RabbitMQ – components, benefits and application

AUTHORSHIP: MARC CERVERA ROSELL
JOSEP LLUÍS LÉRIDA – DISTRIBUTED COMPUTING

BACHELOR'S DEGREE IN COMPUTER ENGINEERING – UNIVERSITY OF LLEIDA

Contents

Introduction

- The main aim of this investigation is to discover RabbitMQ.
- There’re three main essential attributes when designing and developing software to offer a trustworthy product: scalability, resiliency and interoperability.
  - RabbitMQ complains each one of these attributes.
- RabbitMQ is an open-source message broker, sometimes called message-oriented middleware.
- “Advanced message queuing protocol” (AMQP) as the core architecture.
- RabbitMQ is a cross-platform software and the code is released under the Mozilla public license.
- Simple definition for RabbitMQ ➔ Defines queues that will store messages until a consumer app gets these messages and process it.

Architecture

- RabbitMQ was implemented for supporting AMQP.
  - Extended by a plug-in architecture support ➔ STOMP, MQTT, HTTP, WebSockets and RabbitMQ streams.
- HTTP is not actually a messaging protocol ➔ RabbitMQ can transmit messages over HTTP in three ways:
  - Using the web STOMP plug-in.
  - Using the MQTT plug-in.
  - Using the Management plug-in ➔ Supports a simple HTTP API to send and receive messages.

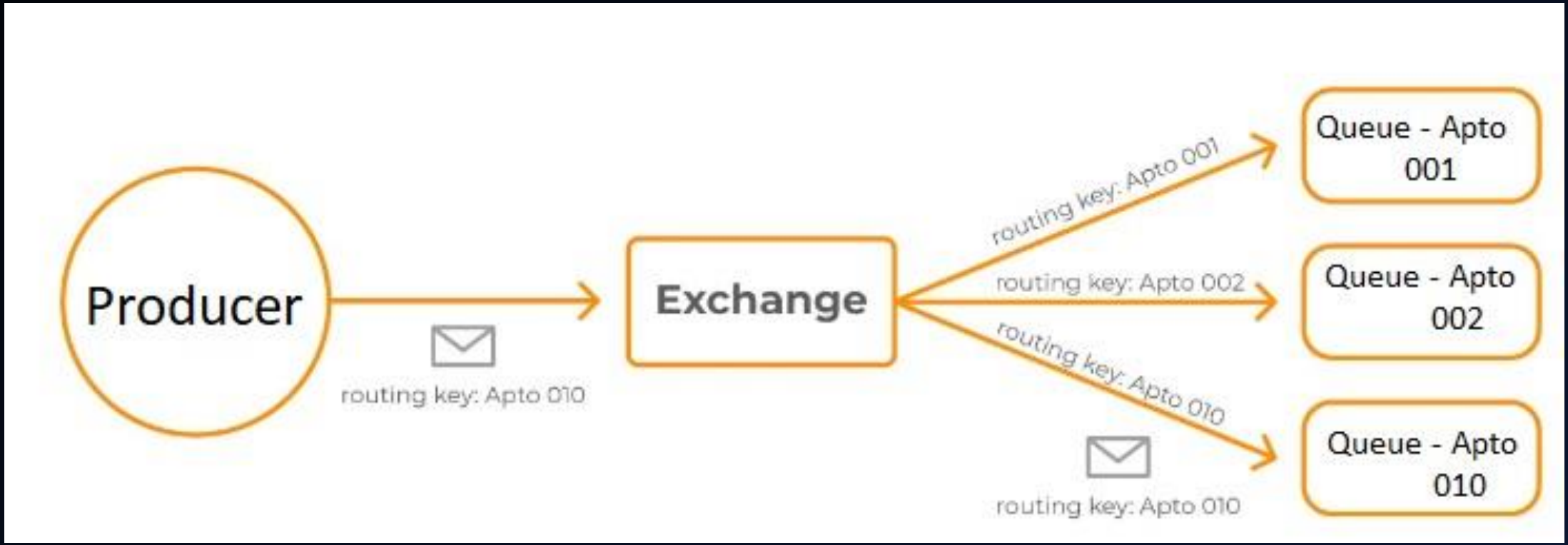
Architecture - AMQP

- Is the core protocol supported by the broker.
- AMQP is a binary protocol and defines quite string messaging semantics.
  - Reasonably simply to implement for clients.
- AMQP stands out for its fidelity.
  - Used by big corporations that have to process million of messages.
- Before AMQP were other message-oriented middleware, such is, JMS, but AMQP has become the standard one.

