

Build a Producer / Consumer Scenario with RabbitMQ

<https://developers.sap.com/mission.scp-6-rabbitmq.html>

How microservices communicate ?

- Microservices based application is a distributed system
 - Each service instance is typically a process
- Services must interact with each other using communication protocol such as
 - HTTP (Synchronous)
 - AMQP (Asynchronous)

Another consideration

- Single receiver – Each request is processed by exactly one receiver
- Multiple receivers – Each request can be processed by multiple receivers
 - (must be asynchronous)

What is RabbitMQ ?

- AMQP is **an asynchronous protocol** supported by many operating systems and cloud environments
- RabbitMQ is a messaging broker and it implements the **AMQP protocol**
 - **Asynchronous protocol**
 - Decouples applications and improves scalability
 - Applications can send and receive messages in a standardized manner
 - Messages have a safe place to live until received
 - Most popular open source message broker

Common Terminologies

Message

- Package of information consisting of
 - header (key-value pair)
 - body (actual message)

Producer

- Creates and sends message

Consumer

- Receives and reads the message

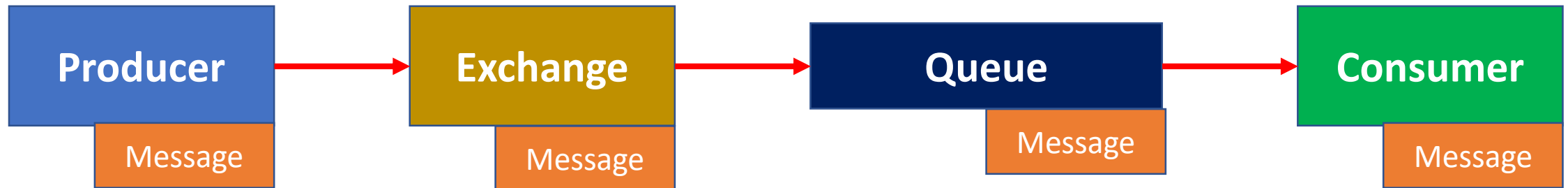
Common Terminologies

Queue

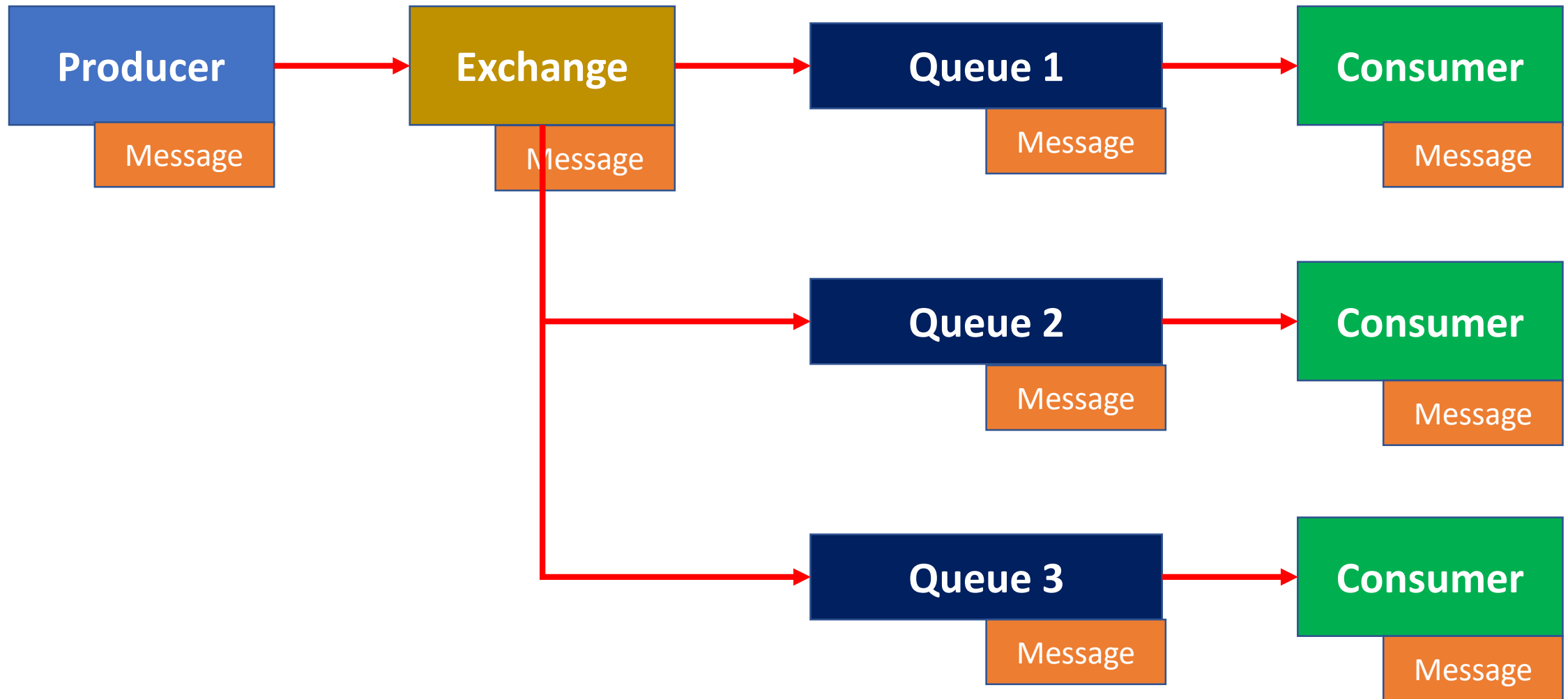
- Communication channel

Exchange

- Abstracts the Queue from the Producer



Multiple queues



Exchange Types

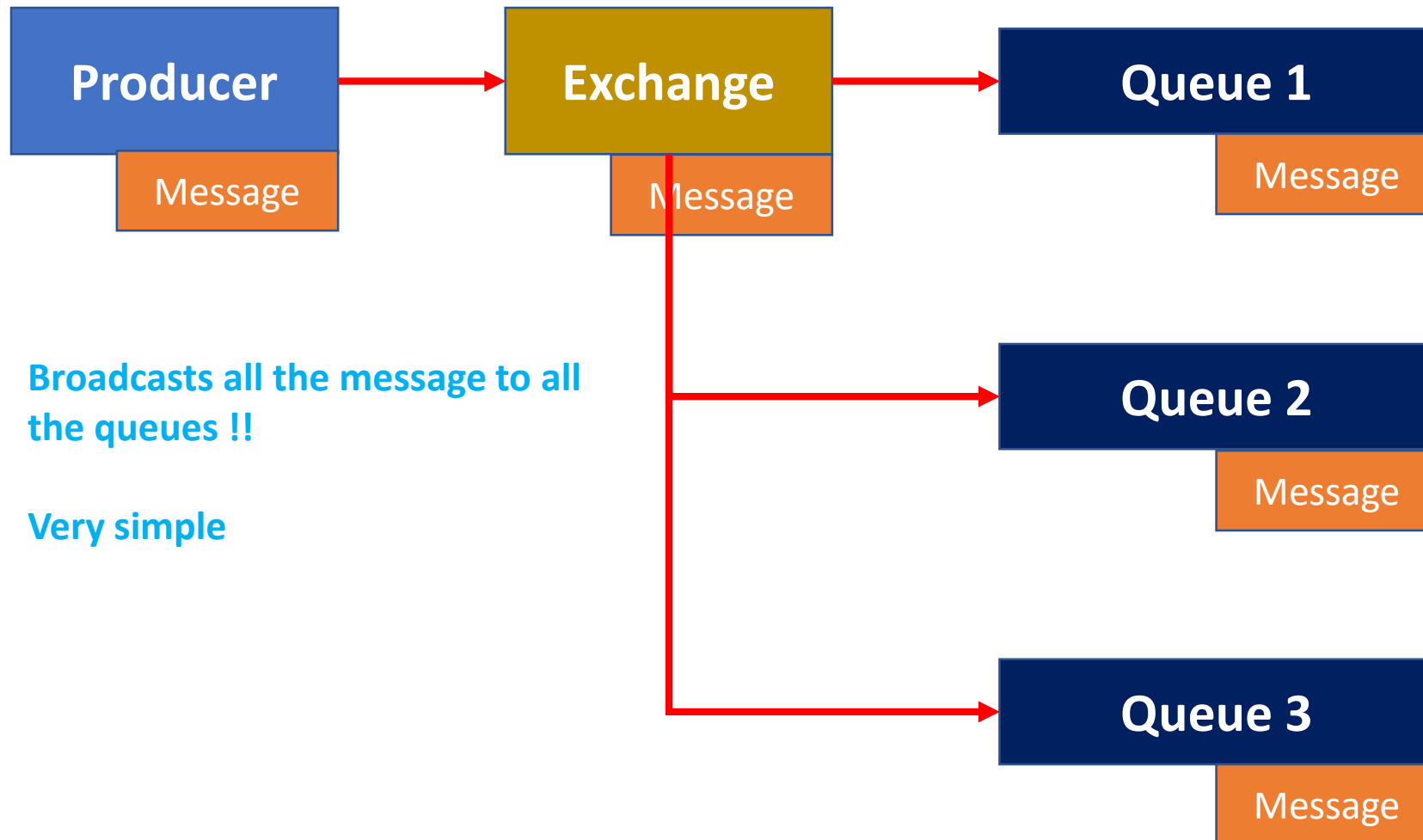
Exchange must know exactly what to do with a message it receives

