Designing with Patterns

Synopsis:

In this exercise, your task is to propose a design for the Messaging Infrastructure layer of the new Brokerage Information System (BIS) for the BizCo company. To accomplish this, you must select a pattern and instantiate it, modifying the architecture to meet the quality attribute requirements for the company.

BizCo Business Goals

To establish BizCo as the industry leader in brokering appraisal for commercial and residential properties. We must:

- Increase market share of brokerage services
- Dramatically decrease response time to changing market conditions
- Have direct, secure access to brokerage information 24/7
- Have reporting that aggregates, collates, and communicates timely information clearly and as needed

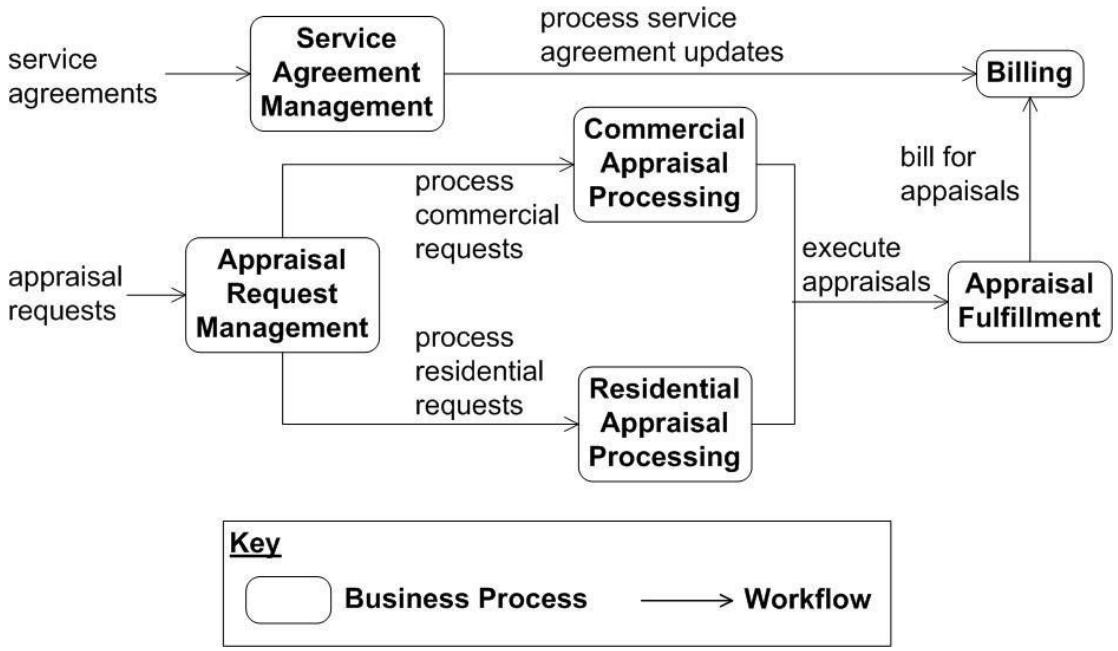
BizCo Business Processes

A real estate appraisal is needed to secure a mortgage loan when purchasing a property. A real estate appraisal is performed by licensed appraisers who work as independent contractors. As a real estate appraisal broker, BizCo connects the lender with an independent appraiser.

The typical workflow at a BizCo branch office begins with a Service Agreement between BizCo and lenders, and between Bizco and appraisers. The Service Agreement Management business process establishes and records the service agreements.

The Appraisal Request Management business process records appraisal requests and routes residential and commercial requests differently pairing commercial and residential appraisals with commercial and residential appraisers and routes the requests accordingly.

Once the property has been appraised, the Appraisal Fulfilment process records the appraisal results, and notifies the Billing process to bill the lender and pay the appraiser according to their service agreement.



BizCo Information Systems

Three standalone systems operate at each BizCo branch. Please note that the systems are not integrated within the branch and do not integrate externally.

Table 2 - BizCo Existing Information Systems per Branch

BizCo Branch Systems	Description
Commercial Property System (CPS)	Services requests for data about appraisal requests submitted, assigned, and fulfilled for commercial properties including request details, lender/appraiser assignments, and status.
Residential Property System (RPS)	Services requests for data about appraisal requests submitted, assigned, and fulfilled for residential properties including request details, lender/appraiser assignments, and status.