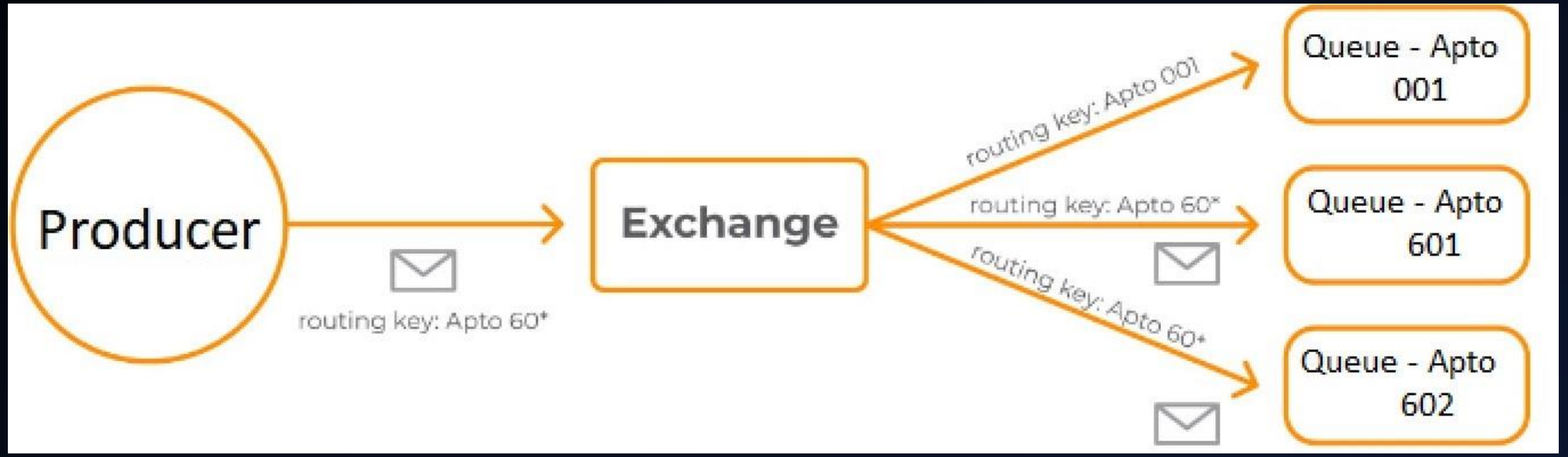
Architecture – AMQP – Exchange

- Is in charge of receiving the messages that have been sent by the producers and its responsible of place them in the proper queue according to a routing key.
  - This means that a producer sends the messages to an exchange not to a queue. Is the exchange who sends the messages to the queues depending on a routing key.
- If a producer wants to send a message to more than one queue, the exchange is responsible for distributing this message to each one of the queues.

