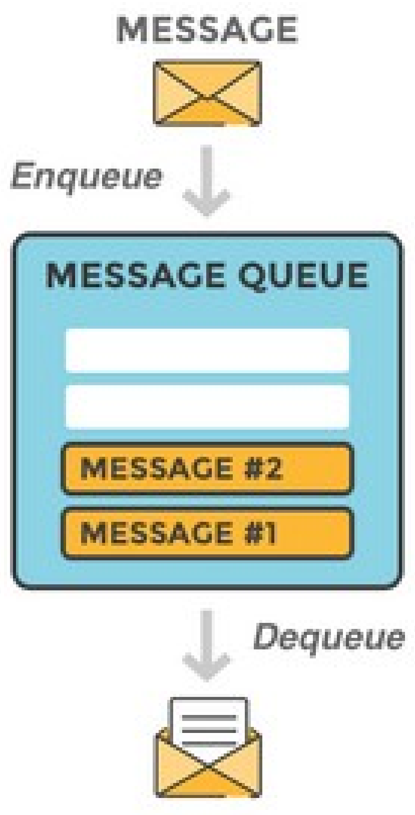
10 – RabbitMQ

- https://www.cloudamqp.com/blog/2015-05-18-part1-rabbitmq-for-beginners-what-is-rabbitmq.html
- https://www.rabbitmq.com/management.html
- https://www.rabbitmq.com/getstarted.html

RabbitMQ

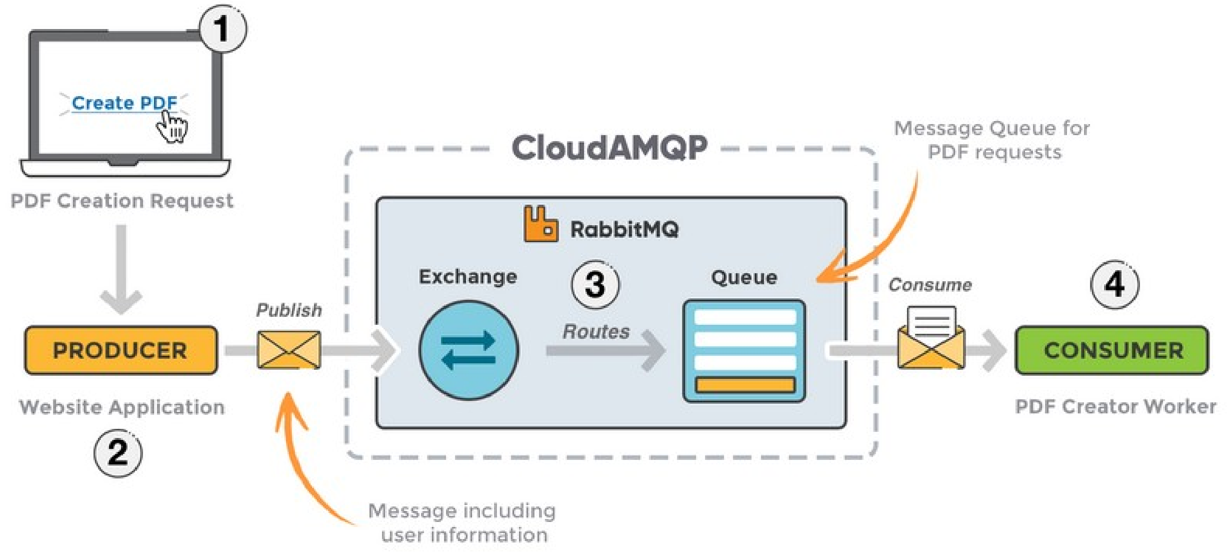
- Message-queueing software
  - message broker or queue manager.
- queues can be defined
- applications may connect to the queue and transfer a message onto it.



When to use

- Allows fast respose instead of being forced to perform resource-heavy procedures
- Is also good when you want to distribute a message to multiple recipients for consumption
- Is also good when you want to balance loads between workers.
- Is good to add low coupling between the sender and the receiver.