Brokerage Billing System (BBS)	Services requests for data about branch office, lender, and appraiser business addresses and contacts, contract terms and conditions, and accounts receivable.

Table 3. X indicates system support for a business process

Business Process	CPS	RPS	BBS
Appraisal Request Management	**X**	**X**	
Commercial Appraisal Processing	**X**		
Residential Appraisal Processing		**X**	
Appraisal Fulfillment	**X**	**X**	
Billing			**X**
Service Agreement Management			**X**

BizCo Information Reporting

CPS, RPS, and BBS at each branch generate reports independently

- Summarize brokerage activities
- Biweekly
- Hard-copy
- Sent to main office in Pittsburgh via courier

BizCo Brokerage Information System (BIS) – the Proposed System

BizCo needs to gather more extensive information from the branch offices' existing systems in a timely manner. To this end, your organization plans to develop the new **Brokerage Information System (BIS).**

The BIS must allow the main office to display information on:

- Branch office, lender, and appraiser business address and contacts
- Contract terms and conditions for lenders and appraisers
- Appraisal requests submitted, assigned, and fulfilled
- Brokerage activities across and for individual branch offices
- Brokerage activities across and for individual lenders, appraisers, and brokers

- Account receivables aggregated among all offices

BizCo BIS Software Architecture

The BizCo systems architect has designed a Layered Architecture, consisting of four layers, as a module structure for the new system, encapsulating the three existing systems, CPS, RPS, and BBS.

