RabbitMQ for .Net Developers – Part 1

Advanced Message Exchange Patterns

Michael Stephenson
@michael_stephen
Michael_stephensonuk@yahoo.co.uk

pluralsight
hardcore developer training

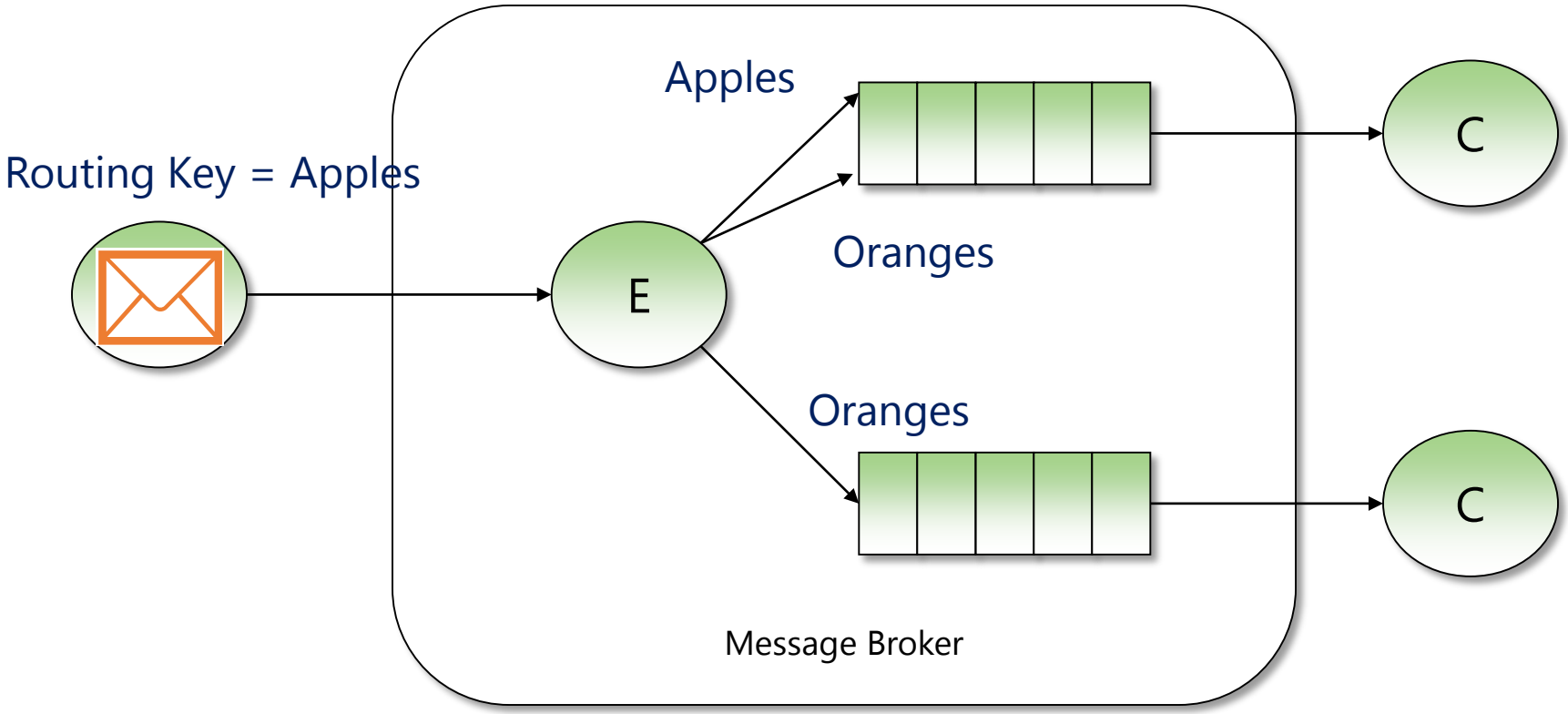
Agenda

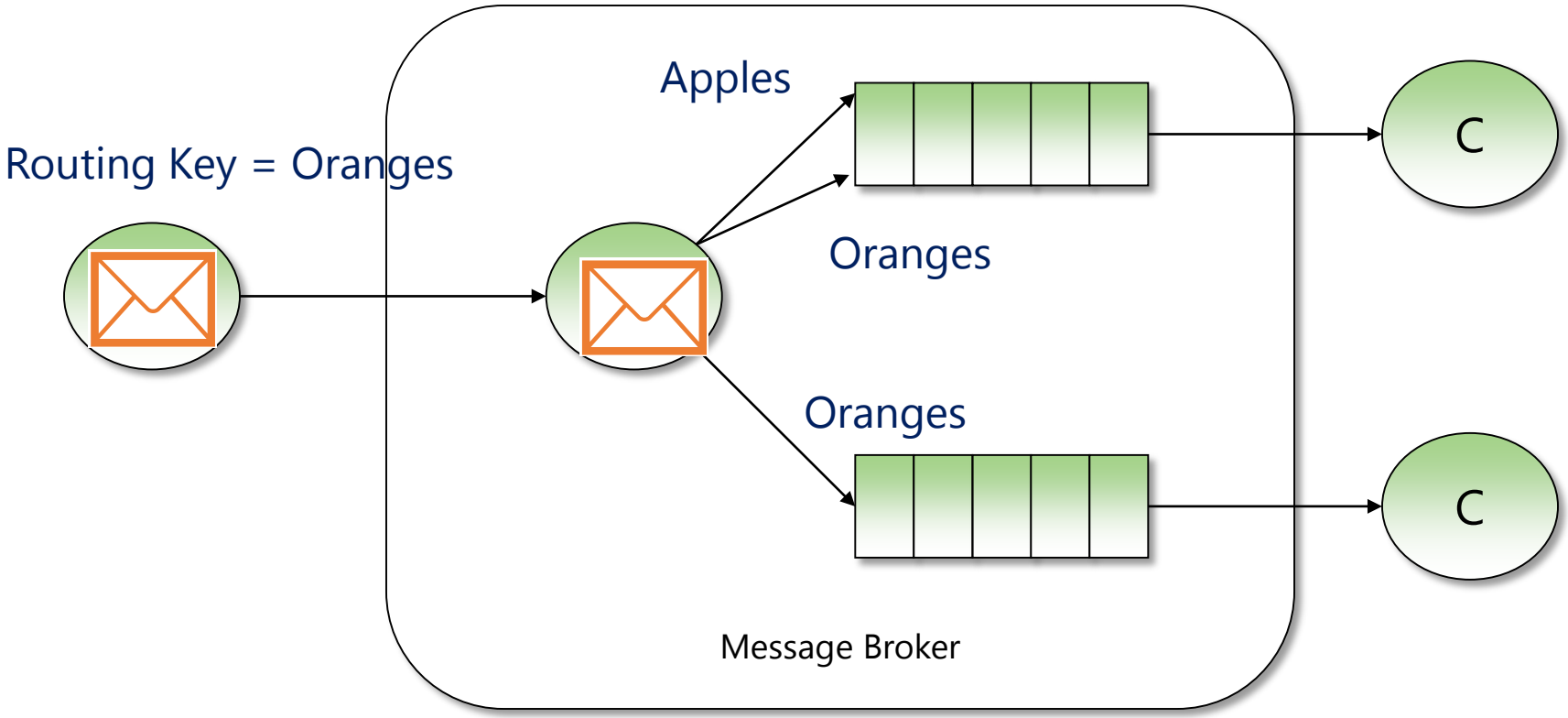
- Routing
- Topic
- Headers
- Scatter Gather