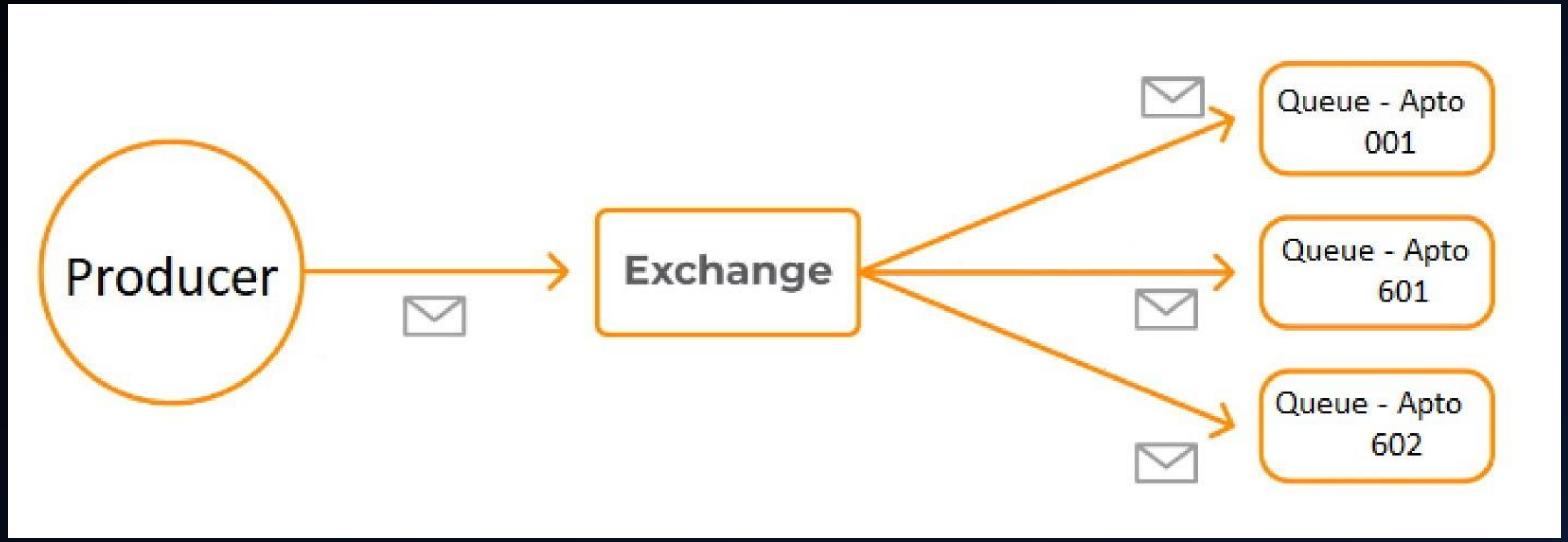
Architecture – AMQP – Exchange



Architecture – AMQP – Exchange



Architecture – AMQP – Exchange



Architecture – AMQP – Routing key

- Identifier that uses the exchange to know the route.
- Identifier that uses a queue to know with which exchange is associated with.

Architecture – AMQP – Queue

- Data structure that imitates real queues as the ones we can see in the mailbox.
- A message queue is an asynchronous communication way that is used on microservices architectures.
  - The messages are stored in the queue until are received and deleted being every message processed only once.

Architecture – AMQP – Binding

- Is a relationship between an exchange and a queue.
  - In an easy way: The queue is interested in messages from this exchange.
- Bindings can take an extra routing key.

```
Channel.queue_bind(exchange = exchange_name, queue = queue_name,
routing_key = ‘black’)
```

- The fanout exchanges ignore the `routing_key` value.

Architecture – AMQP – Virtual host

- VH refers to the practice of running more than one web site on a single machine.
- RabbitMQ's VH are very similar to the Apache's.
  - The difference ➔ In Apache, VH are defined in the configuration file. In RabbitMQ are created and deleted using *rabbitmqctl* or HTTP API.
- When an AMQP client connects to RabbitMQ, it specifies a VH name to connect to. If the username is not granted permissions, the connection is refused.

