

Bài tập lớn II: Viết hệ thống chứng khoán (đặc tả + code) như 1 phần miêu tả trong sách enterprise integration patterns.

Tham khảo: Mục 13. **INTEGRATION PATTERNS IN PRACTICE** (528)

Mô Tả:

Hệ thống giao dịch cổ phiếu là một ứng dụng thời gian thực (real-time) cho phép các nhà giao dịch mua và bán cổ phiếu trên sàn giao dịch cổ phiếu. Nó bao gồm nhiều thành phần và điểm tích hợp khác nhau để hỗ trợ đặt lệnh, thực hiện lệnh và theo dõi thời gian thực giá cổ phiếu. Hệ thống cần xử lý một lượng giao dịch lớn một cách hiệu quả đồng thời đảm bảo độ tin cậy, bảo mật và độ trễ thấp.

EIP là một danh mục các design patterns để phát triển hệ thống nhằm tích hợp phần mềm mới và hiện có trong môi trường kinh doanh. Nó mô tả 65 patterns được cấu trúc thành 9 loại để sử dụng tích hợp ứng dụng doanh nghiệp và phần mềm trung gian hướng thông báo dưới dạng mẫu và bao gồm icon-based pattern language.

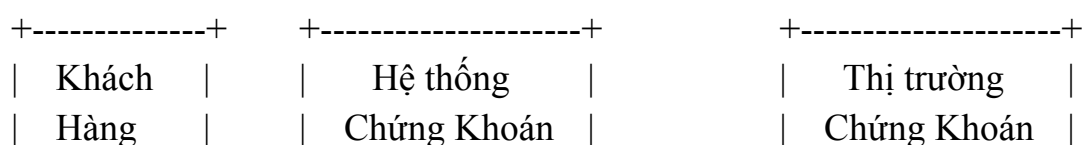
- Client/ Server
 - TIB (TIBCO information bus 401) -> JMS - socket - technique (539)
 - Overview (535)
 - Internal Structure (537)
 - update (543)
- Crash