Routing

Pattern

- **Overview**
  - Message sent to an exchange along with a routing key
  - A copy of message sent to queues which exactly match the routing key
  - Each queue will have a receiver processing messages
- **Characteristics**
  - Exchange = The message is sent to a named exchange
  - Exchange Type = Direct
  - Routing Key = Piece of information which can direct messages
  - Message only goes to queues which match the routing key


Routing Key = Apples
E
Apples
Oranges
Oranges
C
C
Message Broker

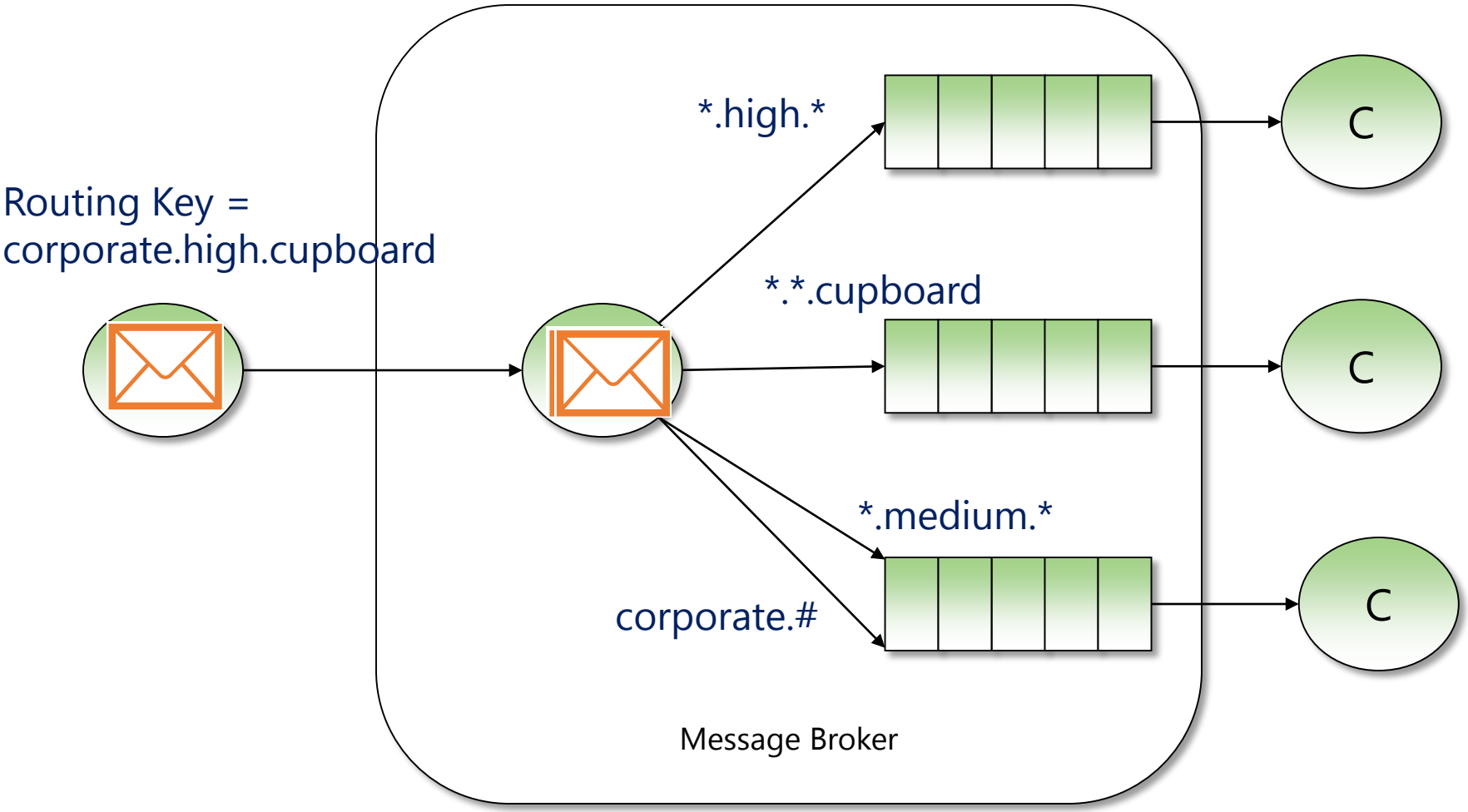

Routing Key = Oranges
Apples
Oranges
Oranges