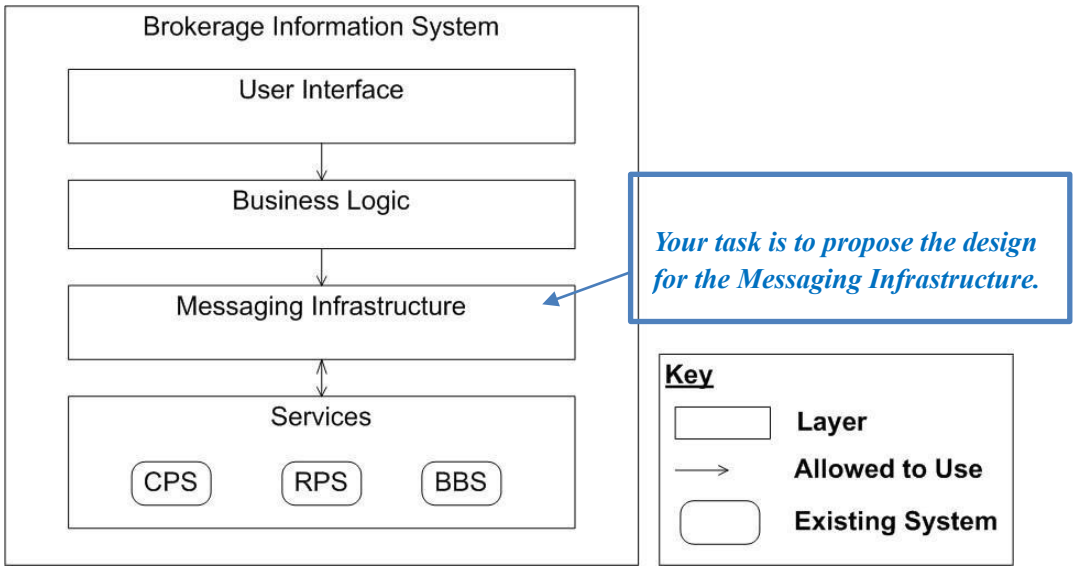


Figure 2 BIS Architecture

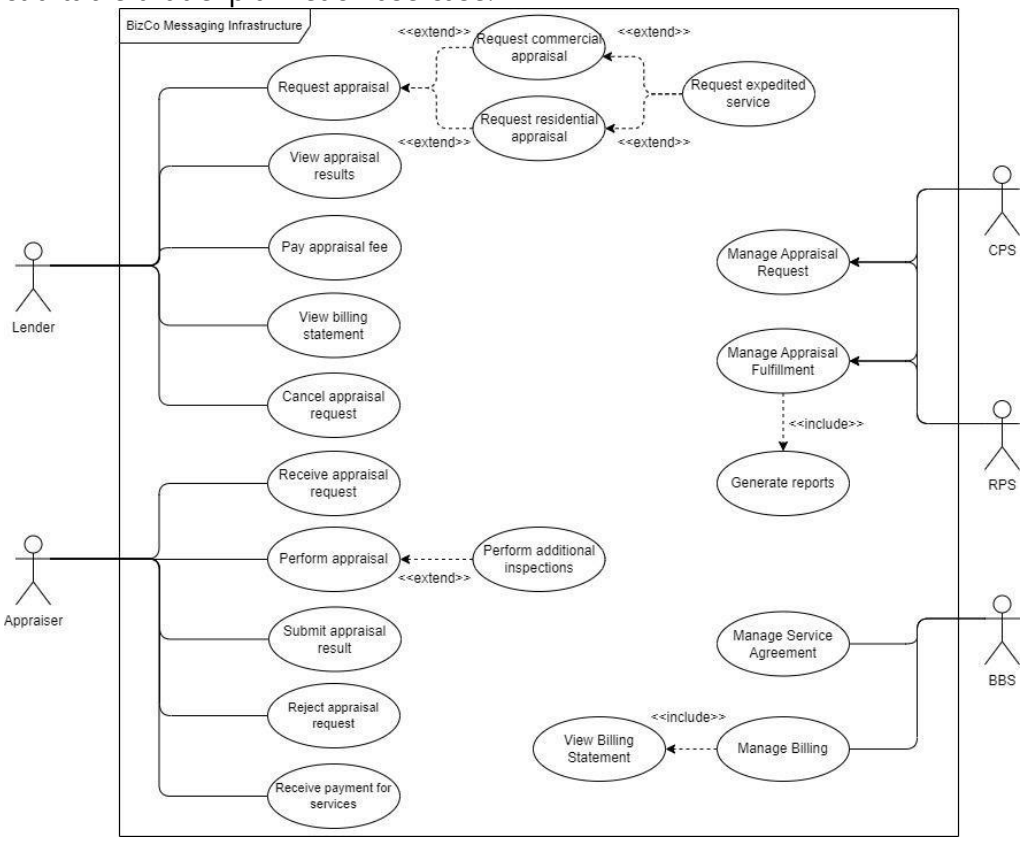
Table 4 BIS Architecture Layers

Layer	Description
User Interface	Allows users to select, customize, and display brokerage data and activity reports.
Business Logic	Interacts with remote services to collect data and collates the data into brokerage reports.
Message Infrastructure *Your design task.*	Provides access to remote services.
Services	A collection of services that provide brokerage data. CPS,

Do the Following:

A. Use case Diagram

From your understand of the system and assumption, draw use case diagram and construct a table that explain each use case.



This use case does not include Authentication to simplify it.

Use Case	Description	Actor	Input	Output
Request Appraisal	Lender requests appraisal for a property	Lender	Property information, appraisal requirements	Appraisal request confirmation
Request Residential Appraisal	Lender requests residential appraisal for a property	Lender	Property information, residential appraisal requirements	Residential appraisal request confirmation
Request Commercial Appraisal	Lender requests commercial appraisal for a property	Lender	Property information, commercial appraisal requirements	Commercial appraisal request confirmation

