực tiếp với nhau mà thay vào đó ứng dụng kết nối với bus. Bus cung cấp kết nối giữa các ứng dụng. Với  $N$  ứng dụng tích hợp qua ESB chỉ cần  $N$  kết nối. Ví dụ như hình 2.2 với 6 ứng dụng chỉ cần có 6 kết nối để tích hợp.



Hình 2.2 Mô hình kết nối sử dụng ESB

Một ưu điểm khi tích hợp qua ESB là việc thêm mới hay loại bỏ các ứng dụng khỏi hệ thống tích hợp linh động, dễ dàng. Khi thêm mới hoặc loại bỏ ứng dụng chỉ cần thêm mới kết nối hoặc loại bỏ kết nối của ứng dụng đó với ESB mà không ảnh hưởng đến các ứng dụng khác.

Một ưu điểm khác của ESB là tính linh hoạt, nhà cung cấp dịch vụ không cần quan tâm tới ứng dụng nào đang gọi dịch vụ và ứng dụng không cần quan tâm tới nhà cung cấp dịch vụ. Với cấu trúc dịch vụ bus không cần phải chỉ rõ địa chỉ mạng của dịch vụ tới ứng dụng, bus cung cấp khả năng tìm kiếm địa chỉ của dịch vụ cuối dựa trên nội dung và bối cảnh của các yêu cầu nhận được từ ứng dụng client.

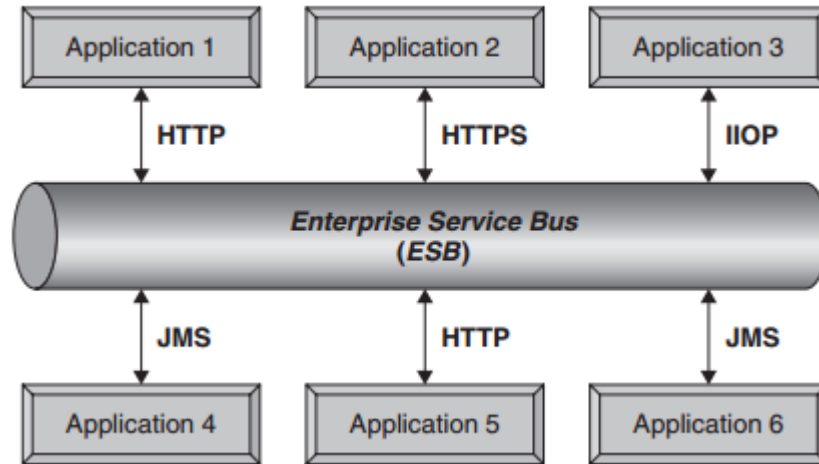
Có 2 loại BUS:

- Loại đầu tiên sử dụng ORBs (Object Request Broker) (máy chủ ứng dụng) như là xương sống. Loại này có ưu điểm là dễ dàng thiết lập và ít tốn kém. Tuy nhiên chức năng mà nó cung cấp không tỷ lệ với giao dịch có liên quan nên loại này phù hợp với hệ thống có lượng giao dịch thấp. Loại này chỉ được thiết kế để sử dụng với Web service, XML, Java RMI.
- Loại thứ hai dựa trên hệ thống tin nhắn không đồng bộ (asynchronous message system). Nó có giá thành đắt hơn và đòi hỏi thiết lập phức tạp hơn loại trên. Nó có ba ưu điểm vượt trội hơn so với loại trên:
  - + Khả năng mở rộng về khối lượng giao dịch, vì thế thường được dùng cho các hệ thống có tỷ lệ giao dịch lớn.
  - + Được sử dụng để tích hợp đa dạng các ứng dụng.
  - + Đảm bảo thông điệp được truyền giữa các ứng dụng.

### **2.2.2. Chuyển đổi giao thức**

Trong các tổ chức lớn, các ứng dụng khác nhau được phát triển riêng rẽ sử dụng những giao thức khác nhau như HTTP, HTTPS, JMS,... Khi đó sự không phù hợp về giao thức để tích hợp giữa ứng dụng là một khó khăn lớn. Lý tưởng nhất là tổ chức thực hiện chuẩn hóa chỉ sử dụng chung một giao thức thống nhất cho tất cả các ứng dụng. Tuy nhiên việc này đòi hỏi nguồn lực lớn về tài chính, con người. Thêm nữa chất lượng và tính an toàn của dịch vụ khi đó có thể không được đảm bảo.

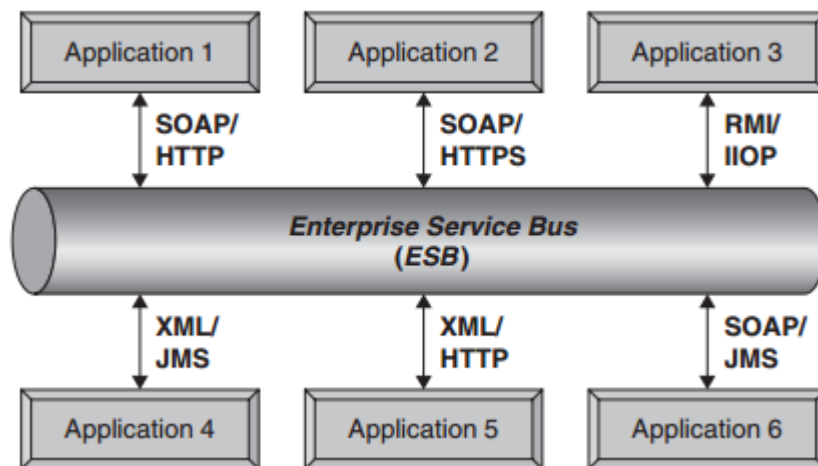
ESB đã giải quyết được vấn đề này. Một trong những chức năng cốt lõi của ESB chính là khả năng chuyển đổi giao thức, cho phép các ứng dụng có thể sử dụng các giao thức khác nhau mà vẫn có thể kết nối.



Hình 2.3 Các ứng dụng sử dụng các giao thức khác nhau kết nối qua ESB

### 2.2.3. Chuyển đổi dữ liệu/ thông điệp

Khi kết nối thông qua ESB, các ứng dụng có thể dễ dàng kết nối và tương tác với nhau ngay cả khi các định dạng dữ liệu/thông điệp khác nhau



Hình 2.4 Các ứng dụng có định dạng dữ liệu khác nhau kết nối qua ESB

### 2.2.4. Các nhóm ảo hóa

Từ ba chức năng trên, ESB có thể hỗ trợ các nhóm ảo hóa:

- Ảo hóa về vị trí và danh tính: các ứng dụng yêu cầu dịch vụ không cần biết vị trí của các ứng dụng cung cấp dịch vụ, các ứng dụng cung cấp dịch vụ

cũng không quan tâm tới danh tính của ứng dụng yêu cầu dịch vụ. Các yêu cầu có thể được cung cấp bởi nhiều ứng dụng. Điều đó cho phép có thể thêm mới hoặc xóa bỏ một ứng dụng cung cấp dịch vụ khỏi cấu trúc tích hợp mà không gây gián đoạn hệ thống.

- Ảo hóa về giao thức tương tác: ứng dụng yêu cầu dịch vụ và nhà cung cấp dịch vụ không cần chia sẻ giao thức truyền thông giống nhau.

- Ảo hóa về giao diện giao tiếp: ứng dụng yêu cầu dịch vụ không cần phải gửi yêu cầu theo đúng giao diện mà ứng dụng cung cấp dịch vụ yêu cầu.

Việc ảo hóa ở các khía cạnh trên cho phép ESB có thể cung cấp dịch vụ minh bạch tới ứng dụng yêu cầu dịch vụ cả về mặt thời gian phát triển và thời gian triển khai.

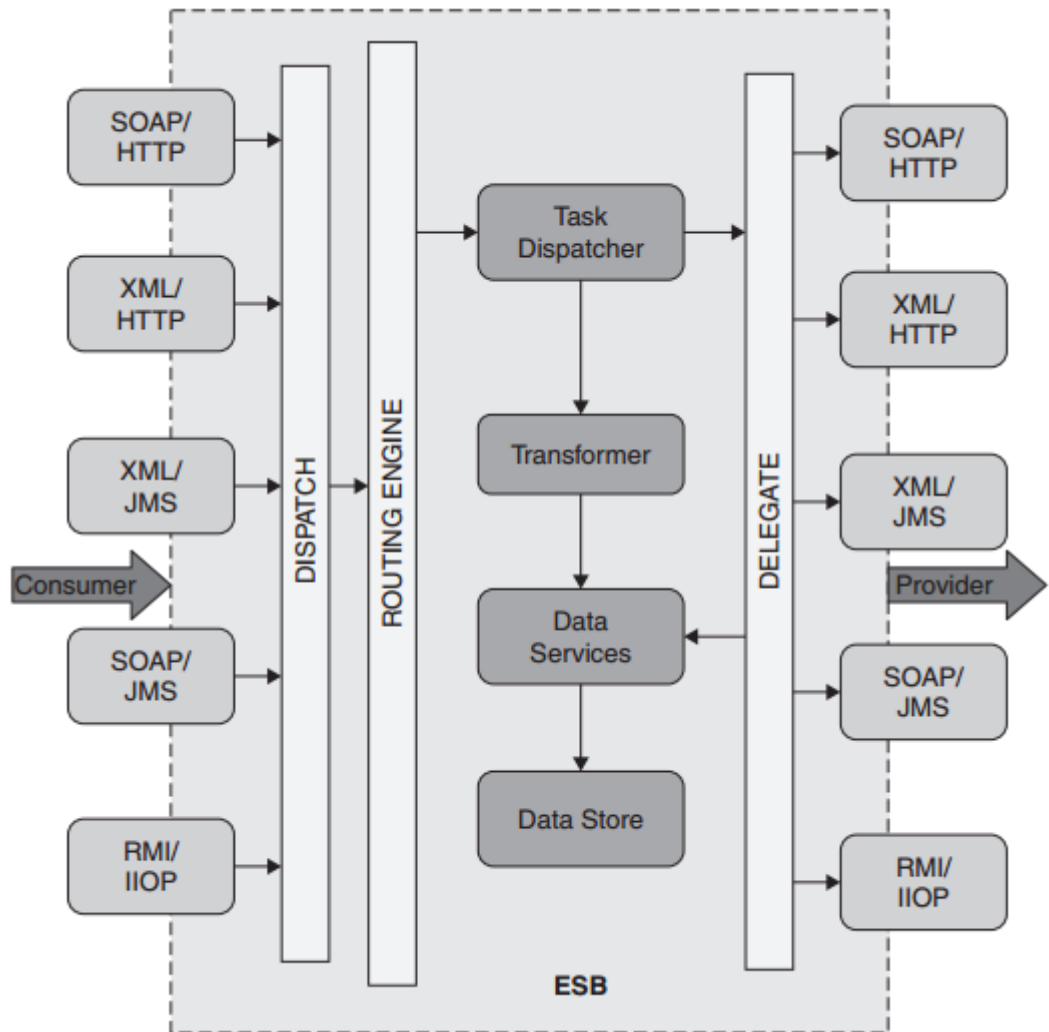
#### **2.2.5. Các yêu cầu phi chức năng đối với ESB:**

Yêu cầu về hiệu suất và khả năng đáng tin cậy: bao gồm thời gian phản ứng của một dịch vụ không vượt quá một mức cố định nào đó. Khả năng tin cậy của hệ thống có thể hoạt động đến bao nhiêu %.

Yêu cầu về an ninh bảo mật của dịch vụ: ESB cung cấp một vài dịch vụ về an ninh, bảo mật:

- Mã hóa dữ liệu;
- Xác thực khi yêu cầu dịch vụ;
- Toàn vẹn dữ liệu;
- Thẩm tra dịch vụ.

## 2.3. Các thành phần logic



Hình 2.5 Các thành phần logic của ESB

### 2.3.1. Bộ chuyển đổi - Adapter

Là thành phần quan trọng nhất của ESB, nằm ở vùng ngoại vi của ESB. Tất cả các yêu cầu đi vào và đi ra đều sử dụng adapter. Adapter cho phép ESB tương tác với nhiều cơ chế đầu vào và đầu ra. Mỗi adapter phục vụ nhu cầu của một dịch vụ cụ thể. Điều này cho phép ESB nhận đầu vào và gửi đầu ra ở bất kỳ giao thức nào.