When to use



- The consumer can take a message of the queue and start the processing of the PDF
  - at the same time as the producer is queueing up new messages on the queue.
- The consumer can be on a totally different server than the publisher,
  - or they can be located on the same server.
- The request can be created in one programming language and handled in another programming language

It uses the AMQP protocol

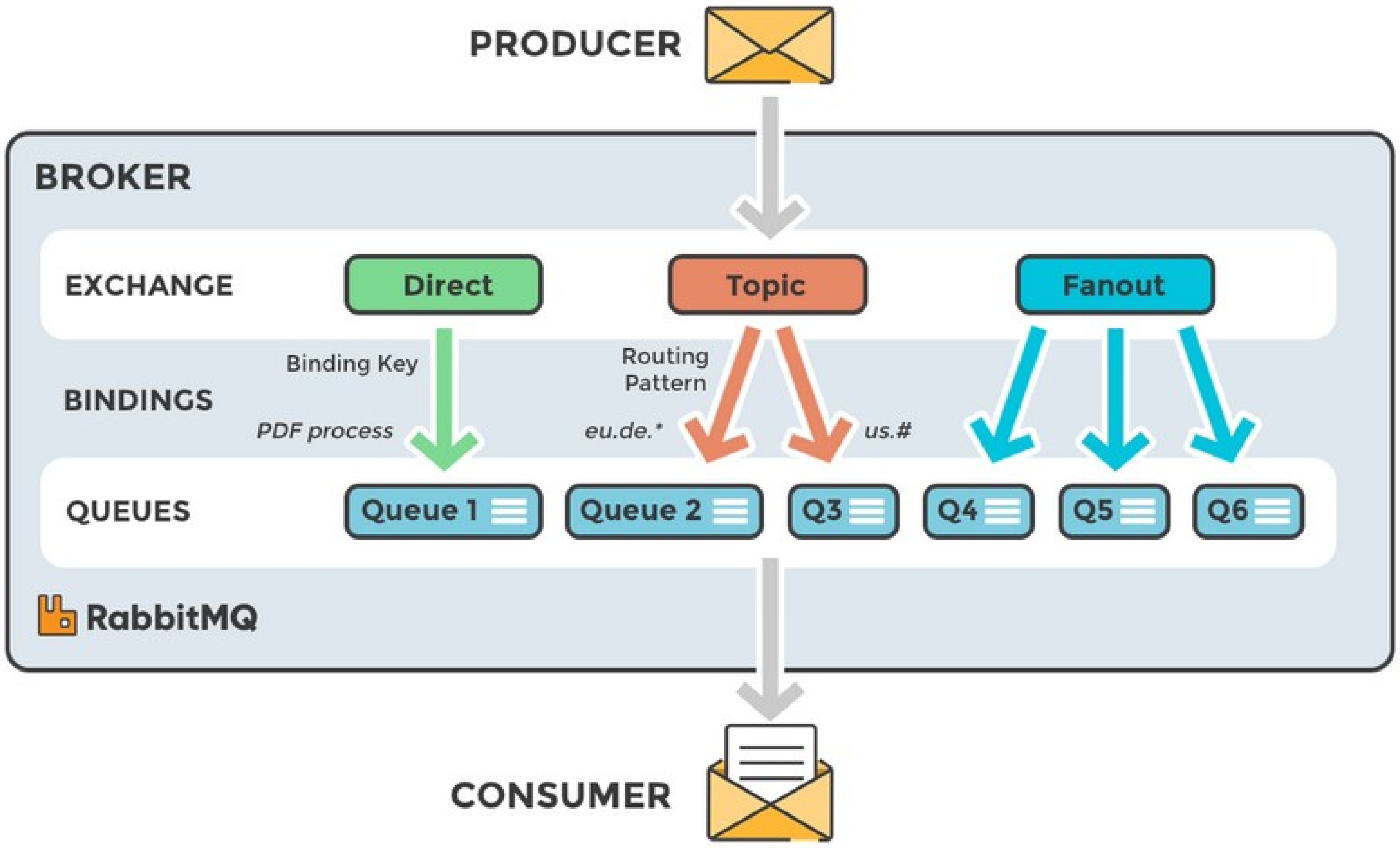
- Advances Message Queueing Protocol
  - First developed at JPMorgan Chase
  - Began as a collaborative effort and eventually involved Bank of America, RedHat, Cisco, Credit Suisse, Goldman Sachs, Microsoft and Novell -- which became known as the AMQP Working Group
- Conceived for trading and risk management systems