C
C
Message Broker

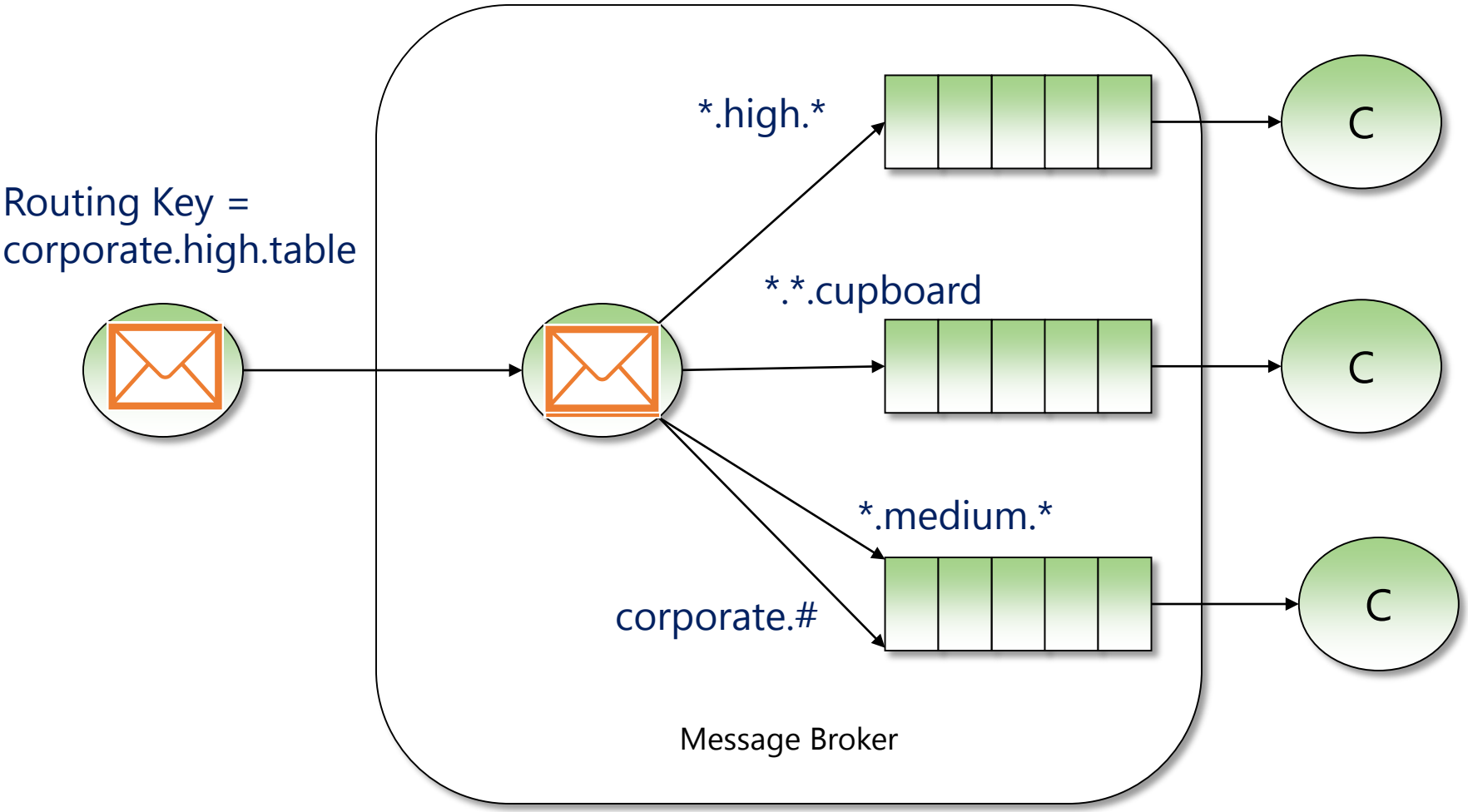
Demo – Routing

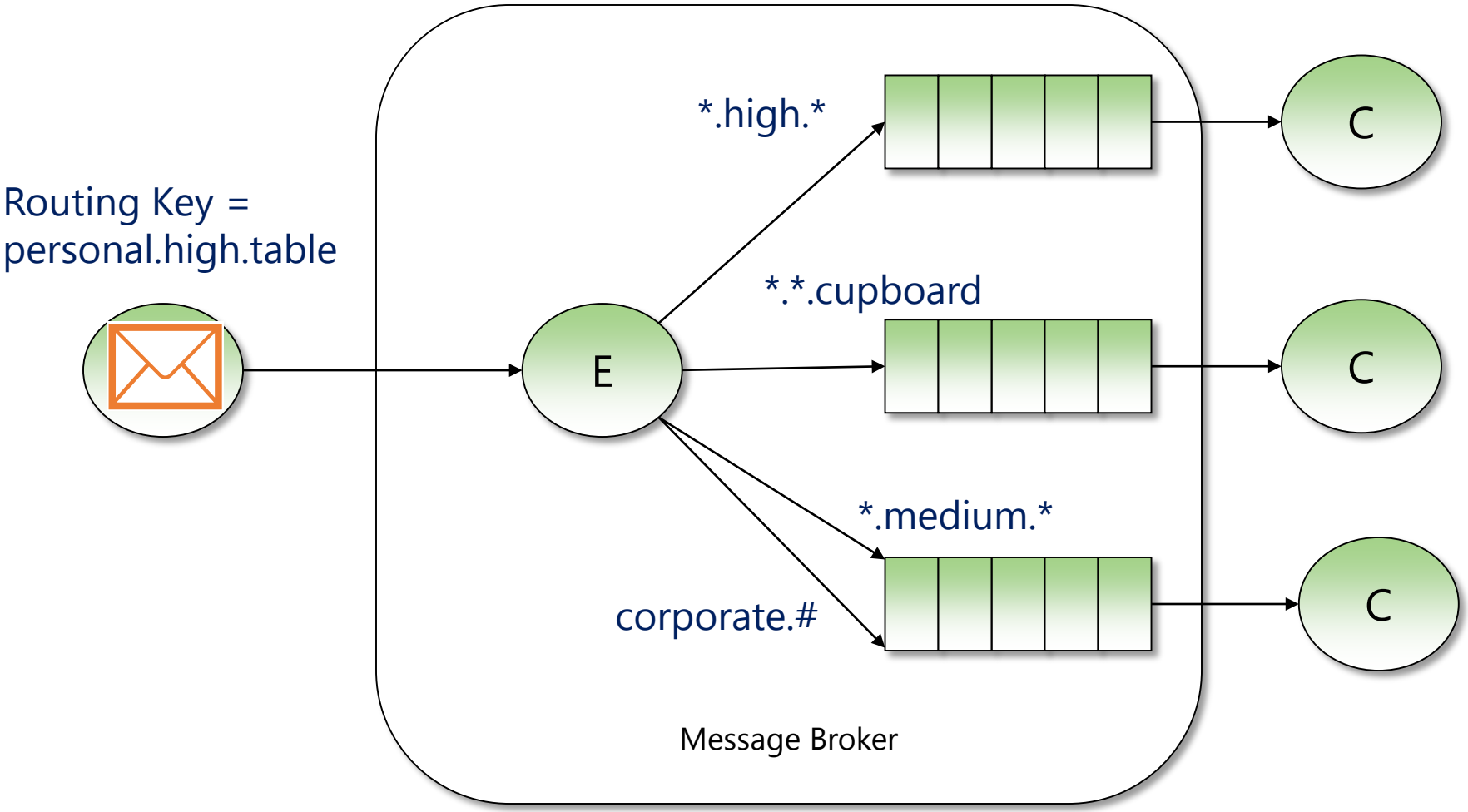
Topic

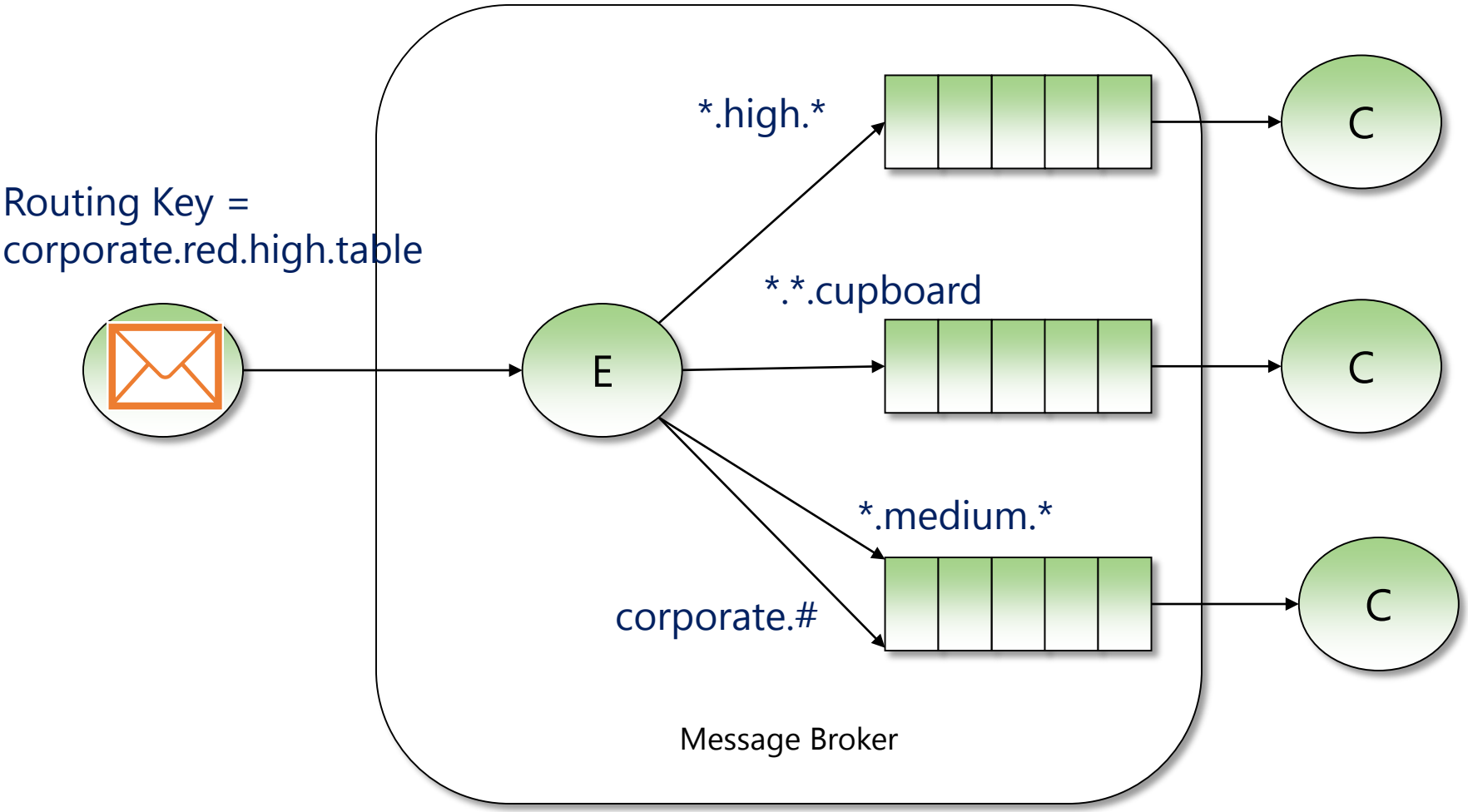
Pattern

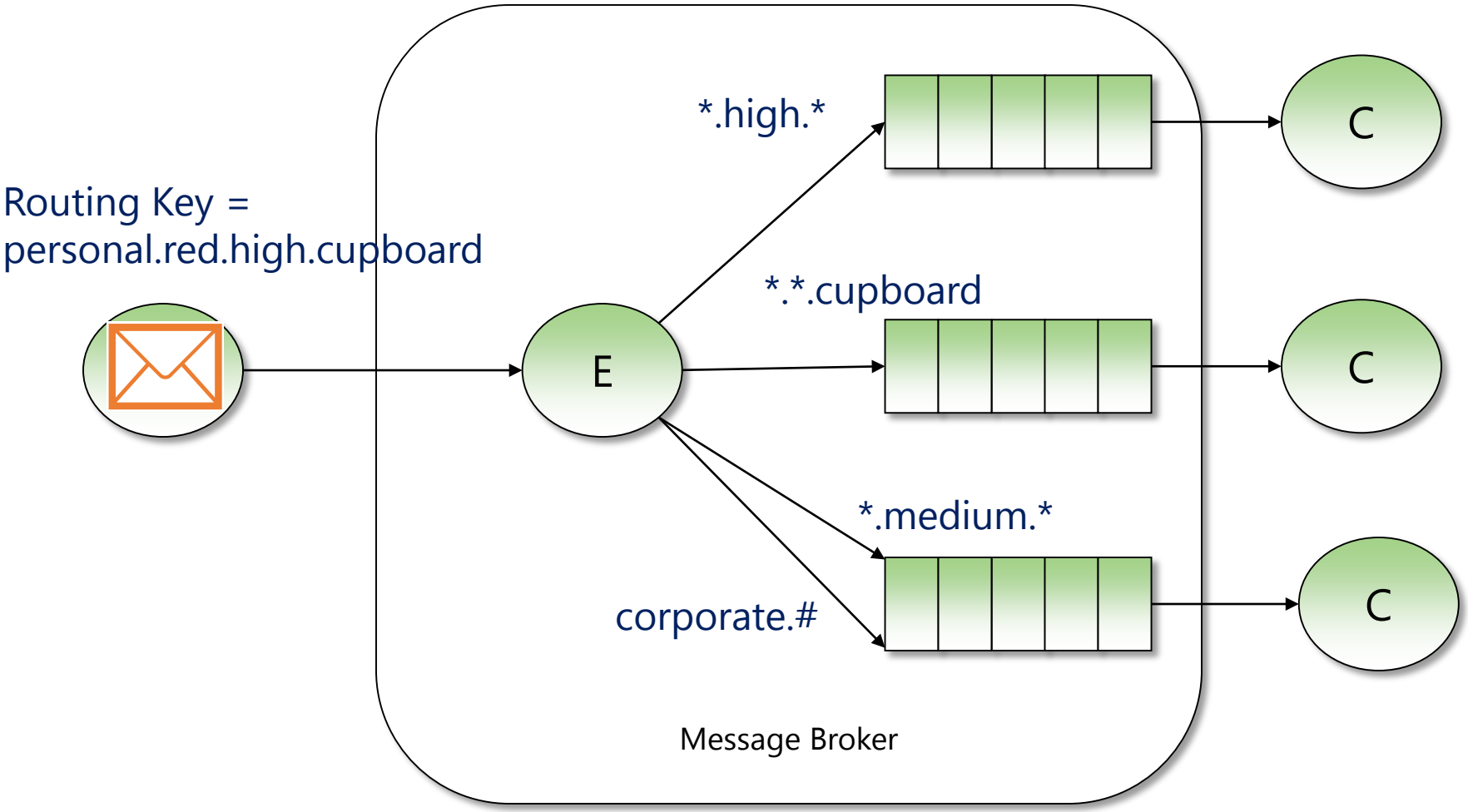
- **Overview**
  - Message sent to an exchange along with a routing key
  - A copy of message sent to queues which match expressions against the routing key
  - Each queue will be processed by receivers
- **Characteristics**
  - Exchange = The message is sent to a named exchange
  - Exchange Type = Topic
  - Routing Key = Piece of information which can direct messages
  - Routing Key includes special characters
    - * can replace one word
    - # can replace zero or more words


Routing Key =
corporate.high.cupboard
.high.
..cupboard
.medium.
corporate.#
C
C
C
Message Broker


Routing Key =
corporate.high.table
.high.
..cupboard
.medium.
corporate.#
C
C
C
Message Broker


Routing Key =
personal.high.table
E
.high.
..cupboard
.medium.
corporate.#
C
C
C
Message Broker


Routing Key =
corporate.red.high.table
E
.high.
..cupboard
.medium.
corporate.#
C
C
C
Message Broker


Routing Key =
personal.red.high.cupboard
E
.high.
..cupboard
.medium.
corporate.#
C
C
C
Message Broker

