Presentation on TIB / Rendezvous

Group Members

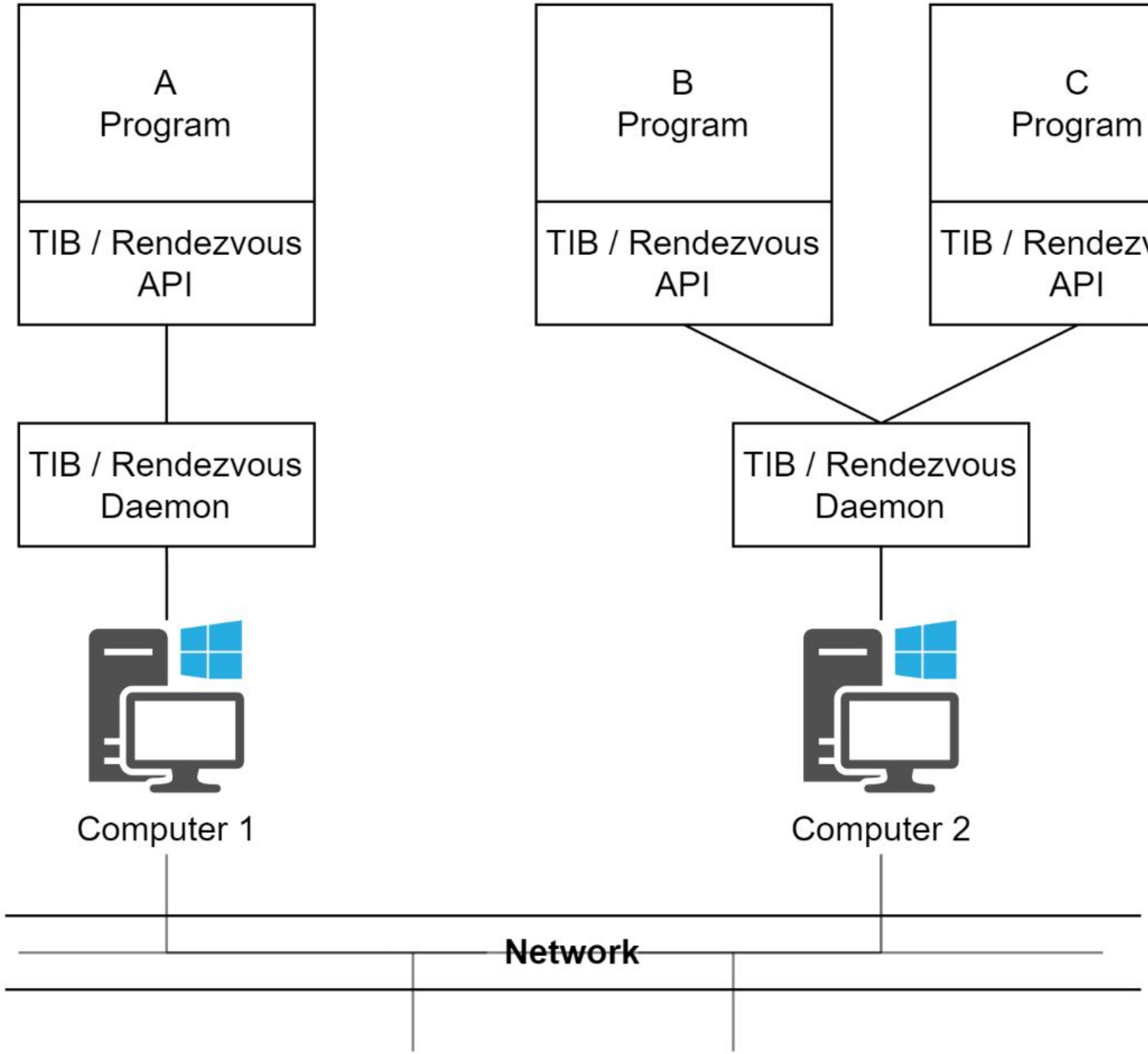
Prajesh Shrestha	THA076BCT028
Pratigya Paudel	THA076BCT029
Sushank Ghimire	THA076BCT047
Ujjwal Paudel	THA076BCT048