Concepts

- Producer:
  - Application that sends the messages.
- Consumer:
  - Application that receives the messages.
- Queue:
  - Buffer that stores messages.
- Message:
  - Information that is sent from the producer to a consumer through RabbitMQ.
- Connection:
  - A connection is a TCP connection between your application and the RabbitMQ broker.
- Channel:
  - A channel is a virtual connection inside a connection. When you are publishing or consuming messages from a queue - it's all done over a channel.
- Exchange:
  - Receives messages from producers and pushes them to queues depending on rules defined by the exchange type. To receive messages, a queue needs to be bound to at least one exchange.

Basics

- Producers send Messages with Routing Keys and Exchange Names to Brokers
- Brokers use Exchange rules to route / filter Messages
- Brokers then use Queues to store and forward Messages for Consumers
- Consumers receive Messages from the Broker for known Queues
- A RoutingKey (Producing) is congruent to a QueueName (Consumption)
- Exchanges only matter for routing / filtering rules


PRODUCER
BROKER
EXCHANGE
Direct
Topic
Fanout
BINDINGS
Binding Key
PDF process
Routing Pattern
eu.de.*
us.#
QUEUES
Queue 1
Queue 2
Q3
Q4
Q5
Q6
RabbitMQ
CONSUMER

- Sender only cares about
  - Broker
  - Exchange
  - Message Body (& properties)
  - Routing Key
  - Queue (sometimes)

- Receiver only care about
  - Broker
  - Exchange
  - Bindings
  - Queue
  - Message Body

Main objectives

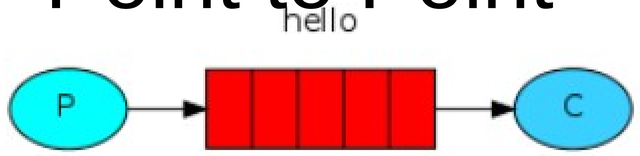
- "Business messaging is provided by infrastructure and not by integration experts"
  - Ubiquity - all the same everywhere
  - Safety - nobody look at my messages
  - Fidelity - trust in the delivery!
  - Applicability - the more you use it, the more payoff
  - Interoperability - easiest integration, evar
  - Manageability - defined wire protocol, easy for reporting / managing

Binding

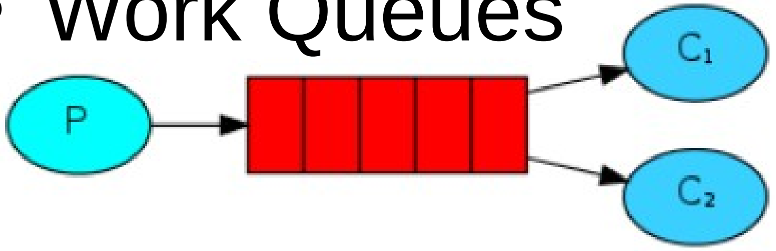
- Binds a queue to a particular exchange
  - Unconditional
    - all messages from the exchange
  - Conditional fixed
    - routingKey must match queueName
  - Conditional pattern
    - routingKey matches some pattern in the queueName
  - Conditional multi-* -
    - routingKey must match a number of names / patterns
  - Conditional algorithm -
    - method using headers to match