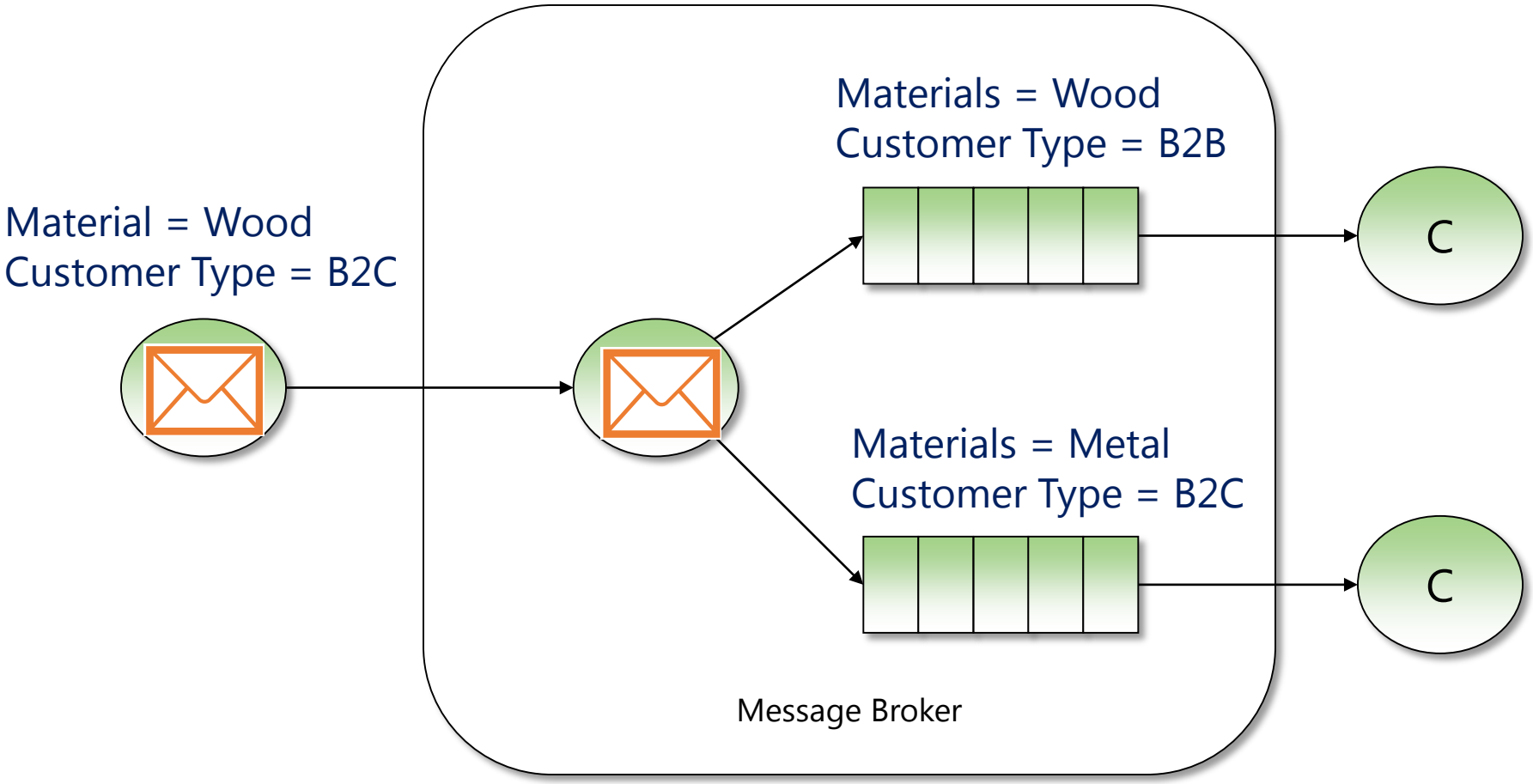
Demo – Topic

Headers

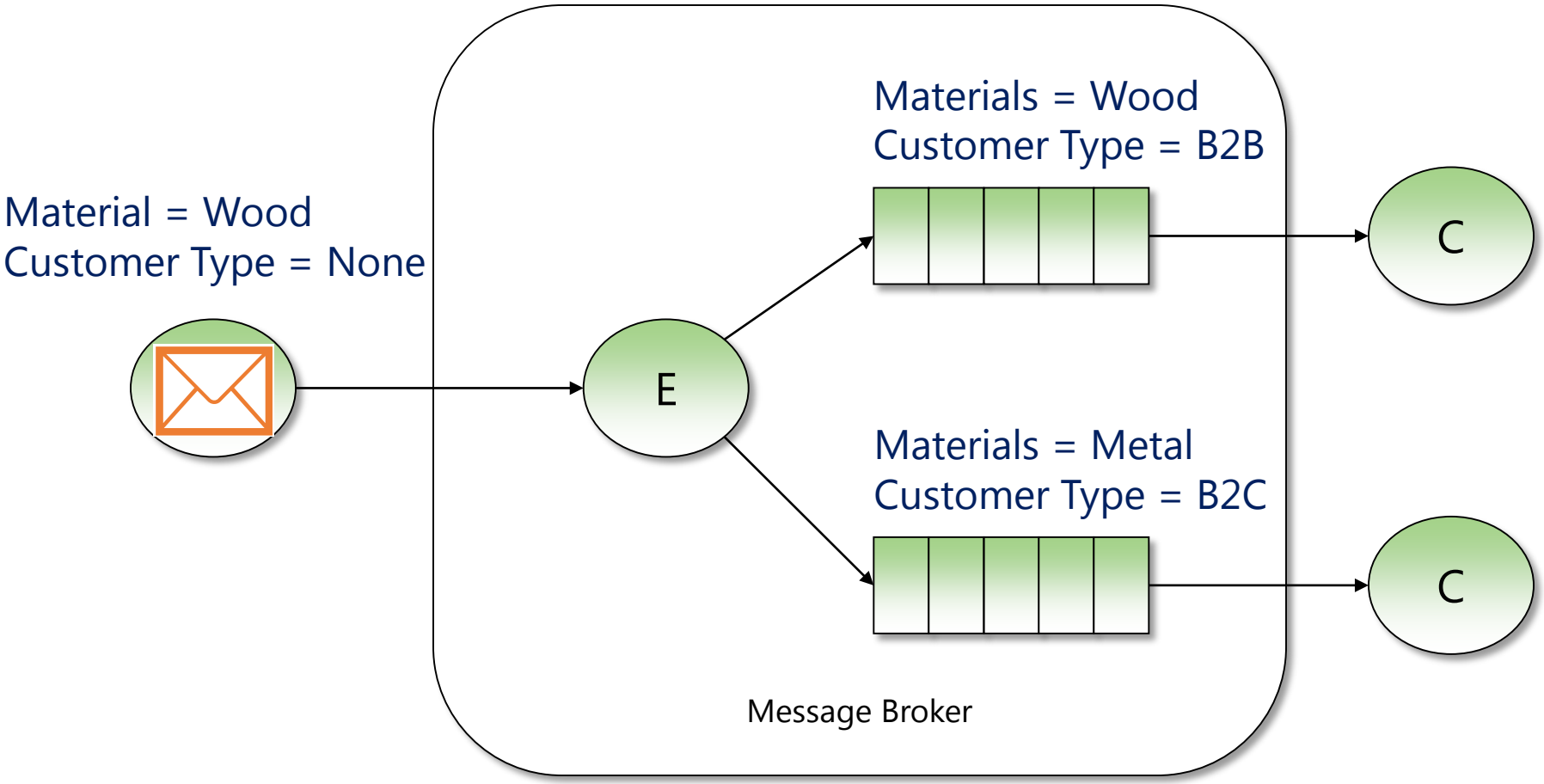
Pattern

- **Overview**
  - A message is sent to an exchange including some headers
  - A copy of the message is sent to queues which match the headers
  - Each queue is then processed by receivers
- **Characteristics**
  - Exchange = The message is sent to a named exchange
  - Exchange Type = Headers
  - Routing Key = should not be set
  - The message will have some properties specified
  - Match type indicates if all or any header must match