### 2.3.2. Thành phần điều phối - Dispatcher

Dispatcher hoạt động như các điểm trung tâm. Dispatcher chịu trách nhiệm cho việc lấy đầu vào từ các adapter và đi qua nó để thực hiện việc định tuyến, chuyển đổi thực thi tác vụ. Dispatcher gửi yêu cầu tới bộ quản lý yêu cầu



(request handle) và cùng với bộ quản lý yêu cầu cung cấp định tuyến dựa trên nội dung.

### **2.3.3. Thành phần quản lý yêu cầu - Request Handle**

Mỗi dịch vụ có thành phần quản lý yêu cầu riêng. Công cụ định tuyến nhận tham số dịch vụ từ thành phần quản lý yêu cầu, sau đó thành phần quản lý yêu cầu xử lý bàn giao yêu cầu cho công cụ định tuyến để thực hiện nhiệm vụ thích hợp

### **2.3.4. Công cụ định tuyến - Routing and Rule Engine**

Công cụ định tuyến có trách nhiệm thực hiện việc chuyển đổi và định tuyến các công việc tới các dịch vụ đại diện thích hợp.

### **2.3.5. Dịch vụ đại diện - Service Delegates**

Tương tự như Adapter nhưng nó được sử dụng ở đầu ra của ESB.

### **2.3.6. Công cụ chuyển đổi - Transformation Engine**

Thực hiện việc chuyển đổi định dạng của thông điệp/dữ liệu đầu vào thành định dạng phù hợp với ứng dụng cung cấp dịch vụ.

### **2.3.7. Enrichment Component**

Thành phần này cho phép ESB tăng thêm trọng tải thông điệp từ một nguồn bên ngoài để phù hợp với yêu cầu của ứng dụng cung cấp dịch vụ. Ví dụ, dịch vụ yêu cầu chứa địa chỉ không có tên quốc gia, tuy nhiên ứng dụng cung cấp dịch vụ cần địa chỉ đầy đủ bao gồm cả tên quốc gia. Khi đó, ESB có thể thêm vào thông điệp tên quốc gia từ một nguồn dữ liệu khác. Đôi khi công cụ chuyển đổi và enrichment được kết hợp thành một thành phần.

### **2.3.8. Ghi nhật ký - Logging Component**

Thành phần này cho phép ghi nhật ký để hỗ trợ các thành phần ESB.

### **2.3.9. Xử lý ngoại lệ - Exception-Handling Component**

Thành phần này xử lý tất cả các ngoại lệ được tạo ra bởi các thành phần khác nhau của ESB.

## **2.4. Phân loại ESB**

### **2.4.1. ESB dựa trên máy chủ ứng dụng**

Đây là loại ESB sử dụng máy chủ ứng dụng như xương sống. Ngoài chức năng đồng bộ điển hình, nó còn hỗ trợ thông điệp không đồng bộ.

*Ưu điểm:*

- Giá thành rẻ;
- Dễ dàng cài đặt, thiết lập .

Loại ESB này thích hợp nhất với ngôn ngữ XML và Java. Tuy nhiên nó cũng đáp ứng được các loại ứng dụng khác khi cần tích hợp. Loại này thường sử dụng với số lượng nhỏ ứng dụng.

### **2.4.2. ESB dựa trên hệ thống thông điệp**

Với loại này xương sống chính là hệ thống thông điệp. Nó hỗ trợ cả thông điệp đồng bộ và thông điệp không đồng bộ.

*Ưu điểm:*

- Có khả năng mở rộng trong việc tích hợp ứng dụng cũng như hỗ trợ cho số lượng lớn các giao dịch;
- Hỗ trợ tích hợp đa dạng các ứng dụng bao gồm ứng dụng viết bằng Java, C/C++, COBOL;
- Đảm bảo thông điệp được chuyển giữa ứng dụng yêu cầu dịch vụ và nhà cung cấp dịch vụ.

*Nhược điểm:*

- Tồn chi phí hơn so với loại ESB khác;
- Cấu hình, cài đặt phức tạp hơn.

### **2.4.3. ESB dựa trên hạ tầng phần cứng**

Loại này dựa trên hạ tầng phần cứng để xử lý.

*Ưu điểm:*

- Dễ cài đặt
- Tăng an ninh bảo mật và hiệu quả xử lý [7].

## **2.5. Một số nền tảng hỗ trợ tích hợp dịch vụ theo ESB**

### **2.5.1. IBM WebSphere ESB**

Là một giải pháp trên máy chủ ứng dụng WebSphere dựa trên ESB, sử dụng JMS bus của máy chủ ứng dụng là chính. Giải pháp này được sử dụng chủ yếu cho môi trường dịch vụ Web. Giải pháp cung cấp web service dựa trên tích hợp hướng dịch vụ, chủ yếu làm việc với các ứng dụng J2EE/JAVA và WebService.

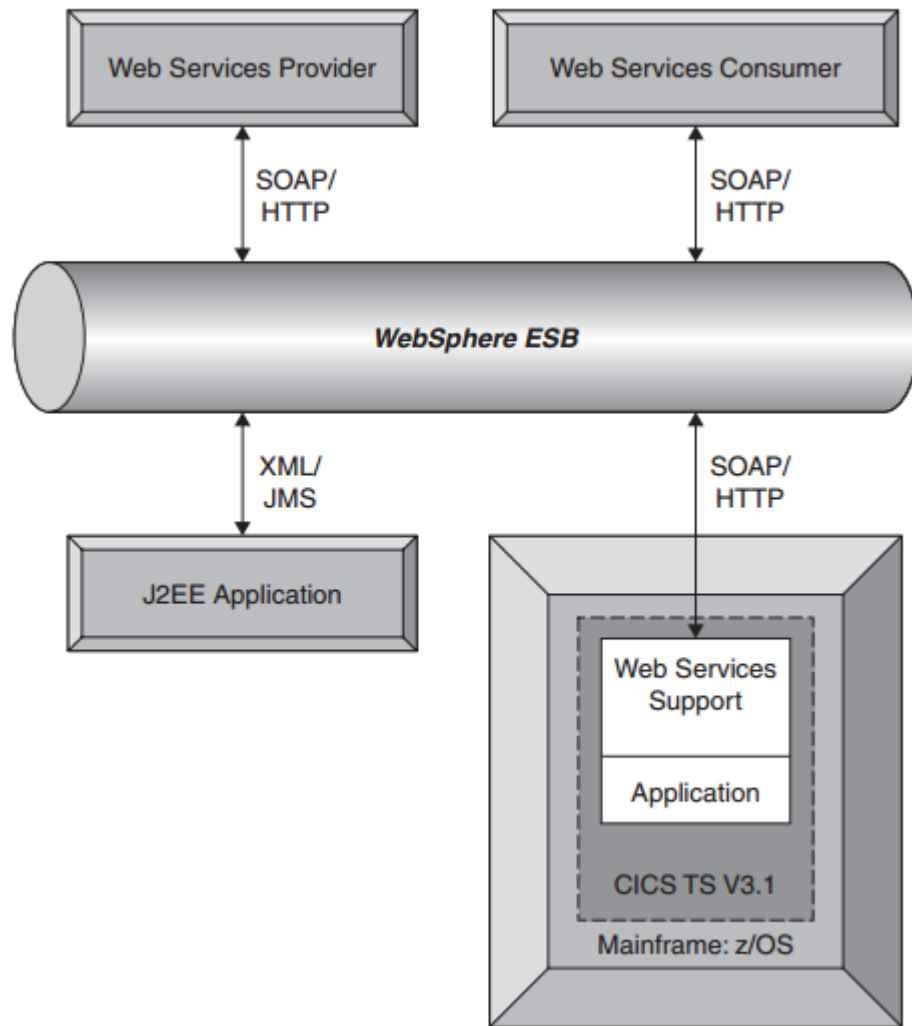
WebSphere ESB hỗ trợ tương tác giữa các dịch vụ cuối trên 3 cấp độ: kết nối rộng, mô hình và chất lượng tương tác phổ rộng, khả năng đàn xếp.

Các chức năng:

- Hỗ trợ API với ứng dụng Java Message Service (JMS) với giao thức TCP/IP, SSL, HTTP, HTTPS.
- Hỗ trợ kết nối các chuẩn web service chuẩn như: SOAP/HTTP, SOAP/JMS, WSDL 1.1, UDDI 3.0.
- Hỗ trợ giao thức chuyển đổi cho các thông điệp nhận được: HTTP, JMS, IIOP.
- Hỗ trợ chuyển dữ liệu giữa các giao thức XML, JMS, SOAP và nhiều chuẩn khác thông qua adapter.

*Ưu điểm:*

- Chi phí thấp;
- Dễ cài đặt, thiết lập: có thể cài trên nhiều môi trường hệ điều hành như Windows, Linux, zLinux, HP-UX, AIX, Solaris;
- Hỗ trợ giao thức HTTPS và SSL[6].



Hình 2.6 Mô hình tích hợp cho ứng dụng CICS mainframe

### 2.5.2. Talend ESB

Là giải pháp nguồn mở tốt nhất cho ESB.

*Ưu điểm:*

- Giá thành rẻ hơn so với các phần mềm thương mại;
- Cho phép người dùng truy cập trực tiếp vào mã nguồn;
- Tuân thủ chặt chẽ các chuẩn mở cho khả năng tương tác tối đa;
- Khả năng phát triển, nâng cao nhanh hơn so với phần mềm thương mại.

Talend nổi lên như một nhà cung cấp mã nguồn mở hàng đầu về giải pháp ESB và SOA với các dịch vụ cung cấp cho khách hàng:

- Lựa chọn, kiểm tra, chứng nhận: Talend phát triển giải pháp nguồn mở cẩn thận, kiểm tra khả năng tích hợp và chứng nhận các giải pháp mang lại sự an tâm cho khách hàng.

- Đóng gói và cấu hình trước: Talend tích hợp và tăng cường mã nguồn mở cho kỹ thuật ESB, đóng gói, cấu hình trước cho các môi trường phát triển khác nhau nên dễ dàng cài đặt.

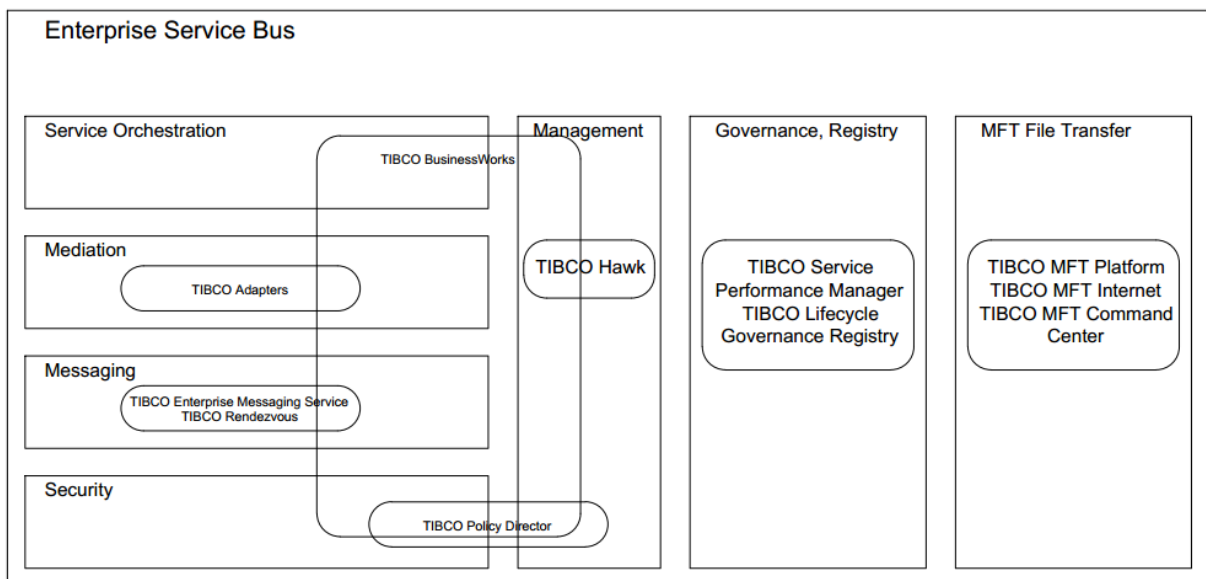
- Đào tạo, hỗ trợ và tư vấn: Talend cung cấp tài liệu và các lớp học trực tuyến để hỗ trợ kỹ thuật.

