Paulo Gandra de Sousa
psousa@dei.isep.ipp.pt

Mestrado em Engenharia Informática

Programação de Sistemas Distribuidos

Disclaimer

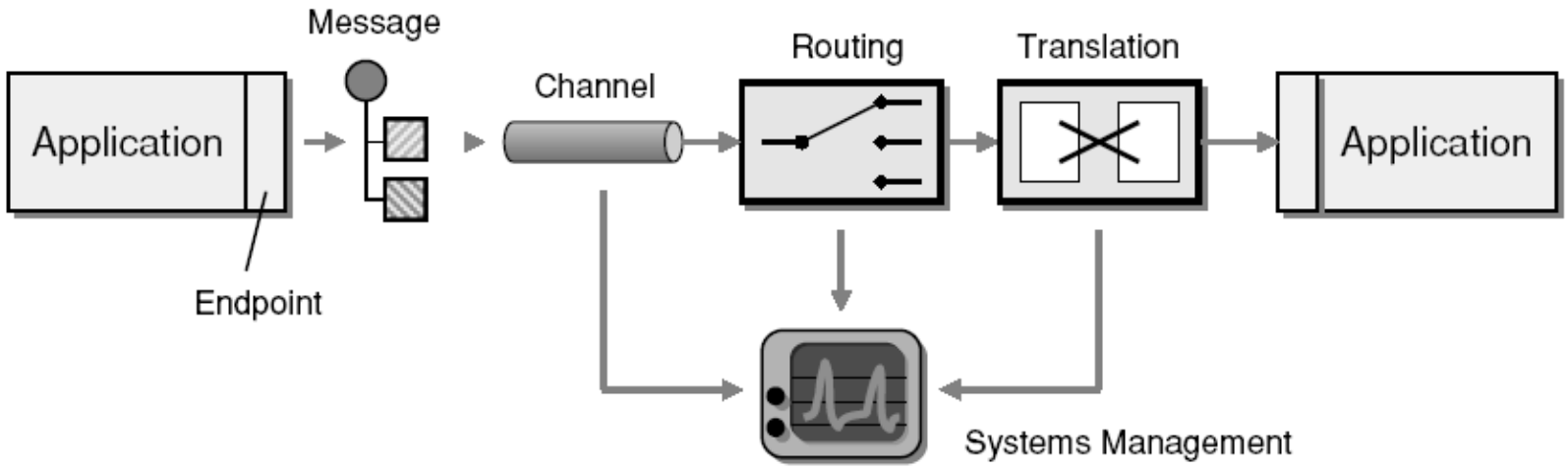
- Parts of this presentation are from:
  - Miguel Losa (INTS)
- The content of this presentation is exclusively from the book:
  - *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*
  - Gregor Hohpe, Bobby Woolf
  - www.eaipatterns.com

Today's lesson

- Introduction to Enterprise Integration Patterns

Integration Patterns

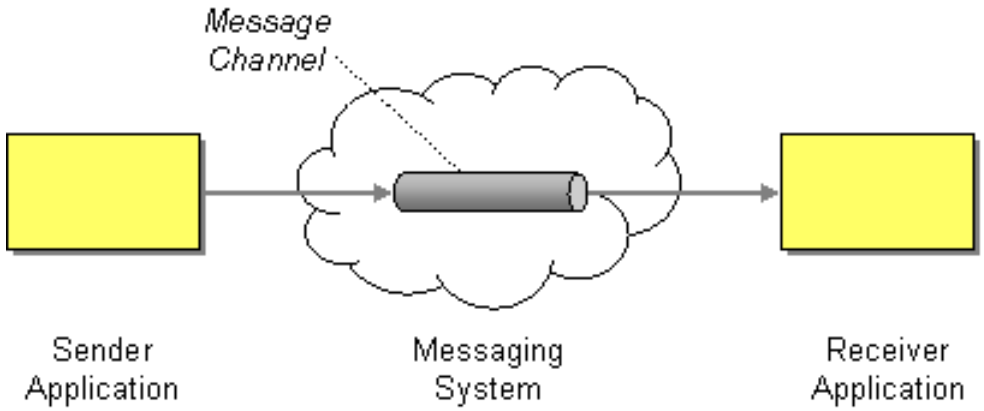
Loosely coupled template



Basic Elements of Message-Based Integration

Message Channel

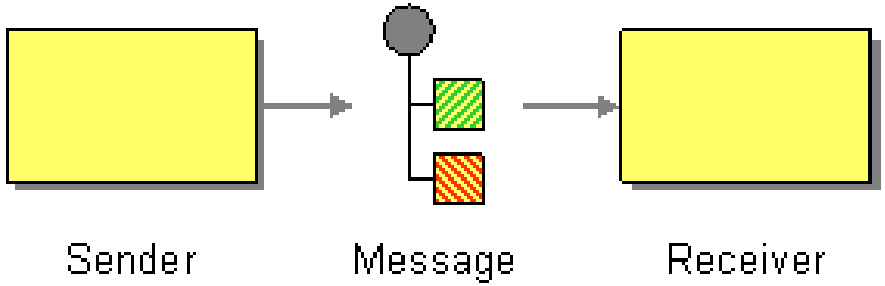
- How does one application communicate with another using messaging?



- Connect the applications using a Message Channel, where one application writes information to the channel and the other one reads that information from the channel.

Message

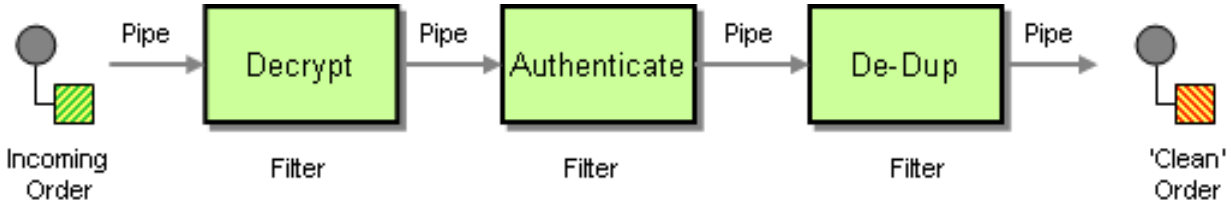
- How can two applications connected by a message channel exchange a piece of information?



- Package the information into a Message, a data record that the messaging system can transmit through a message channel.

Pipes and Filters

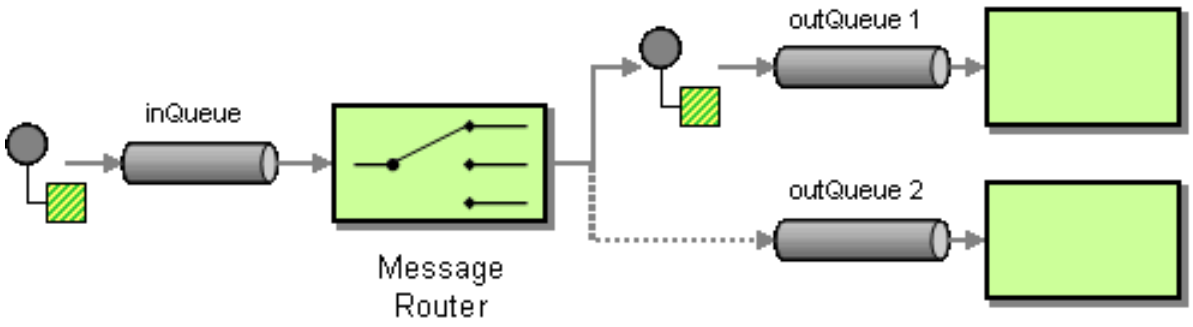
- How can we perform complex processing on a message while maintaining independence and flexibility?



- Use the Pipes and Filters architectural style to divide a larger processing task into a sequence of smaller, independent processing steps (Filters) that are connected by channels (Pipes).

Message Router

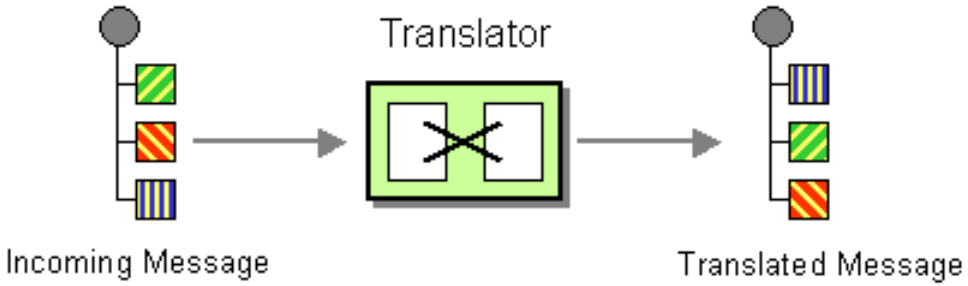
- How can you decouple individual processing steps so that messages can be passed to different filters depending on a set of conditions?



- Insert a special filter, a Message Router, which consumes a Message from one Message Channel and republishes it to a different Message Channel depending on a set of conditions.