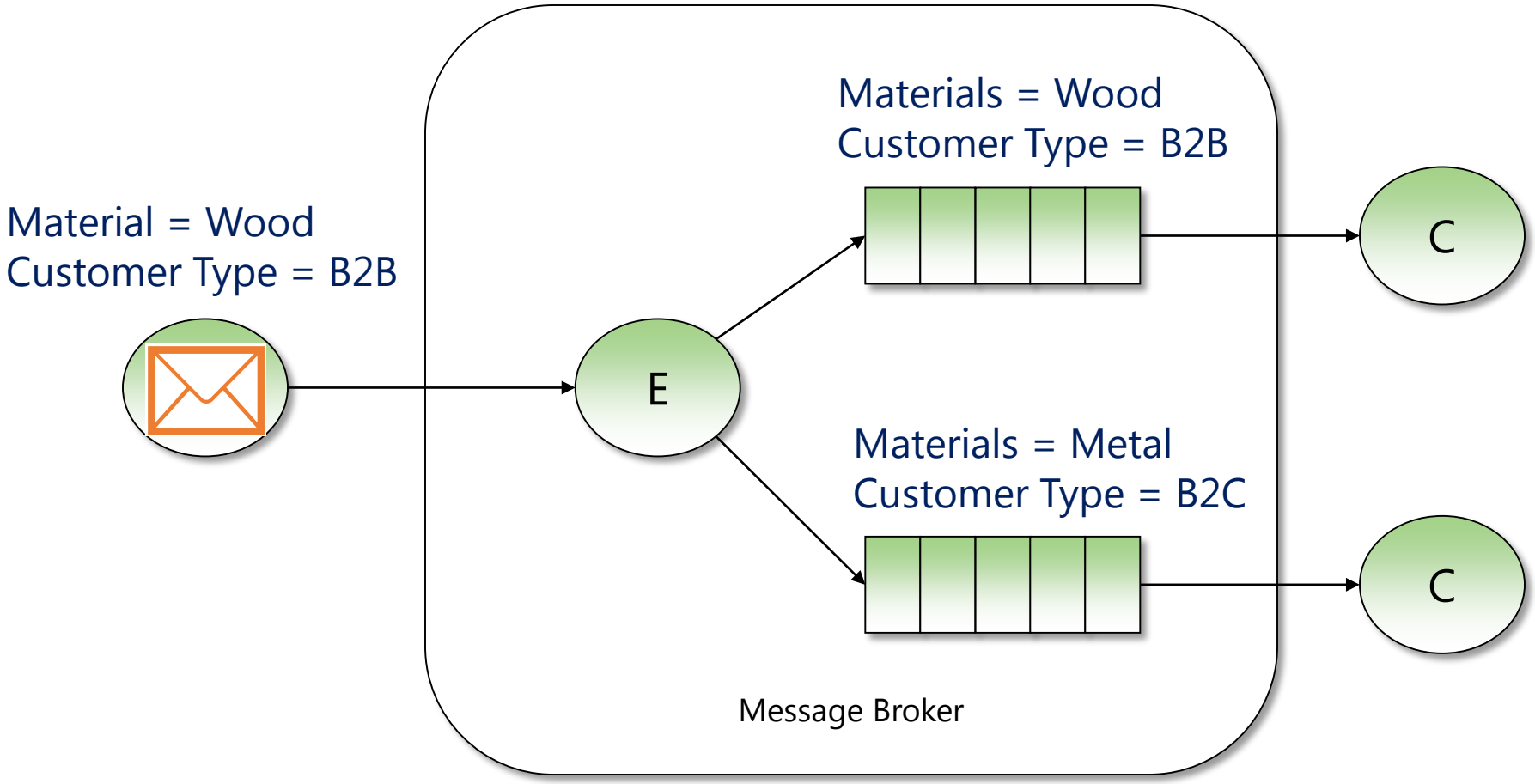
x-match = any


Material = Wood
Customer Type = B2C
Materials = Wood
Customer Type = B2B
Materials = Metal
Customer Type = B2C
C
C
Message Broker