Department of Electronics and Computer Engineering
Institute of Engineering, Thapathali Campus

August, 2023

Introduction

- Messaging Middleware by TIBCO Software Inc.
- Enables seamless cross-enterprise application connections.
- Publish-Subscribe messaging for real-time info exchange.
- Enhances scalability and adaptability.
- Patented tech: Publish/Subscribe, Subject-Based Addressing, Self-Describing Data Messages.
- Supports hardware and software platform like TIB/Rendezvous API, TIB Rendezvous daemon.



Components of TIB

TIB / Daemon

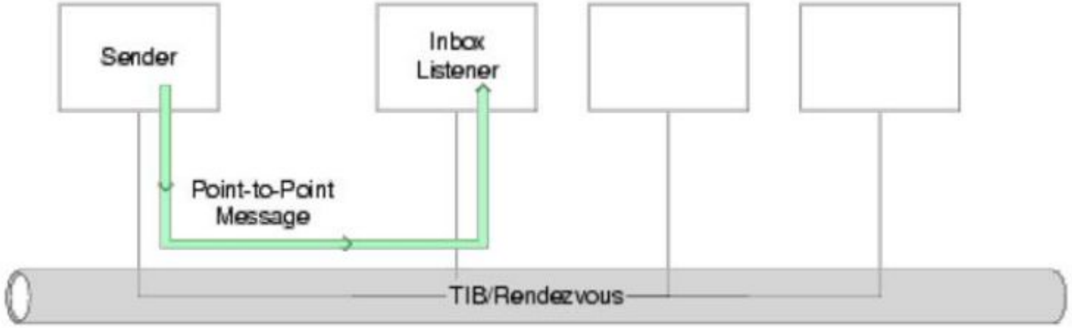
- Background process providing TIB/Rendezvous functionality.
- Message Routing:
  - Delivers messages based on subjects, maintains subject-subscriber directory.
  - Transmits outbound message from program processes to the network
  - Delivers inbound message from the network to program processes
- Load Balancing: Distributes message traffic for fault tolerance and load sharing in multi-daemon setups.
- Fault Tolerance: Multiple daemons collaborate, ensuring continuous message delivery if one fails.
- Security: Ensures message confidentiality and integrity through authentication and encryption.

TIB / Rendezvous Language Interface(TLI)

- Set of APIs for app interaction.
- Available in C, C++, Java, and .NET.
- Allows the application to perform different tasks:
  - Create/Manage Message Queues: Handle reliable message storage and delivery.
  - Send/Receive Messages: Seamless communication and data exchange.
  - Subscribe/Unsubscribe: Filter incoming messages by topics/subjects.
  - Control Message Flow: Manage traffic with rate limiting and prioritization.
  - Monitor System Health: Access metrics, stats, and diagnostics.
- Implement Async Communication: Continue processing while waiting for messages.