- Which queue should get the message ?
- Should the message be sent to multiple queues ?
- Should the message be discarded

Rules are defined by the Exchange Type

- Direct
- Topic
- Headers
- Fanout

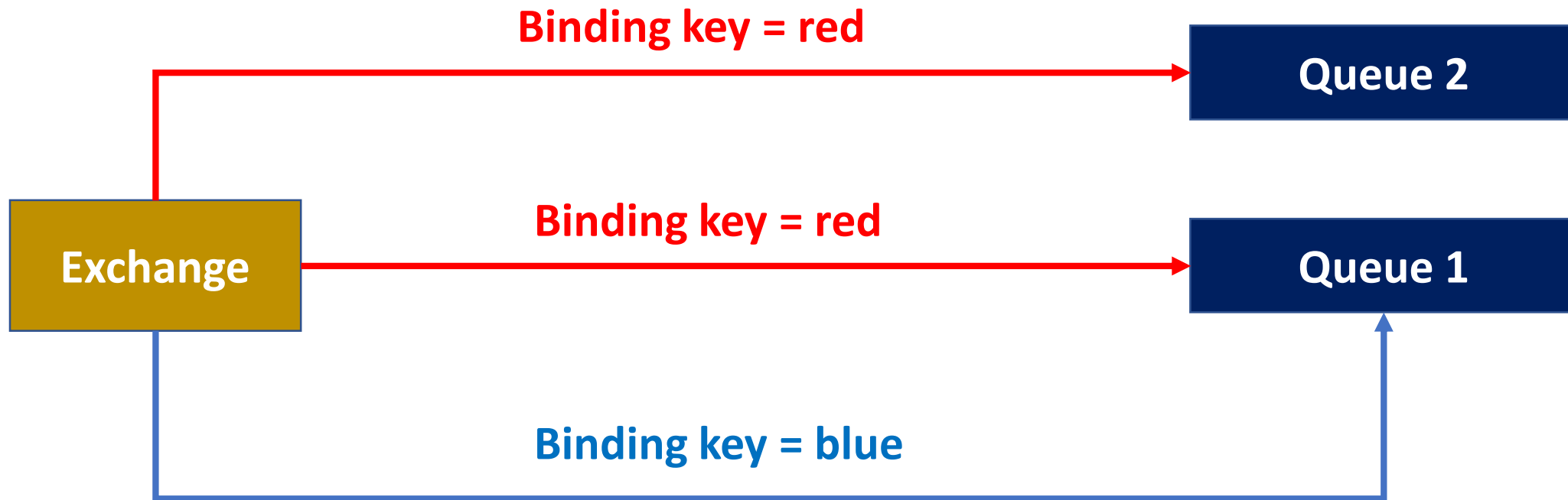