Message Translator

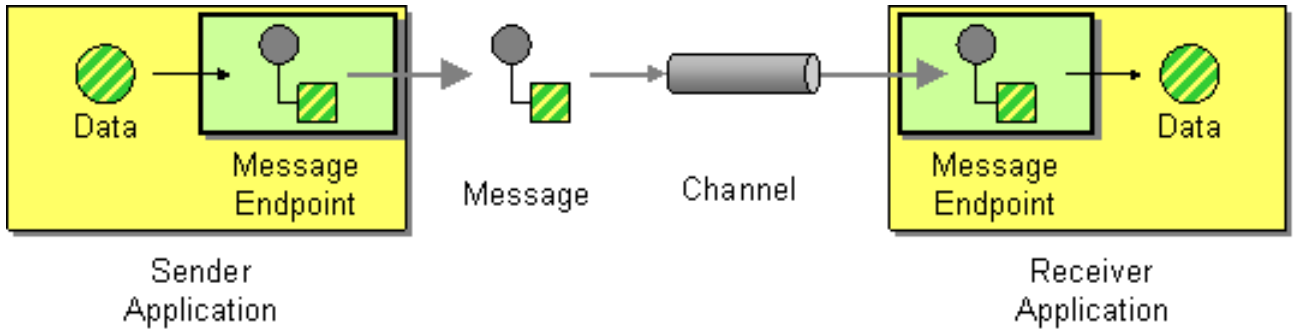
- How can systems using different data formats communicate with each other using messaging?



- Use a special filter, a Message Translator, between other filters or applications to translate one data format into another.

Message Endpoint

- How does an application connect to a messaging channel to send and receive messages?



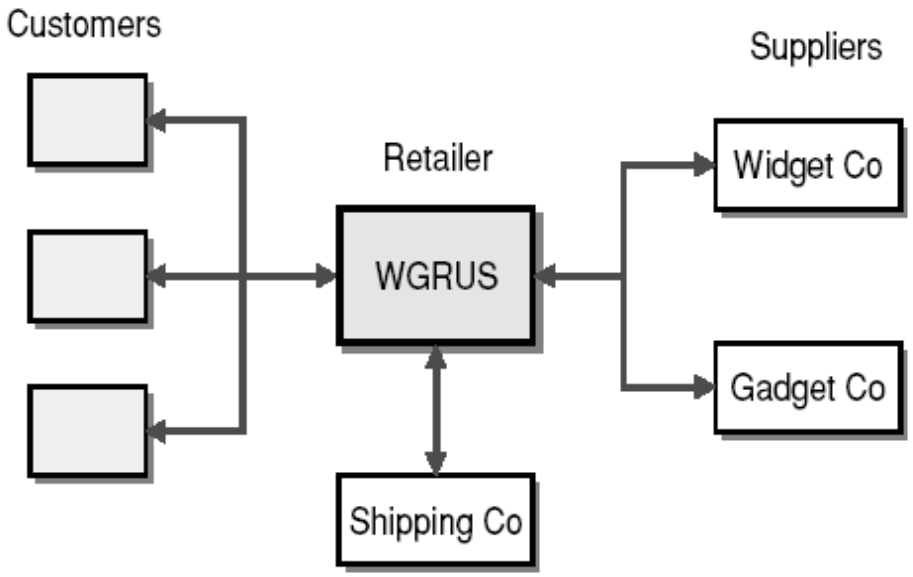
- Connect an application to a messaging channel using a Message Endpoint, a client of the messaging system that the application can then use to send or receive messages.

Sample Application

Widgets & Gadgets 'r Us

Sample application

- Widgets & Gadgets R' US



WGRUS Ecosystem

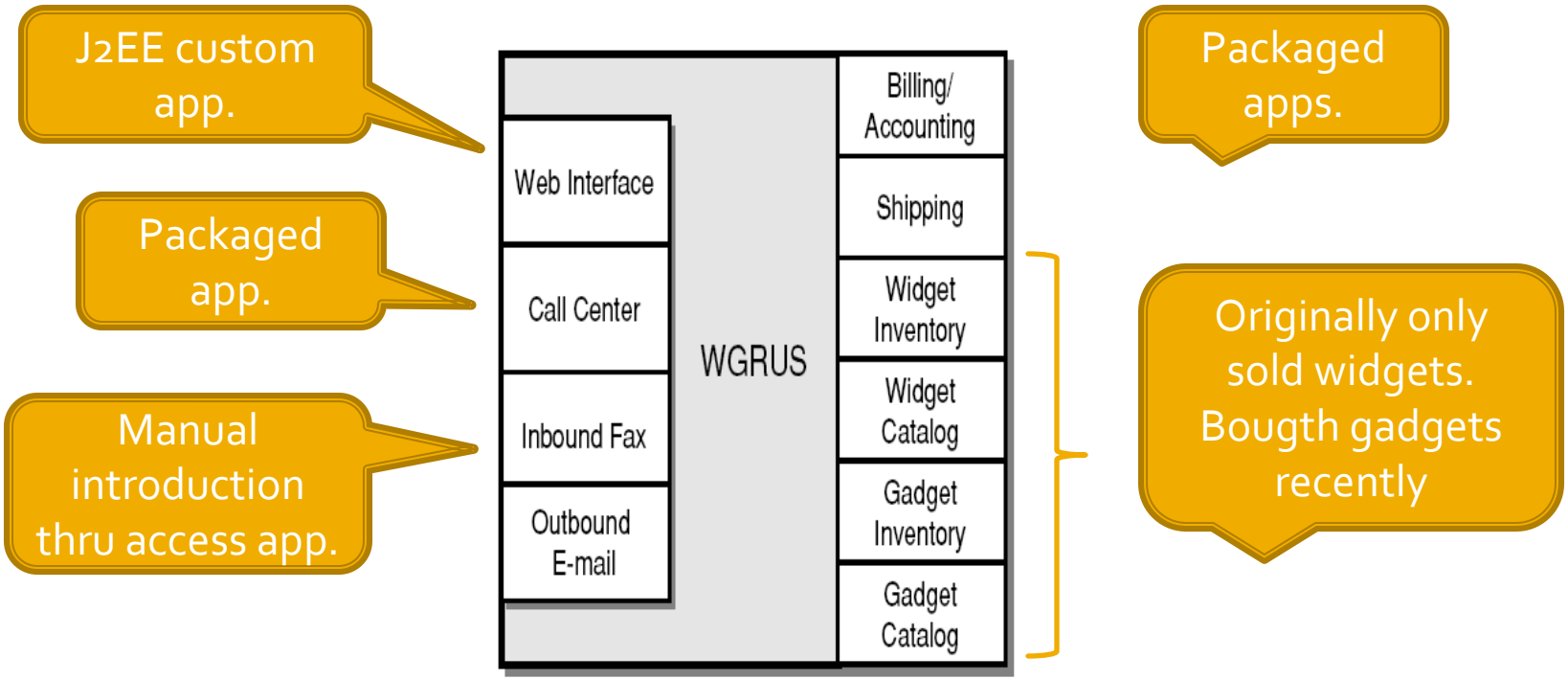
WGRUS
Business processes

- Take Orders
  - Customers can place orders via Web, phone or fax
- Process Orders
  - Processing an order involves multiple steps, including verifying inventory, shipping the goods and invoicing the customer
- Check Status
  - Customers can check the order status
- Change Address
  - Customers can use a Web front-end to change their billing and shipping address

WGRUS
Business processes (2)

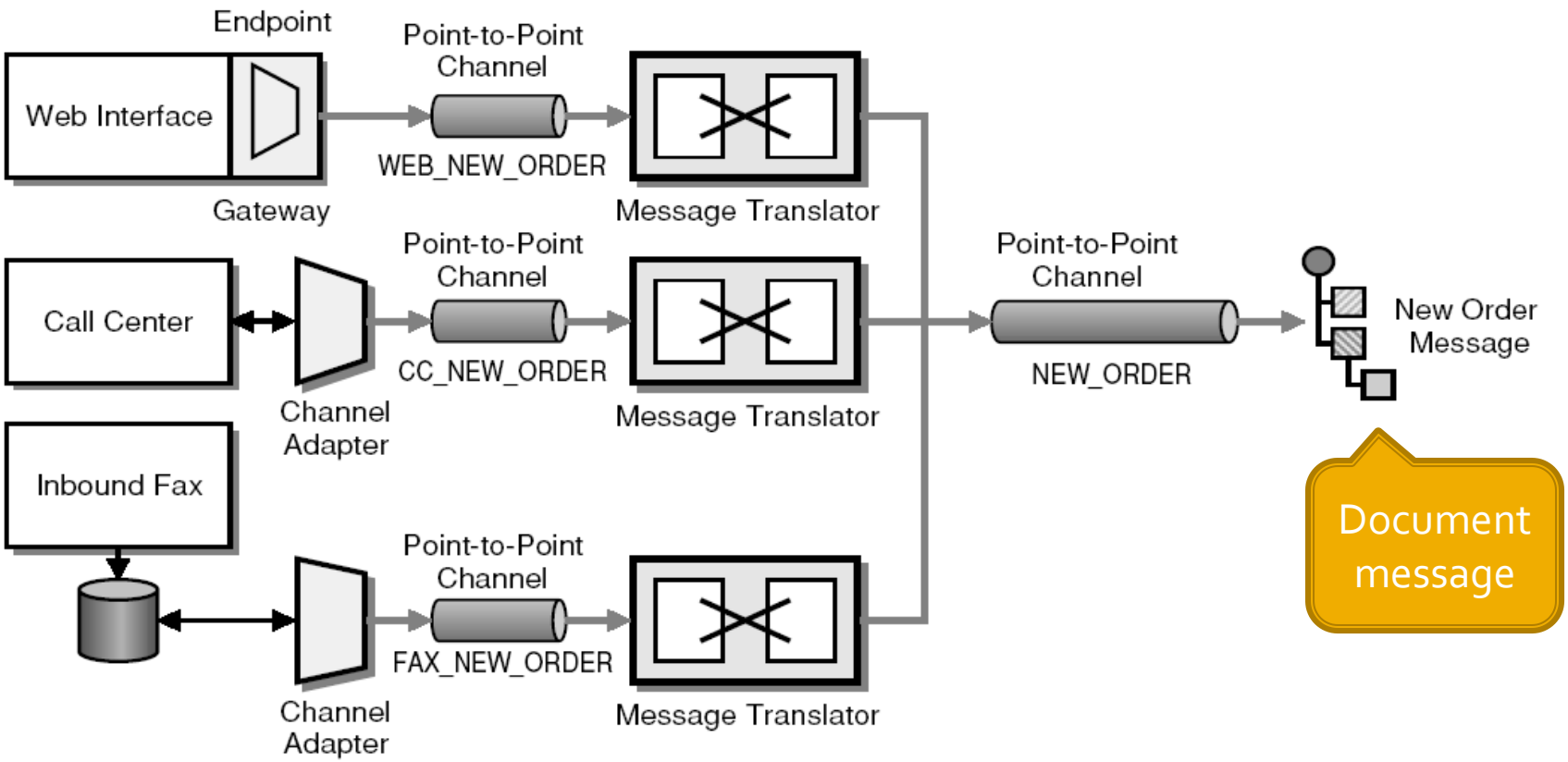
- New Catalogue
  - The suppliers update their catalogue periodically. WGRUS needs to update its pricing and availability based in the new catalogues.
- Announcements
  - Customers can subscribe to selective announcements from WGRUS.
- Testing and Monitoring
  - The operations staff needs to be able to monitor all individual components and the message flow between them.