Giải pháp mã nguồn mở ESB của Talend thích hợp cho các tổ chức vì:

- Kiến trúc phân tán, nhẹ nên dễ dàng tích hợp trên các hệ thống có sẵn;
- Thiết kế module hóa cho phép phát triển theo nhu cầu;
- Cho phép tùy chỉnh các quy tắc định tuyến, tích hợp;
- Hỗ trợ các dịch vụ nhanh chóng, linh hoạt;
- Tuân thủ các chuẩn mở đảm bảo khả năng tương tác;
- Cung cấp nhiều gói giải pháp phù hợp với nhu cầu của tổ chức[10].

### 2.5.3. TIBCO

*Kiến trúc logic : bao gồm các thành phần*



Hình 2.7 Kiến trúc logic của trực tích hợp Tibco ESB

- Service Orchestration: Tầng này có nhiệm vụ phân nhánh các dịch vụ tùy theo nghiệp vụ, các dịch vụ có thể gọi lẫn nhau để xử lý các chức năng khác nhau, cũng như tận dụng những dịch vụ có sẵn. Công cụ thực hiện là TIBCO ActiveMatrix BusinessWorks.

- Mediation: Tầng mediation thực hiện chuyển đổi định dạng thông điệp, hoặc cấu trúc dịch vụ tương ứng với yêu cầu của từng dịch vụ cụ thể. Những hệ

thống cần adapter để kết nối để thì sẽ được sử dụng trong tầng này. Công cụ thực hiện là TIBCO ActiveMatrix BusinessWorks và TIBCO Adapters.

- Messaging: Tầng messaging có nhiệm vụ truyền tải thông điệp giữa các hệ thống. Các thông điệp sau khi được xử lý sẽ được truyền tải sang hệ thống đích thông qua TIBCO Enterprise Messaging Service sử dụng hàng đợi JMS tương ứng.

- Security: Tầng security thực hiện các chính sách bảo mật dịch vụ ở nhiều mức khác nhau (mức truyền tải dữ liệu, mức dịch vụ và mức thông điệp).

- Management: Các dịch vụ sử dụng trong hệ thống được quản lý, theo dõi hoạt động thông qua công cụ TIBCO Hawk.

- Governance, registry: Công cụ giúp quản lý các phiên bản dịch vụ, các địa chỉ dịch vụ được công khai lên hệ thống, thuận lợi cho việc tra cứu, sử dụng lại các dịch vụ đã phát triển.

- MFT File Transfer: Phục vụ hoạt động gửi nhận tệp trong và ngoài hệ thống.

*Các kiểu tích hợp:* Giải pháp hỗ trợ tích hợp:

- Tích hợp đồng bộ và không đồng bộ;
- Publish và Subscribe sử dụng JMS.

*Các chuẩn hỗ trợ:* Giải pháp hỗ trợ các chuẩn:

- Dịch vụ: hỗ trợ WSDL, XML Schema trong việc định nghĩa các Web Service;

- Service registry: hỗ trợ UDDI v3 registry (sử dụng TIBCO Business Works);

- Giao thức truyền tải: hỗ trợ SOAP over HTTP/HTTPS và SOAP over JMS;

- Chất lượng dịch vụ: hỗ trợ WS-Security, WS-Reliable Messaging.

*Ưu điểm:*

- Nhanh hơn, tốn ít chi phí hơn: phát triển các ứng dụng và dịch vụ nhanh hơn, giảm chi phí với yêu cầu tối thiểu về môi trường.

- Tăng truyền thông: dàn xếp kết nối giữa ứng dụng và dịch vụ bởi bộ định tuyến và chuyển đổi định dạng dữ liệu, giao thức truyền khác nhau.

- Độ phức tạp giảm: giảm khả năng tương tác giữa các ứng dụng và công nghệ không đồng nhất, thúc đẩy việc trao đổi thông tin thời gian thực.
- Quy tắc đổi: giảm các tác động của giao diện bằng cách đơn giản hóa xử lý.
- Nâng tầm nhìn: dễ dàng theo dõi các kết nối không thành công và quản lý lỗi.
- Thúc đẩy tái sử dụng[8, 11].

## **2.6. Kết luận**

Chương này giới thiệu sâu hơn về công nghệ ESB trong tích hợp dịch vụ, các thành phần lõi, thành phần logic của ESB, phân loại ESB đồng thời giới thiệu ba giải pháp tích hợp dựa trên ESB là IBM Websphere ESB, Talend ESB và Tibco ESB.

## **CHƯƠNG 3. GIẢI PHÁP TÍCH HỢP MỘT SỐ HỆ THỐNG NGHIỆP VỤ NGÂN HÀNG**

### **3.1. Bài toán tích hợp hệ thống nghiệp vụ ngân hàng**

Ngân hàng Nhà nước Việt Nam (NHNN) là cơ quan ngang bộ của Chính phủ, là Ngân hàng Trung ương của nước Cộng hòa xã hội chủ nghĩa Việt Nam thực hiện chức năng quản lý nhà nước về tiền tệ, hoạt động ngân hàng và ngoại hối, thực hiện chức năng của Ngân hàng Trung ương về phát hành tiền, ngân hàng của các tổ chức tín dụng và cung ứng dịch vụ tiền tệ cho Chính phủ.

Do sự bùng nổ nhanh chóng của ngành Ngân hàng, các sản phẩm và dịch vụ của Ngân hàng ngày càng đa dạng và phức tạp, niềm tin hướng tới một nền kinh tế thị trường, thỏa thuận thương mại song phương của Việt Nam khi gia nhập WTO, các vấn đề này đặt ra yêu cầu về một Ngân hàng Trung ương với nhiều quyền tự chủ hơn trong việc đưa ra chính sách và các quyết định cũng như năng lực mạnh mẽ hơn để thực hiện những hành động kịp thời và hiệu quả. Thực hiện nhiệm vụ này, NHNN phải đối mặt với các vấn đề mà căn nguyên của nó đã tồn tại từ trước và cơ sở hạ tầng quản lý thông tin hiện tại thiếu tập trung, khối lượng thông tin thu thập được nhiều nhưng không thể cung cấp dữ liệu một cách hiệu quả và kịp thời. Việc phát triển các hệ thống Công nghệ thông tin phân tán, rời rạc đã làm khó khăn trở nên trầm trọng.

Chính vì vậy nhu cầu cấp thiết đặt ra là phải tích hợp được các hệ thống Công nghệ thông tin tại NHNN nhằm tạo ra một hệ thống thống nhất có khả năng cung cấp thông tin đầy đủ, chính xác, nhanh chóng phục vụ công tác điều hành nhà nước được kịp thời.

NHNN có nhiều hệ thống nghiệp vụ trong đó có 4 hệ thống cần thiết phải tích hợp được mô tả ở phần sau.

#### **3.1.1. Hệ thống ngân hàng lõi**

Hệ thống ngân hàng lõi tại NHNN đang sử dụng là sản phẩm T24 của Temenos. Hệ thống cung cấp tích hợp liền mạch giữa các đơn vị chức năng trong hoạt động ngân hàng trung ương của NHNN tại Sở giao dịch bao gồm 63 chi nhánh tỉnh, thành phố và các đơn vị tại Hội sở. Hệ thống cung cấp các chức năng:



- Quản lý thông tin khách hàng: Thông tin khách hàng của NHNN được quản lý tập trung. Thông tin khách hàng bao gồm: thông tin về nhân thân/pháp lý (số đăng ký kinh doanh, mã nhóm khách hàng, mã ngành), thông tin về tên và địa chỉ khách hàng, thông tin liên lạc khách hàng (số điện thoại, địa chỉ thư điện tử, số fax, người liên lạc), thông tin tín dụng khách hàng (xếp hạng tín dụng do Stand and Poor, Moody và NHNN xếp hạng), thông tin về tài khoản khách hàng (tổng số dư tài khoản, hạn mức tín dụng, số tiền bị phong tỏa,...), mối quan hệ giữa các khách hàng (hội sở, chi nhánh, phòng giao dịch), chữ ký của khách hàng.

- Quản lý tiền gửi/dự trữ: Quản lý các khoản tiền gửi dự trữ bằng ngoại tệ, tài khoản tiền gửi bằng vàng tại nước ngoài, tài khoản tiền gửi thanh toán của TCTD, tài khoản tiền gửi dự trữ bắt buộc, tài khoản đầu tư chứng khoán và các tài khoản tiền gửi khác. Các chức năng của nghiệp vụ này bao gồm: gửi tiền, rút tiền, thanh toán lãi/phí, chứng nhận gửi/rút tiền.

- Quản lý giao dịch ngoại hối/vàng: quản lý các giao dịch ngoại hối/vàng.

- Quản lý các khoản vay/đi vay: định nghĩa các khoản vay, lãi suất, cách tính lãi suất.

### **3.1.2. Hệ thống sổ sách kế toán và kế toán tài chính**

Giao dịch tài chính trong hệ thống xử lý giao dịch sẽ tạo ra các sự kiện kế toán để bắt đầu hạch toán các bút toán tài chính thông qua giải pháp công cụ kế toán. Công cụ kế toán cho phép khởi tạo các bút toán cho sổ cái phù hợp với chuẩn mực kế toán Việt Nam GAAP và các chuẩn mực kế toán khác được xác định trước như IFRS. Hệ thống đang sử dụng sản phẩm E-Business Suite (EBS) của Oracle.

Hệ thống cung cấp các chức năng:

- Quản lý tài sản cố định: nhằm đảm nhiệm theo dõi tài sản cố định của NHNN cho mục đích dự báo, lập ngân sách tài chính bảo dưỡng và thay thế tài sản, bao gồm các chức năng: định nghĩa tài sản, tính và ghi nhận khấu hao của tài sản, quản lý nhóm tài sản, thực hiện điều chuyển, chia nhỏ hay thanh lý tài sản, quản lý các thông tin về tài chính và bảo hiểm của tài sản.

- Quản lý tài khoản phải thu, phải trả: theo dõi các khoản phải trả cho nhà cung cấp và nhà phân phối và theo dõi các khoản phải thu đến hạn từ khách

hàng. Bao gồm các chức năng: tra cứu tài khoản, ghi nhận hóa đơn, thanh toán và xử lý hóa đơn, xử lý chứng từ.

- Quản lý sổ kế toán: sổ kế toán bao gồm các bút toán kế toán trên góc độ toàn NHNN, bao gồm cả các đơn vị phụ thuộc cho mục đích hợp nhất toàn NHNN. Chức năng này cho phép xác định và thiết lập các thông số Sổ kế toán, xử lý hạch toán, xử lý cuối ngày, báo cáo.

- Quản lý ngân sách nhà nước: chức năng này nhằm mục đích hỗ trợ quy trình lập kế hoạch, lập ngân sách, dự báo, tổng hợp và báo cáo ngân sách. Chức năng này cho phép định nghĩa, cấu hình tài khoản ngân sách, khởi tạo quy trình lập ngân sách, phân tích và kiểm soát ngân sách, đưa ra các cảnh báo khi có phát sinh.

### **3.1.3. Hệ thống thanh toán điện tử liên ngân hàng IBPS (Inter-Bank Payment System)**

Hệ thống thanh toán điện tử liên ngân hàng được triển khai theo mô hình client – server với trung tâm xử lý IBPS, CI-TAD là phần mềm client được cài đặt tại máy trạm của thành viên tham gia trực tiếp vào hệ thống.

Đối tượng tham gia: Sở giao dịch NHNN, NHNN chi nhánh tỉnh, thành phố, các tổ chức tín dụng và công ty cho thuê tài chính. Trong đó Sở giao dịch NHNN và NHNN chi nhánh tỉnh, thành phố đóng vai trò như một tổ chức tín dụng, tham gia trực tiếp vào hệ thống IBPS.