Fanout Exchange



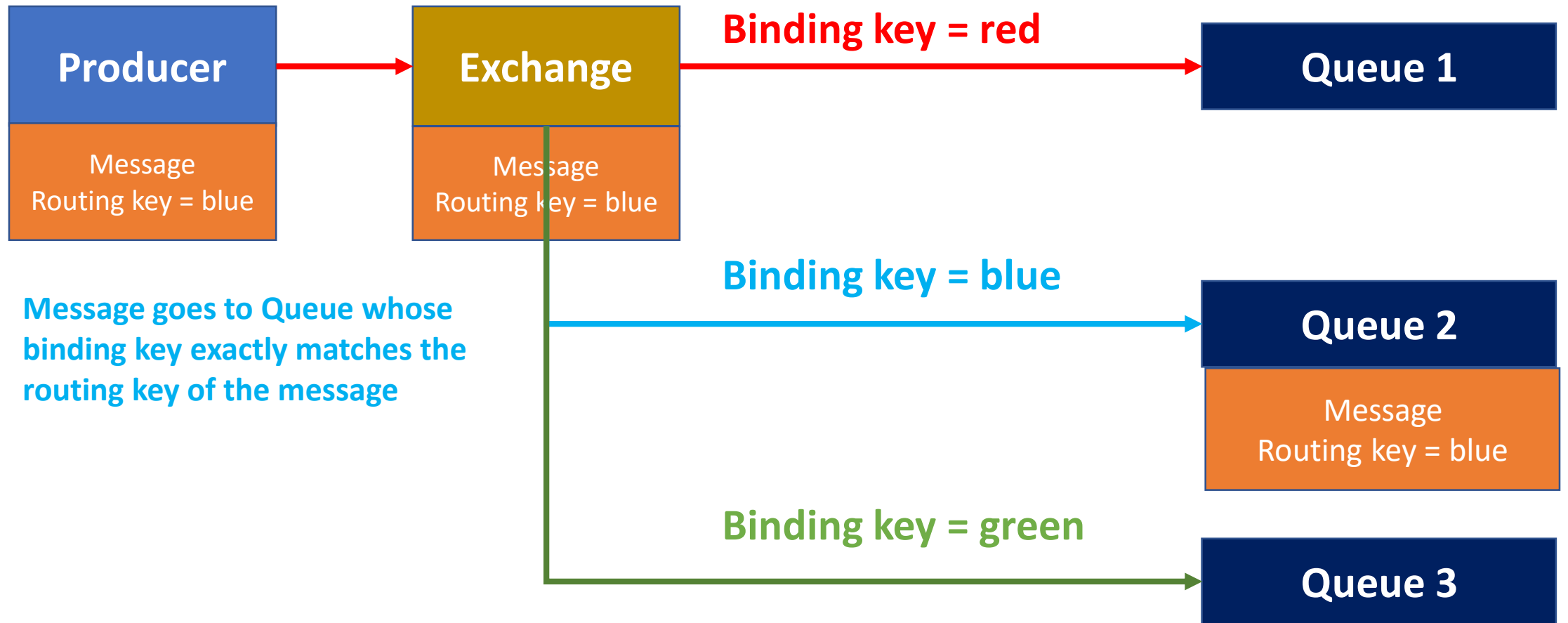
Common Terminologies

Binding : Relation between Exchange and Queue

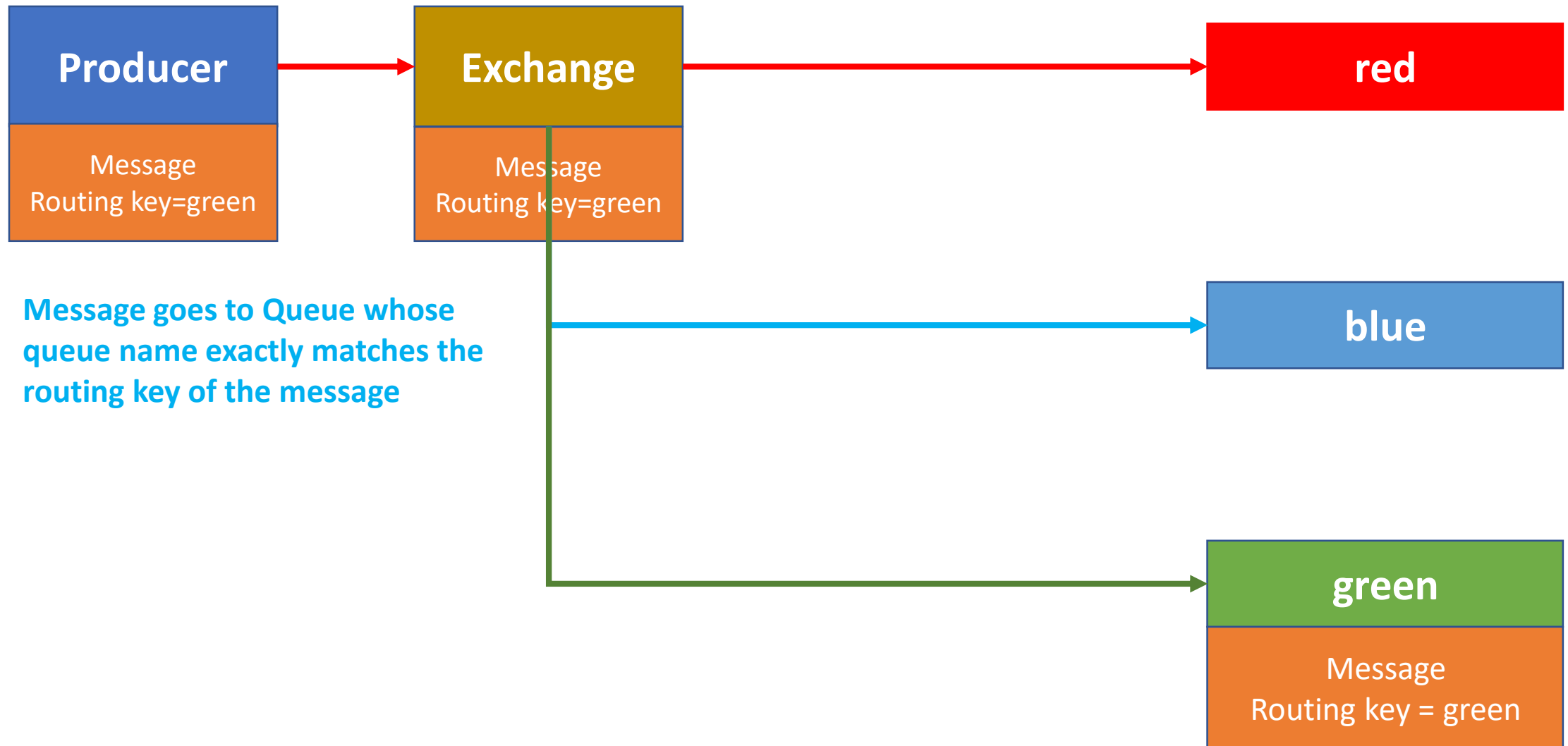
Binding Key : The key that binds the Exchange and Queue