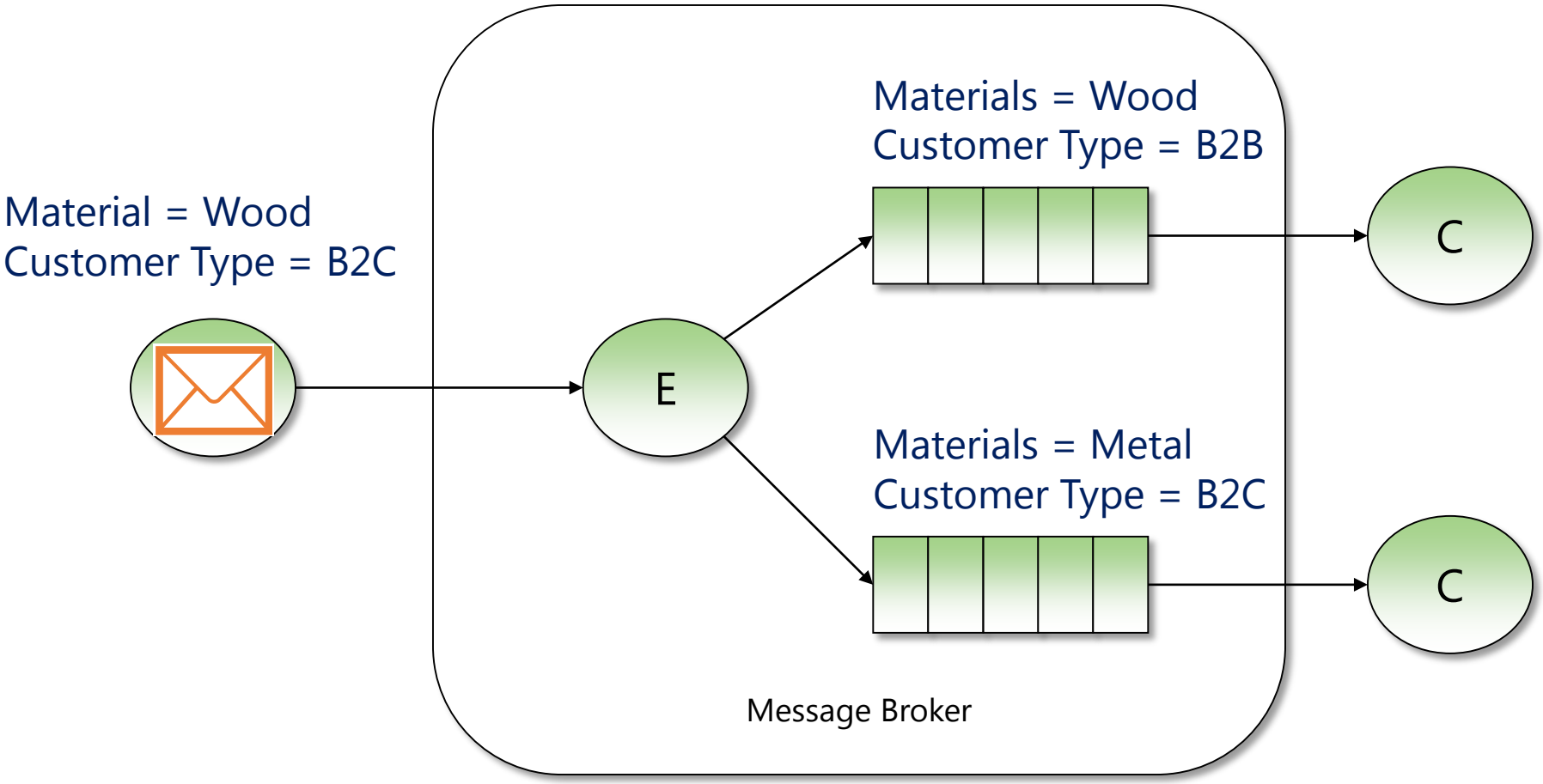
x-match = any


Material = Wood
Customer Type = None
E
Materials = Wood
Customer Type = B2B
C
Materials = Metal
Customer Type = B2C
C
Message Broker

x-match = all


Material = Wood
Customer Type = B2B
E
Materials = Wood
Customer Type = B2B
Materials = Metal
Customer Type = B2C
C
C
Message Broker

x-match = all

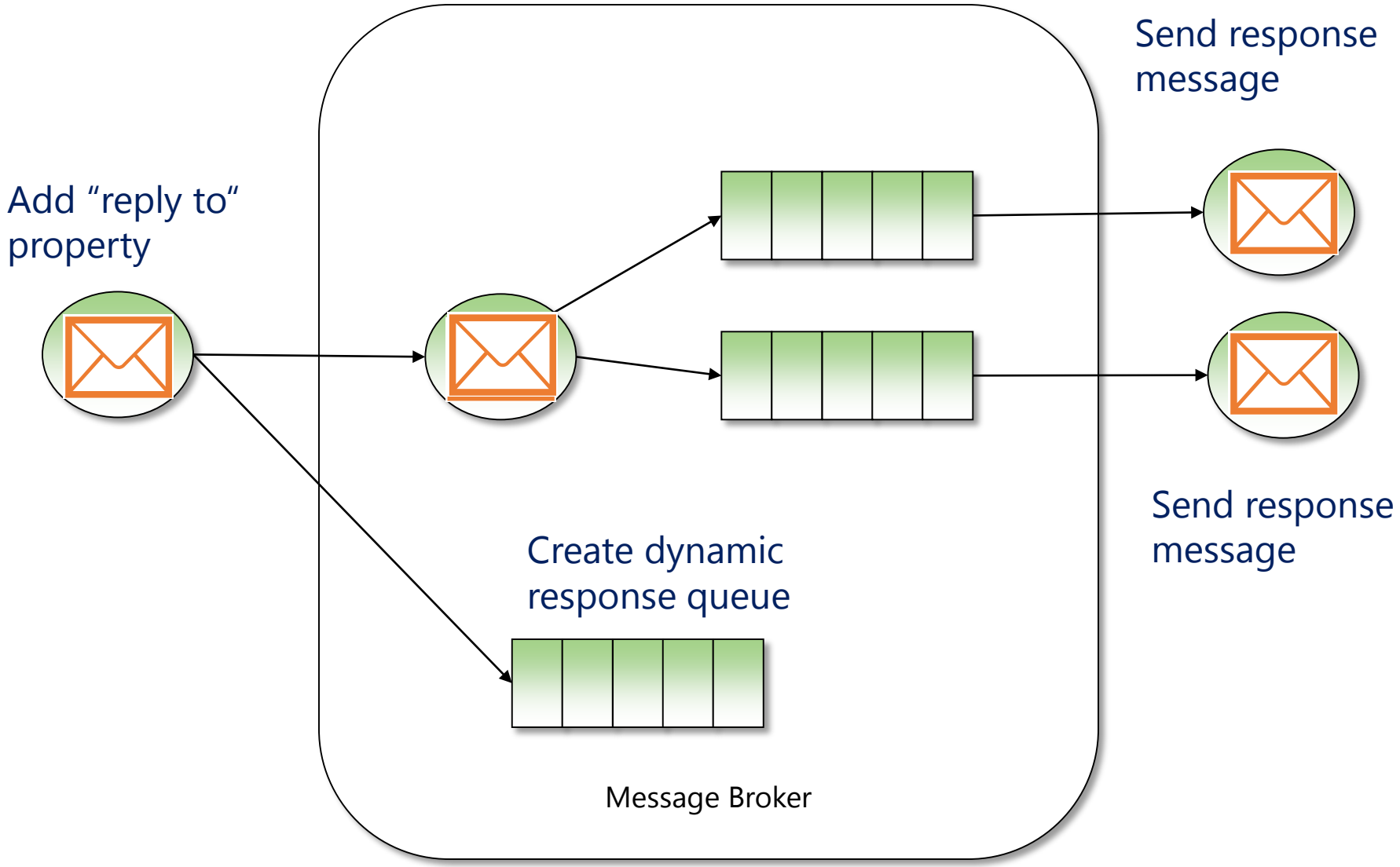

Material = Wood
Customer Type = B2C
E
Materials = Wood
Customer Type = B2B
Materials = Metal
Customer Type = B2C
C
C
Message Broker

Demo – Headers

Scatter Gather

Pattern

- **Overview**
  - The sender will start polling a response queue
  - The sender will send a request message to an exchange
  - The message will be copied to queues with a matching binding
  - The receivers will be process messages and send responses to the response queue
  - The sender will get its collection of responses
- **Characteristics**
  - Exchange = This will be a names exchange
  - Exchange Type = Can be any type (Fanout, Direct, Headers, Topic)
  - Routing Key = is optional depending on exchange type
  - Response = A collection of messages


Send response message
Add “reply to“ property
Send response message
Create dynamic response queue
Message Broker

Demo – Scatter Gather

Summary

- **Complex message exchange patterns**
  - Routing
  - Topics
  - Headers
  - Scatter Gather
- **Not so difficult?**