Request Expedited Appraisal	Lender requests expedited appraisal for the property	Lender	Property information, expedited appraisal requirements	Expedited appraisal request confirmation
View Appraisal Results	Lender views appraisal results for a property	Lender	Property information, appraisal details	Appraisal results
Pay Appraisal Fee	Lender pays appraisal fee for a property	Lender	Appraisal fee, payment information	Confirmation of payment
View Billing Statement	Lender views billing statement for appraisal services	Lender	Billing statement request	Billing statement
Receive Appraisal Request	Appraiser receives appraisal request from BizCo	Appraiser	Appraisal request	Acknowledgement of receipt
Perform Appraisal	Appraiser performs appraisal for a property	Appraiser	Property details, appraisal requirements	Appraisal report
Submit Appraisal Result	Appraiser submits appraisal report to BizCo	Appraiser	Appraisal report	Confirmation of submission
Receive Payment for Services	Appraiser receives payment for appraisal services rendered	Appraiser	Appraisal fee, payment information	Confirmation of payment
Manage Service Agreement	BBS manages service agreements with lenders and appraisers	BBS	Service agreement details	Confirmation of service agreement
Manage Appraisal Request	CPS and RPS manages appraisal	CPS & RPS	Appraisal request details	Confirmation of appraisal request
Use Case	**Description**	**Actor**	**Input**	**Output**
	requests from lenders to appraisers			
Manage Appraisal Fulfilment	CPS and RPS manages appraisal fulfilment process and appraisal results	CPS & RPS	Appraisal report details, payment information	Confirmation of appraisal fulfilment
Manage Billing	BBS manages billing for appraisal services rendered	BBS	Billing details, payment information	Confirmation of billing

Generate Reports	CPS and RPS generates reports on appraisal requests, fulfilment, and billing	CPS & RPS	Report request	Report output

B. Identify the quality attribute requirements.

1) Create a utility tree for the current system. Consider a minimum of **four** different quality attributes. Ensure that **the goals and scenarios** that you elucidated at the leaf nodes have explicit responses and response measures.

Quality Attribute	**Attribute Refinement**	**ASR**
Availability	System accessibility	Users will be notified of system maintenance in advance and it will be scheduled during non-peak hours. (H.H)
Performance	Response time	The system can complete transactions in 2 seconds to help BizCo respond faster to market changes. (M.H)
Scalability	Flexibility	The system should be able to handle varying loads and traffic patterns, such as seasonal spikes or sudden increases in usage. (M, H)
Security	Authentication	user is required to enter a username and password, as well as a one-time code sent to his phone or email for certain transactions (M.M)

Usability	Ease of Use	System training should take no longer than 1 day. (L, L)

2) Using the following table, write down what you think are the most important quality attributes for the Messaging Infrastructure layer and prioritize their relative importance on a scale from 1 to n, where 1 is most important quality attribute requirement for the messaging infrastructure and n is the least important. Then you should justify your choice for each quality attribute.

*Hint: Review the **business goals**!*

RELATIVE IMPORTANCE	QUALITY ATTRIBUTE	JUSTIFICATION (SOURCE)
1	Scalability	BizCo's messaging system must be able to handle large volumes of messages quickly and reliably.
2	Performance	Messaging infrastructure must efficiently manage traffic to ensure timely delivery, especially during peak hours.
3	Availability	To ensure that the system can provide direct access to information 24/7 .

4	Security	To ensure that the system is protected from unauthorized access and data breaches .
5	Usability	Messaging infrastructure should be user-friendly to facilitate efficient communication .

C. **Design the Messaging Infrastructure Layer**

Now you can begin designing the Messaging Infrastructure layer using one of three patterns: **Messaging, Publisher-Subscriber, or SOA**. In this step you must first consider the patterns, weighing the pros and cons of each, and the tradeoffs from a *quality attribute perspective*.

Read about and consider the Messaging, Publisher-Subscriber, and SOA patterns.

The Messaging Pattern

The Messaging Pattern	
Context	Some distributed systems are composed of services that were developed independently. To form a coherent system, however, these services must interact reliably, but without incurring overly tight dependencies on one another.
Problem	Integrating independently developed services, each having its own business logic and value, into a coherent application requires reliable collaboration between services. However, since services are developed independently, they are generally unaware of each other's specific functional interfaces. Furthermore, each service may participate in multiple integration contexts, so using them in a specific context should not preclude their use in other contexts.

Solution	Connect the services via a message bus that allows them to transfer data messages asynchronously. Encode the messages (request data and data types) so that senders and receivers can communicate reliably without having to know all the data type information statically.