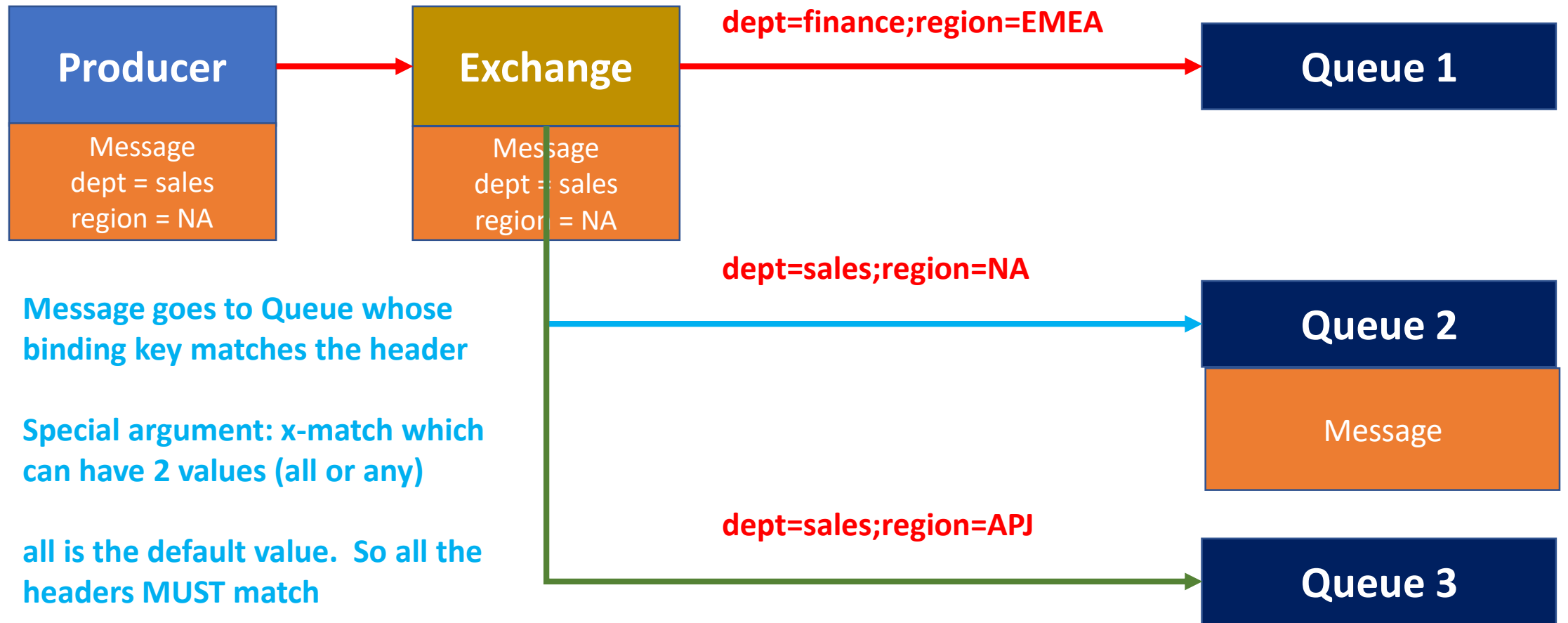
Direct Exchange



