

MESSAGE QUEUE

TS. Huỳnh Hữu Nghĩa

luckerhuynhvn@gmail.com

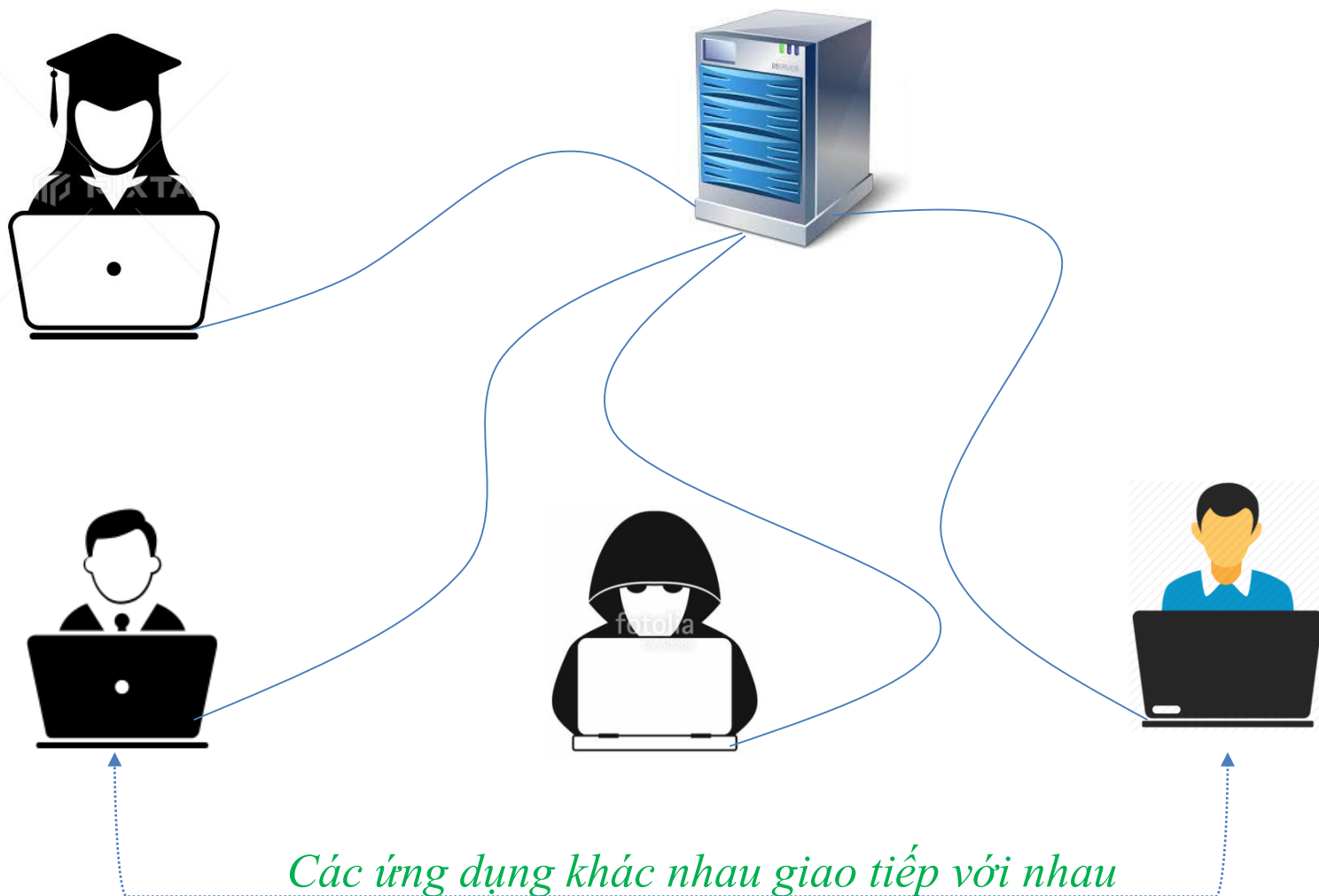


Nội dung:

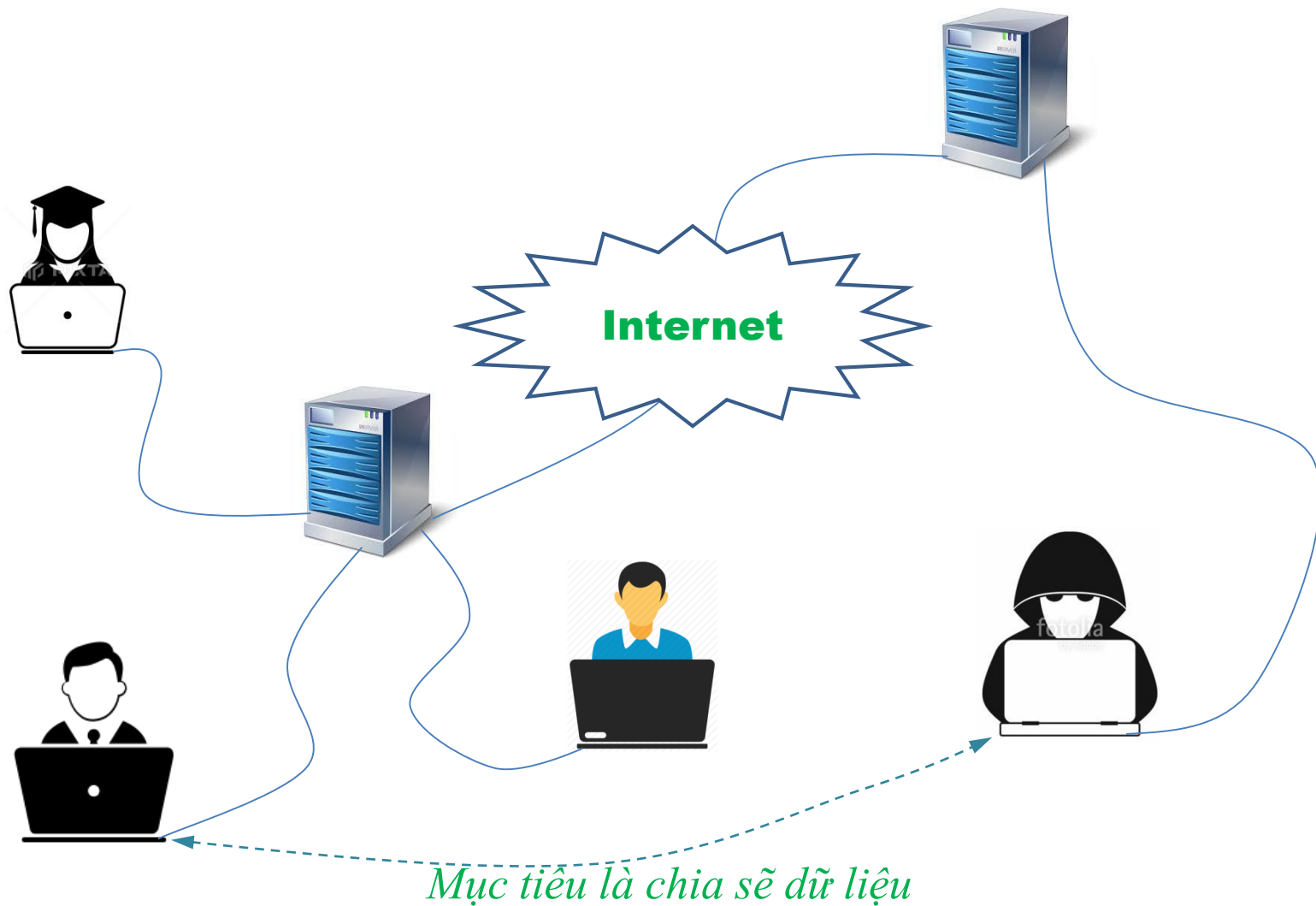
- ❖ Microsoft Message Queue (MSMQ)
- ❖ Apache ActiveMQ

Microsoft Message Queue (MSMQ)

Message Queuing là gì?