WGRUS
Internal systems



WGRUS IT Infrastructure

WGRUS
Take orders



Taking Orders from Three Different Channels

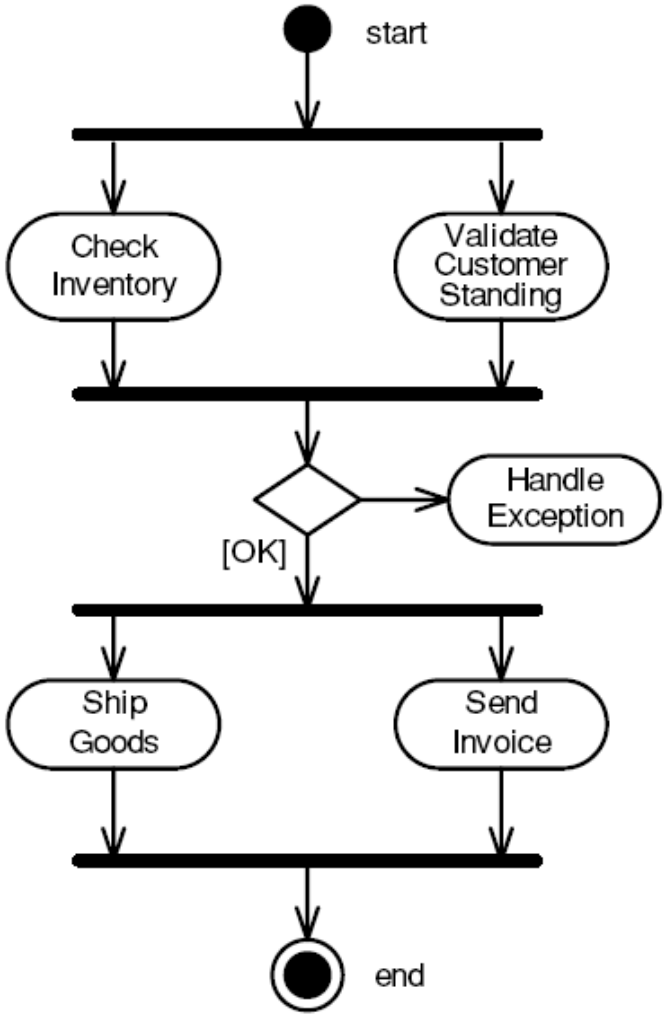
WGRUS
Patterns used in Take orders

- Message endpoint
  - A way to connect na application to a message channel to send and/or receive messages
- Message gateway
  - isolate the application code from the messaging-specific code
- Channel adapter
  - attach to an application and publish messages to a Message Channel whenever an event occurs inside the application
- Point-to-point channel
  - Guarantees only one receiver will receive the message

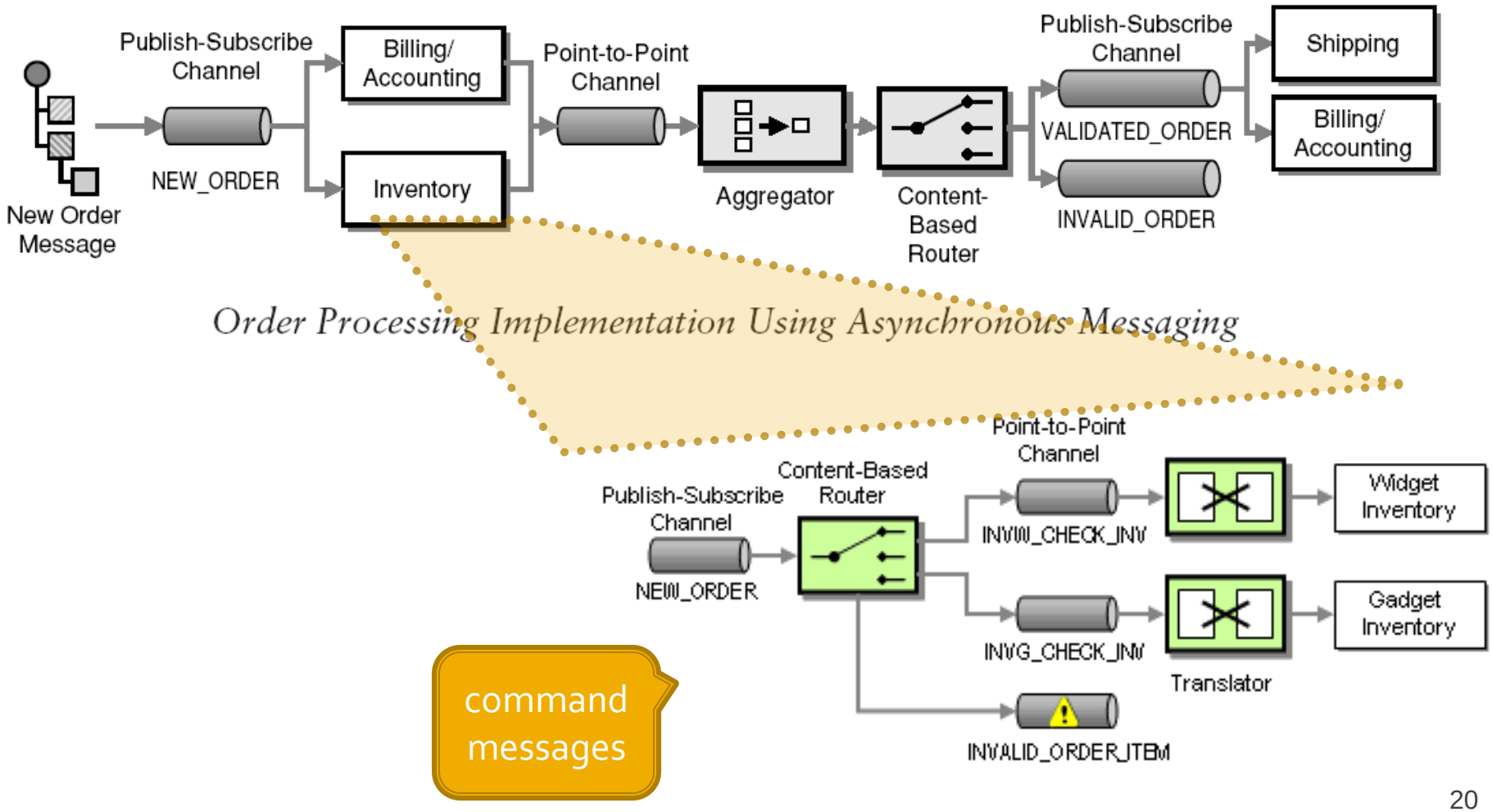
WGRUS
Patterns used in Take orders (2)

- Message translator
  - Translates between two data formats
- Canonical data model
  - Common data model independent of any specific application
- Datatype channel
  - All the messages in the channel are of the same type (format)
- Document message
  - Represents the data; the receiver will decide what to do with it.