Applications

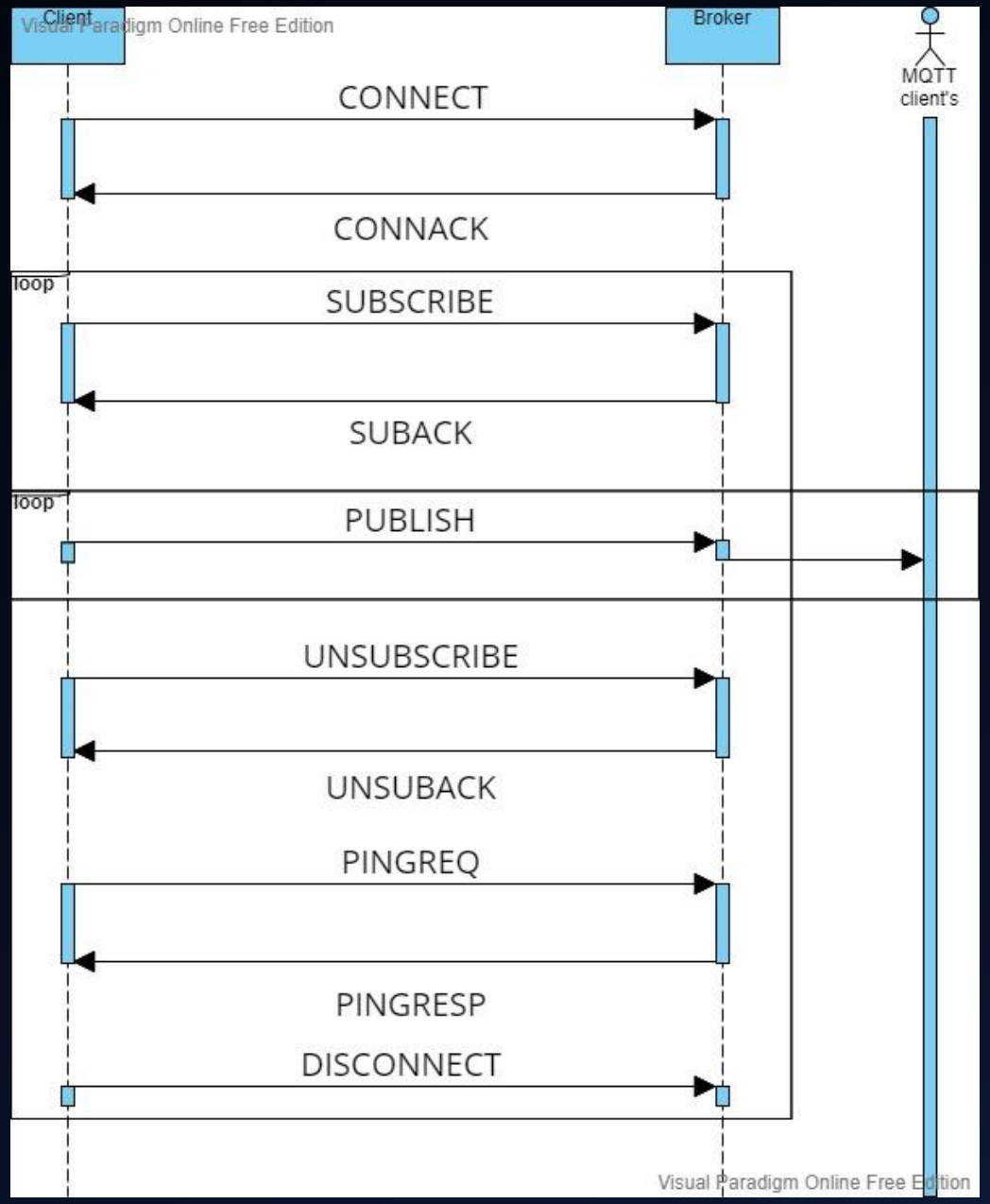
- Some protocols are going to be seen.
  - These protocols are plug-ins.
- RabbitMQ can work properly without this set of protocol, but cannot work without AMQP

Applications - MQTT

- Based on the TCP stack as the communication base.
- Is a push messaging service with the pub-sub pattern.
  - In this type of infrastructures the clients they connect with a central broker.
- To filter the messages that are sent to each client, the messages are arranged in topics hierarchically organized. A client can publish a message in a specific topic. Other clients can subscribe to this topic and the broker will send the subscribed messages.
- The clients open a TCP/IP connection with the broker until the client ends it.

APPLICATIONS - MQTT

Simple example of MQTT protocol through a SSD.