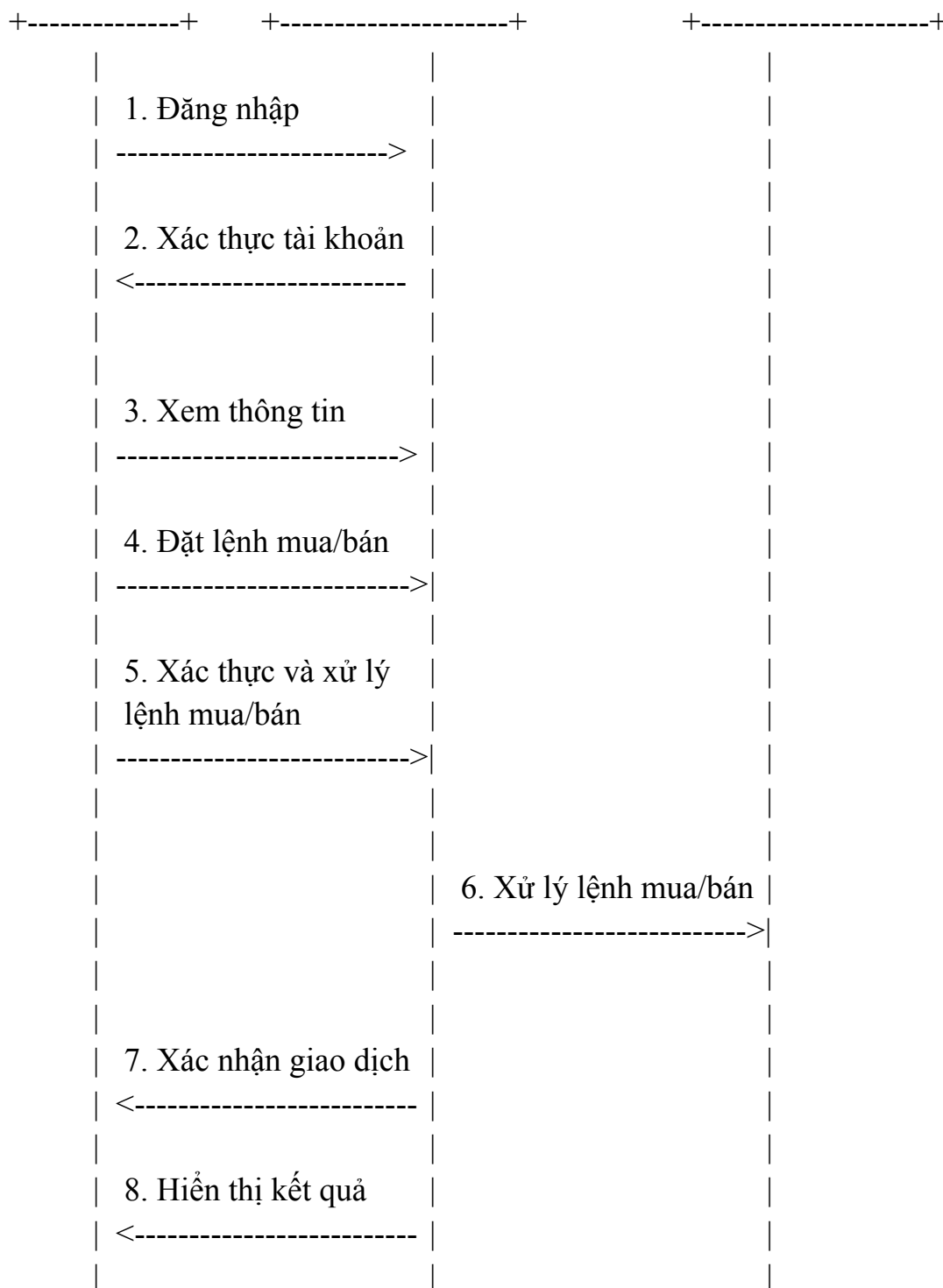
***Yêu cầu:**

Xác định Yêu cầu Hệ Thống Chứng Khoán:

Yêu cầu cụ thể cho hệ thống chứng khoán, bao gồm: quy trình giao dịch, giao dịch trực tuyến, tài khoản khách hàng, theo dõi chứng khoán, và quản lý rủi ro.

Đây là luồng đi của quy trình giao dịch:





- **Khách hàng đăng nhập** vào hệ thống.
- **Xem thông tin:** Khách hàng có thể xem thông tin tài khoản của họ và tình hình thị trường chứng khoán.
- **Đặt lệnh mua/bán:** Khách hàng đặt lệnh mua hoặc bán chứng khoán thông qua giao diện hệ thống.

- **Xác thực tài khoản:** Hệ thống chứng khoán xác thực tài khoản của khách hàng để đảm bảo tính bảo mật và đủ tiền cho giao dịch.
- **Xác thực và xử lý lệnh mua/bán:** Hệ thống xác thực và xử lý lệnh mua/bán theo yêu cầu của khách hàng. Điều này bao gồm kiểm tra tính khả thi của giao dịch và xử lý tài chính.
- **Xử lý lệnh mua/bán:** Hệ thống gửi lệnh mua/bán đến thị trường chứng khoán để thực hiện giao dịch.
- **Xác nhận giao dịch:** Hệ thống nhận thông báo xác nhận từ thị trường chứng khoán rằng giao dịch đã thành công.
- **Hiển thị kết quả:** Hệ thống hiển thị kết quả của giao dịch cho khách hàng, bao gồm số lượng cổ phiếu mua/bán, giá, và thông tin liên quan khác.