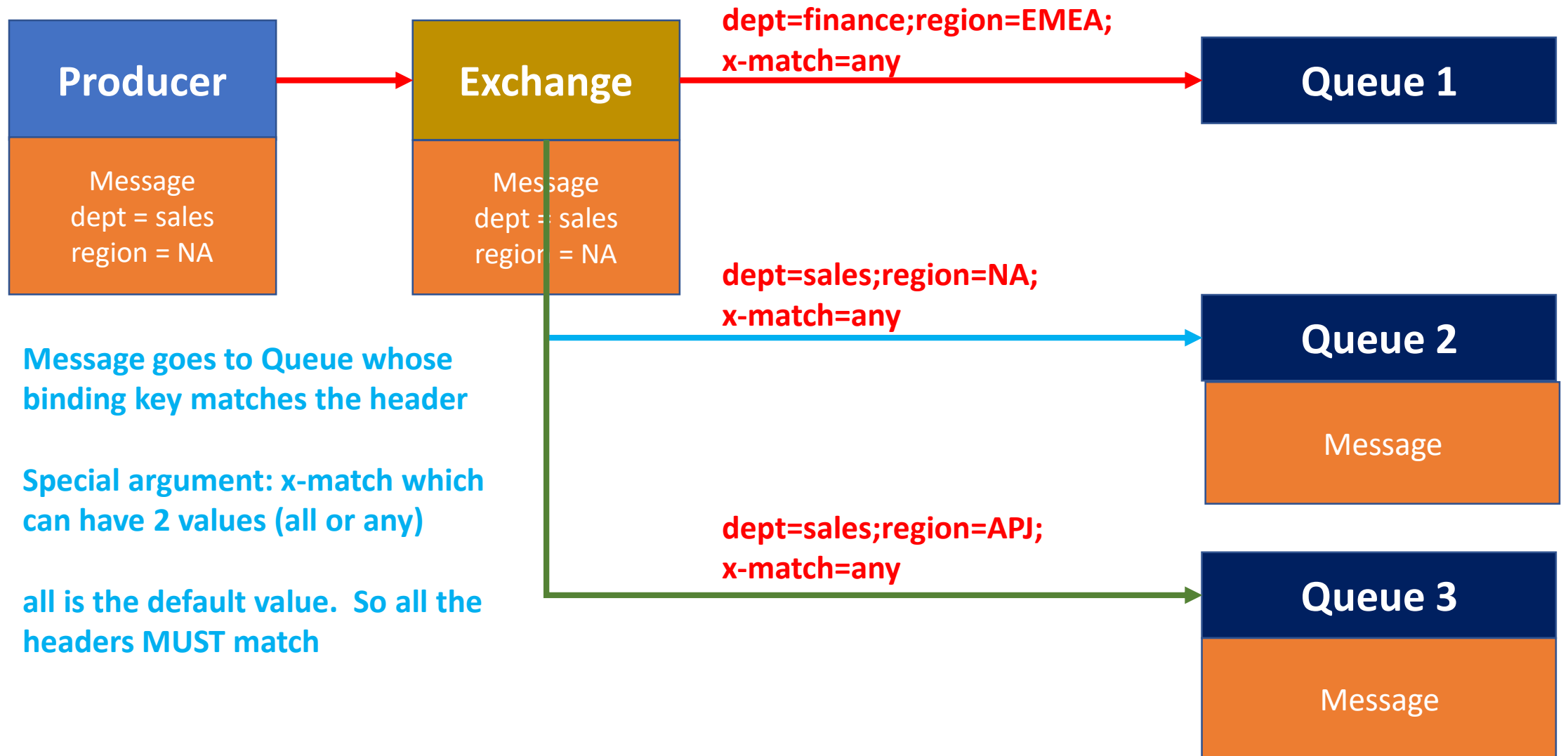
Direct Exchange (Nameless – Default)