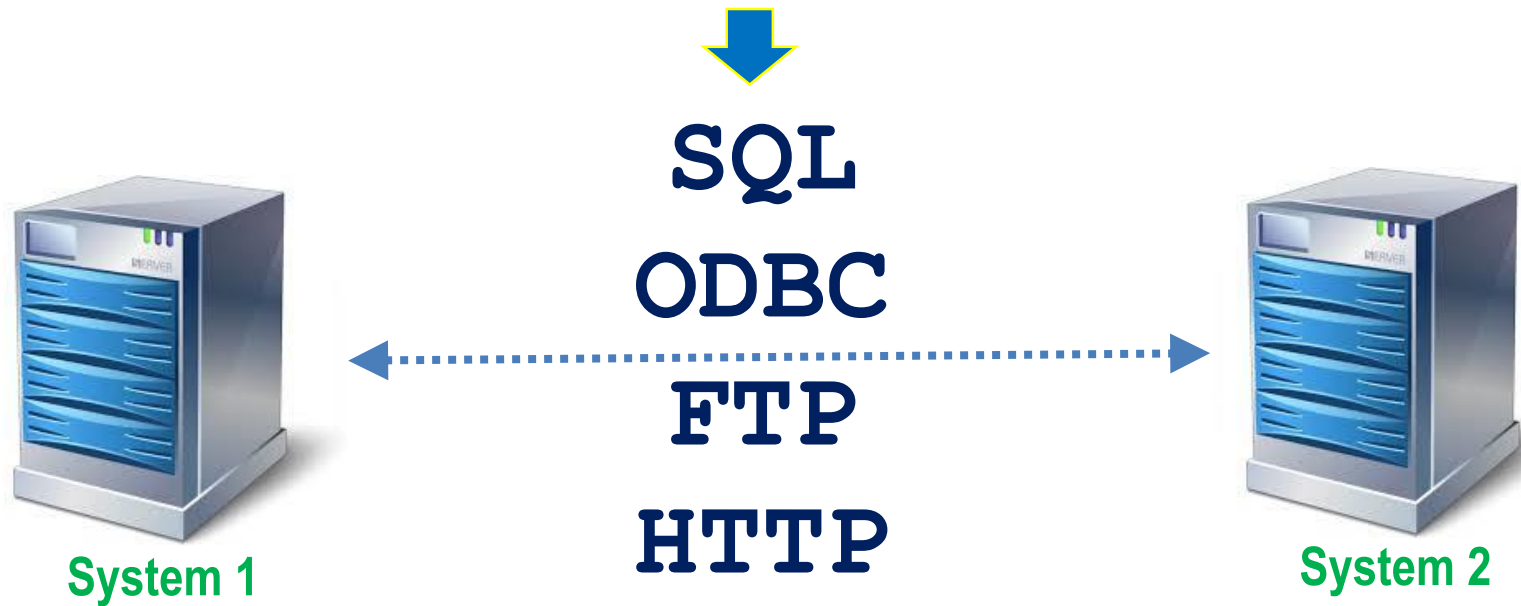


Message Queuing là gì?



Message Queuing là gì?

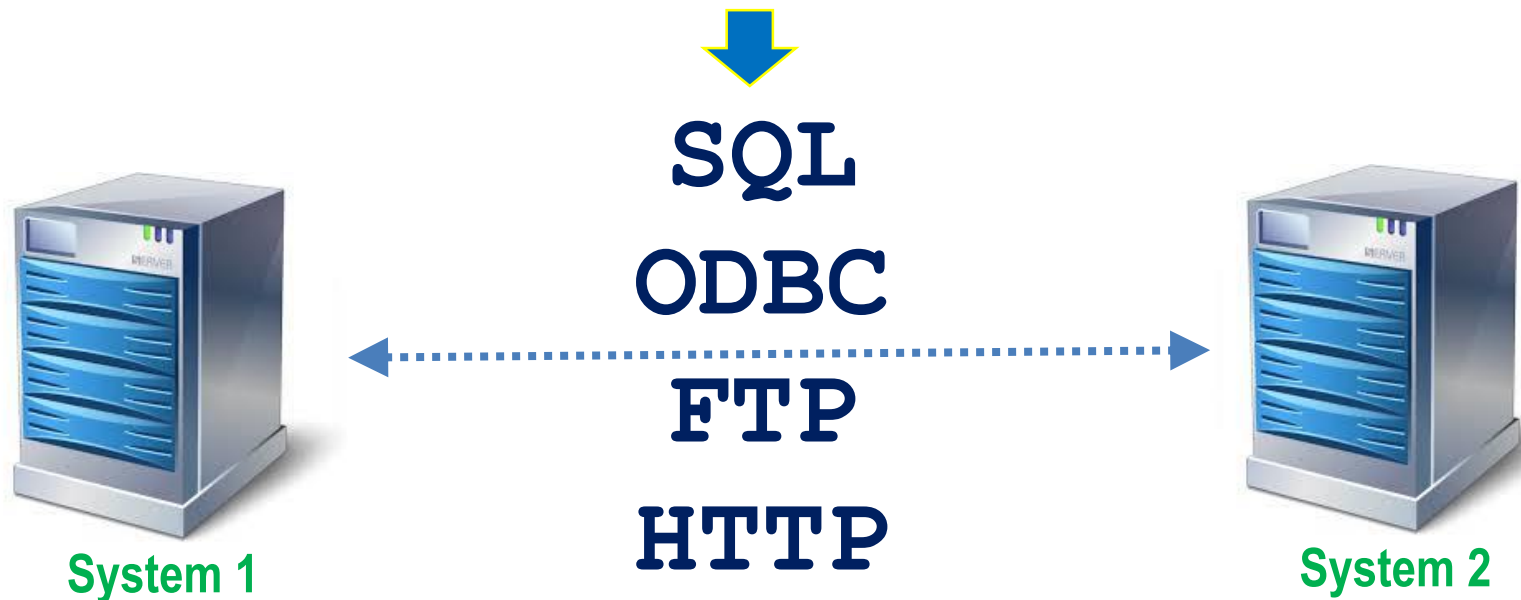
Các công nghệ chuyển đổi dữ liệu qua các hệ thống



đều cung cấp các cơ chế khác nhau cho dữ liệu được di chuyển giữa các hệ thống trong môi trường không đồng nhất

Message Queuing là gì?

Các công nghệ chuyển đổi dữ liệu qua các hệ thống



Hạn chế: Các ứ/d cả 2 đầu phải hiểu chính xác giao thức cùng nhau theo cách định dạng dữ liệu và phải tham gia giao tiếp ở cùng thời điểm

→ mạng giao tiếp giữa 2 hệ thống cần được xác thực.

Message Queuing là gì?

*Các công nghệ chuyển đổi thông tin
giữa các ứng dụng*



COM

(Component Object Model)

CORBA

(Common Object Request Broker Architecture)



System 1



System 2

*Giao tiếp có thể diễn ra khi cả hai ứng dụng phải
chạy và kết nối không tổn hại nhau.*

Message Queuing là gì?

