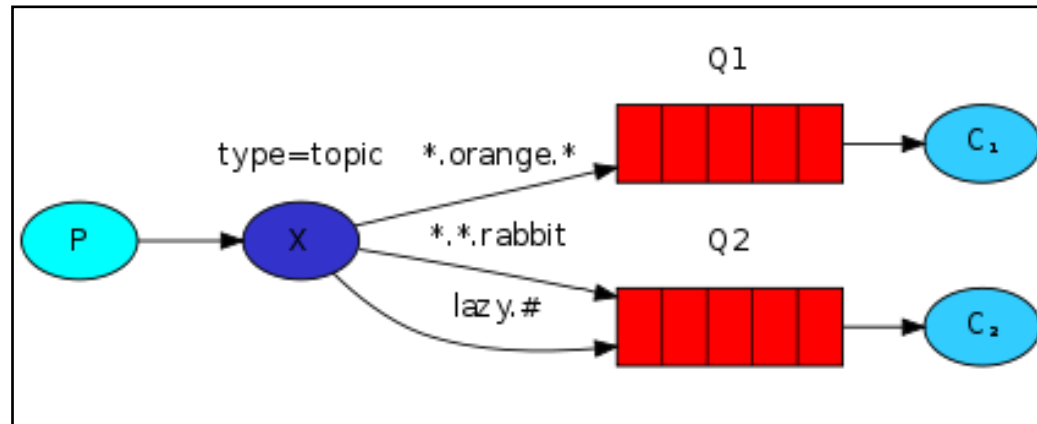
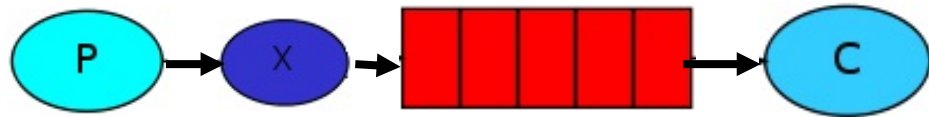
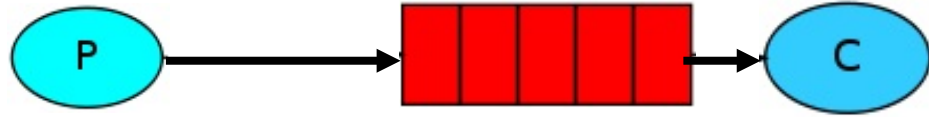


- Qué Es un intermediario de mensajería.
- Para Transmitir mensajes entre aplicaciones o procesos sin importar sus diferencias.
- Por qué Open Source, multiprotocolo, multiplataforma y en evolución.



## Queue me

▼ Overview

Queued messages (chart: last minute) (?)



Message rates (chart: last minute) (?)

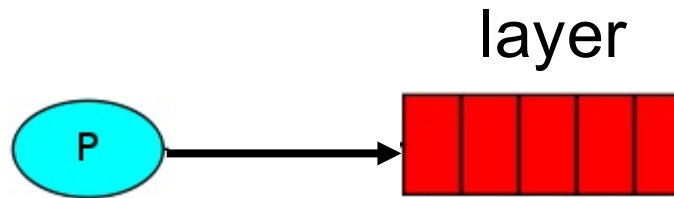


Details

Features	durable: true	State	idle	Total	Ready	Unacked	In memory	Persistent
Policy		Consumers	1	Messages (?)	0	0	0	0
		Consumer utilisation (?)	N/A	Message body bytes (?)	0B	0B	0B	0B
				Process memory (?)	14kB			

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import pika

user = 'usertest'
password = 'password'
credentSend = pika.PlainCredentials(user, password)
hostSend = 'amqp. es'
portSend = 5672
queue = 'layer'
```



```
message = '{ "target": "datagramServer", "vep_data": [ { "layerData": { "_id": "PM4444", "system": "LAYERS", "subsystem": "PUTDATA", "function": "REPLACE", "layer": { "owner": "mobilitylabs.usertest", "type": "SHARED", "name": "BICISENS.ontology" }, ... } ] }'
```

```
connection = pika.BlockingConnection(pika.ConnectionParameters(hostSend, portSend, '/', credentSend))
channel = connection.channel()
# channel.tx_select()
```

```
print "se envia mensaje"
try:
    channel.basic_publish(exchange="",
                          routing_key= queue ,
                          body= message,
                          properties=pika.BasicProperties(delivery_mode = 2, user_id = user))
```

```
    # channel.tx_commit()
    print "Sent"
except Exception as e:
    print "The message can't be published"
    print e
```

```
connection.close()
```

- **PRIVATE** Lectura y escritura sólo para el propietario.
- **PUBLIC** Lectura para todos, escritura sólo para el propietario.
- **SHARED** Lectura y escritura para todos.