Thiết kế cơ sở dữ liệu:

Đây chỉ là csdl tham khảo, trong quá trình làm nếu cần thêm gì thì mọi người bổ sung nhé:

Bảng Người dùng (User):

Field	Data Type	Describe
User_ID	INT (PRIMARY KEY)	Khóa chính, ID người dùng
Username	VARCHAR(50)	Tên đăng nhập người dùng
Password	VARCHAR(100)	Mật khẩu người dùng (đã băm)

Full_Name	VARCHAR(100)	Tên đầy đủ của người dùng
Email	VARCHAR(100)	Địa chỉ email của người dùng
Role	VARCHAR(20)	Quyền của người dùng (ví dụ: người mua, người bán)
...		

Bảng Tài khoản (Account Table):

Field	Data Type	Describe
Account_ID	INT (PRIMARY KEY)	Khóa chính, ID tài khoản
User_ID	INT (FOREIGN KEY)	ID người dùng liên kết đến tài khoản

Account_Type	VARCHAR(20)	Loại tài khoản (ví dụ: tiết kiệm, giao dịch)
Balance	DECIMAL(10, 2)	Số dư trong tài khoản
Status	VARCHAR(20)	Trạng thái tài khoản (ví dụ: hoạt động, đóng)
...		

Bảng Chứng khoán (Stock Table):

Field	Data Type	Describe
Stock_ID	INT (PRIMARY KEY)	Khóa chính, ID chứng khoán
Stock_Name	VARCHAR(100)	Tên chứng khoán

Stock_Type	VARCHAR(20)	Loại chứng khoán (ví dụ: cổ phiếu, trái phiếu)
Current_Price	DECIMAL(10, 2)	Giá hiện tại của chứng khoán
...		

Bảng Lệnh (Order Table):

Field	Data Type	Describe
Order_ID	INT (PRIMARY KEY)	Khóa chính, ID lệnh
User_ID	INT (FOREIGN KEY)	ID người dùng tạo lệnh
Stock_ID	INT (FOREIGN KEY)	ID chứng khoán liên quan đến lệnh

Order_Type	VARCHAR(20)	Loại lệnh (mua hoặc bán)
Quantity	INT	Số lượng chứng khoán
Price	DECIMAL(10, 2)	Giá đặt lệnh
Status	VARCHAR(20)	Trạng thái lệnh (đã thực hiện, đang chờ)
...		

Bảng Giao dịch (Transaction Table):

Field	Data Type	Describe
Transaction_ID	INT (PRIMARY KEY)	Khóa chính, ID giao dịch

Buyer_ID	INT (FOREIGN KEY)	ID người mua
Seller_ID	INT (FOREIGN KEY)	ID người bán
Stock_ID	INT (FOREIGN KEY)	ID chứng khoán liên quan đến giao dịch
Quantity	INT	Số lượng chứng khoán
Price	DECIMAL(10, 2)	Giá giao dịch
Date_Time	DATETIME	Thời gian giao dịch

Thiết kế Hệ Thống Chứng Khoán:

Thiết kế kiến trúc sử dụng EIP: (Message Routing, Message Transformation, Messaging Channels, Messaging Endpoints, Messaging Systems, System Management, Message Construction) để mô hình hóa hệ thống giao tiếp và trao đổi dữ liệu.

Tạo bản đặc tả thiết kế hệ thống chứng khoán bao gồm kiến thức về kiến trúc hệ thống, giao diện người dùng, tích hợp với thị trường chứng khoán, quản lý tài khoản, và các chức năng chính.

Triển Khai Cơ Sở Hạ Tầng:

Xây dựng cơ sở hạ tầng hệ thống, bao gồm máy chủ, mạng, cơ sở dữ liệu, và hệ thống an toàn.

Xây Dựng Ứng Dụng Chứng Khoán:

Phát triển các ứng dụng máy khách và máy chủ để thực hiện các chức năng giao dịch, theo dõi chứng khoán, và quản lý tài khoản.

Tích Hợp Với Thị Trường Chứng Khoán: Tích hợp hệ thống chứng khoán của bạn với thị trường chứng khoán hoặc các sàn giao dịch.

Xây Dựng Giao Diện Người Dùng: Phát triển giao diện người dùng cho khách hàng và nhà đầu tư để thực hiện giao dịch và theo dõi tình hình thị trường.