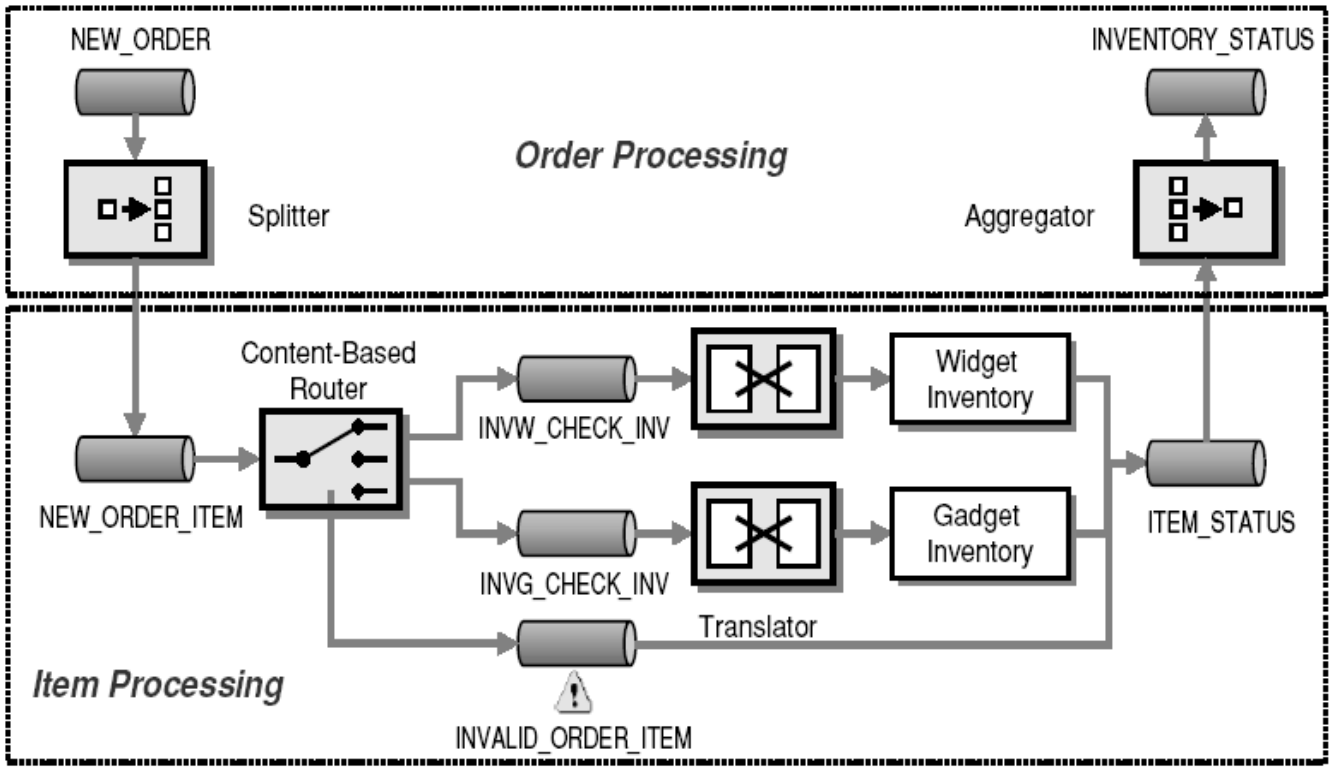
WGRUS
Process Orders



WGRUS
Process Orders (2)



WGRUS
Process Orders (3)

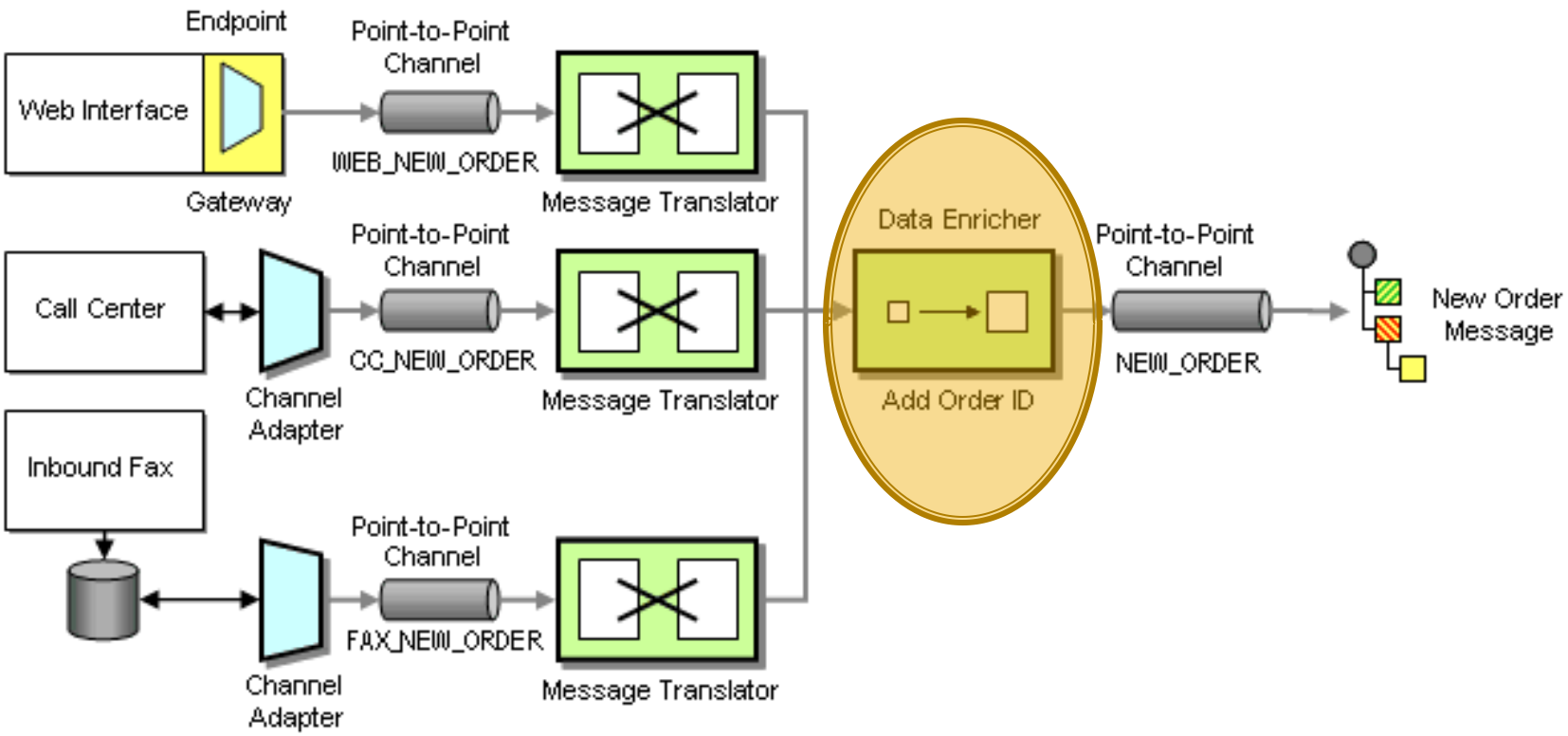


Processing Order Items Individually

Aggregators

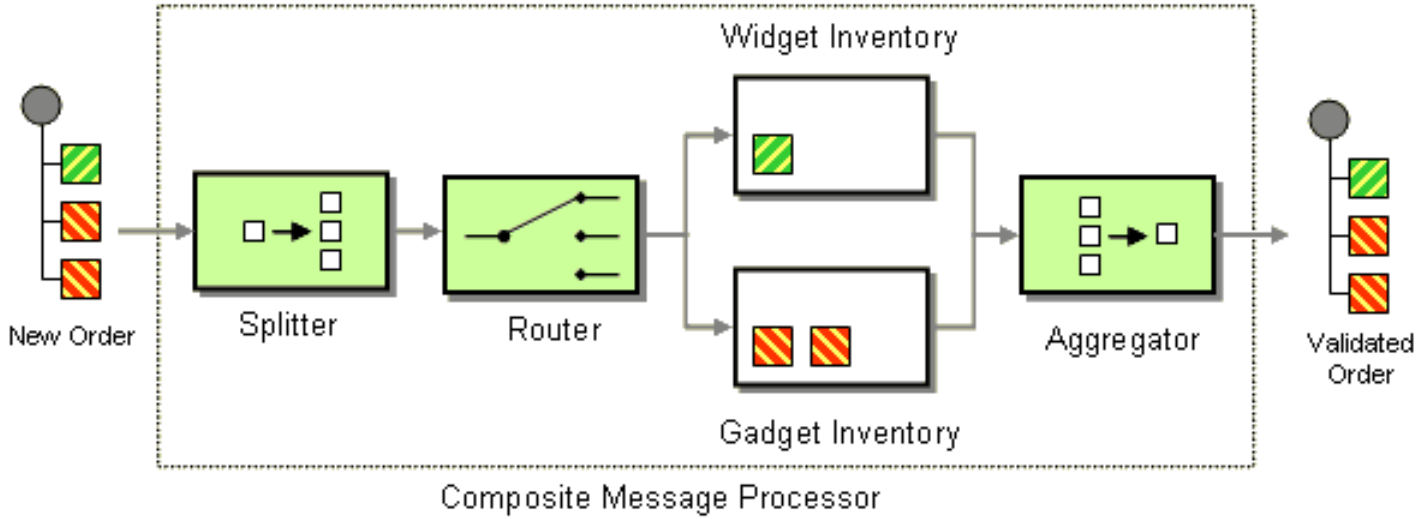
- Which messages belong together?
  - "correlation"
- How do we determine that all messages are received?
  - the "completeness condition"
- How do we combine the individual messages into one result message?
  - the "aggregation algorithm"