Các chức năng chính:

- Xử lý lệnh thanh toán: Lệnh thanh toán là lệnh chuyển tiền đi hoặc đến từ một đơn vị thành viên trực tiếp của hệ thống IBPS tới một đơn vị thành viên trực tiếp khác của hệ thống IBPS. Các loại lệnh thanh toán bao gồm:

- + Lệnh chuyển nợ giá trị thấp
- + Lệnh chuyển có giá trị thấp
- + Lệnh chuyển nợ giá trị cao
- + Lệnh chuyển có giá trị cao
- + Bù trừ giấy chuyển nợ
- + Bù trừ giấy chuyển có
- + Các lệnh đến tương ứng

- Xử lý giao dịch sai sót: là các giao dịch được tạo ra nhằm xử lý các sai sót trong quá trình xử lý các lệnh thanh toán, bao gồm:
  - + Yêu cầu hủy giao dịch
  - + Hoàn chuyển lệnh thanh toán
  - + Tra soát lệnh thanh toán
- Truy vấn, vấn tin trong ngày: là giao dịch được tạo ra nhằm truy vấn các thông tin về giao dịch hoặc thông tin về tình trạng thanh toán trong ngày, bao gồm:
  - + Truy vấn tình trạng giao dịch
  - + Vấn tin lệnh chuyển tiền đi
  - + Vấn tin khả năng thanh toán
  - + Vấn tin hạn mức
  - + Vấn tin cảnh báo số dư và hạn mức tổng thể

#### **3.1.4. Trung tâm lưu ký chứng khoán CSD (Central Securities Depository)**

Trung tâm lưu ký chứng khoán được thiết kế là hệ thống lưu ký chứng khoán, cung cấp dịch vụ phát hành và lưu ký đối với chứng khoán được sử dụng trong giao dịch với NHNN. Trung tâm lưu ký chứng khoán đồng thời sử dụng để theo dõi và quản lý chứng khoán trong và ngoài nước.

Hệ thống cung cấp các chức năng:

- Định nghĩa và cấu hình các loại chứng khoán như trái phiếu NHNN, trái phiếu Chính phủ, tín phiếu kho bạc.
- Định giá chứng khoán.

### **3.2. Yêu cầu đặt ra**

#### **3.2.1. Yêu cầu của hệ thống**

Các hệ thống trong NHNN đang được xây dựng riêng biệt, với các chức năng riêng, yêu cầu đặt ra là một hệ thống tổng thể hướng tới người dùng gồm 2 đối tượng là người dùng thuộc NHNN và người dùng thuộc các TCTD.

Hệ thống cần đảm bảo một số chức năng cơ bản:

- Kho dữ liệu về khách hàng của NHNN được quản lý tập trung.
- Các thông tin về giao dịch, tài khoản được quản lý tập trung trên sổ cái.

- Có khả năng truy xuất báo cáo kế toán và báo cáo toàn hàng nhanh chóng, chính xác.

Yêu cầu về tích hợp hệ thống:

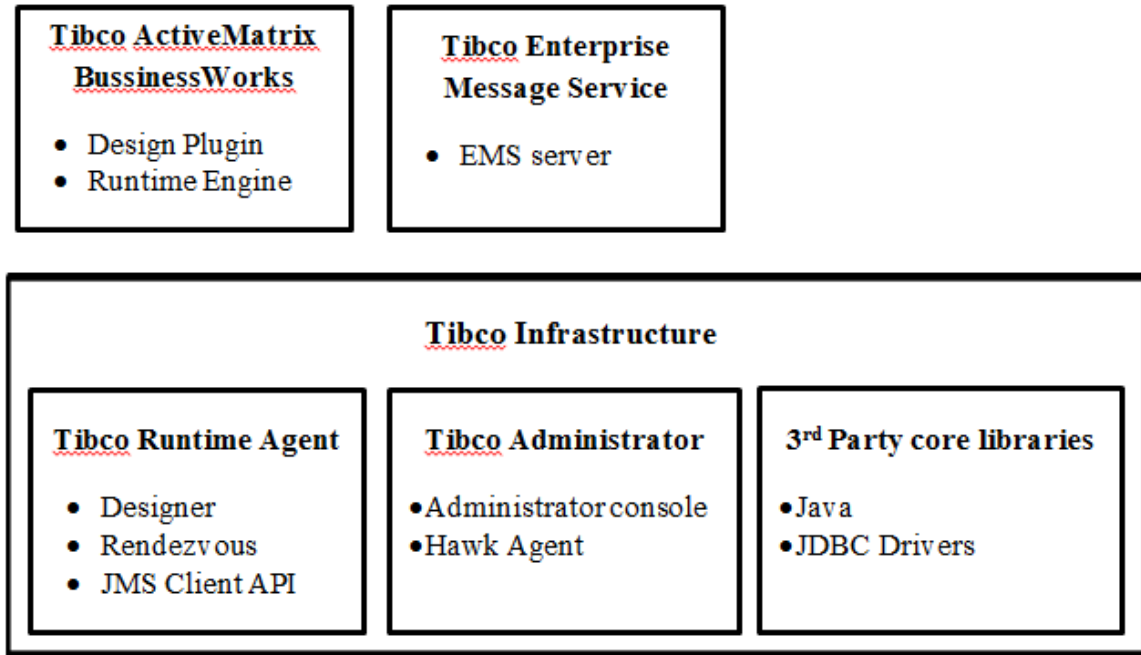
- Mô hình tích hợp cơ bản cho các hệ thống được kế thừa và các hệ thống mới sẽ triển khai

- Theo dõi và kiểm soát đường trao đổi thông tin giữa các ứng dụng

- Tích hợp dịch vụ giữa các hệ thống thuộc NHNN: ngân hàng lõi, hệ thống kế toán, hệ thống thanh toán điện tử liên ngân hàng, hệ thống quản lý trung tâm lưu ký chứng khoán.

### **3.2.2. Môi trường thực nghiệm**

Dựa trên khảo sát, đánh giá 3 giải pháp tích hợp hướng dịch vụ dựa trên ESB, nhóm kỹ thuật NHNN chọn giải pháp tích hợp ESB của Tibco (sản phẩm Tibco Active Matrix Bussiness Work và Tibco Enterprise Message service) để tích hợp dữ liệu giữa các hệ thống ngân hàng lõi, sổ sách kế toán, trung tâm lưu ký và thanh toán liên ngân hàng. Việc giao tiếp giữa các hệ thống thông qua trực dịch vụ tổng thể ESB được thực hiện nhờ các adapter cung cấp bởi TIBCO, cũng như sử dụng các chuẩn Web service.



Hình 3.1 Các phần mềm ứng dụng cài đặt

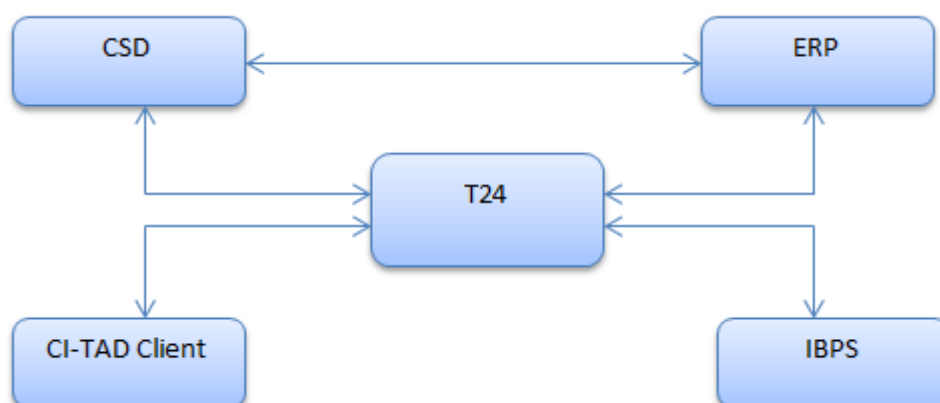
- Tibco Active Matrix Bussiness Works: cung cấp một framework tích hợp hoàn thiện, sử dụng giao diện đồ họa nên dễ dàng sử dụng và nhìn thấy cấu trúc logic. BussinessWorks cũng hỗ trợ những chuẩn mới nhất như J2EE (JMS, EJB, INDI), giao thức (SOAP, WSDL, HTTP, HTTPS), thông điệp (JMS, Tibco Rendezvous), mô tả dữ liệu, hiển thị dữ liệu (XML) và chuyển dữ liệu.

+ Tibco Designer: là một giao diện cho việc thiết kế quy trình và cấu hình adapter, luồng nghiệp vụ và thông điệp. Nó sử dụng định dạng XML để lưu nên có thể cấu hình các giải pháp tích hợp với ít dòng lệnh nhất. Tibco Designer cho phép nhiều lập trình viên làm việc trên cùng một project. Nó sử dụng tệp để chia sẻ hay khóa hoặc quản lý phiên bản.

- Tibco Enterprise Message Service: phần mềm quản lý thông điệp, cung cấp các thành phần của máy chủ JMS và APIs cho JSM client.

### 3.3. Mô hình giải pháp tích hợp

### 3.3.1. Mô hình liên thông giữa các hệ thống nghiệp vụ

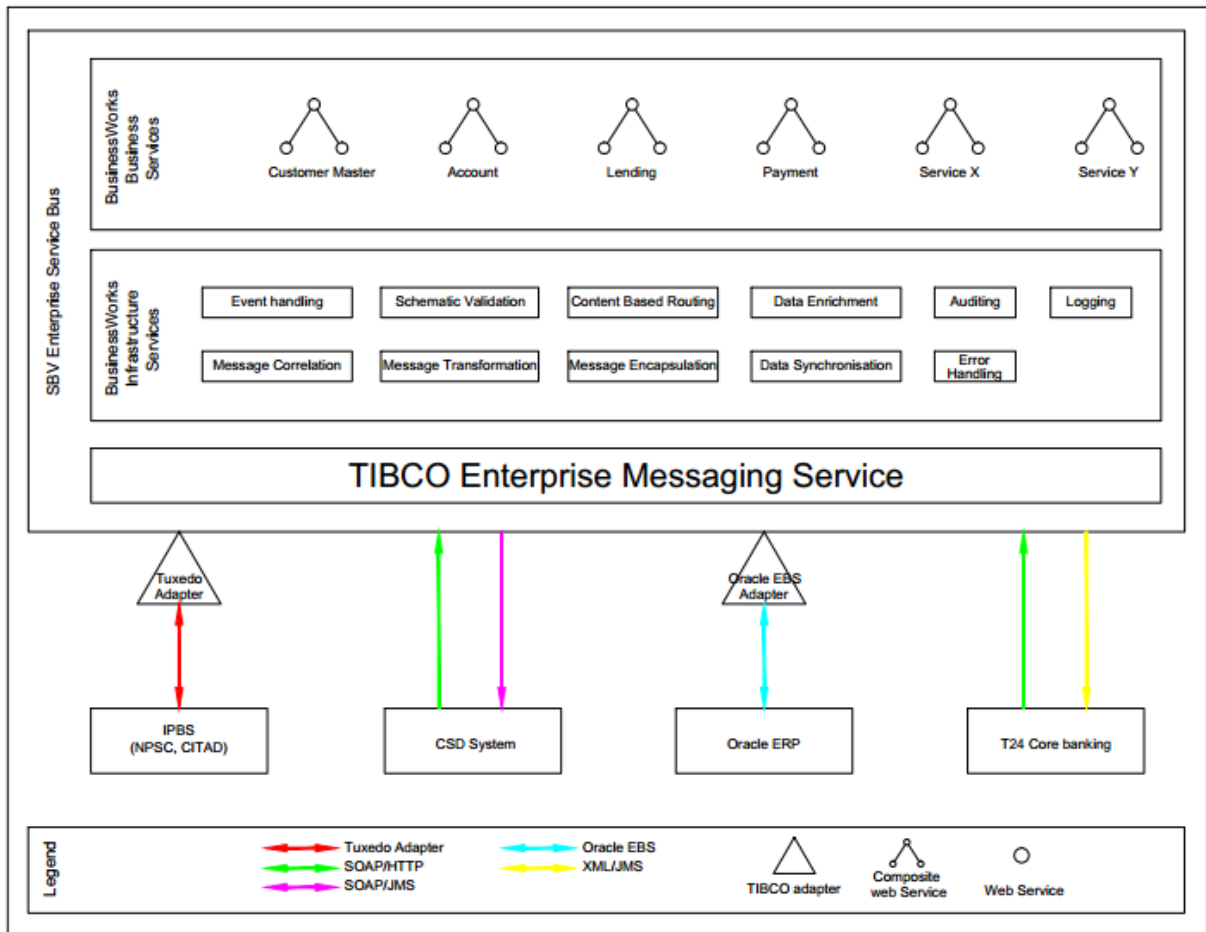


Hình 3.2 Mô hình tương tác giữa các hệ thống nghiệp vụ

Mô tả:

- Thông tin về khách hàng được lưu tập trung trên T24.
  - Các giao dịch thực hiện trên các hệ thống sẽ được tích hợp sang ERP để ghi nhận trong sổ cái.
  - Thông tin từ T24 tích hợp sang CSD: thông tin về những khách hàng tham gia thị trường mở, thông tin về tài khoản khách hàng, hạn mức của khách hàng.
  - Thông tin từ T24 tích hợp sang ERP: thông tin về khách hàng, tài khoản, sổ cái, giao dịch (bao gồm giao dịch chuyển khoản, tiền mặt, ngoại hối...)
  - Thông tin từ T24 tích hợp sang CI-TAD client: giao dịch chuyển tiền, thông tin tài khoản khách hàng, hạn mức tài khoản.
  - Thông tin từ CSD tích hợp sang T24: thông tin về tài khoản khách hàng sau khi thực hiện giao dịch chứng khoán.
  - Thông tin từ ERP tích hợp sang T24: giao dịch bắt nguồn từ phải thu phải trả, sổ cái trên ERP được tích hợp sang T24 để thực hiện thanh toán bằng giao dịch chuyển khoản hoặc giao dịch tiền mặt.
  - Thông tin từ CI-TAD client tích hợp sang T24: giao dịch chuyển tiền, thông tin về ngân hàng, mã ngân hàng, thành viên trực tiếp, gián tiếp tham gia IBPS.
- Các thông tin tích hợp phải được ghi nhận ký, có trạng thái trả lại thành công hay không.