Bảo Mật và Quản Lý Phiên: Đảm bảo tính bảo mật cho hệ thống chứng khoán, bao gồm bảo vệ thông tin cá nhân của khách hàng và quản lý phiên đăng nhập.

Kiểm Tra và Tối Ưu Hóa Hệ Thống: Kiểm tra và tối ưu hóa hệ thống để đảm bảo tính ổn định và hiệu suất.

TÌM HIỂU HỆ THỐNG TRÁI PHIẾU TRONG SÁCH (bond trading system)

- **Xây dựng hệ thống:**
 - + Xác định các thành phần và dịch vụ trong hệ thống, bao gồm quản lý tài khoản, đặt lệnh mua/bán, xử lý giao dịch, quản lý dữ liệu thị trường và hệ thống thông điệp. Xác định cách tích hợp các thành phần sử dụng mô hình kiến trúc dựa trên thông điệp và sử dụng các hệ thống messaging như JMS.
 - + Vẽ luồng giao dịch
- **Kiến trúc hệ thống với patterns**
 - + Kế thừa: server cung cấp data về giá thị trường, công cụ phân tích (Analytics Engine), Contribution Server

Nền tảng Java cung cấp sự tích hợp cho việc Remote Procedure Invocation và Messaging.

-> Tích hợp RPC-style đạt được bằng cách sử dụng Remote Method Invocation (RMI), CORBA hoặc Enterprise Java Beans (EJB).

Dịch vụ nhắn tin Java (JMS) là API để tích hợp Messaging-style

=> Việc messaging đơn giản hơn

Cầu nối từ máy chủ TIB đến máy chủ MQSeries yêu cầu giao tiếp giữa C++ và Java. Chúng ta có thể sử dụng CORBA, nhưng còn việc nhắn tin thì sao?

Một cái nhìn gần hơn về mô hình **Trình dịch tin nhắn (Message Translator)** cho thấy nó có liên quan đến **Channel Adapter** trong việc sử dụng các giao thức truyền thông.

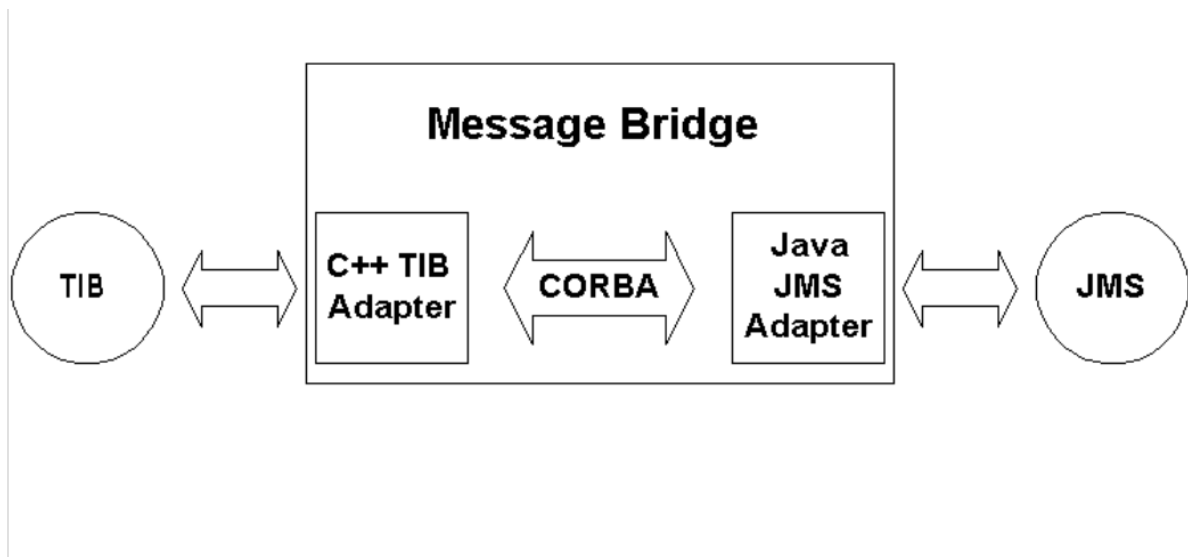
=> Trái tim của một **Channel Adapter** là kết nối các *hệ thống không nhắn tin* với *hệ thống nhắn tin*.