Multiple interactions

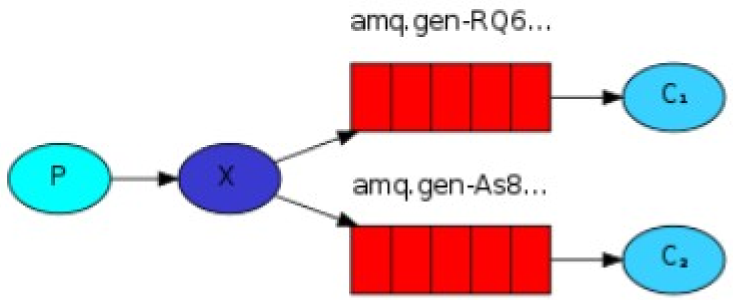
- Point to Point



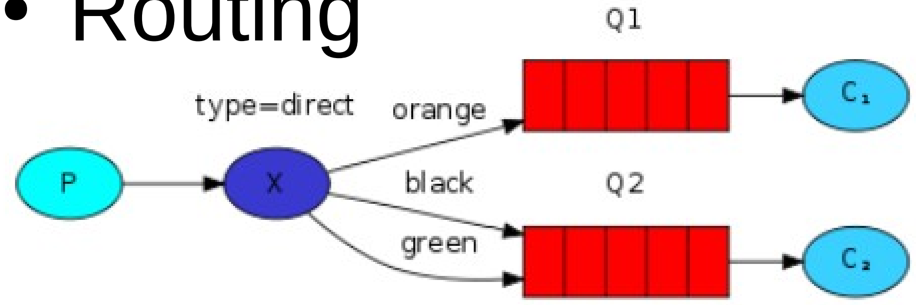
- Work Queues



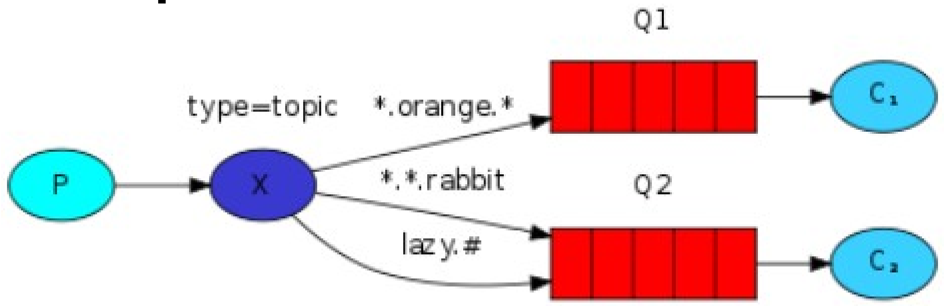
- Publish/Subscribe



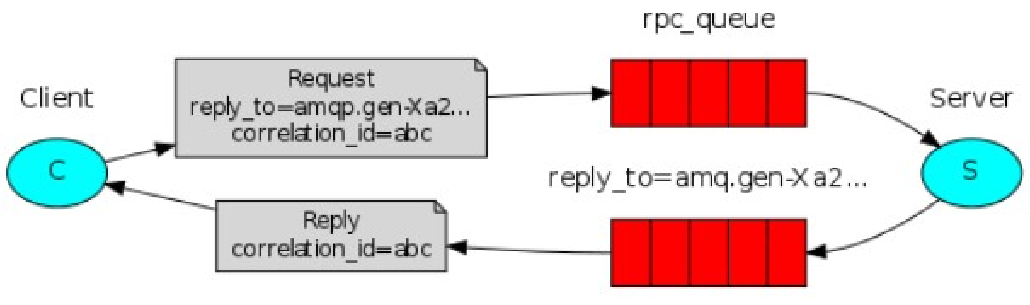
- Routing



- Topics



- RPC



Message Queues Brokers

- Implementation – Server on the network
- Scope - global
- No Duplex
- Time-uncoupling
- Space-uncoupling
- Explicit
- Synchronization – No
- Process relation - unrelated
- Identification – Broker IP address
- API – specific API

Setup

- Install RabbitMQ
  - zypper install rabbitmq-server
  - zypper install rabbitmq-server-plugins
- Enable plugins
  - rabbitmq-plugins enable rabbitmq_management
- Run rabbitmq
  - Rabbitmq-server
- Access console
  - http://localhost:15672/ (guest / guest)
- Install client
  - pip3 install pika