### 3.3.2. Kiến trúc tích hợp



Hình 3.3 Kiến trúc tích hợp

- Hệ thống T24 cung cấp các chức năng sử dụng SOAP thông qua HTTP Web service.
- Hệ thống thanh toán điện tử liên ngân hàng IBPS tích hợp với ESB thông qua Adapter Tuxedo.
- Hệ thống CSD tích hợp với ESB sử dụng SOAP thông qua HTTP Web service.
- Hệ thống kế toán ERP tích hợp với ESB sử dụng Oracle EBS Adapter.

## 3.4. Xây dựng và phát triển hệ thống thử nghiệm

### 3.4.1. Cài đặt hệ thống thử nghiệm

STT	Ứng dụng	Số lượng	Yêu cầu
1	BussinesWork	02	- CPU: 2GHz

			<ul style="list-style-type: none"> <li>- Bộ nhớ: &gt;= 16GB DDR</li> <li>- Ổ cứng: &gt;100GB</li> <li>- Hệ điều hành: Linux RHEL 6.4</li> </ul>
2	EMS	02	<ul style="list-style-type: none"> <li>- CPU: 2GHz</li> <li>- Bộ nhớ: &gt;= 16GB DDR</li> <li>- Ổ cứng: &gt; 100GB</li> <li>- Phân vùng share: 150GB</li> <li>- Hệ điều hành: AIX 7.1</li> </ul>
3	Hawk	01	<ul style="list-style-type: none"> <li>- CPU: 2GHz</li> <li>- Bộ nhớ: 8GB DDR</li> <li>- Ổ cứng: &gt; 100GB</li> <li>- Hệ điều hành: Microsoft Windows server 2012</li> </ul>
4	Database	02	<ul style="list-style-type: none"> <li>- CPU: 2GHz</li> <li>- Bộ nhớ: 8GB DDR</li> <li>- Ổ cứng: &gt;= 200GB</li> <li>- Phần mềm: Oracle rack 11g</li> <li>- Hệ điều hành: Linux RHEL 6.4</li> </ul>

Bảng 3.1 Danh sách máy chủ cài đặt hệ thống thực nghiệm

### 3.4.2. Phương thức quản lý người dùng trên các hệ thống

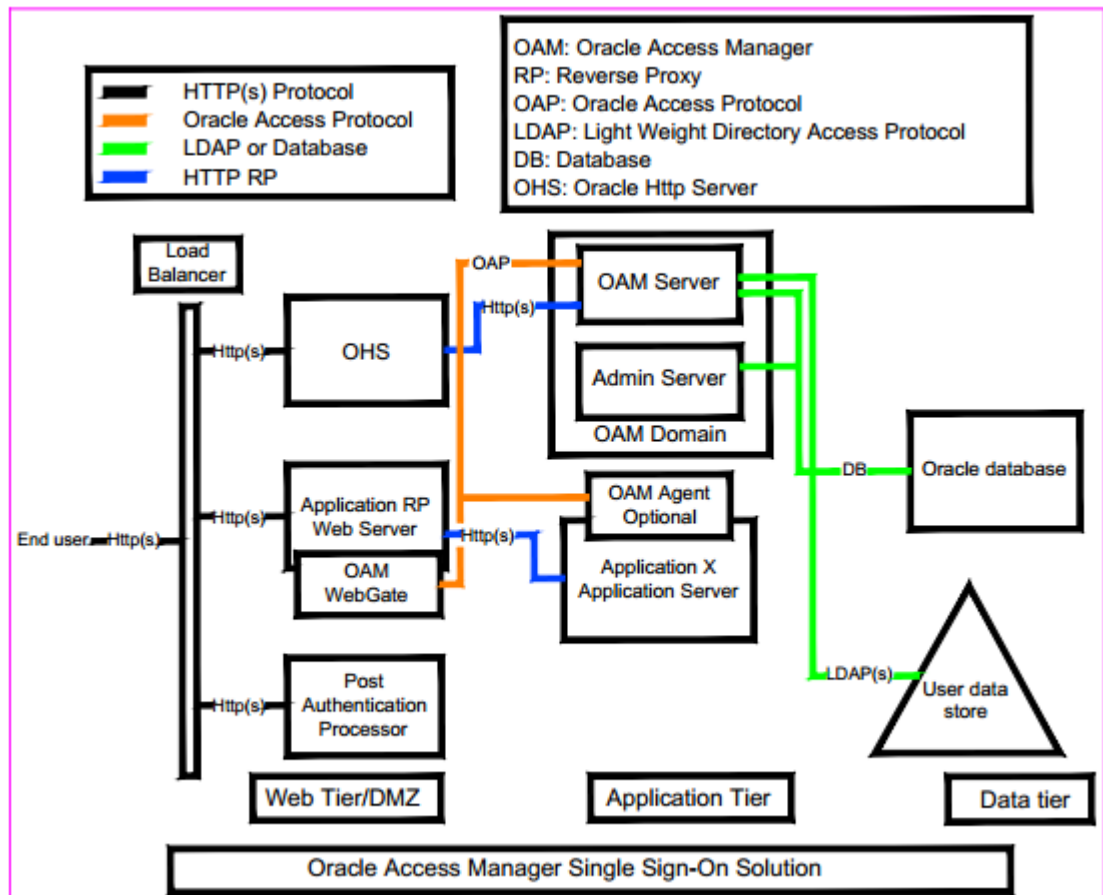
NHNN có nhiều hệ thống nghiệp vụ khác nhau, mỗi hệ thống có chức năng quản lý người dùng riêng nên khi truy cập vào các hệ thống này người dùng cần có tài khoản riêng. Việc đó gây ra một số các khó khăn nhất định như:

- Đối với người quản trị: phải tạo nhiều tài khoản cho người dùng trên từng ứng dụng dịch vụ riêng dẫn đến việc khó quản lý, mất thời gian.
- Đối với người dùng: phải ghi nhớ nhiều tài khoản, mật khẩu trên các hệ thống khác nhau.

Giải pháp đưa ra để giải quyết bài toán này hệ thống quản lý người sử dụng bao gồm bộ sản phẩm Oracle Access Manager Suite (OAM) cung cấp các dịch vụ quản lý đăng nhập một lần SSO (Singler Sign On).

*OAM bao gồm những thành phần chính sau:*

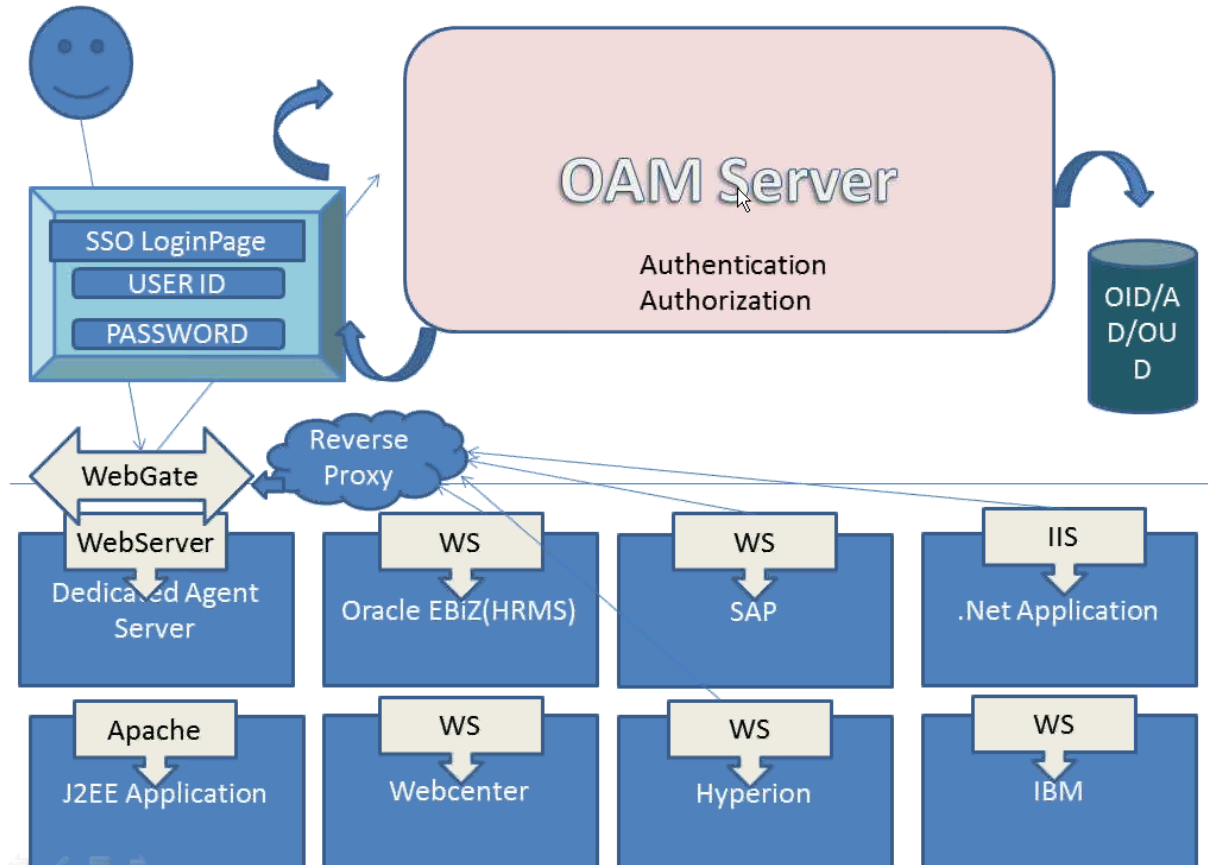




Hình 3.4 Mô hình hệ thống SSO

- Access Server (OAM Server): Là máy chủ đóng vai trò như một cổng truy cập, cung cấp khả năng xác thực và dịch vụ audit.
- Webgate: Là một plugin biên dịch các yêu cầu truy cập vào các tài nguyên Web (HTTP) và chuyển tiếp chúng tới cho Access Server để xác thực.
- Identity Assertion Provider (Application Agent): Là nơi cung cấp định danh của người dùng dựa trên các thông tin header được thiết lập bởi các bên xác thực, được cài đặt trên máy chủ ứng dụng cần SSO. Để thực hiện SSO, OAM phải được cấu hình với ít nhất một IAP.