Applications - STOMP

- “Simple Text Oriented Message Protocol”
- Provides an interoperable wire format so that STOMP clients can communicate with any STOMP broker to provide easy messaging interoperability among many languages, platforms and brokers.
- STOMP is a very simple and easy to implement protocol, coming from the HTTP school of design.
  - The server is hard to implement well.
  - Very easy to implement a client (in a couple of hours)

Applications - WebSockets

- Is a communication protocol over a TCP connection.
- WebSockets are designed to work over HTTP ports 443 and 80 as well as to support HTTP proxies and intermediaries thus making it compatible with HTTP.

Applications – RabbitMQ streams

- The RabbitMQ streams protocol allows communicating with streams.
  - The RabbitMQ stream is a Java library to communicate with the RabbitMQ stream plug-in.
  - It allows creating and delete streams, as well as to publish and consume from these streams.
- A stream is a sequence of elements from a source that supports data processing operations.
  - Sequence of elements: while collections are about data, streams are about computations.
  - Source: streams consume data from a providing source.

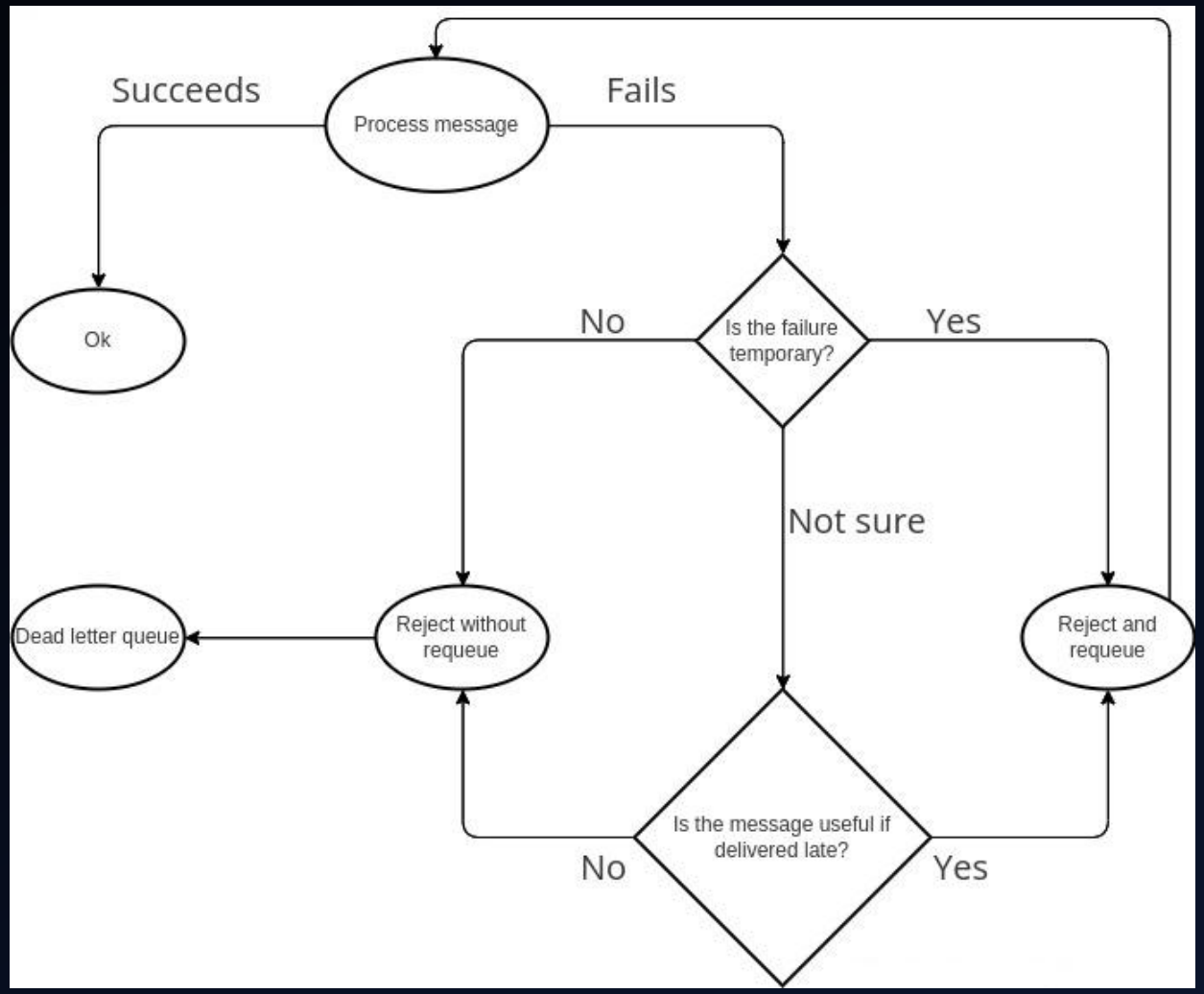
RabbitMQ in the market

- This section has three parts:
  - Advantages of RabbitMQ
  - Disadvantages of RabbitMQ
  - Competition in the market

RabbitMQ in the market - Advantages

- Decoupling in time and in space.
  - Integration of apps through messages in an asynchronous way and from various locations.
- Reliability.
  - Several characteristics that guarantee the delivery of the messages.
- Cluster creation.
  - Sometimes, RabbitMQ has to be able to process more than thousands of messages per second without impacting on the app's performance. For this, RabbitMQ allows the creation of clusters to scale horizontally the solution
- Security.
  - RabbitMQ uses the Transport Layer Security (TLS) protocol.
    - Is a cryptographic protocol that provides secure connections through a network
- Available.
  - The queues can be replicated in several nodes of the cluster and in case of failure of a node the broker can keep receiving and sending messages.
- Affordable: Completely free.
- Fault-tolerance (next slide)

RabbitMQ in the market - Advantages



RabbitMQ in the market - Advantages