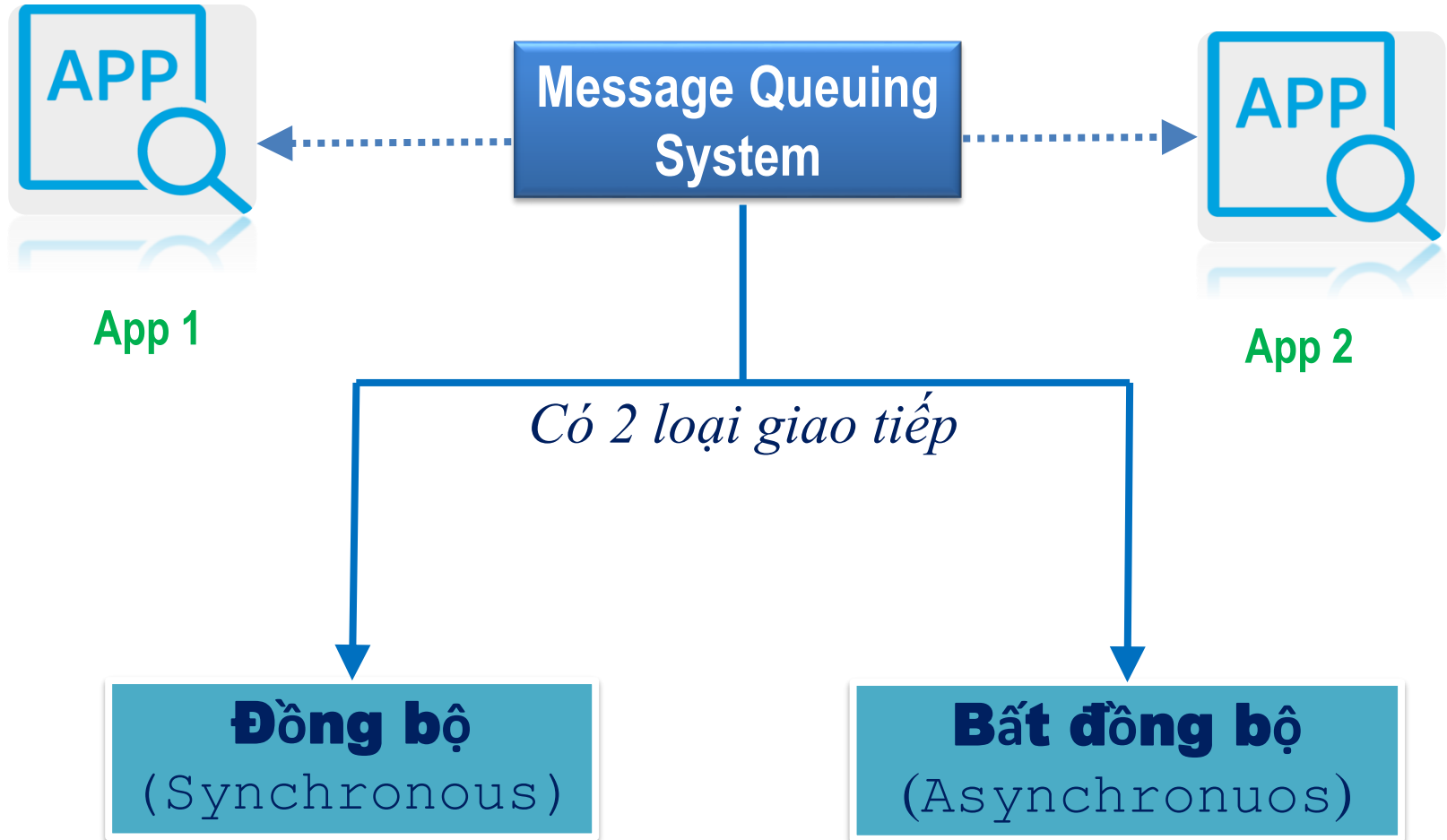
Điều gì sẽ xảy ra nếu một trong các hệ thống không có sẵn hoặc kết nối giữa hai hệ thống không hoạt động?