Messaging Pattern

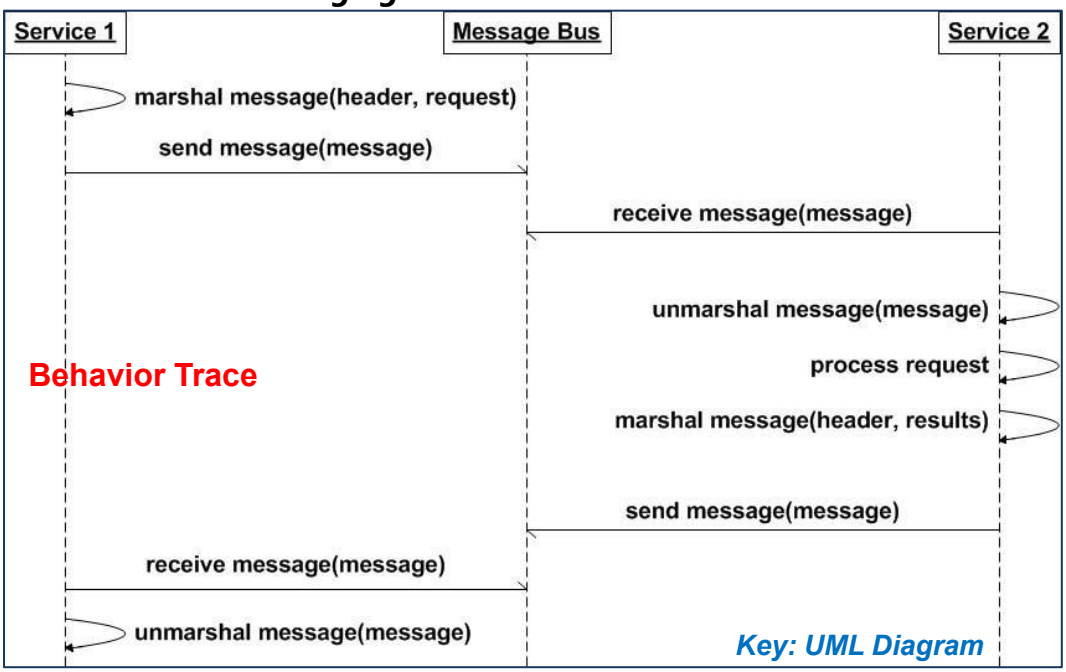


Figure 3 Messaging Pattern, Sequence Diagram of Services Interactions

Messaging Pattern Benefits	Messaging Pattern Liabilities
Services can interact without having to deal with networking and service location concerns.	Lack of statically typed interfaces makes it hard to validate system behavior prior to runtime.
Asynchronous messaging allows services to handle multiple requests simultaneously without blocking.	Service requests are encapsulated within self-describing messages that require extra time and space for message processing.

Allows services to participate in multiple application integration and usage contexts.	

The Publisher-Subscriber Pattern

Publisher-Subscriber Pattern	
Context	Components in some distributed applications are loosely coupled and operate largely independently. If such applications need to propagate information to some or all of their components, a notification mechanism is needed to inform the components about state changes or events that affect or coordinate their own computation.
Problem	The notification mechanism should not couple application components too tightly, or they will lose their independence. Components want to know only that another component is in a specific state, not which specific component is involved. Components that disseminate events often do not care which other components want to receive the information. Components should not depend on how other components can be reached or on their specific location in the system.
Solution	Define a change propagation infrastructure that allows publishers in a distributed application to disseminate events that may be of interest to others. Notify subscribers interested in those events whenever such information is published.
Publisher-Subscriber Pattern	