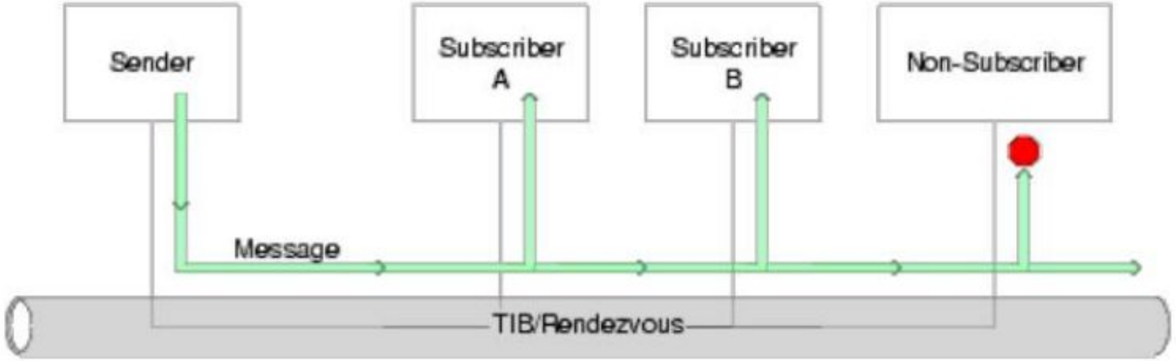
Messaging Patterns : Point-to-Point

- **Direct Communication:** Point-to-point pattern ensures direct transmission of messages between sender and recipient.
- **Dedicated Channel:** Establishes a dedicated communication channel for one-to-one message delivery.
- **Targeted Information:** Ideal for scenarios requiring specific information delivery to a single recipient.
- **TIBCO Rendezvous:** Enables clients to send messages directly, ensuring precise and focused communication.

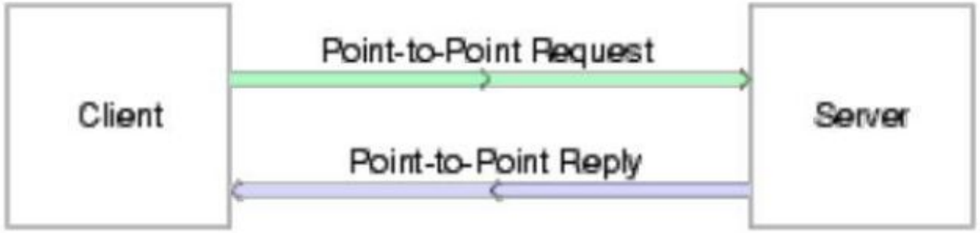


Messaging Patterns : Multicast Messaging

- **Broadcast to Subscribers:** Multicast sends messages to multiple subscribers interested in a specific subject.
- **Simultaneous Delivery:** All subscribers for the subject receive the message at the same time.
- **Efficient Group Communication:** Ideal for targeted information sharing without individualized messages.
- **Use Cases:** Effective for market data distribution and real-time updates in distributed systems, akin to radio broadcasts.