COM và CORBA

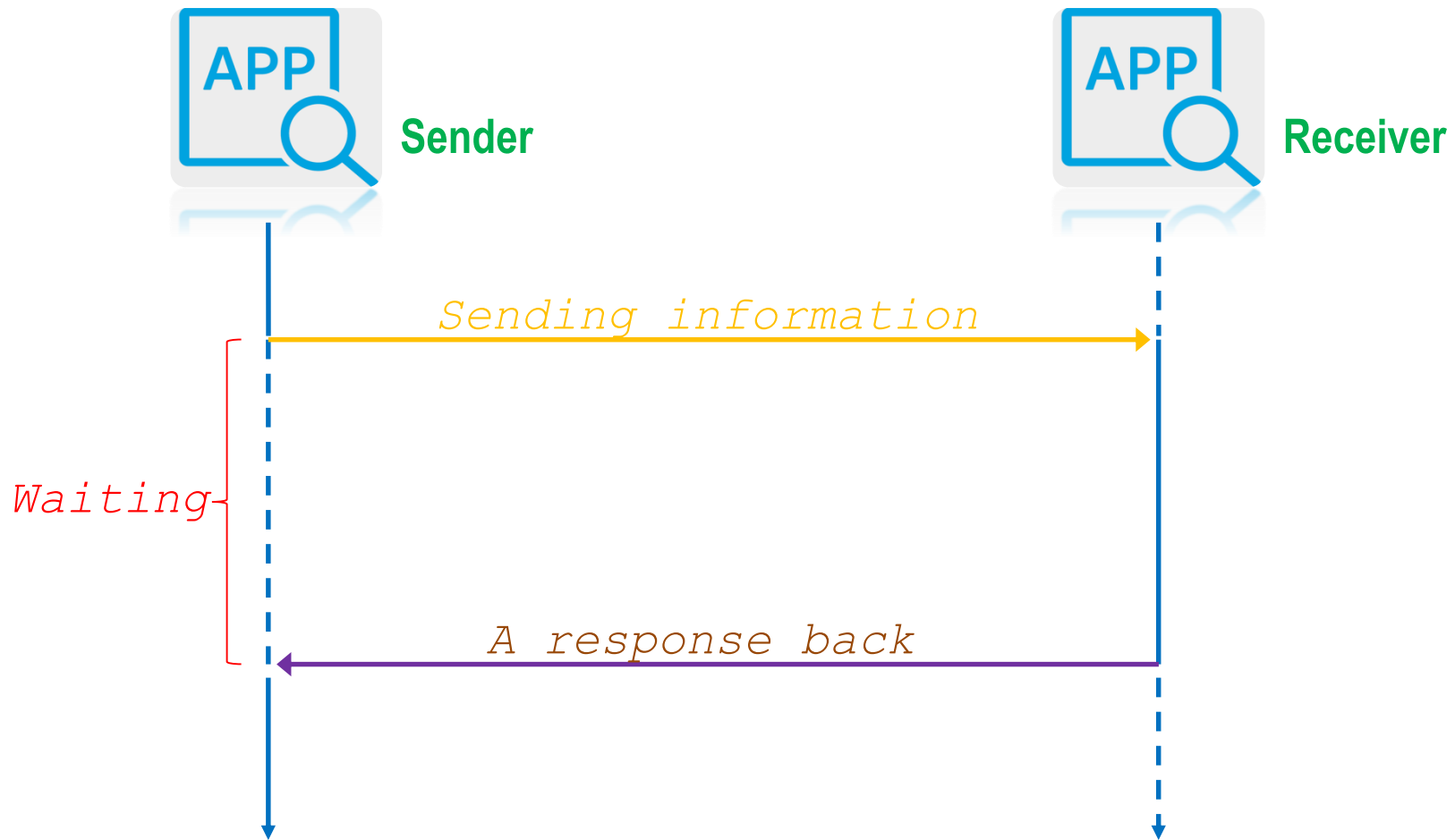


Hệ thống **Message Queuing**

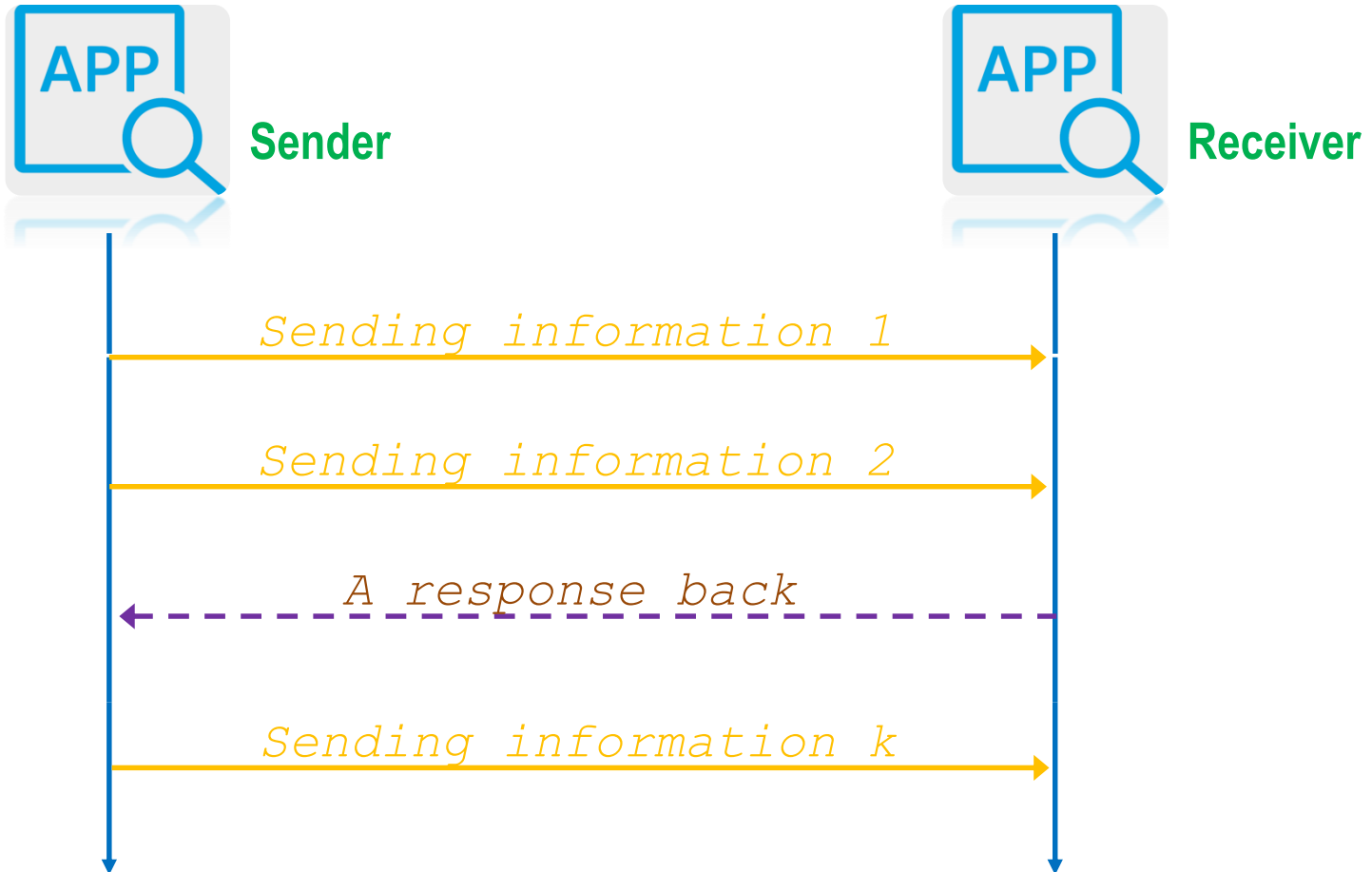
Các Kiểu Giao Tiếp



Đồng Bộ (*Synchronous*)



Bất Đồng Bộ (*Asynchronous*)



Tại sao sử dụng *Asynchronous*?

Có 3 lý do chính:

1. Nếu người gửi luôn phải đợi trả lời thì hệ thống sẽ dành nhiều thời gian đợi hơn là xử lý. Giao tiếp không đồng bộ sẽ làm hệ thống nhanh hơn, vì nó dành thời gian xử lý hơn chờ đợi.

Tại sao sử dụng *Asynchronous*?

Có 3 lý do chính:

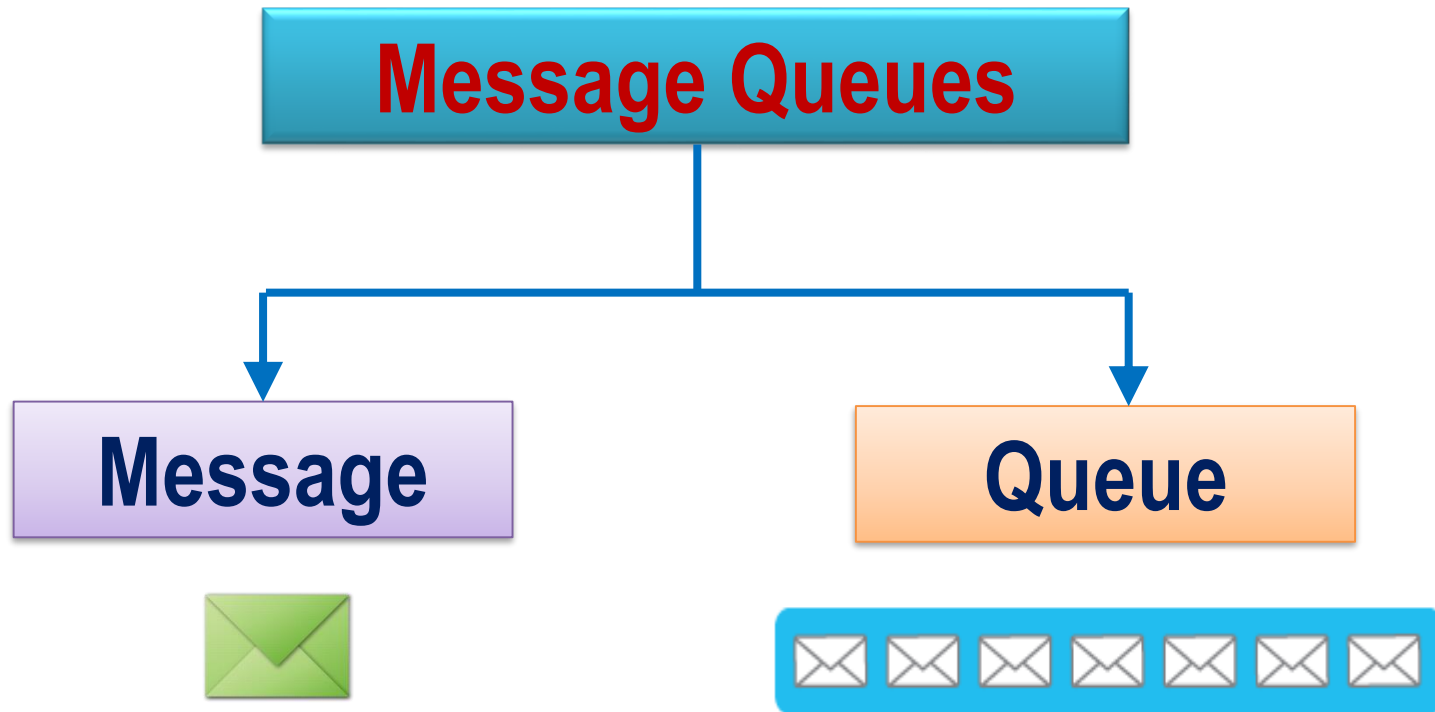
2. Khi người gửi không quan tâm khi người nhận xử lý message như thế nào. Nó giống như ủy quyền công việc cho một ai đó.

Tại sao sử dụng *Asynchronous*?

Có 3 lý do chính:

3. Nếu ứng dụng chạy trong môi trường kết nối giữa hai ứng dụng không được xác thực, hoặc nếu 1 ứng dụng đang chạy trên máy tính không được kết nối. Các ứng dụng chỉ chạy trong giờ nghỉ.