*Tích hợp OAM với các ứng dụng:*



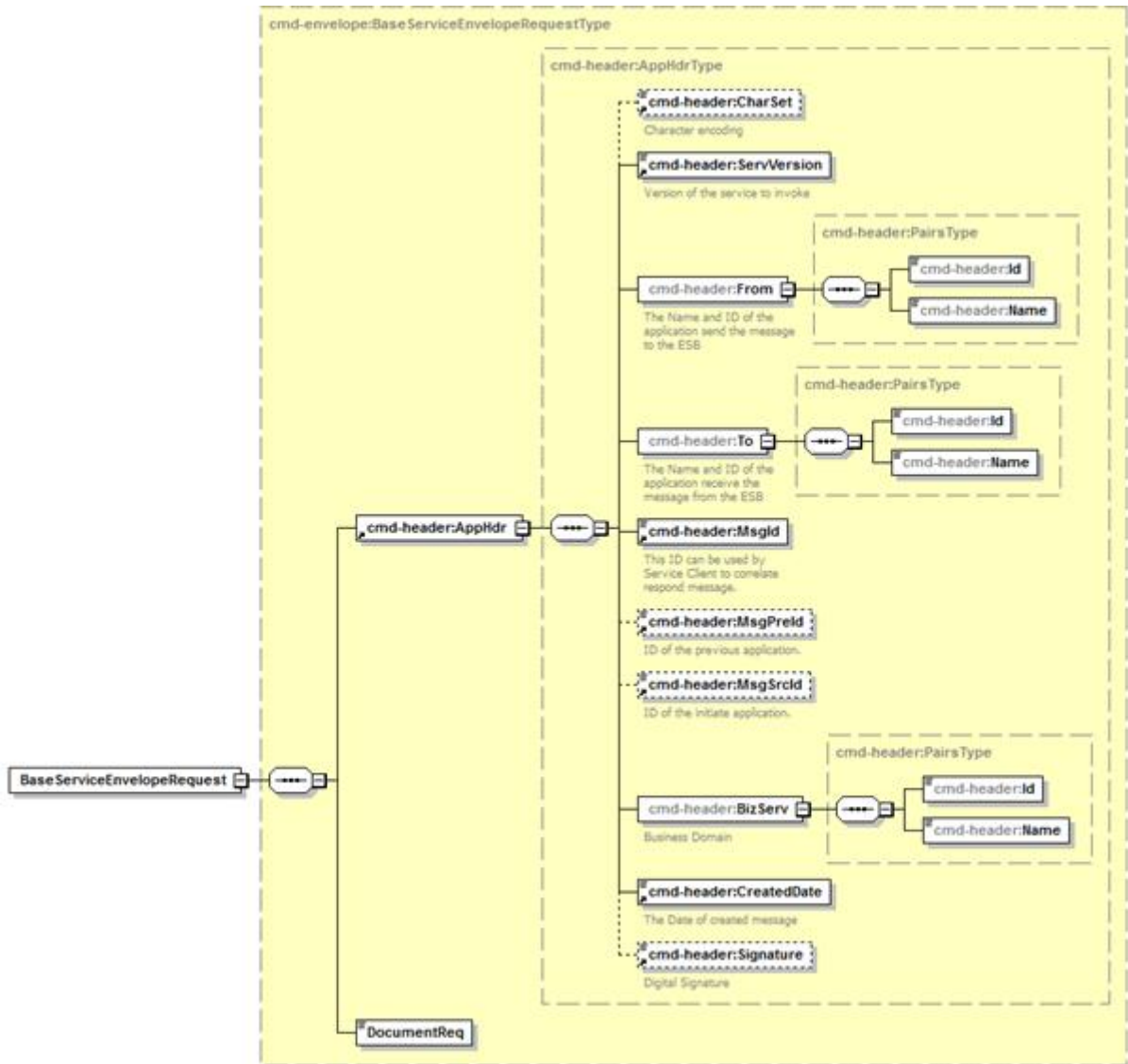
Hình 3.5 Mô hình tích hợp OAM với các ứng dụng

Khi người dùng truy cập vào một ứng dụng ví dụ t24.sbv.gov.vn, webgate tại tầng http của ứng dụng đó sẽ chuyển link đến địa chỉ đăng nhập của SSO ví dụ là sso.sbv.gov.vn là địa chỉ cân bằng tải của tầng http AOM. Yêu cầu được gửi xuống http AOM, chuyển tiếp xuống AOM server, các thông tin về tên đăng nhập, mật khẩu, cookies được lưu trong cơ sở dữ liệu OAM. AOM server hỏi xuống Active Directory để xác thực người dùng.

### 3.4.3. Tích hợp dịch vụ qua Tibco ESB sử dụng WebService

Các ứng dụng tích hợp qua Tibco ESB sử dụng WebService có cấu trúc thông điệp, cách đặt tên dịch vụ, mã lỗi tuân theo chuẩn quy định sau:

*Cấu trúc thông điệp gửi đi*



Hình 3.6 Cấu trúc thông điệp gửi đi

Tên thẻ	Mô tả	Kiểu dữ liệu
AppHdr	Chứa thông tin chung của dịch vụ phục vụ nhận biết nghiệp vụ, nguồn, đích của dịch vụ	AppHdrType
DocumentReq	Nội dung thông điệp chứa các thông tin cần trao đổi (theo cấu trúc XML)	#Any

Bảng 3.2 Mô tả thẻ trong cấu trúc thông điệp gửi đi

Cấu trúc AppHdr:

Tên thẻ	Mô tả	Kiểu dữ	Độ dài	Giá trị
---------	-------	---------	--------	---------

XML		liệu	tổ đa	
Charset	Bao gồm đặc tả kí tự sử dụng trong các thành phần text-based ở nội dung thông điệp	String	20	Basic_Latin
ServVersion	Phiên bản dịch vụ mong muốn sử dụng	String	5	
From	Tên của hệ thống gửi thông điệp đến trực ESB  Đối với dịch vụ nội bộ, mã hệ thống và tên hệ thống nằm trong danh sách qui định  Đối với dịch vụ cung cấp cho các TCTD, mã hệ thống và tên hệ thống theo qui định chung với các TCTD	String		
Id	Mã hệ thống	String	10	T24/ERP/AO M,...
Name	Tên hệ thống	String	50	
To	Tên của hệ thống nhận thông điệp  Đối với dịch vụ nội bộ, mã hệ thống và tên hệ thống nằm	String		

	<p>trong danh sách qui định.</p> <p>Đối với dịch vụ cung cấp cho các TCTD, mã hệ thống và tên hệ thống theo qui định chung với các TCTD</p>			
Id	Mã hệ thống	String	10	T24/ERP/AOM,...
Name	Tên hệ thống	String	50	
MsgId	<p>Định danh thông điệp do hệ thống từng hệ thống sinh ra khi gọi dịch vụ từ ESB</p> <p>Qui tắc sinh: MÃ HỆ THỐNG + YYMMDD + Seq (6 character)</p>	String	30	AOM201509021245
MsgPreId	Định danh thông điệp do hệ thống trước gửi đến	String	30	
MsgSrcId	Định danh thông điệp do hệ thống ban đầu sinh ra	String	30	
BizServ	Tên chức năng nghiệp vụ của dịch vụ	String		
Id	Mã nghiệp vụ	String	10	

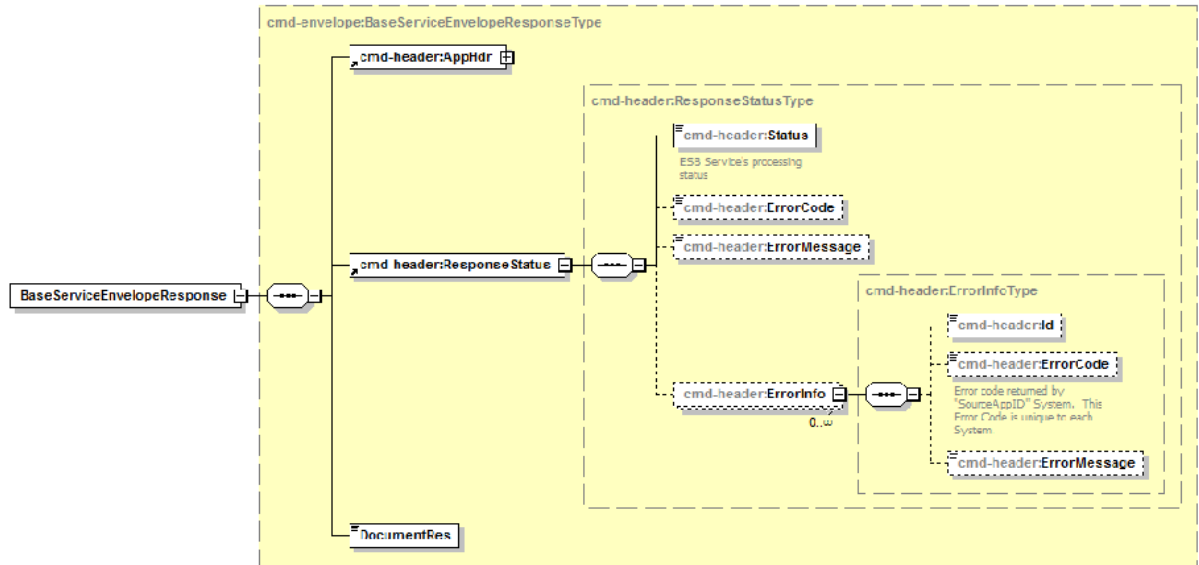
Name	Tên nghiệp vụ	String	50	Business service name. Eg: Loan, FXRate,..etc
CreatedDate	Ngày/giờ Bussiness Message Header được tạo	DateTime (yyyy-MM-dd HH:mm:ss)	30	2015-06-30 12:10:20
Signature	Bao gồm chữ ký số (nếu cần) của hệ thống gửi thông điệp	String	Độ dài qui định theo qui ước kí 2048 bit giữa hai hệ thống	

Bảng 3.3 Cấu trúc AppHdr

Quy tắc điền Id của các trường MsgID, MsgPreId, MsgSrcId như sau:

- MsgId là định danh thông điệp được sinh ra tại hệ thống đang xử lý message
- MsgPreId là định danh thông điệp của hệ thống trước đó gửi đến.
- MsgSrcId là định danh thông điệp của hệ thống khởi tạo tin điện.

***Cấu trúc thông điệp nhận về***



Hình 3.7 Cấu trúc thông điệp nhận về

Tên thẻ XML	Mô tả	Kiểu dữ liệu	Độ dài tối đa
AppHdr	Các trường trong thẻ AppHdr giống như thẻ AppHdr của thông điệp gửi đi		
Document	Nội dung thông điệp chứa các thông tin cần trao đổi (theo cấu trúc XML)	#Any	
ResponseStatus		Type	
➤ Status	Trạng thái thực hiện của hệ thống: 1= có lỗi, 0: thành công	String	1
➤ ErrorCode	Mã lỗi trả về của hệ thống xử lý	String	20
➤ ErrorMessage	Mô tả lỗi	String	200
➤ ErrorInfo	Bao gồm các thông tin lỗi trả về từ hệ thống đích	Type	
+ Id	Mã hệ thống gây lỗi	String	10
+ ErrorCode	Mã lỗi trả về từ hệ thống gây lỗi	String	20

+ ErrorMessage	Mô tả lỗi trả về	String	200
----------------	------------------	--------	-----

Bảng 3.4 Mô tả cấu trúc thông điệp nhận về

**Cách đặt tên dịch vụ**

&lt;Tên hệ thống&gt;\_&lt;Tên Service&gt;\_&lt;Số version&gt;

Trong đó :

- Tên hệ thống : Tên duy nhất cho từng hệ thống, tối đa 4 ký tự
- Tên Service : Tên nghiệp vụ liên quan : MM, FT, FX ....
- Số version : Số version theo thời gian release : tối đa 3 số

VD : T24\_CUSTOMER\_001

**Cấu trúc mã lỗi**

ERR.&lt;Tên hệ thống&gt;.&lt;Mã lỗi&gt;

Trong đó :

- Tên hệ thống : Tên duy nhất cho từng hệ thống, tối đa 4 ký tự.
- Mã lỗi : Gồm 3 số từ 000 – 999 đặt theo lỗi cụ thể

VD : ERR.T24.001

**3.4.4. Tích hợp dịch vụ qua Tibco ESB sử dụng Adapter**

Các ứng dụng tích hợp qua ESB sử dụng Adapter được hỗ trợ sẵn cần cấu hình tham số kết nối và luồng dữ liệu. Tùy từng yêu cầu nghiệp vụ khác nhau mà có các luồng nghiệp vụ khác nhau tương ứng. Sau đây là ví dụ về adapter tuxedo được sử dụng để tích hợp hệ thống thanh toán liên ngân hàng và luồng nghiệp vụ hệ thống thanh toán liên ngân hàng yêu cầu sang hệ thống T24 để lấy số dư tài khoản.

