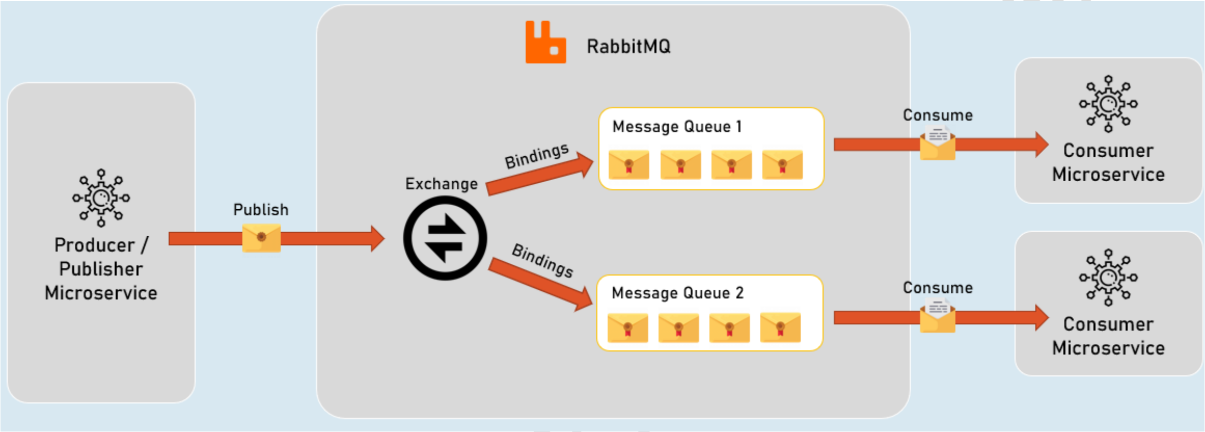
.NET Core Microservices – True Ultimate Guide

Section 11 – Async Microservice Communication using RabbitMQ – Cheat Sheet

Introduction to RabbitMQ

RabbitMQ is an open-source message broker that facilitates the exchange of messages between microservices to implement indirect / asynchronous communication.



Terminology

Producer/Publisher Microservice: Publishes messages to an exchange.

Exchange: Routes messages to the appropriate queues based on bindings.

Bindings: Connections between routing keys (events or actions) and message queues within a specific exchange.

Message Queues: Hold the messages until they are consumed.

Consumer Microservices: Consume (read) messages from the queues.

RabbitMQ Exchanges

Fanout Exchange

Routes messages to all queues bound to the exchange.

Direct Exchange

Routes messages to queues based on an exact match between the message's routing key and the queue's binding key.

Headers Exchange

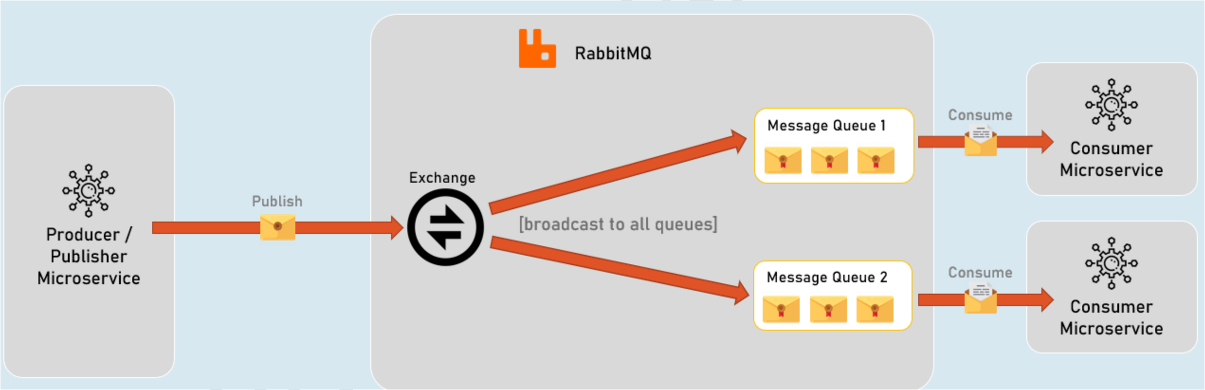
Routes messages based on matching message headers (metadata), instead of the routing key.

Topic Exchange

Routes messages to queues based on pattern matching between the routing key and the binding key using wildcards (* for a single word and # for zero or more words).