=> **Messaging Bridge** là một cặp Channel Adapter kết nối hai hệ thống nhắn tin.

Mục đích của một **Messaging Bridge** là chuyển tin nhắn từ hệ thống nhắn tin này sang hệ thống nhắn tin khác. Chúng ta có thể triển khai ngôn ngữ chéo **Messaging Bridge** sử dụng kết hợp **Channel Adapter** và CORBA.

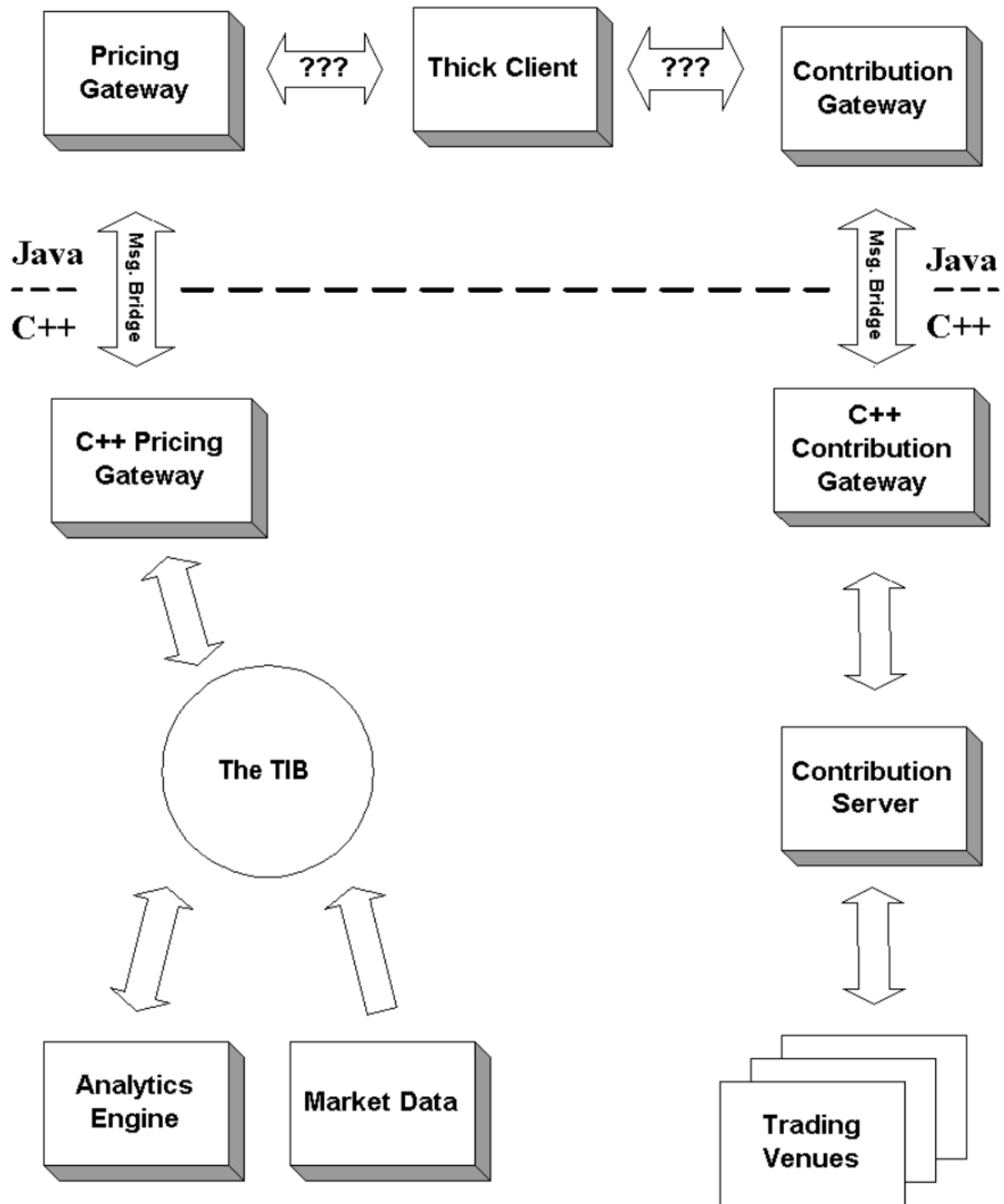
=> Chúng ta cần xây dựng hai server **Channel Adapter**, một server C++ quản lý giao tiếp với TIB và một server Java quản lý giao tiếp với JMS. Hai **Channel Adapter** này là **Message Endpoints**, chúng liên lạc với nhau thông qua CORBA.