Hệ thống Message Queue



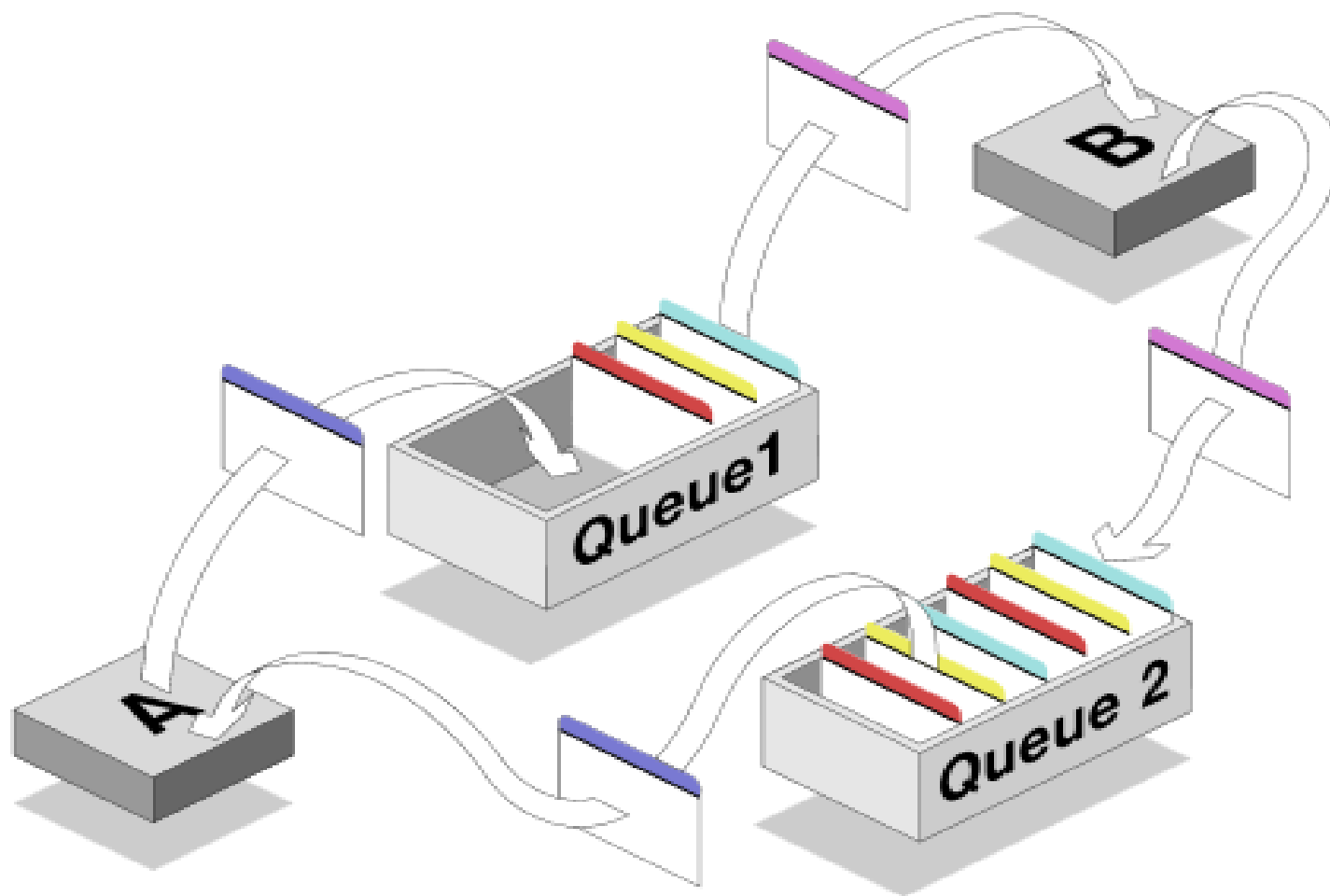
Nội dung Message

- Một message được tạo từ 1 số thành phần khác nhau
- Thành phần chính của message là nội dung
- Dữ liệu có thể là văn bản hay nhị phân
- Định dạng nội dung bởi 2 ứng dụng truyền message
- Không có phương thức chuẩn nào để định dạng nội dung của một message

Nội dung Message

- Phần thân của message được diễn tả bởi hệ thống, ứng dụng có thể đọc nội dung của message là phải biết chính xác nội dung có gì.
- Một message cũng bao gồm thông tin người gửi và người nhận. Và dấu hiệu thời gian khi gửi
- Có thể gồm một thời gian hết hạn, sau đó nó không được đọc thì sẽ tự xóa.

Message Queue



Message Queuing vs. E-Mail

Nhìn chung thì tương tự:

- ✓ Cả 2 hệ thống có message với senders và receivers
- ✓ Hệ thống E-Mail có mailboxes và MQ có queues
- ✓ Bộ quản lý queue cũng giống e-mail server.

Cái gì là khác nhau?

Message Queuing vs. E-Mail

Message Queuing	E-Mail
<ul style="list-style-type: none">✓ Giao tiếp: application to application✓ Các ứng dụng phải có khả năng diễn giải các kết quả của message✓ Message phải cung cấp kiểu thông tin theo một chuẩn mà ứng dụng có thể đọc và hiểu message.✓ Nội dung được hiểu bởi ứng dụng gửi và ứng dụng nhận✓ Thông tin có thể được viết tắt bởi hai ứng dụng, không phải viết đầy đủ để con người hiểu✓ Có nhiều biện pháp bảo vệ message đi đến nơi.	<ul style="list-style-type: none">✓ Giao tiếp: person to person✓ Thông tin gửi có thể được đọc và giải thích bởi con người.✓ Nội dung được hiểu bởi con người

Message Queuing vs. E-Mail

- **MSMQ** cung cấp hai phương pháp:
 - **Express:** message chỉ nằm trong bộ nhớ (vì sử dụng ít tài nguyên) và có thể di chuyển qua hệ thống MQ nhanh hơn.
 - **Recoverable:** message được ghi vào bộ lưu trữ cố định ở mỗi bước từ người gửi đến đích.

Message Queuing vs. E-Mail

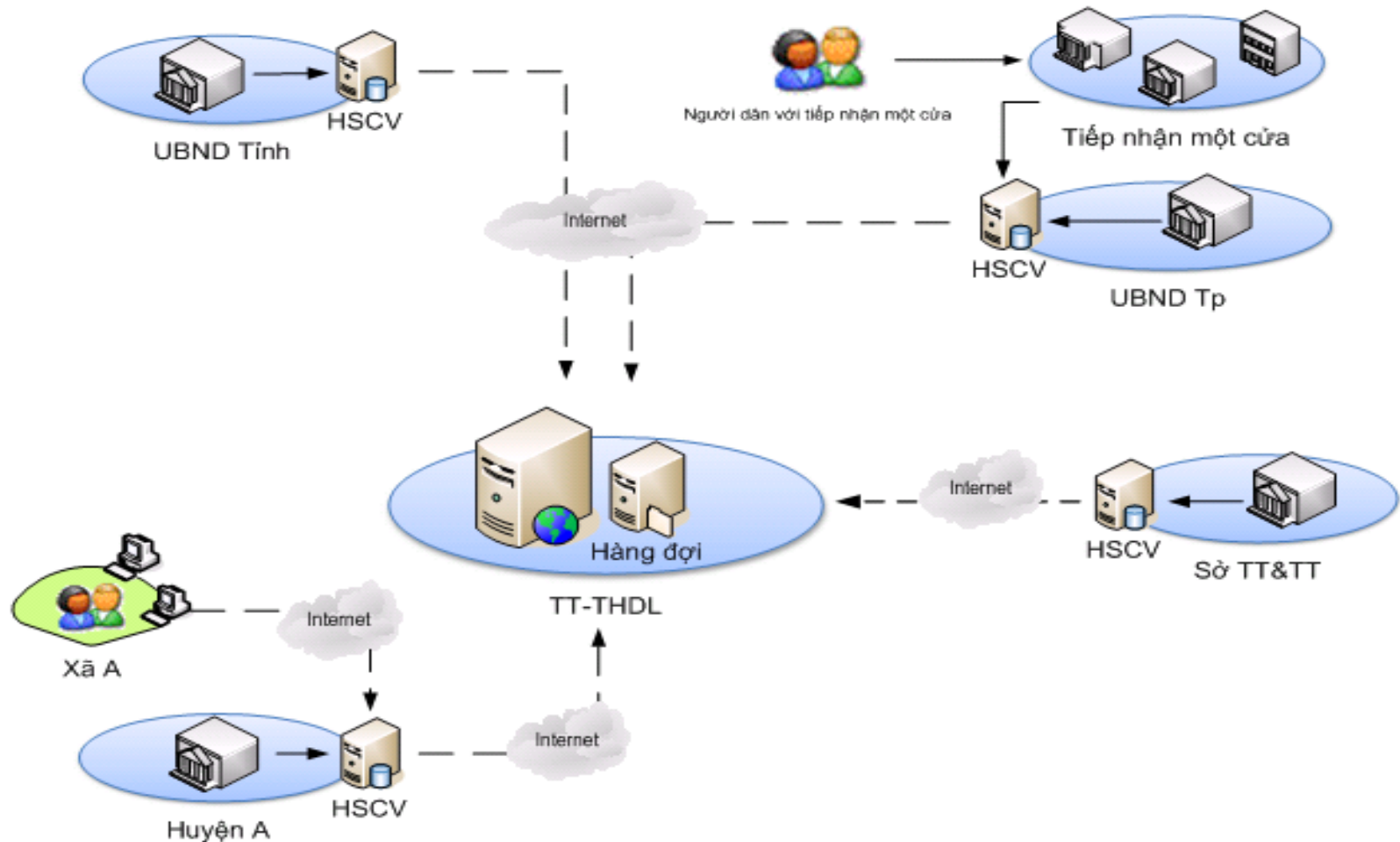
Nếu 2 message (1 Express và 1 Recoverable) ở trong máy chủ queue đang chờ để chuyển đến máy chủ queue khác và máy chủ bị chết.