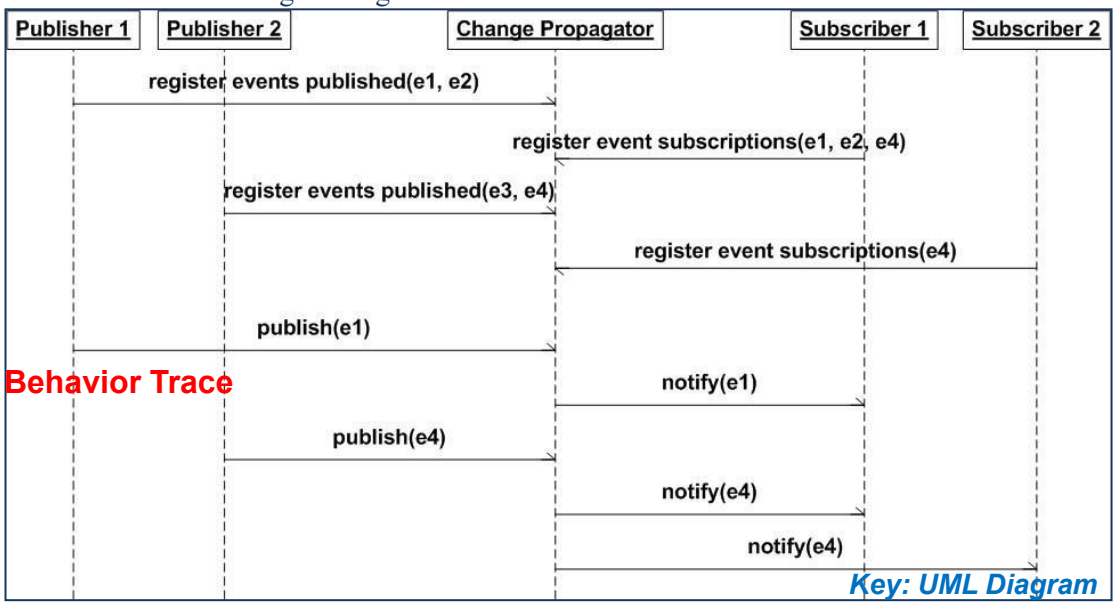


Figure 4 Publisher-Subscriber Pattern, sequence diagram of services interactions

Publisher-Subscriber Pattern Benefits	Publisher-Subscriber Pattern Liabilities
Publishers can asynchronously transmit events to Subscribers without blocking.	Publishing can cause unnecessary overhead if Subscribers are interested in only a specific type of event.
Asynchronous communication decouples Publishers from Subscribers, allowing them to be active and available at different times.	Filtering events to decrease event publishing and notification overhead can result in other costs (e.g., decrease in throughput, unnecessary notifications, breakdown of anonymous communication model).

Publishers and Subscribers are unaware of each other's location and identity.	

Dynamic Routing Pattern (SOA)

Dynamic Routing Pattern (SOA)	
Context	It is often necessary to build complex business processes by wiring together a set of relatively simple services in a dynamic way.
Problem	Routing messages through a distributed system based on filtering rules is inefficient because messages are sent to every destination's filter and router for inspection and rules resolution, whether or not the message could be processed.
Solution	Define a message router that includes both filtering rules and knowledge about the processing destination paths so that messages are delivered only to the processing endpoints that can act on them. Unlike filters, message routers do not modify the message content and are concerned only with message destination.

Dynamic Routing Pattern (SOA)