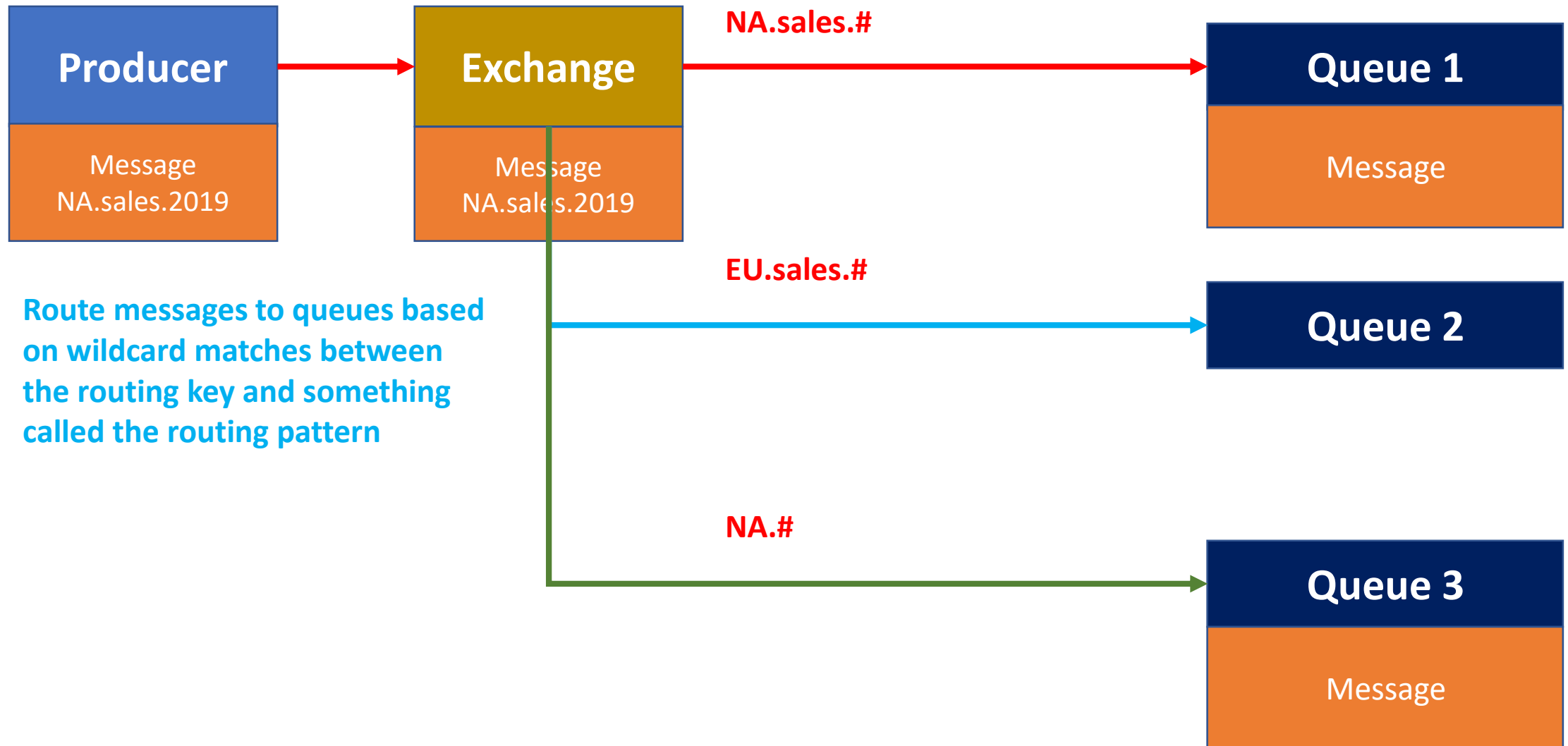
Header Exchange



Header Exchange (Contd.)



Topic Exchange



Header Exchange