WGRUS
Back to Take Orders…



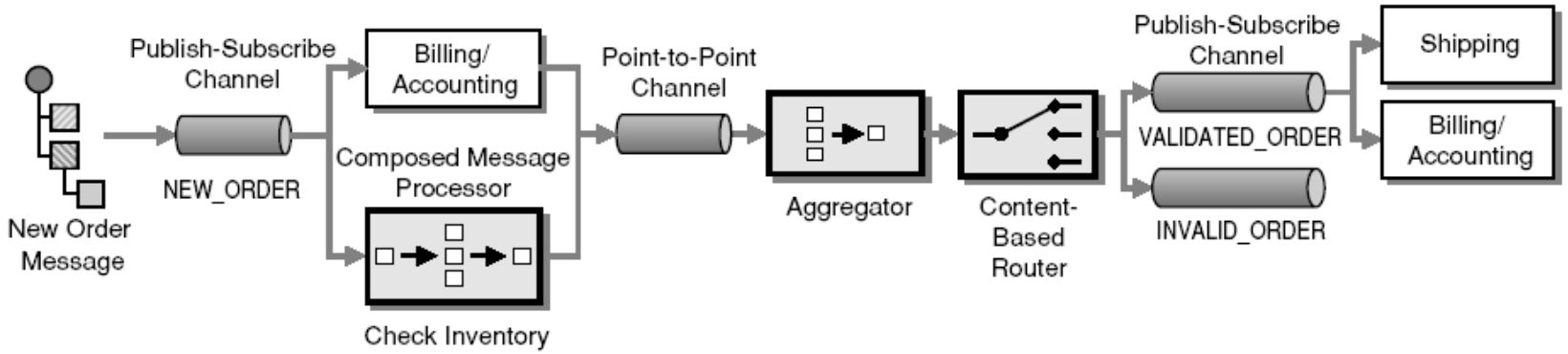
Composed Message Processor

- How can you maintain the overall message flow when processing a message consisting of multiple elements, each of which may require different processing?



- The Composed Message Processor splits the message up, routes the sub-messages to the appropriate destinations and re-aggregates the responses back into a single message.

WGRUS
Revised Process Orders

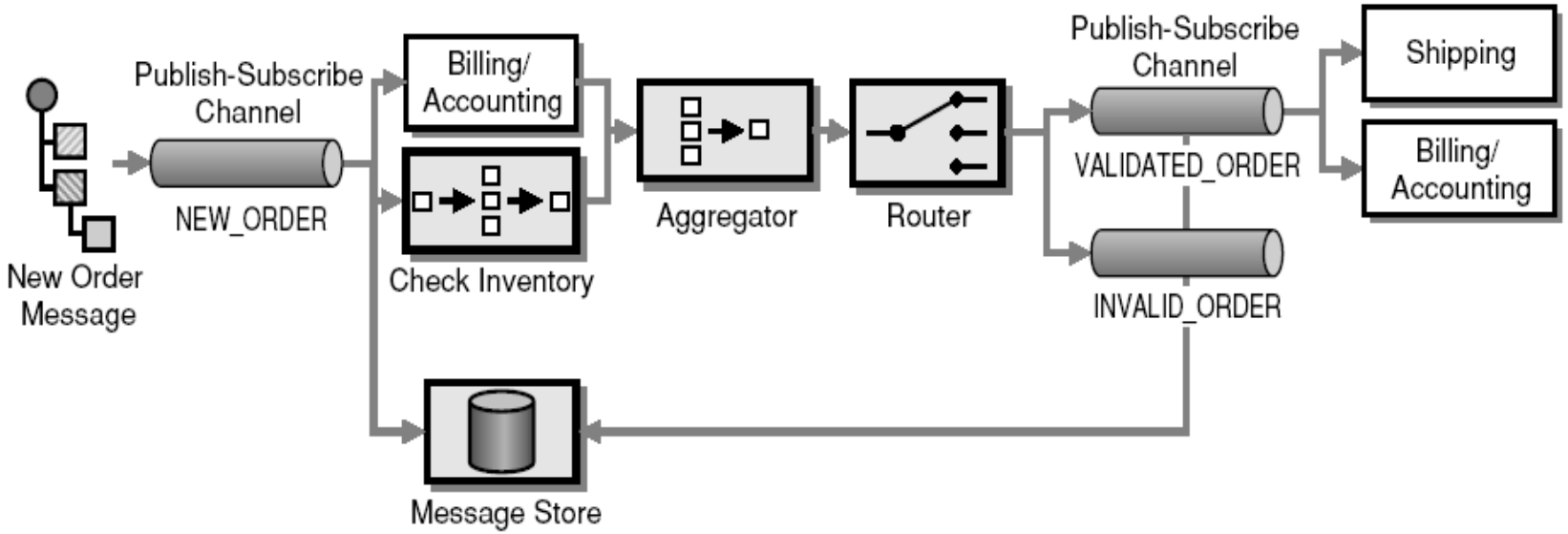


Revised Order Process Implementation

WGRUS
Patterns in Process Orders

- Publish-Subscribe channel
  - One application sends a message and the Pub-Sub channel delivers it to all active receivers
- Aggregator
  - Generates a single message distilled from several input messages
- Content-based router
  - Delivers each message to the correct receipient based on message content
- Command message
  - Represents the data and the operation that is beeing requested to the receiver
- Splitter
  - Splits a single message and its content into several messages
- Content Enricher
  - Adds content to a message (tipically a correlation id)

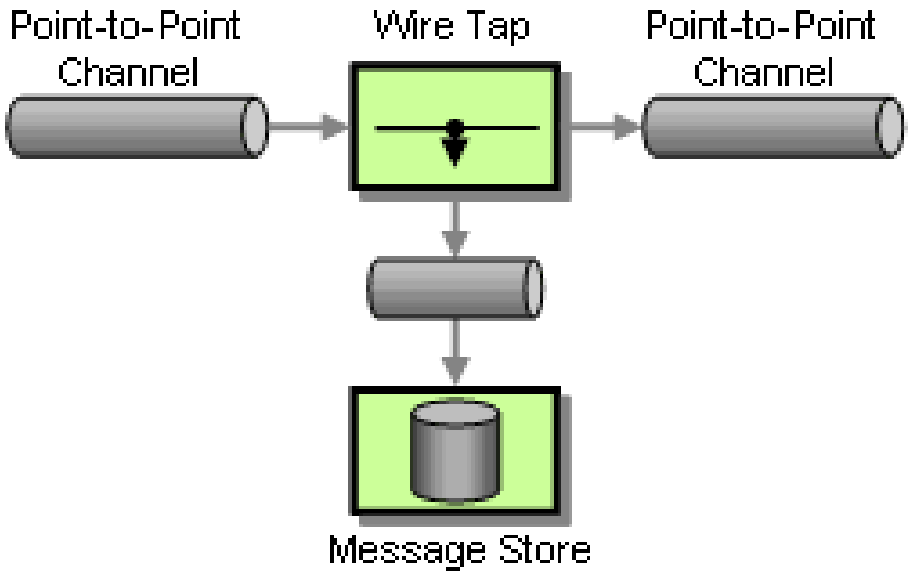
WGRUS
Track Order Status



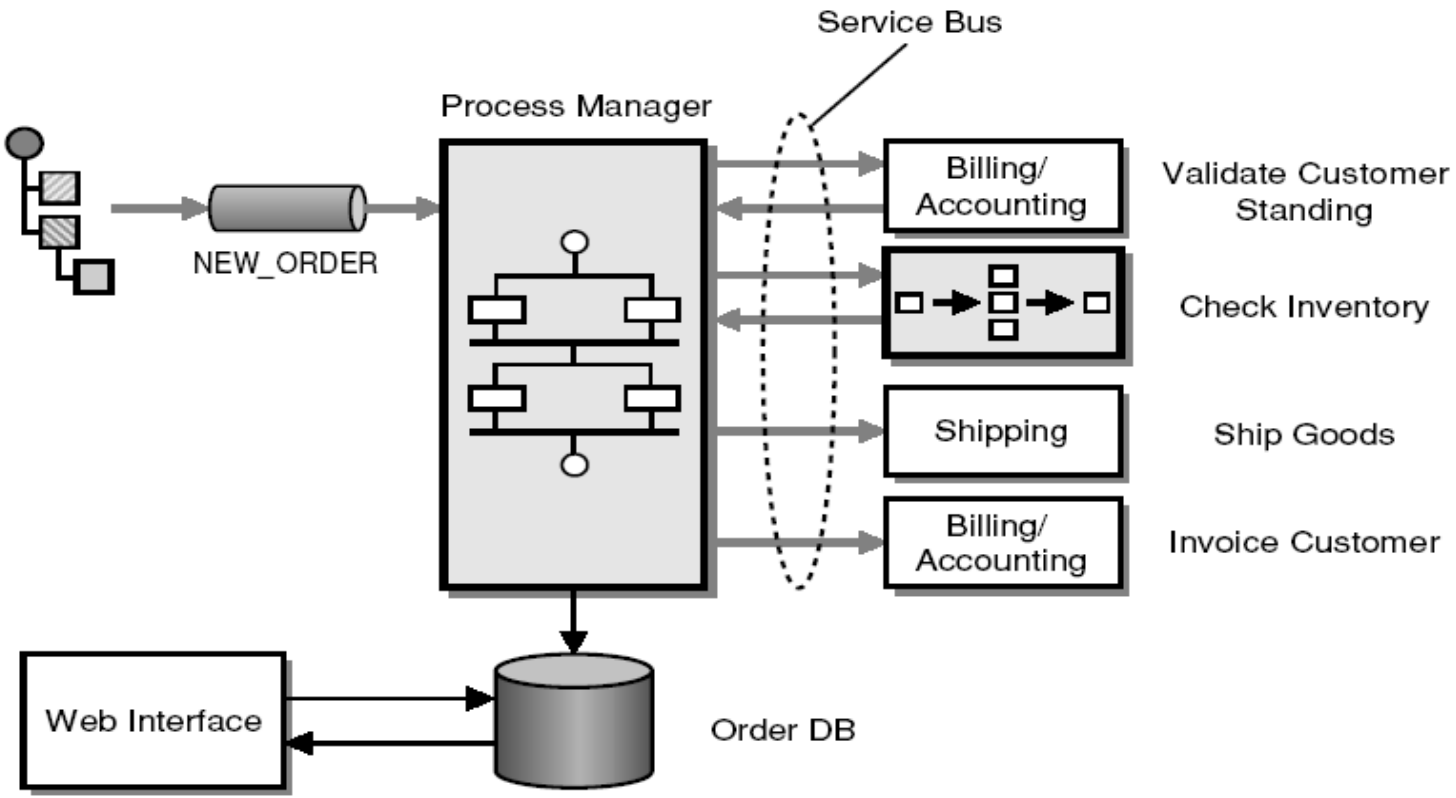
Adding a Message Store to Track Order Status

