

**Project Report**

Data Engineering – Big Data Architecture

Rabbit MQ

Ms. Peenal Gupta (11037400)

Applied Data Science and Analytics

**Rabbit MQ**

**Introduction**

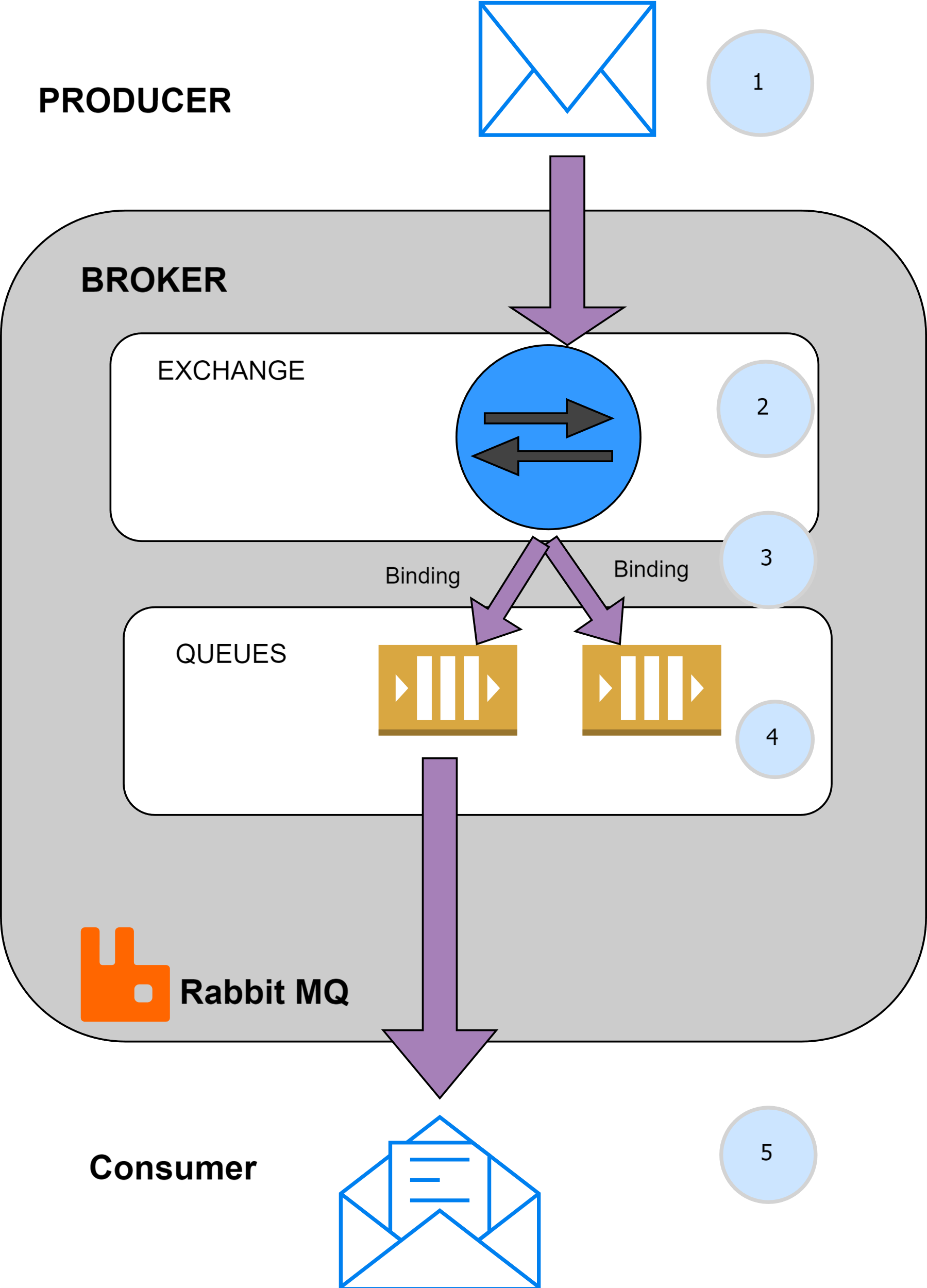
Rabbit MQ is a message broker system that facilitates communication between distributed systems. It is a message-oriented middleware that functions as a message broker or queue manager. It provides a platform for applications to connect and exchange messages in the form of queues. By doing so, RabbitMQ enables applications to communicate and process data asynchronously.