Tại sao sử dụng **Message Queue**?

- ✓ Cần kiểm tra xem các công nghệ có sẵn khác có thể đóng vai trò như hệ thống MQ.
- ✓ Cũng cần xem khi nào nên sử dụng MQ hơn là sử dụng các công nghệ khác.
- ✓ Cần nhìn vào sự thay đổi kiến trúc của các hệ thống hiện nay.
- ✓ Nên cân bằng sự lựa chọn MQ hơn công nghệ khác.

Tại sao sử dụng Message Queue?

Các hệ thống phân tán ngày càng phổ biến



Tại sao sử dụng **Message Queue**?

Khi phân tán quá trình xử lý và dữ liệu trong một hệ thống:

- Lo lắng về việc các máy tính kết nối nhau.
- Các kết nối có thể không vĩnh viễn.
- Kết nối và chuyển đổi dữ liệu
- Người dùng có thể ngắt kết nối để thực hiện các công việc cục bộ với ứng dụng của họ.

➔ ***MQ là công nghệ quan trọng cho các ứng dụng kinh doanh trong tương lai.***

Khi nào sử dụng **Message Queue**?

- ✓ Nếu 2 ứng dụng đang giao tiếp không chạy đồng thời.
- ✓ Nền message quan trọng và sẽ có vấn đề khi nó bị mất.
- ✓ Nếu ứng dụng gửi không thường xuyên được kết nối đến ứng dụng nhận.
- ✓ Nếu ứng dụng gửi không thể duy trì việc kết nối mạng xác thực đến ứng dụng nhận.
- ✓ Thực hiện việc giao tiếp với các ứng dụng khác bất đồng bộ và có thể không quan tâm đến phản hồi.

Thực hành Message Queue

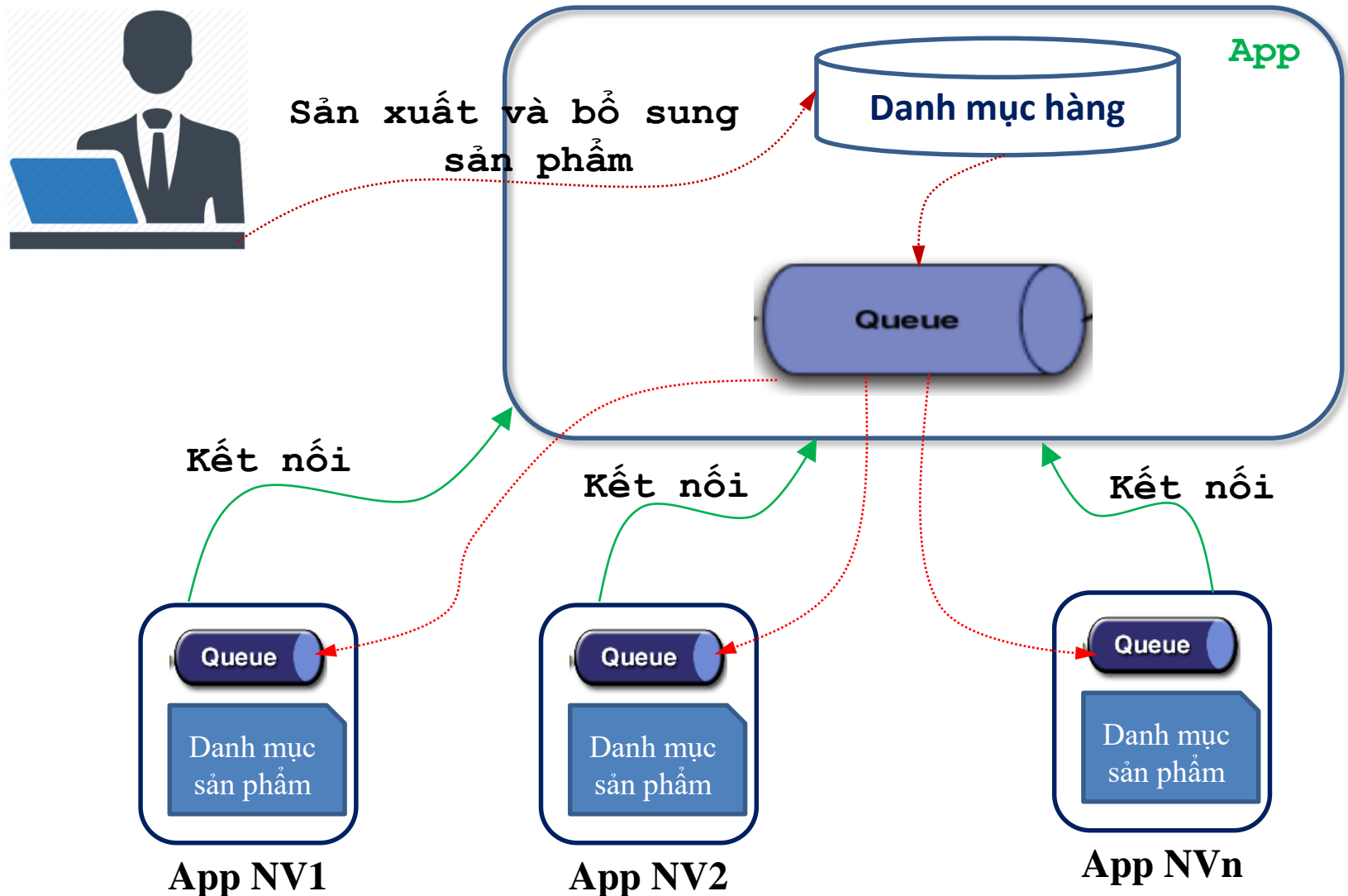
Lưu ý:

- ✓ Không phải mọi ứng dụng đều thích hợp sử dụng MQ.
- ✓ Nếu sử dụng sai chỗ cho ứng dụng sai, MQ đôi khi có thể gây ra nhiều thiệt hại hơn lợi.