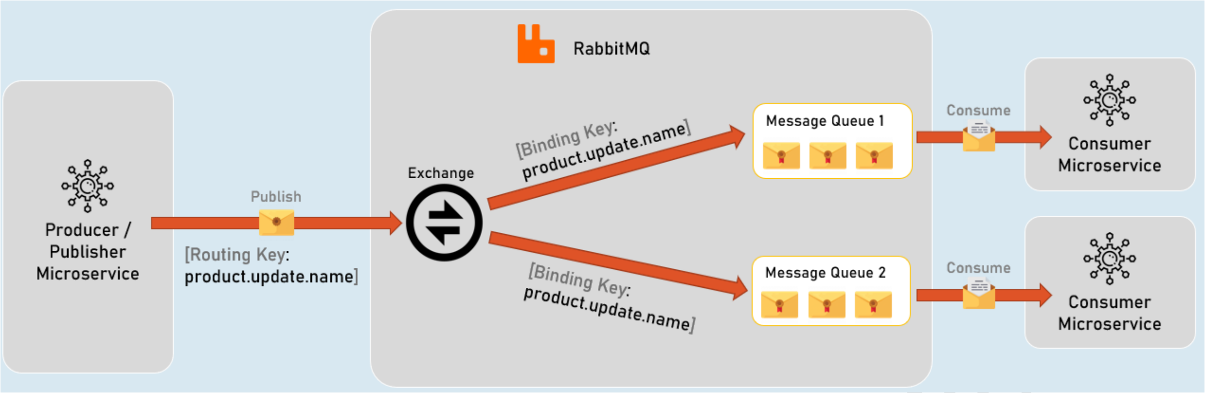
Fanout Exchange

Routes messages to all queues bound to the exchange.



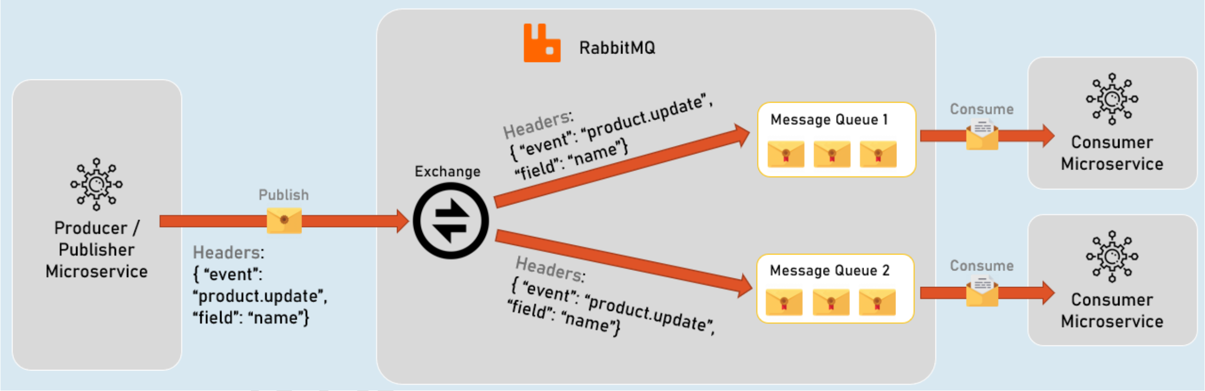
Direct Exchange

Routes messages to queues based on an exact match between the message's routing key and the queue's binding key.


RabbitMQ
Producer / Publisher Microservice
Publish
[Routing Key: product.update.name]
Exchange
[Binding Key: product.update.name]
[Binding Key: product.update.name]
Message Queue 1
Message Queue 2
Consume
Consume
Consumer Microservice
Consumer Microservice

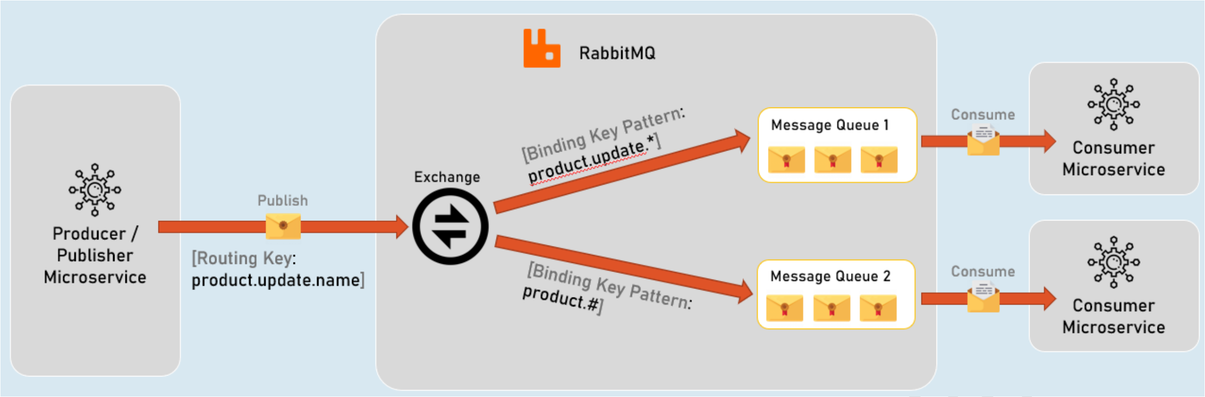
Headers Exchange

Routes messages based on matching message headers (metadata), instead of the routing key.


RabbitMQ
Producer / Publisher Microservice
Publish
Headers: { "event": "product.update", "field": "name"}
Exchange
Headers: { "event": "product.update", "field": "name"}
Headers: { "event": "product.update", "field": "name"}
Message Queue 1
Message Queue 2
Consume
Consume
Consumer Microservice
Consumer Microservice

Topic Exchange

Routes messages to queues based on pattern matching between the routing key and the binding key using wildcards (* for a single word and # for zero or more words).


RabbitMQ
Producer / Publisher Microservice
Publish
[Routing Key: product.update.name]
Exchange
[Binding Key Pattern: product.update.*]
[Binding Key Pattern: product.#]
Message Queue 1
Message Queue 2
Consume
Consume
Consumer Microservice
Consumer Microservice