- Complexity: RabbitMQ is user-friendly and it’s easy to modify the configurations to suit the expected porpoise.
- Latency: Has the lowest latency of the market.
- Performance: Has one of the best yields of the market

RabbitMQ in the market - Disadvantages

- Latency: The consumers will balloon in memory as they buffer all the messages in their own RAM.
  - A big buffer results in a lot of extra latency if the network performs normally
  - And huge amounts of extra latency if the client suddenly starts taking longer to process the messages.
- Resource sharing: The management plug-in has to be enabled. ➔ By default the UI app will refuse to access to websites hosted on origins different from its own.
- The AMQP need a certain processing capacity.
- RabbitMQ is implemented in Erlang.

RabbitMQ in the market – Competition in the market

Conclusions

1. RabbitMQ can support a lot of messaging protocols.
2. RabbitMQ is uncomplicated to use, reliable, scalable, available, secure, affordable, fault-tolerant and efficient.
3. There are many libraries available to implement it.
4. The exchange is a key element in a message-broker.
5. Many similarities between Apaches' VH and RabbitMQ's VH.
6. Every message has an "id" that makes it unique.
7. Is not perfect, but there are more advantages than disadvantages.
8. Has a lot of competition in the market, but is the most chosen.

References

- https://en.wikipedia.org/wiki/RabbitMQ
- https://www.rabbitmq.com
- https://www.sdos.es/blog/microservicios-mensajes-spring-rabbitmq
- https://www.pragma.com.co/academia/lecciones/conozcamos-sobre-rabbitmq-sus-componentes-y-beneficios
- https://www.researchgate.net/publication/325119432/figure/fig5/AS:626093459505153@1526283721309/AMQP-architecture-34.png
- https://www.rabbitmq.com/protocols.html
- https://www.rabbitmq.com/vhosts.html
- https://www.luisllamas.es/que-es-mqtt-su-importancia-como-protocolo-iot/
- https://en.wikipedia.org/wiki/WebSocket
- http://stomp.github.io
- https://programmerclick.com/article/80671335987/
- https://blog.devgenius.io/scalable-system-implementation-using-rabbitmq-java-and-mysql-2d5fe0fa182e
- https://blog.rabbitmq.com/posts/2012/05/some-queuing-theory-throughput-latency-and-bandwidth/
- https://medium.com/codait/handling-failure-successfully-in-rabbitmq-22ffa982b60f
- https://www.rabbitmq.com/management.html
- https://geekflare.com/es/top-message-brokers/