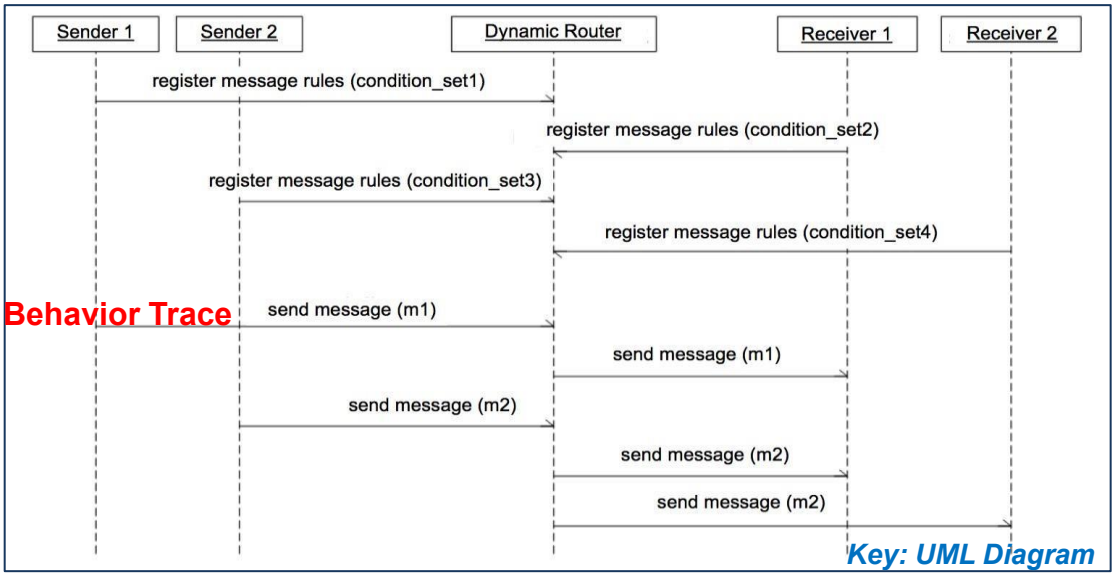


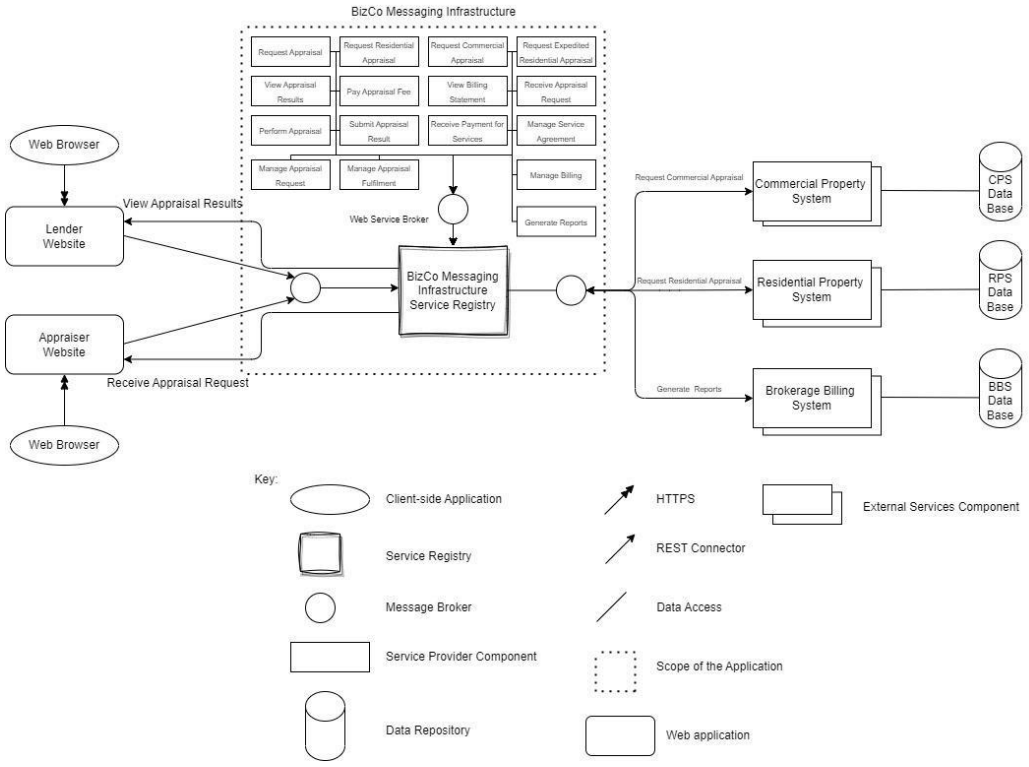
Figure 5 Dynamic Routing Pattern, Sequence diagram for services interactions

Dynamic Routing Pattern Benefits	Dynamic Routing Pattern Liabilities
Services do not need to deal with networking concerns or know each other's locations since all requests are handled through the message router.	Performance overhead.
Communications can be optimized as services dynamically become available or unavailable.	Single point of failure.
Efficient, predictive routing.	Potentially complex, unintuitive behavior when rules conflict.

Do the following:

1) Using the above description, select and draw the pattern that you think will best meet the messaging infrastructure requirements. Instantiate the pattern by describing the roles of its constituent participants, their responsibilities and relationships, and the ways in which they collaborate.

Service-Oriented Architecture (SOA) Pattern



Roles:

- Service Provider Component: This component provides a service that can be consumed by other components or systems. It exposes the service through a standardized interface and implements the necessary logic to execute the service.
- Service Consumer Component: This component consumes the service provided by the Service Provider Component. It sends requests to the service and receives responses back.
- Service Registry: Manages the registration of available services and allows Service Consumers to discover and locate services they need.
- Message Broker: Provides a means of routing messages between Service Providers and Service Consumers. Serves as the intermediary between them.