WGRUS
Track Order Status (2)

- If we were using a point-to-point channel (not a pub-sub)



WGRUS
Process Orders with Process Manager



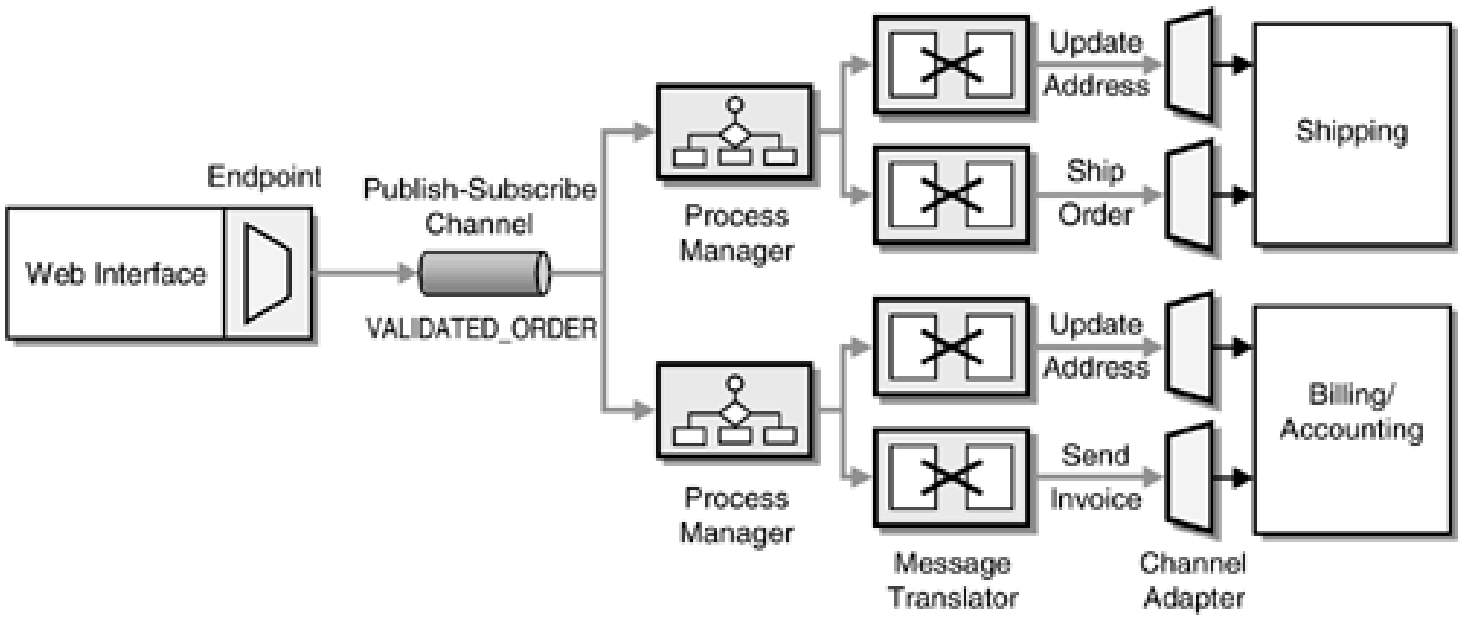
Processing Orders with a Process Manager

WGRUS
Patterns in Check Order status

- Wire-Tap
  - Creates a detour in a point-to-point message channel to publish the same message to more than one recipient
- *Message-Store*
  - Allows reporting against message information without disturbing the loosely coupled and transient nature of a messaging system
- Process Manager
  - Allows routing a message through multiple processing steps when the required steps may not be known at design-time and may not be sequential thru a central processing unit that maintains the state of the sequence and determine the next processing step based on intermediate results

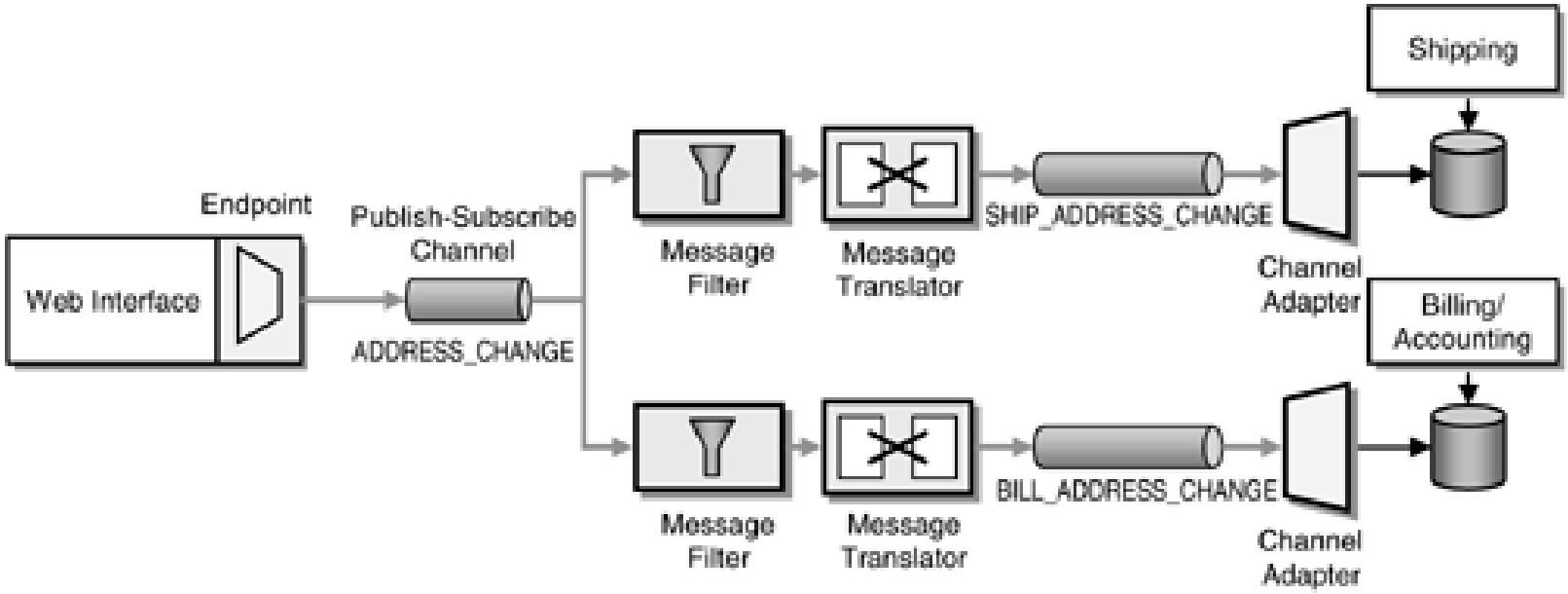
WGRUS
Change Address

- Alternative 1
  - Include the billing and shipping address with every message



WGRUS
Change Address (2)

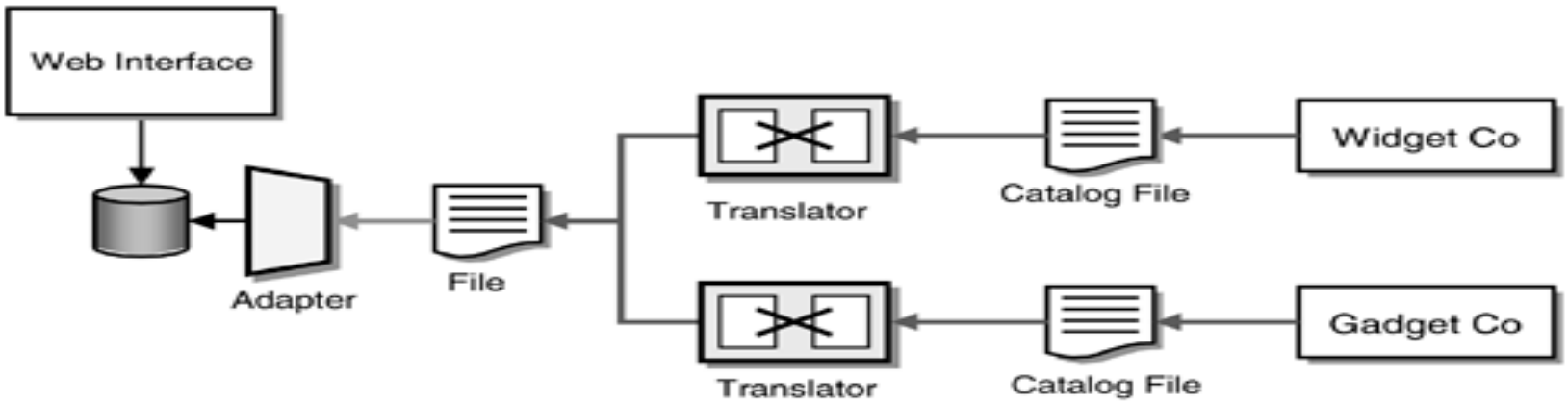
- Alternative 2
  - Address is saved in every system and changes are replicated