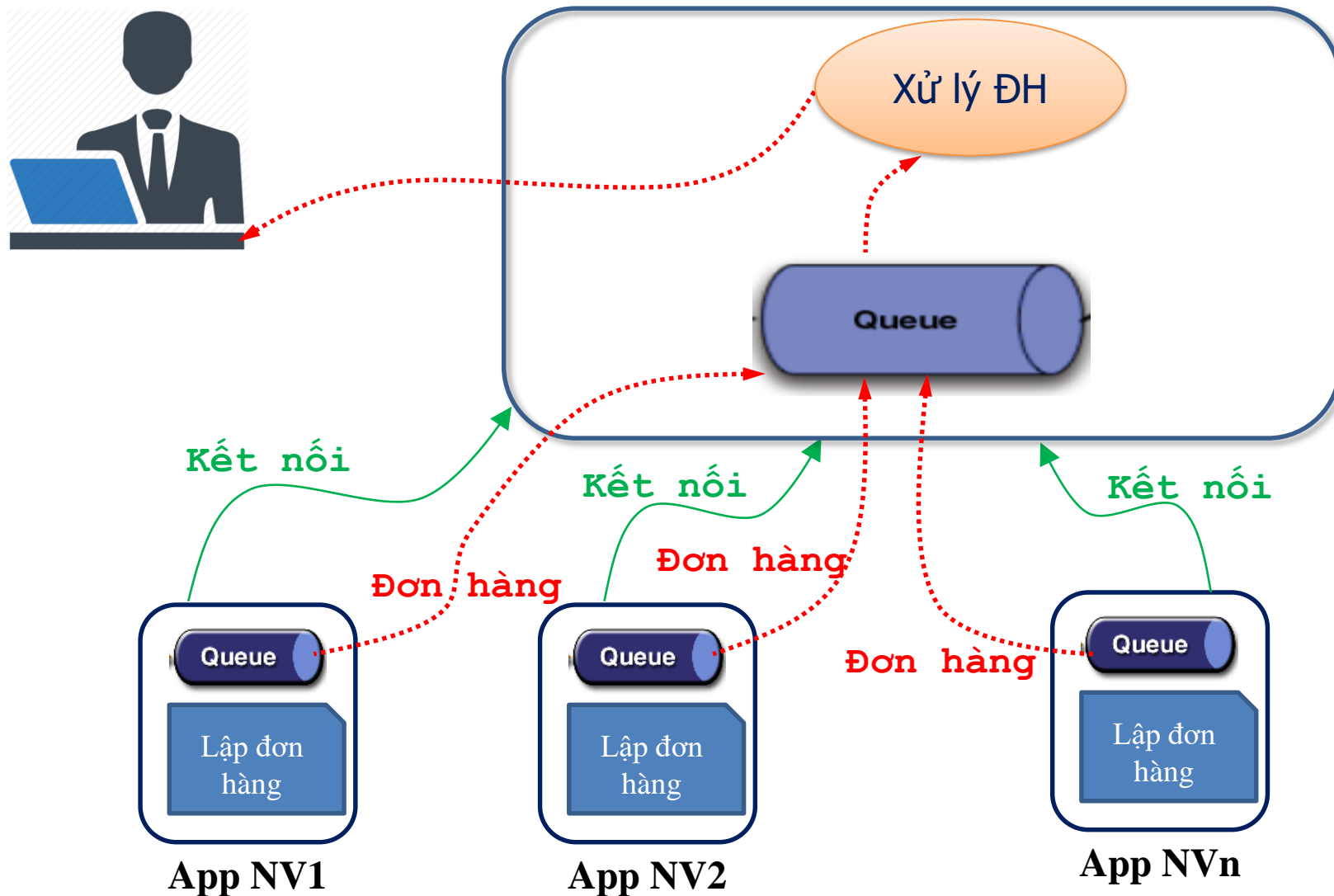
Thực hành Message Queue

Giả sử một tổ chức dựa vào một nhóm nhân viên bán hàng di động. Các nhân viên bán hàng giới thiệu các sản phẩm đến những người mua tiềm năng và tạo ra các đơn hàng dựa trên cái gì mà người mua muốn. Yêu cầu là nhân viên bán phải cập nhật thông tin sản phẩm có sẵn để bán. Vấn đề quan trọng là giới thiệu sản phẩm và tạo các đơn hàng trong điều kiện máy tính xách tay không kết nối đến đường dây điện thoại.

Ứng dụng thứ 1



Ứng dụng thứ 2



Ứng dụng Web

- Có thể sử dụng MQ trong ứng dụng Web.
- Nếu mỗi trang web hay nhiều trang web mà người dùng có thể truy cập dưới dạng một ứng dụng, thì việc chuyển đổi thông tin qua lại giữa các ứng dụng có thể sử dụng Queuing.
- Công nghệ ASP cho phép giao tiếp giữa các trang trong cùng ứng dụng thông qua đối tượng Session.
- Ví dụ bán sách qua mạng.

Apache ActiveMQ

Tổng quan

- ✓ Phần mềm nhắn tin doanh nghiệp đã tồn tại từ những năm 1980. Nhắn tin không chỉ là một kiểu giao tiếp giữa các ứng dụng mà còn là kiểu tích hợp.
- ✓ Do đó, nhắn tin đáp ứng đầy đủ nhu cầu cả thông báo cũng như sự tương tác giữa các ứng dụng.
- ✓ ActiveMQ là một giải pháp cung cấp khả năng cho các ứng dụng giao tiếp một cách không đồng bộ và lỏng lẻo.

ActiveMQ là gì?

- ActiveMQ là một phần mềm mã nguồn mở trung gian hướng message (MOM) trên nền tảng phần mềm Apache cung cấp khả năng sẵn sàng, hiệu quả, khả năng mở rộng, độ tin cậy và bảo mật cho message doanh nghiệp.
- Mục tiêu của ActiveMQ là cung cấp dựa trên các tiêu chuẩn, sự tương tác ứng dụng hướng message dựa trên nhiều nền tảng và ngôn ngữ.

Các đặc điểm ActiveMQ

- **Tuân thủ JMS.** ActiveMQ là một hiện thực của JMS 1.1 spec, dựa trên các tiêu chuẩn MOM phù hợp với JMS.
- **Kết nối.** ActiveMQ cung cấp nhiều tùy chọn kết nối, hỗ trợ các giao thức như: HTTP/S, multicast, SSL, Stomp, TCP, UDP, XMPP và một số cái khác
- **Khả năng bền bỉ và bảo mật.** ActiveMQ cung cấp nhiều tính năng bền bỉ và tùy chọn. Bảo mật trong ActiveMQ có thể tùy chỉnh hoàn toàn cho loại xác thực và ủy quyền phù hợp.

Các đặc điểm ActiveMQ

- **Client APIs.** ActiveMQ cung cấp client API cho nhiều ngôn ngữ lập trình gồm C/C++, .NET, Perl, PHP, Python, và ngôn ngữ khác.
- **Nhóm broker.** Nhiều ActiveMQ broker có thể làm việc cùng nhau như một mạng lưới liên kết brokers nhằm mục tiêu mở rộng. Được xem là mạng lưới brokers và có thể hỗ trợ nhiều cấu trúc liên kết khác nhau.

Các đặc điểm ActiveMQ

- **Các đặc trưng broker tiên tiến và tùy chọn client.**
ActiveMQ cung cấp nhiều tính năng tinh vi cho cả broker và clients kết nối đến brokers.
- **Đơn giản hóa quản trị.** ActiveMQ được thiết kế với những nhà phát triển tâm huyết, nó không yêu cầu một quản trị viên chuyên nghiệp do nó cung cấp nhiều tính năng quản trị dễ sử dụng nhưng rất mạnh.

Khi nào dùng ActiveMQ

- Các nhà phát triển nguồn mở hợp nhất để hình thành Apache Geronimo. Geronimo cần cài đặt JMS do tương thích với J2EE. Nhưng họ không có giấy phép của BSD.
- Các doanh nghiệp không đủ khả năng chi phí để theo đuổi MOM thương mại.
- Cho nên các nhà phát triển muốn có một MOM nguồn mở sử dụng giấy phép Apache.
- → Apache ActiveMQ. ActiveMQ được phát triển như JMS hỗ trợ giao tiếp từ xa giữa các ứng dụng phân tán.

Khớp nối lỏng lẻo và ActiveMQ

- ActiveMQ cung cấp lợi ích khớp nối lỏng lẻo cho kiến trúc ứng dụng nhằm giảm thiểu các khớp nối chặt chẽ cổ điển của RPC (Remote Procedure Calls).
- Việc thiết kế lỏng lẻo được xem là bất đồng bộ.
- ActiveMQ hoạt động như người trung gian cho phép tích hợp không đồng nhất và tương tác một cách không đồng bộ.

Khớp nối lỏng lẻo và ActiveMQ

- Các công nghệ như COM, CORBA, DCE và EJB sử dụng các kỹ thuật được gọi RPC. Sử dụng RPC, khi một ứng dụng gọi một ứng dụng khác, bên gọi sẽ bị khóa cho đến khi đối tượng được gọi trả lại quyền kiểm soát cho bên gọi.