Responsibilities and Relationships:

- Service Provider: Implements and exposes a service through a standardized interface. Registers the service with the Service Registry.
- Service Consumer: Discovers and locates the service it needs from the Service Registry. Sends requests to the Service Provider and receives responses back.
- Service Registry: Maintains a registry of available services and their interface descriptions. Allows Service Consumers to discover and locate the services they need.
- Message Broker: Receives requests from Service Consumers and routes them to the appropriate Service Provider. Receives responses from Service Providers and routes them back to the Service Consumer.

Collaborate:

- The Service Provider accesses any necessary data sources to retrieve or update data needed to fulfill the request from the Service Consumer.
- If the service provided by the Service Provider requires data analytics and reporting capabilities, it may access a separate analytics component to provide those capabilities.
- The Service Provider updates any necessary data sources with any changes resulting from the service's execution.
- The Service Consumer may also retrieve data from necessary data sources via the Service Provider to fulfill its request.
-

2) Fill out the table and identify the tradeoffs using this pattern.

Pattern	Service-Oriented Architecture (SOA)
Overview	The Service-Oriented Architecture pattern enables components to communicate by providing and consuming services in a decoupled manner.
Elements	• Service Provider Component: This component provides a service that can be consumed by other components or systems. It exposes the service through a standardized interface and implements the necessary logic to execute the service. • Service Consumer Component: This component consumes the service provided by the Service Provider Component. It sends requests to the service and receives responses back. • Service Registry: Manages the registration of available services and allows Service Consumers to discover and locate services they need. • Message Broker: Provides a means of routing messages between Service Providers and Service Consumers. Serves as the intermediary between them.

Relations	The Service Provider exposes a service and registers it with the Service Registry. • The Service Consumer discovers and locates the service it needs from the Service Registry. • The Service Consumer sends a request to the Message Broker. • The Message Broker routes the request to the appropriate Service Provider. • The Service Provider executes the service logic and sends a response back to the Message Broker. • The Message Broker routes the response back to the Service Consumer.
Constraints	• • Service Providers and Consumers must use the same set of standardized interfaces. The Service Registry should be able to handle large volumes of services efficiently.
Tradeoffs	**Pros:** **Cons:** • Loose coupling between Service Providers and Consumers. • Flexibility in adding new Services without affecting existing components. • Scalability due to the distributed nature of the Message Broker. • Increased complexity due to the introduction of standardized interfaces and message passing. • Potential for message delivery failures or delays. • Increased overhead due to additional messaging and service processing logic.

D. Applying Tactics

1) **Based on your design decision for the BIzCo BIS, what quality would you like to improve?**

Availability and Performance

Quality Attribute	Text Reference[i]
Availability	p. 87-95
Interoperability	p. 110-112

2) **What tactics would you choose to improve your design of the messaging infrastructure for the system?**

Modifiability	p. 121-125
Performance	p. 135-141
Security	p. 150-154
Testability	p. 164-168
Usability	p. 177-181

Availability:
Active redundancy
Passive redundancy

Performance:
Prioritize events Caching

3) **Consider the tradeoffs. What are the issues associated with the selection of those tactics?**

Availability:

Active redundancy:
Tradeoff: Increased complexity.
Issue: Implementing and managing active redundancy requires additional resources.

Passive redundancy:
Tradeoff: Longer recovery time compared to active redundancy since the spare nodes need to be brought online.
Issue: Passive redundancy requires the activation of spare nodes in case of failure, which may result in a temporary disruption of service until the spare nodes are fully operational.

Performance:

Prioritize events:
Tradeoff: Lower-priority events may experience delays or have longer processing times compared to higher-priority events.
Issue: Depending on the prioritization strategy, there is a risk of delaying less critical events, which may impact the overall responsiveness of the system.

Caching:
Tradeoff: Increased complexity of managing the cache and the potential for cache inconsistency if the underlying data changes.
Issue: Caching requires careful management to ensure the cache remains consistent and up to date with the underlying data. Inconsistent caching can lead to incorrect or outdated results being served to consumers.