WGRUS
Patterns in Change Address

- Message Filter
  - eliminates undesired messages from a channel based on a set of criteria to avoid receiving uninteresting messages

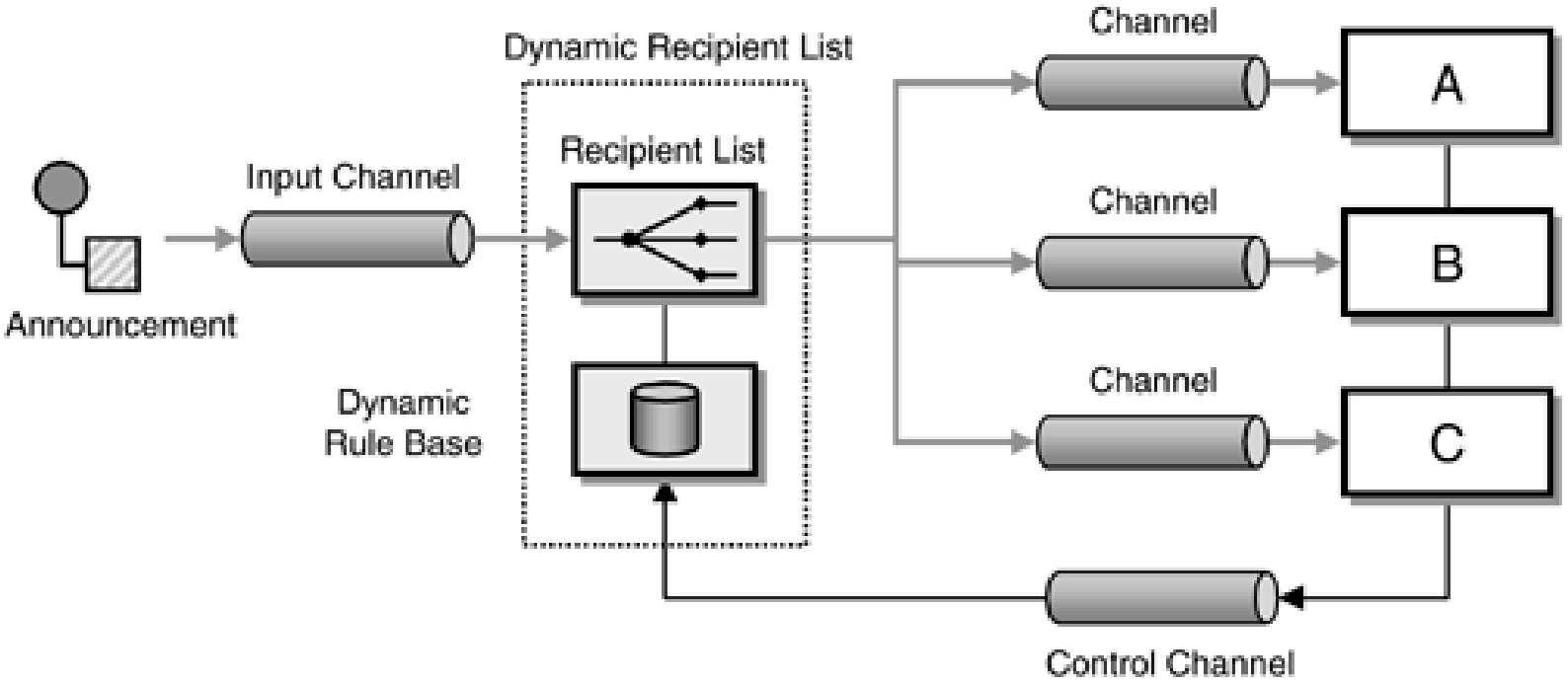
WGRUS
New Catalogue



WGRUS
Patterns in New Catalogue

- File Transfer
  - each application produces files containing information that other applications need to consume. Integrators take the responsibility of transforming files into different formats. Produce the files at regular intervals according to the nature of the business.