**RabbitMQ Architecture**

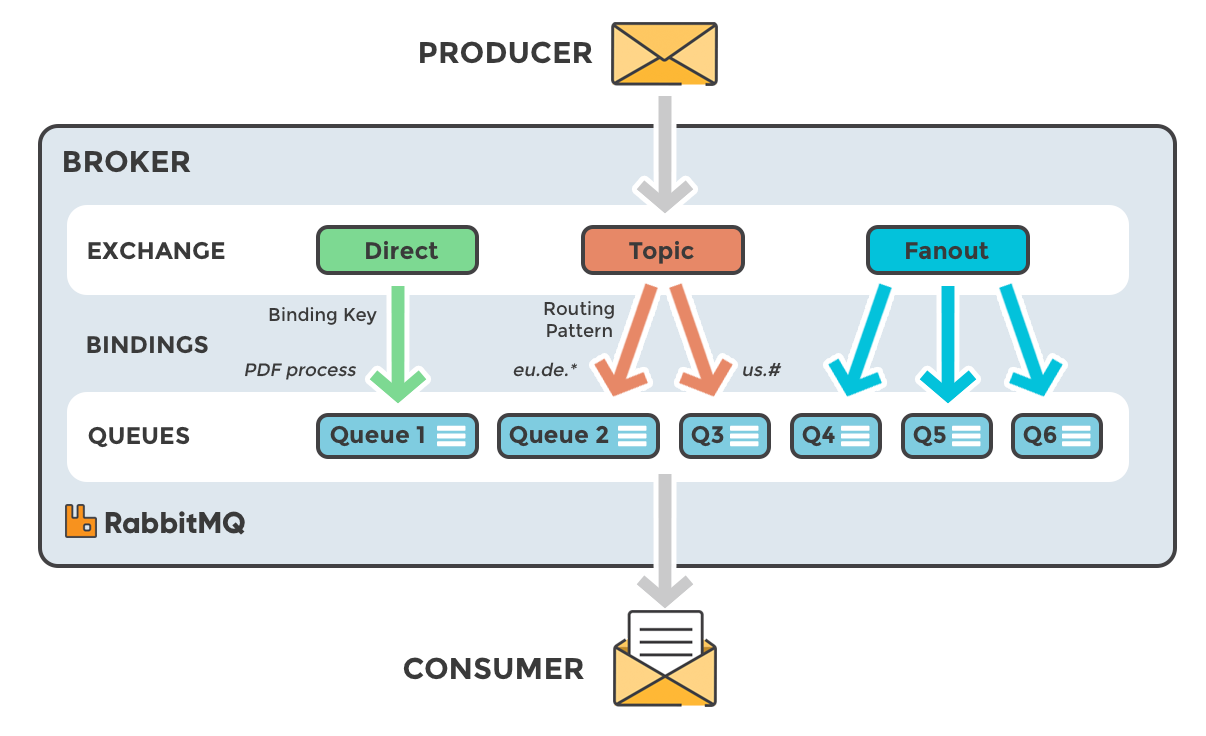


**RabbitMQ EXCHANGES**

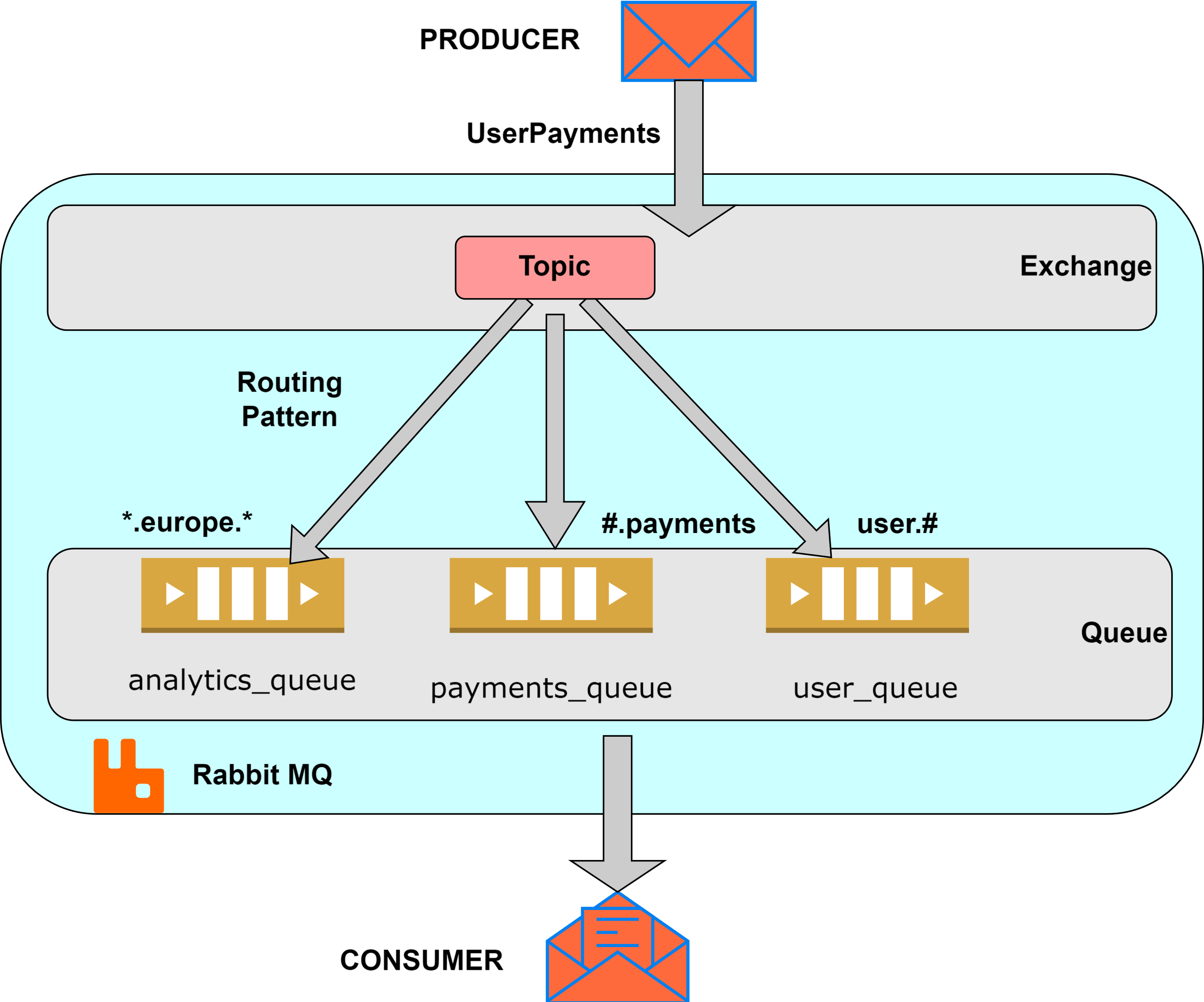
Producers do not directly publish messages to a queue, instead they are sent to an exchange. Exchanges are used to route messages to different queues through bindings and routing keys. Bindings are what connects a queue to an exchange.



**Types of Exchanges**



**My Solution – Topic Exchange**



The image illustrates an architecture using RabbitMQ with a topic exchange. **Exchange Type**: The exchange is of type **Topic**.

Topic exchanges route messages to queues based on wildcard matches between the routing key and the routing pattern specified in the binding.

* **Message Flow**:
  + Messages are sent to the Topic exchange.
  + Based on the routing key, the message is routed to one or more queues.
* **Queues and Routing Patterns**:
  + **analytics\_queue**:
    - **Routing Pattern:** \*.europe.\*
    - Receives messages that have a routing key matching the pattern where the second word is europe.
  + **payments\_queue**:
    - **Routing Pattern:** #.payments
    - Receives messages that have a routing key ending in .payments, regardless of the preceding words.
  + **user\_queue**:
    - **Routing Pattern:** user.#
    - Receives messages that have a routing key starting with user. followed by zero or more words.
* **Message Delivery**:
  + A message is published with a specific routing key.
  + The exchange uses the routing key to determine which queues should receive the message.
  + Messages are then delivered to the appropriate queues as per the routing patterns.
* **RabbitMQ Broker**:
  + The entire setup is managed by RabbitMQ, which is the message broker handling the message exchange and queue management.

This architecture supports routing messages to different consumers based on flexible routing keys, enabling complex and dynamic routing scenarios.

A screenshot of a computer

Description automatically generated