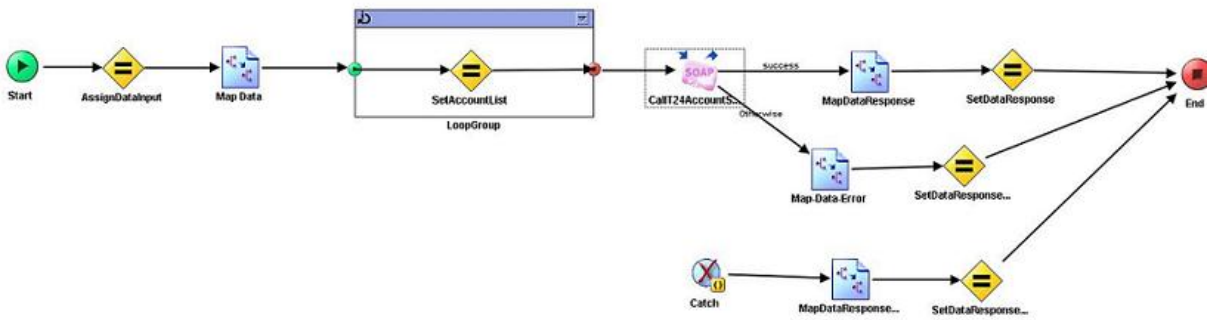
Cấu hình Adapter Tuxedo	Ý nghĩa các trường
	<ul style="list-style-type: none"> <li>- ClientName: tên ứng dụng Tuxedo cần tích hợp</li> <li>- Flag: cơ chế thông báo và truy cập client</li> <li>- GroupName: Nếu adapter là workstation client thì giá trị GroupName là Null, nếu là native client thì giá trị GroupName được xác định</li> </ul>



<pre> ABC adtuxedo.connection.clientName = "E ABC adtuxedo.connection.flags = "" ABC adtuxedo.connection.groupName = "" ABC adtuxedo.connection.password = "" ABC adtuxedo.connection.userName = "1; ABC adtuxedo.connection.userPassword ABC numOfMaxPendingInvcns = "50" ABC pollIntervalForOutStandingInvcn = "20 ABC pollIntervalForReplies = "1000" ABC pollingInterval = "3000" </pre>	<ul style="list-style-type: none"> <li>- Password: Mật khẩu tương ứng với tên đăng nhập truy cập ứng dụng Tuxedo</li> <li>- UserName: tên đăng nhập truy cập vào ứng dụng Tuxedo</li> <li>- NumOfMaxPendingInvcn: số lượng lời gọi tối đa đang chờ, giá trị mặc định là 50.</li> <li>- PollIntervalForOutStandingInvcn: xác định mức độ thường xuyên của dịch vụ yêu cầu – phản hồi của Tuxedo ở chế độ không đồng bộ; giá trị mặc định là 2000 mili giây.</li> <li>- PollIntervalForReplies: xác định mức độ thường xuyên của dịch vụ yêu cầu – phản hồi từ trả lời của Tuxedo trong chế độ không đồng bộ; giá trị mặc định là 1000 mili giây.</li> <li>- PollInterval: xác định mức độ thường xuyên của kết nối giữa adapter và máy chủ Tuxedo; giá trị mặc định là 3000 mili giây.</li> </ul>
--	--

Bảng 3.5 Cấu hình tham số Adapter Tuxedo

Luồng nghiệp vụ hệ thống thanh toán liên ngân hàng yêu cầu số dư tài khoản từ T24

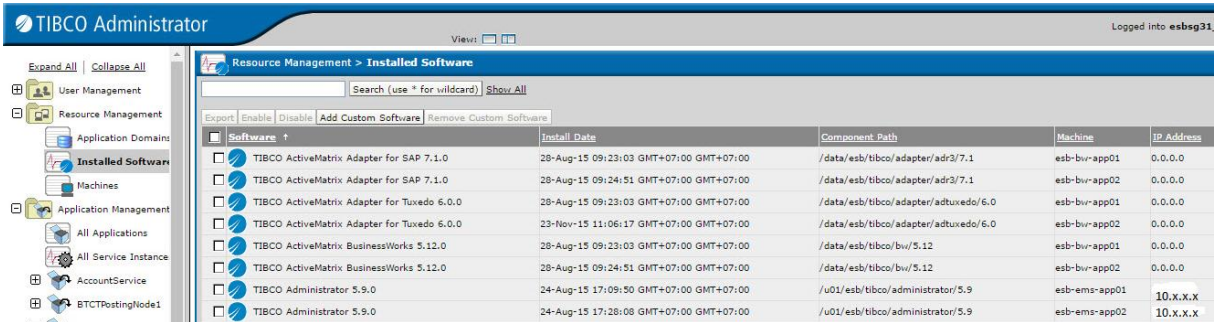


Hình 3.8 Mô hình luồng nghiệp vụ hệ thống thanh toán liên hàng yêu cầu số dư tài khoản từ T24

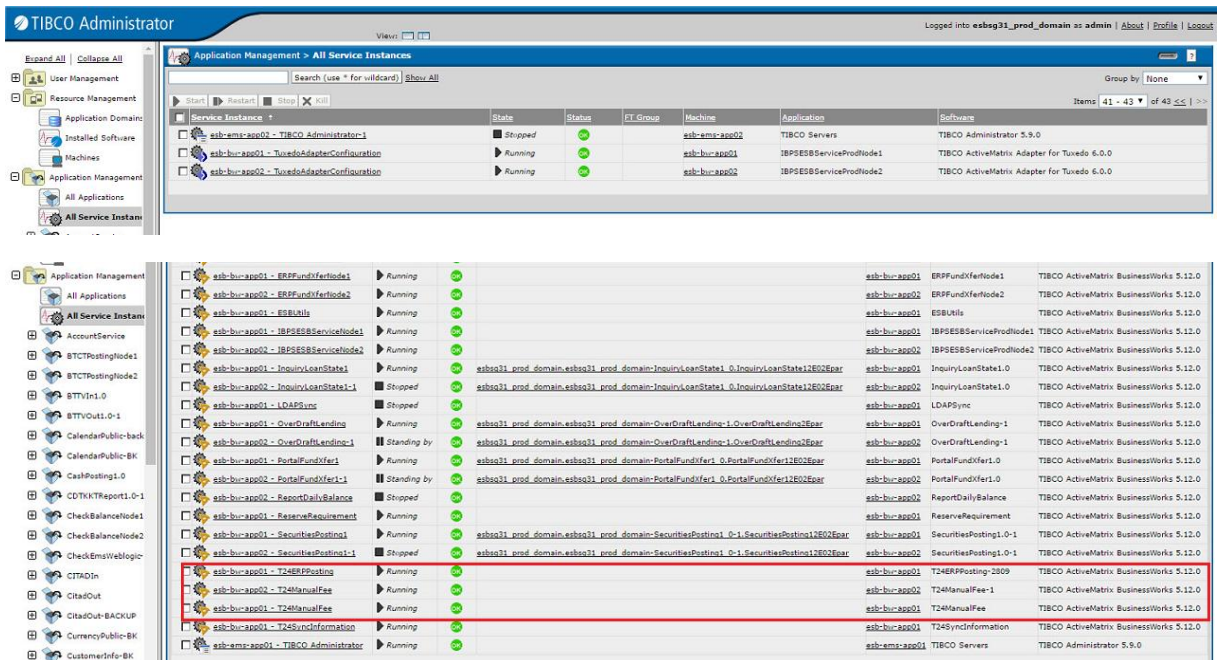
### 3.4.5. Giao diện quản trị của Tibco



Hình 3.9 Các máy chủ ứng dụng



Hình 3.10 Các phần mềm cài đặt



Hình 3.11 Các dịch vụ cài đặt

### 3.5. Kết quả thử nghiệm và đánh giá

Sau khi tiến hành cài đặt và phát triển, toàn bộ bốn hệ thống nghiệp vụ đã được tích hợp liên thông.

Để đánh giá thử nghiệm hệ thống tích hợp, chúng tôi đã xây dựng một số kịch bản kiểm thử và tiến hành thử nghiệm. Các kết quả thử nghiệm được trình bày cụ thể sau đây:





#### 3.5.1. Giao dịch chuyển tiền từ T24 sang CITAD:

- Tiền chuyển từ tài khoản của ngân hàng thương mại cổ phần Bưu điện Liên Việt chi nhánh Hòa Bình mở tài khoản tại NHNN chi nhánh Hòa Bình 5.000.000 VNĐ đến trung tâm thanh toán của Ngân hàng thương mại cổ phần Bưu điện Liên Việt qua kênh thanh toán điện tử liên ngân hàng.

- Giao dịch được lập trên T24, sau khi được phê duyệt sẽ được chuyển sang hệ thống IBPS bao gồm các thông tin về giao dịch: số bút toán tương ứng trên T24, ngân hàng gửi, ngân hàng nhận, tài khoản nợ, tài khoản có, số tiền gửi, ghi chú.

- Người dùng tiến hành phê duyệt trên màn hình CITAD client để chuyển giao dịch tới ngân hàng nhận.

- Sau khi xử lý cuối ngày trên hệ thống T24, thông tin về giao dịch và số dư tài khoản sẽ được chuyển sang hệ thống ERP qua ESB để lên báo cáo liệt kê giao dịch tại đơn vị.

Truy vấn FT		FT/16301/6T7CM	
Truy vấn FT		Kiểm soát	
Loại giao dịch	BC	LCC CITAD ĐI	
Loại lệnh	201001		
Tài khoản ghi nợ	 128368		TG NH
Loại tiền ghi nợ	 VND	VIETNAM DONG	
Số tiền ghi nợ	5,000,000,000		
Ngày giá trị	27 OCT 2016	27 OCT 2016	
Tài khoản ghi có	 105686		TTLNH
Loại tiền ghi có	 VND	VIETNAM DONG	
Ngày ghi có	27 OCT 2016	27 OCT 2016	
Mã NH gửi	17101001	NHNN, CN tỉnh Hoa Binh	
Mã NH nhận	01357001		
Đơn vị chuyển.1	NH TMCP BUU DIEN LIEN VIET - CN HOA		
Đơn vị chuyển.2	BINH		
Tài khoản đơn vị chuyển	128368		
Mã NH gửi gián tiếp	17101001	NHNN, CN tỉnh Hoa Binh	
Tài khoản đơn vị nhận	120111		
Đơn vị nhận.1	NH TMCP BUU DIEN LIEN VIET		
Mã NH nhận gián tiếp	01357001	LIENVIETPOSTBANK, CN Ha Noi	
Mã NH chịu phí	17357001		
Điểm giải	LPB HOA BINH DIEU CHUYEN VON VE HSC		
Số bút toán	90001231		

Hình 3.12 Màn hình giao dịch chuyển nợ trên T24

27/10/2016 **NGAN HANG NHA NUOC VIET NAM** [Thoát](#)

Quản trị hệ thống ▶ Giao dịch đi ▶ Giao dịch đến ▶ Báo cáo đối chiếu ▶ Công việc khác ▶ CI-Gateway ▶ Tra cứu ▶ [EN](#)

**Nội dung lệnh chuyển tiền**

Số bút toán: FT163016T7CM      Loại giao dịch: HF      Lệnh chuyển có giá trị cao

Ngân hàng gửi: 17101001      NHNN CN tỉnh Hoà Bình      Ngày lệnh: 27/10/2016

Ngân hàng nhận: 01357001      NHTMCP Bưu điện Liên Việt CN Ha Noi      TELLER ID: BTH1701

Số tiền chuyển: 5.000.000.000      VND      Số giao dịch: 90001231

Thời điểm gửi lệnh: 27/10/2016 16:28:07      Tình trạng giao dịch: **6000: Thành công**