Messaging Bridge triển khai dịch tin nhắn được mô phỏng hiệu quả giữa các hệ thống nhắn tin dường như không tương thích và các ngôn ngữ khác nhau.



Message Translator using Channel Adapters

Sơ đồ tiếp theo cho thấy thiết kế hệ thống hiện tại bao gồm Cổng và các thành phần khác. Đây là một ví dụ điển hình về ứng dụng mẫu. Chúng tôi đã kết hợp hai **Channel Adapter** với giao thức không nhắn tin để triển khai **Message Translator**, sử dụng hiệu quả một pattern để triển khai một mẫu khác. Ngoài ra, chúng tôi đã thay đổi **Channel Adapter** để liên kết hai hệ thống nhắn tin với giao thức dịch ngôn ngữ chéo không nhắn tin thay vì kết nối hệ thống nhắn tin với hệ thống không nhắn tin



The current system with the Channel Adapters

- Cấu trúc Channel

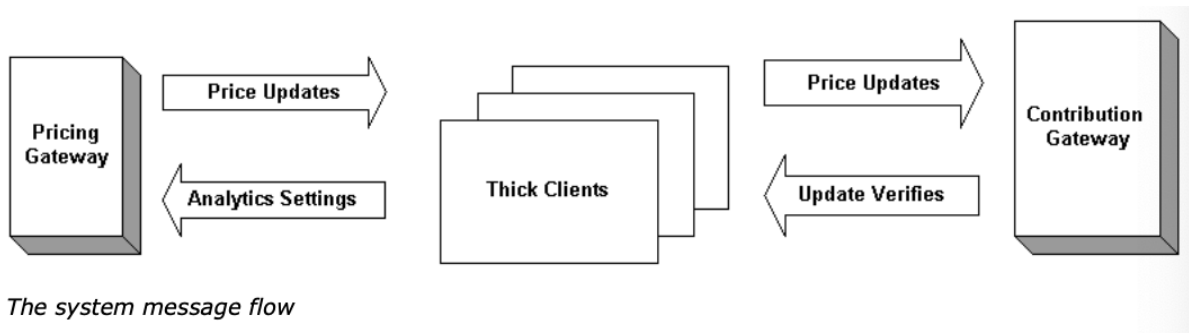
Dữ liệu thị trường theo thời gian thực bắt nguồn từ nguồn cấp dữ liệu thị trường, một máy chủ C++ phát dữ liệu thị trường trên TIB. Nguồn cấp dữ liệu thị trường sử dụng một nguồn cấp dữ liệu riêng [Publish-Subscribe Channel](#) đối với mỗi trái phiếu, đó là giá công bố. Điều này có vẻ hơi cực đoan vì mỗi trái phiếu mới cần có kênh mới riêng. Nhưng điều này không quá nghiêm trọng vì bạn thực sự không cần tạo kênh trong TIBCO.