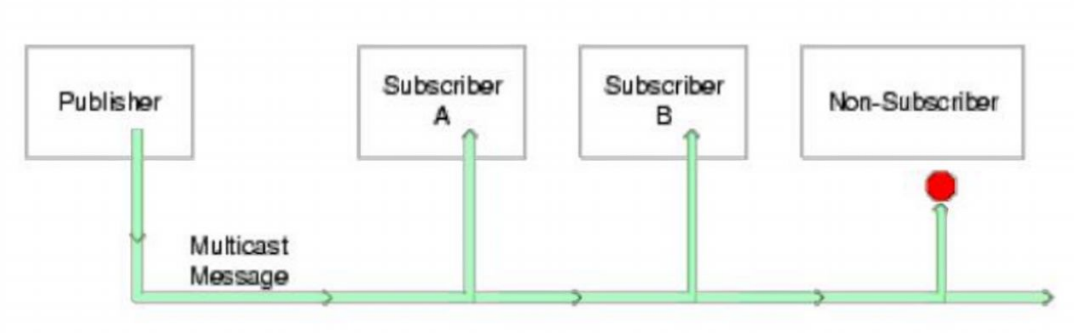


Messaging Interactions: Request-Reply Interactions



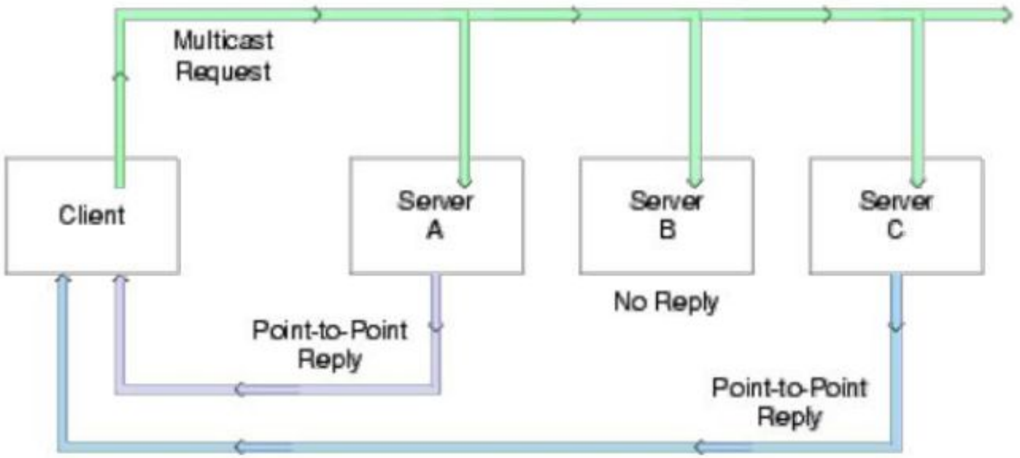
- **Data Coordination:** Request/reply involves close coordination between data producers and consumers.
- **On-Demand Data:** Producers wait for consumer requests before sending data.
- **Inbox Mechanism:** Each program uses specific inbox names for messaging.
- **Targeted Replies:** Server sends replies specifically to the requesting client.
- **Point-to-Point:** TIBCO Rendezvous supports request/reply using point-to-point communication.

Messaging Interactions: Publish-Subscribe Interactions



- **Decoupled Producers and Consumers:** Publish/subscribe decouples data producers from consumers.
- **Subject-Based Coordination:** Producers use subject names for coordination.
- **Standing Requests:** Consumers place standing data requests by subscribing.
- **Flexible Listening:** Consumers can listen for messages on any subject(s) in the network.
- **TIB/Rendezvous Support:** Publish/subscribe is supported by TIBCO Rendezvous, using multicast communication.

Messaging Interactions: Multicast Request / Reply Interactions



- **Multiple Servers, Single Client:** Multicast request/reply allows a client to send a request to multiple servers.
- **Receive Replies:** Client gets a reply message from one server with requested information.
- **Listening for Replies:** The requesting client listens until it receives one or more replies.
- **Stop Listening:** Client stops listening by destroying the reply inbox listener.
- **TIB/Rendezvous Support:** Multicast request/reply is supported by TIBCO Rendezvous, using a combination of multicast and point-to-point communication.

Features of TIBCO Rendezvous

General Characteristics

- **Distributed Queues:** Efficient one-to-many transmission through distributed queues.
- **Secure Transmission:** Encrypted and secure data transfer for confidentiality.
- **Redundancy for Reliability:** Redundancy mechanism for fault tolerance.
- **Cross-Platform Peer-to-Peer:** Peer-to-peer transfer across different platforms.

Communication and Data Characteristics

- **Asynchronous Communication:** Independent messaging, not bound by sender or receiver timing.
- **Publish/Subscribe & Broadcast/Multicast:** Efficient distribution with reliable mechanisms.
- **Peer-to-Peer Interaction:** Support for interactive request/response messaging.
- **Topic-Based Messaging:** Categorized and distributed messages based on subjects.
- **Hardware/OS Agnostic:** Supports custom data across diverse environments.
- **Transparent Data Handling:** Ensures intact and accurate delivery of messages.

Authentication Information Transfer

- **Secure Information Authentication:** Ensures secure transmission to the intended destination.
- **Data Loss Prevention:** Maintains communication integrity during interruptions or restarts.
- **Distributed Queues with Load Balancing:** Even workload distribution for optimal performance.
- **Targeted Data Distribution:** Efficient data distribution to members of specific queues.
- **Asynchronous Queue Processes:** Improves system responsiveness and efficiency.

Fault Tolerance

- **Redundant Processes:** Enhances fault tolerance for continuous operation.
- **Active Monitoring:** Swiftly identifies failures and maintains seamless communication.

Development Features

- **Diverse API Libraries:** Supports Java, C, C++, ActiveX, .NET, and Perl, catering to various developers.
- **Source Code Compatibility:** Consistent source code across platforms for ease of development.
- **Flexible Event Management:** Supports synchronous and asynchronous structures for tailored communication.

Thank You