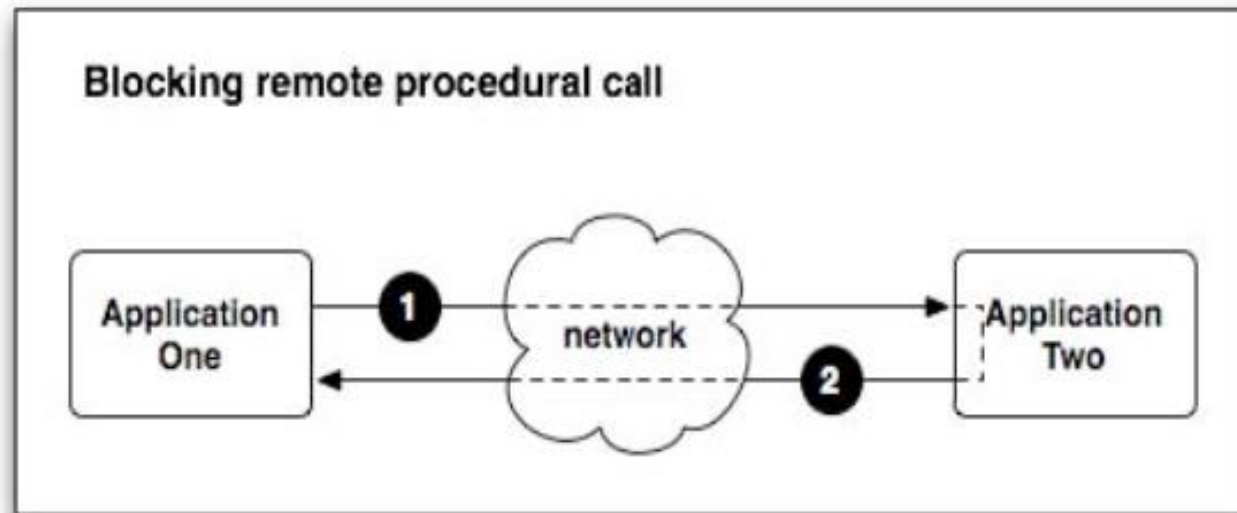


Figure 1 Two tightly-coupled applications using remote procedure calls to communicate

Khớp nối lỏng lẻo và ActiveMQ

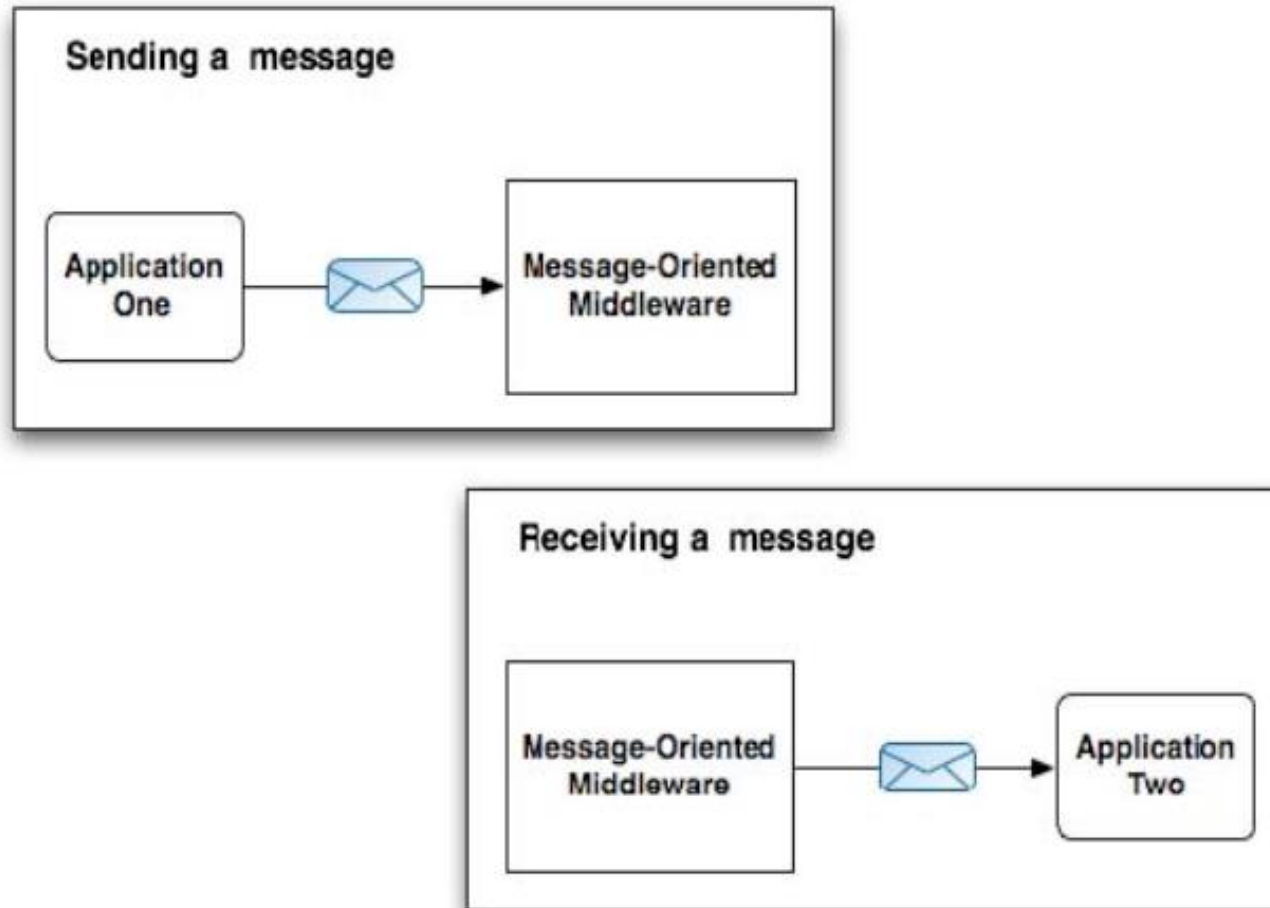


Figure 2 Two loosely-coupled applications using JMS messaging to communicate

Khớp nối lỏng lẻo và ActiveMQ

- Kịch bản là khi có nhiều thể hiện của ứng dụng một và nhiều thể hiện ứng dụng hai, trong đó mỗi thể hiện cư trú trên một máy khác. ActiveMQ được cài đặt trên một máy khác độc lập với ứng dụng một hoặc hai.
- ➔ *Như vậy, các thể hiện bất kỳ ứng dụng một hoặc hai có thể được di chuyển chỗ này sang chỗ kia mà không ảnh hưởng lẫn nhau.*

Khi nào dùng ActiveMQ

Có nhiều kịch bản trong đó ActiveMQ và nhắn tin bất đồng bộ có thể tác động có ý nghĩa đối với kiến trúc hệ thống.

Khi nào dùng ActiveMQ

- **Hợp nhất ứng dụng không đồng nhất.** *Đây là một lợi thế rất lớn khi có thể tích hợp các ứng dụng được viết bằng các ngôn ngữ khác nhau trên các nền tảng khác nhau.* ActiveMQ cung cấp **Java client API, client C/C++, .NET, Perl, PHP, Python, Ruby** và **vài ngôn ngữ khác.**
- **Một sự thay thế cho RPC.** *Các ứng dụng client-server sử dụng RPC bao gồm các máy ATM, hầu hết các ứng dụng web, các hệ thống thẻ tín dụng, các hệ thống bán lẻ, và nhiều thứ khác thường có quy mô hạn chế và làm chậm hệ thống .*

Khi nào dùng ActiveMQ

- **Nơi lòng sự ghép nối giữa các ứng dụng.** *Kiến trúc kết nối chặt chẽ có vấn đề khi thiết kế phân tán. Còn kiến trúc kết hợp lỏng lẻo thể hiện ít phụ thuộc hơn, làm cho nó tốt hơn trong việc xử lý các thay đổi bất ngờ/không biết trước. Khi thay đổi một thành phần trong hệ thống thì không bị gợn sóng trên toàn bộ hệ thống và sự tương tác giữa các thành phần cũng được đơn giản hoá đáng kể.*