[Trước](#)      [Tiếp theo](#)      [Đóng](#)      **Tổng số 1/1**

**Thông tin người gửi**

Tên: NH TMCP BUU DIEN LIEN VIET - CN HOA BINH

Địa chỉ: \_\_\_\_\_

Số tài khoản: 128368      Mã KH: \_\_\_\_\_

Tại: 17101001      NHNN CN tỉnh Hoà Bình

**Thông tin người nhận**

Tên: NH TMCP BUU DIEN LIEN VIET

Địa chỉ: \_\_\_\_\_

Số tài khoản: 120111      Mã KH: \_\_\_\_\_

Tại: 01357001      NHTMCP Bưu điện Liên Việt CN Ha Noi      Mã số thuế: \_\_\_\_\_

Chương - Khoản - Tiêu mục: \_\_\_\_\_

Nội dung: LPB HOA BINH DIEU CHUYEN VON VE HSC

Loại nghiệp vụ 1: 30      Chuyển có      Loại nghiệp vụ 2: 101      CTCN

Thông tin tính phí: 1      Ngân hàng chịu phí: 17101001      NHNN CN tỉnh Hoà Bình

Hình 3.13 Màn hình giao dịch tương ứng chuyển sang CITAD

32	10/27/2016	FT163015SY3 FVND	10/27/2016	FT163015SY3 FVND	Giao dịch hệ thống T24 số: FT163015SY3 Ngày: 27/10/2016 ; Số ID FAH:15118167	VND	60200400 - Thanh toán liên chi nhánh	Có	0	1,873,000.00	0	1,873,000	Tích hợp T24	Đã duyệt
33	10/27/2016	FT163016T7 CMVND	10/27/2016	FT163016T7 CMVND	Giao dịch hệ thống T24 số: FT163016T7 Ngày: 27/10/2016 ; Số ID FAH:15102467	VND	40400100 - Tiền gửi không kỳ hạn	Nợ	5,000,000,000.00	0	5,000,000,000	0	Tích hợp T24	Đã duyệt
34	10/27/2016	FT163016T7 CMVND	10/27/2016	FT163016T7 CMVND	Giao dịch hệ thống T24 số: FT163016T7 Ngày: 27/10/2016 ; Số ID FAH:15102467	VND	60200400 - Thanh toán liên chi nhánh	Có	0	5,000,000,000.00	0	5,000,000,000	Tích hợp T24	Đã duyệt
35	10/27/2016	FT16301H58 1WVND	10/27/2016	FT16301H58 1WVND	Giao dịch hệ thống T24 số: FT16301H58 Ngày: 27/10/2016 ;	VND	60100200 - Thanh toán bù trừ của	Nợ	40,000,000.00	0	40,000,000	0	Tích hợp T24	Đã duyệt

Hình 3.14 Màn hình báo cáo liệt kê giao dịch in trên hệ thống ERP

Đánh giá kết quả đạt được:

- Kết quả đáp ứng được yêu cầu của bài toán đặt ra: thực hiện việc tích hợp giữa các hệ thống nghiệp vụ, trong suốt với người dùng, giúp người dùng không phải nhập dữ liệu nhiều lần trên các hệ thống khác nhau đồng thời giúp giảm thiểu việc sai sót thông tin.
- Cho phép truy xuất báo cáo nhanh, đầy đủ, chính xác từ nhiều hệ thống.

### **3.5.2. Giao dịch chi tạm ứng thực hiện trên phân hệ phải thu phải trả của ERP, tích hợp sang hệ thống T24 để chi tiền mặt cho nhân viên**

- Giao dịch chi tạm ứng cho nhân viên được thực hiện tại NHNN chi nhánh Hòa Bình, thực hiện chi tiền mặt cho nhân viên với số tiền là 25.000.000 đồng
- Sau khi thực hiện phê duyệt hóa đơn, thanh toán hóa đơn trên phân hệ phải thu phải trả của hệ thống ERP, giao dịch được tích hợp sang hệ thống T24. thực hiện chi trả tiền mặt cho nhân viên.
- Người dùng tiến hành phê duyệt giao dịch trên T24 để thực hiện chi tiền mặt cho nhân viên.

PHUONGMT

Đơn vị Hoạt động	Loại YCTT	Đối tác Thu	Số Nhà cung c	Tên địa điểm	Ngày lập YC	Số YCTT	Loại tiền	Số tiền phải trả	TK Phải trả
NHNN Chi nhánh Hồ	Prepayment	Bạch Hồng	113427271	31500100.03	01-11-2016	CHI.BUIPHUC	VND	25.000.000	03000.00.69900101.000.00000

1 Tổng hợp    2 Dòng    3 Treo    4 Xem các khoản thanh toán    5 Khoản thanh toán theo Lịch trình    6 Xem ứng dụ

Tổng hợp

Mặt hàng	25.000.000
Khoản giữ lại	
Kh trả trước đã áp	
Khấu lưu	
Tổng số	25.000.000
Thuế	
Cước phí	
Các khoản khác	
Tổng	25.000.000

Số tiền đã Thanh toán  
VND 25.000.000

In chứng từ

Trạng thái: Khoản trả trước hiện có  
Đã hạch toán: Đã phê duyệt

Treo: 0  
Treo t. toán theo lịch trình: 0

Mô tả: TAM ỨNG CHI TIÊU HCQT (THÚY)

Hình 3.15 Hóa đơn trên phân hệ phải thu phải trả ERP

PHUONGMT

cấp	Địa điểm Nhà cung cấp	Ngày Thanh toán	Tài khoản Ngân h	Số tiền Thanh toán	Phương thức Thanh toán	[ ]	Số Chứng từ	Hồ sơ Quá trình Thanh toán
	31500100.03000	01-11-2016	QuyNVPH-T24	25.000.000	Check		Thờ: 524	US NACHA IAT

Số hiệu: CHILBUIPHUONG\_01/11/2016\_1    Ngày: 01-11-2016

Thu/Chi tiền T24    Liên ch    Hành động... 1

Payment Information

Thông tin thêm: Thông tin thêm

Số chứng từ: 524

Ngày chứng từ: 01-11-2016

Tên người thu hưởng: BẠCH HỒNG THÚY

Địa chỉ người thu hưởng: P.HCNS - NHNN HB

Số tài khoản người thu hưởng:

Ngân hàng người thu hưởng:

Ngân hàng nhận lệnh:

Ngân hàng gửi lệnh:

Ngân hàng phục vụ người gửi:

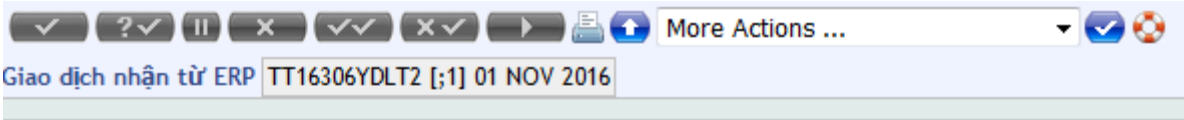
Loại nghiệp vụ: TM0001    Chi tiền từ Quỹ NVPH - tiền đủ TCLT

Số giao dịch T24:

OK    Cancel    Clear    Help

Hình 3.16 Màn hình thực hiện thanh toán hóa đơn và chuyển giao dịch sang T24 để chi tiền mặt





Giáo dịch nhận từ ERP TT16306YDLT2 [;1] 01 NOV 2016	
Loại giao dịch	14
Loại tiền tệ 1	VND
Tài khoản 1.1	VND-10001-0001-0030 TIỀN ĐỦ TCLT, QUỸ NVPH (H
Số tiền.1	25,000,000
Ngày giá trị 1	01 NOV 2016
Loại tiền tệ 2	VND
Tài khoản 2	VND-14810-0001-0030 TK TIỀN MẶT T24-ERP CN HÒ
Ngày giá trị 2	01 NOV 2016
Mã giao dịch ERP	ERP505438_524
Hệ thống chuyển giao dịch	ERP
Phân hệ trên ERP	Payment
Nội dung giao dịch	TẠM ỨNG CHI TIÊU HCQT (THÚY)
Người nhận	BACH HỒNG THÚY
Địa chỉ người nhận	P.HCNS - NHNN HB

Hình 3.17 Màn hình giao dịch tiền mặt tương ứng nhận từ ERP

### 3.6. Kết luận

Chương 3 trình bày bài toán tích hợp các nghiệp vụ ngân hàng tại NHNN và đề xuất giải pháp tích hợp sử dụng trực dịch vụ tổng thể của TIBCO để giải quyết bài toán. Bài toán đã được thực nghiệm và cho kết quả tại NHNN với các hệ thống được tích hợp: hệ thống ngân hàng lõi, hệ thống sổ sách kế toán, hệ thống thanh toán điện tử liên ngân hàng, hệ thống trung tâm lưu ký chứng khoán.

## **CHƯƠNG 4. KẾT LUẬN CHUNG**

### **4.1. Các kết quả đạt được trong luận văn**

Trong chương một, luận văn đã trình bày tổng quan về tích hợp hệ thống, mô hình kiến trúc trong tích hợp hệ thống và một số phương pháp tích hợp hệ thống trong đó chú trọng vào phương pháp tích hợp mức dịch vụ.

Chương hai, luận văn trình bày chi tiết hơn về tích hợp mức dịch vụ sử dụng trực dịch vụ tổng thể ESB, về chức năng, các thành phần logic của ESB đồng thời giới thiệu một số nền tảng ứng dụng ESB như IBM Websphere ESB, Talend ESB và Tibco ESB.

Chương ba trình bày về bài toán đặt ra về tích hợp hệ thống nghiệp vụ tại NHNN, đề xuất giải pháp tích hợp sử dụng sản phẩm Tibco ESB để thực hiện tích hợp bốn hệ thống nghiệp vụ tại Ngân hàng Nhà nước. Các hệ thống nghiệp vụ này đã được tích hợp dựa theo cả công nghệ WebService lẫn trực dịch vụ tổng thể ESB. Giải pháp tích hợp đã được xây dựng và tiến hành thử nghiệm tại Cục Công nghệ tin học, Ngân hàng nhà nước và bước đầu đã hỗ trợ và minh chứng được tính hiệu quả trong liên thông các dịch vụ nghiệp vụ tại Ngân hàng Nhà nước.

### **4.2. Định hướng phát triển trong tương lai**

Sử dụng giải pháp trực dịch vụ tổng thể ESB của TIBCO để tiếp tục tích hợp các hệ thống nghiệp vụ hiện tại khác của NHNN và các hệ thống trong tương lai như:

- Hệ thống mã ngân hàng: thực hiện cấp phát, hủy mã ngân hàng theo quy chuẩn cho các TCTD, chi nhánh, phòng giao dịch, quỹ tín dụng nhân dân mới thành lập.
- Hệ thống báo cáo: lưu trữ, tổng hợp các báo cáo của các TCTD theo các chuẩn, biểu mẫu, tiêu chí khác nhau.
- Hệ thống cổng NHNN: cung cấp cổng thông tin nghiệp vụ cho các TCTD được phép tra cứu thông tin liên quan đến TCTD đó, thực hiện một số nghiệp vụ ngân hàng như chuyển tiền, nộp báo cáo dự trữ bắt buộc, cho vay.

## TÀI LIỆU THAM KHẢO

### Tiếng Việt

[1] PGS.TS. Nguyễn Ngọc Hóa, Bài giảng Tích hợp hệ thống

[2] Quy trình nghiệp vụ ngân hàng tại NHNN

### Tiếng Anh

[3] Carl Jones (2011), “*Do more with SOA Integration: Best of Packt*”, Packt Publishing.

[4] Greg Flurry and Kim Clark (2011), “*The Enterprise Service Bus*”, IBM developer Works.

[5] Kabul Kurniawan, Ahmad Ashari (2015), “Service orchestration using enterprise service bus for real-time government executive dashboard system”, Data and Software Engineering (ICoDSE), 2015 International Conference on, pp 207-212.

[6] Raleigh Center (2011), Redbooks IBM, “*Smart SOA Solutions with WebSphere Enterprise Service Bus Registry Edition, V7.5*”

[7] Ray Harisshankar (2009), “*SOA-Based Enterprise Integration*”, The McGraw-Hill Companies; 1 edition

[8] Tibco Software Inc (2010), “*Tibco Education Programs*”.

### Internet

[9] [https://en.wikipedia.org/wiki/System\\_integration](https://en.wikipedia.org/wiki/System_integration)

[10] <https://www.talend.com/resource/enterprise-service-bus.html>

[11] <http://www.tibco.com/products/automation/application-integration/activematrix-businessworks/enterprise-service-bus>