Chúng ta có thể tạo một hệ thống xuất bản trên một số kênh và người đăng ký chỉ có thể nghe những mức giá mà họ quan tâm. Điều này sẽ yêu cầu người đăng ký sử dụng [Bộ lọc tin nhắn \(Message Filter\)](#) hoặc [Người tiêu dùng có chọn lọc \(Selective Consumer\)](#) để lọc toàn bộ luồng dữ liệu để tìm giá trái phiếu, quyết định xem mỗi thông báo có nên được xử lý khi nhận được hay không. Do dữ liệu thị trường được công bố trên các kênh dành riêng cho trái phiếu nên người đăng ký có thể đăng ký cập nhật hàng loạt trái phiếu. Điều này cho phép người đăng ký "lọc" một cách hiệu quả bằng cách đăng ký có chọn lọc các kênh và chỉ nhận các cập nhật được quan tâm thay vì quyết định sau khi nhận được tin nhắn. Điều quan trọng cần lưu ý là việc sử dụng nhiều kênh để tránh bị lọc là cách sử dụng kênh nhắn tin không chuẩn. Tuy nhiên, trong bối cảnh công nghệ TIBCO, chúng tôi thực sự đang quyết định nên triển khai hay sở hữu các bộ lọc hay sử dụng tính năng lọc kênh được tích hợp trong TIBCO -- thay vì có nên sử dụng nhiều kênh như vậy hay không.

Thành phần tiếp theo cần quan tâm là CÔNG CỤ PHÂN TÍCH

có 2 options: One Channel per Trader HOẶC One Channel per trader per Bond

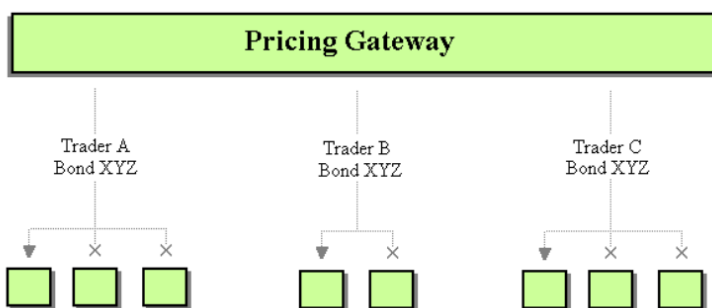
- Lựa chọn Message Channel



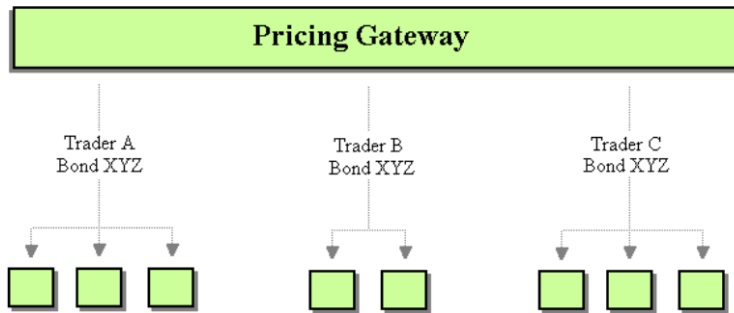
Chúng tôi có hai cổng (Pricing và Contribution) liên lạc với client.

Market Data truyền đến client từ Cổng Pricing và gửi dữ liệu đó đến Cổng Contribution. Ứng dụng khách (client app) gửi tin nhắn đến Cổng Pricing để thay đổi phân tích (analytics setting) được áp dụng cho từng trái phiếu. Cổng Contribution cũng gửi tin nhắn đến client app để chuyển tiếp trạng thái cập nhật giá (price updates) đến các địa điểm giao dịch khác nhau.

Đặc tả JMS mô tả 2 loại **Message Channel** (**Point-to-point Channel** + **Publish-Subscribe Channel**)



Point-to-Point Messaging for Price Updates



Publish-Subscribe Messaging for Price Updates

- Cập nhật FLASHING MARKET DATA