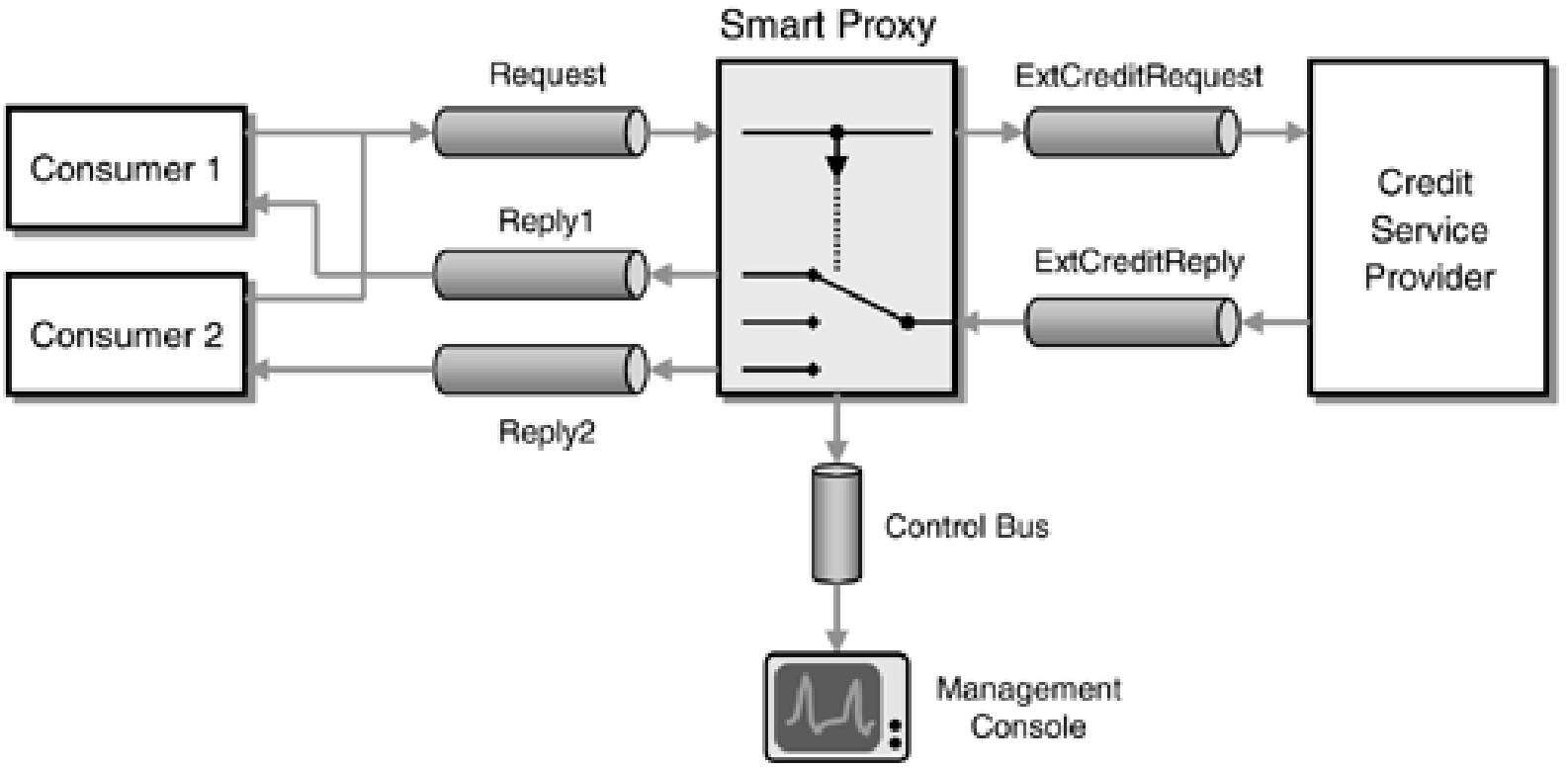
WGRUS Annoucements



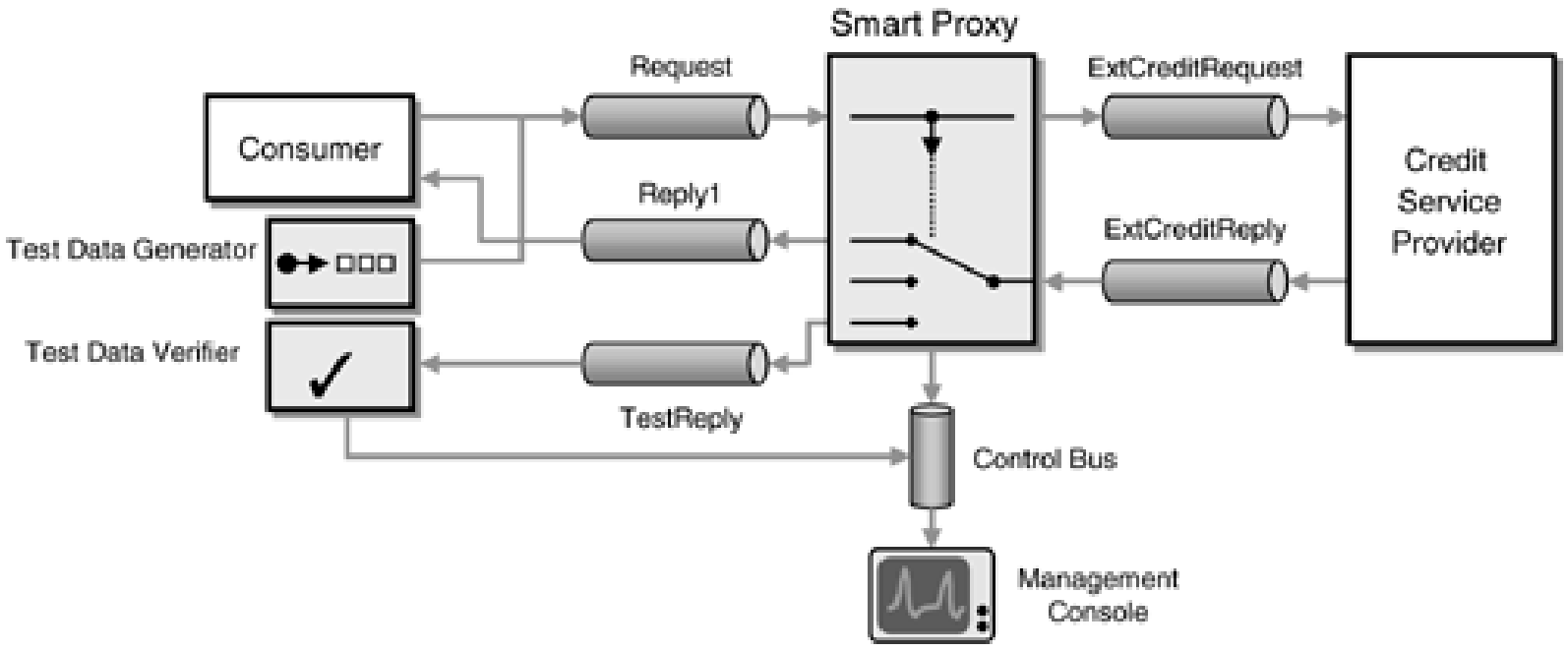
WGRUS
Patterns in Annoucements

- Recipient List
  - Publishes one message to several known recipients (as opposed to pub-sub)
- Dynamic Router
  - a Router that can self-configure based on special configuration messages from participating destinations.

WGRUS
System Monitoring



WGRUS
System Monitoring

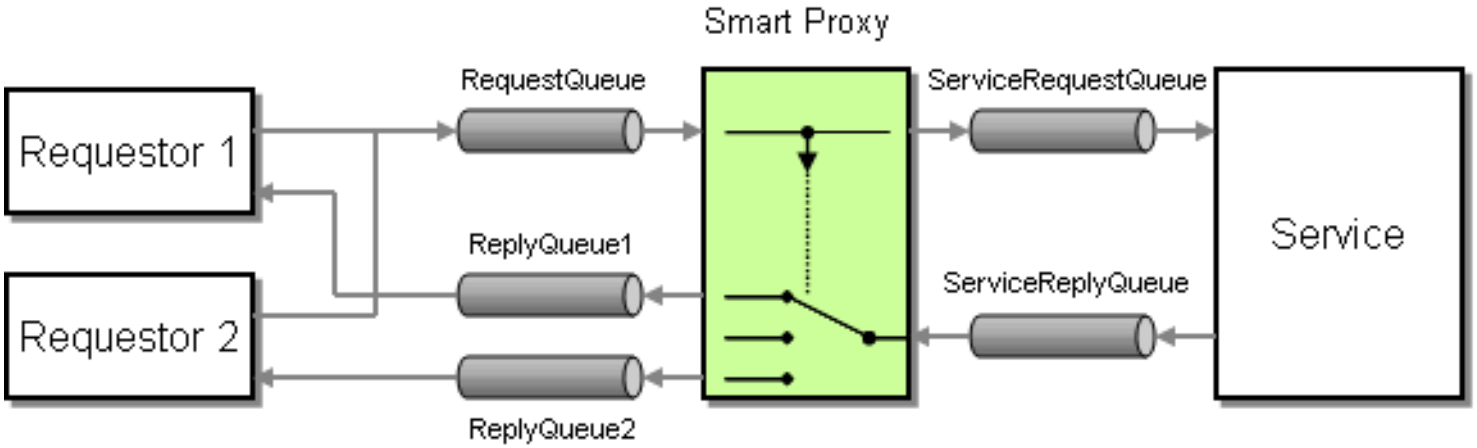


WGRUS
Patterns in System Monitoring

- Return Address
  - The request message contains a Return Address that indicates where the replier should send the reply message
- Control Bus
  - Uses the same messaging infrastructure with specific channels for transmitting control information
- Test Message
  - A message designed to check the "health" of the executing system

Smart Proxy

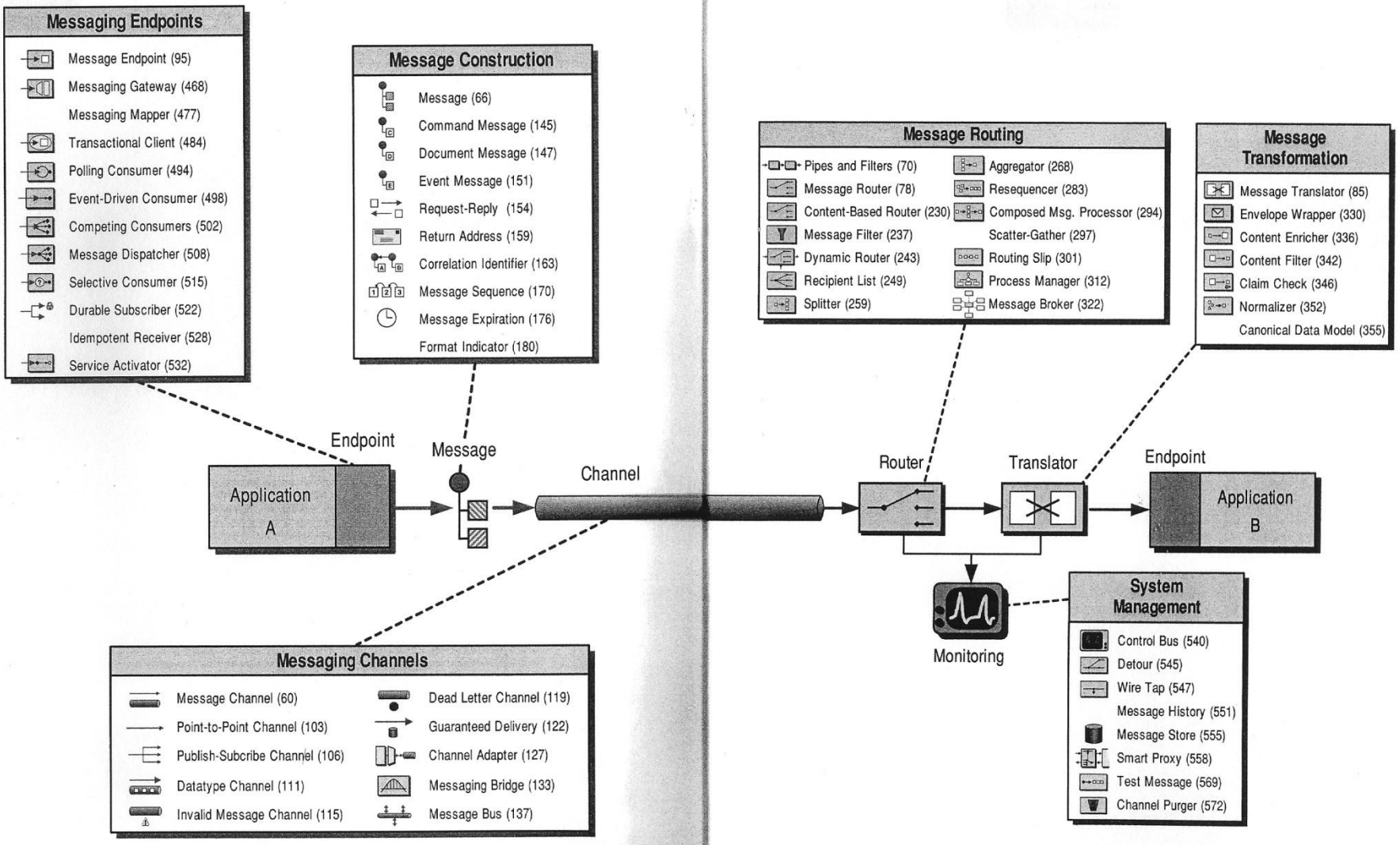
- How can you track messages on a service that publishes reply messages to the Return Address specified by the requestor?



- Use a Smart Proxy to store the Return Address supplied by the original requestor and replace it with the address of the Smart Proxy. When the service sends the reply message route it to the original Return Address.

Enterprise Integration Patterns

Enterprise Integration Patterns



Exercise

- Remember the example DS you provided in the last session.
- Define an hypothetical SOA for that system
  - Define contract
  - Identify where you would use the presented patterns

Bibliography

- Hohpe, G. And Woolf, B. (2004) *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley, ISBN 0321200683.
  - www.eaipatterns.com
  - Chapter 1, http://eaipatterns.com/Chapter1.html