Message queue with Python 101

- Aju Tamang

About me

Aju Tamang

- Software engineer trainee at Insight Workshop (remote)
- Freshman CS Student at ASCOL
- Cloud and open source enthusiast
- Love blogging (csaju.com)

Find me at linkedin: https://www.linkedin.com/in/aju-tamang/

The Problem🤔

What this talk is not about ?

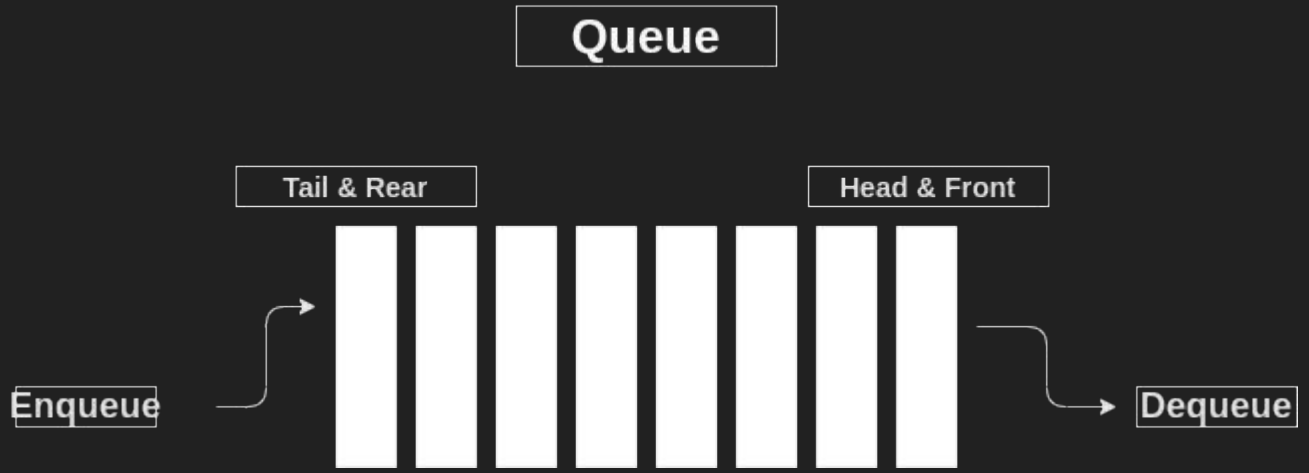
This talk by no means is an exhaustive talk in RabbitMQ, instead this is an 101 guide to help you get started on RabbitMQ with python.

If you are new at Message Broker and wanted to implement in Python, you are at the right place.

Agenda

- Concept of Queue and Message Broker
- Rabbitmq and it's important
- Obviously python and design patterns
- Implementation of Rabbitmq with python
- Resources

Let's refresh our Data structure : Queue



Let's know Message Queue

- Queue of Message (data transported between producer and consumer)

- Think Producer as sender and consumer as receiver connect queue and gets it to be process.

- Uses:

    - asynchronous app to app communication.

    - Helps in managing communication in distributed system

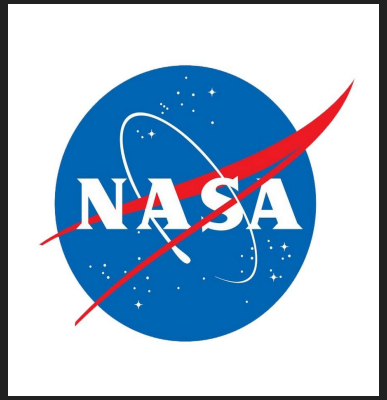
Does every service need MQ ?🤔

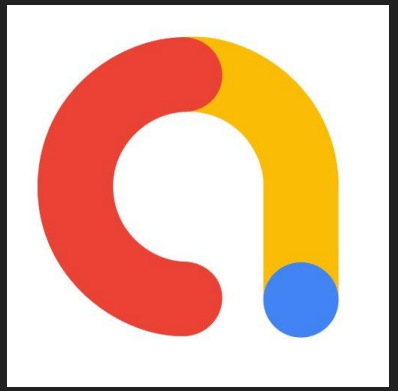
You need message broker (mailman) to implement MQ

Why ?

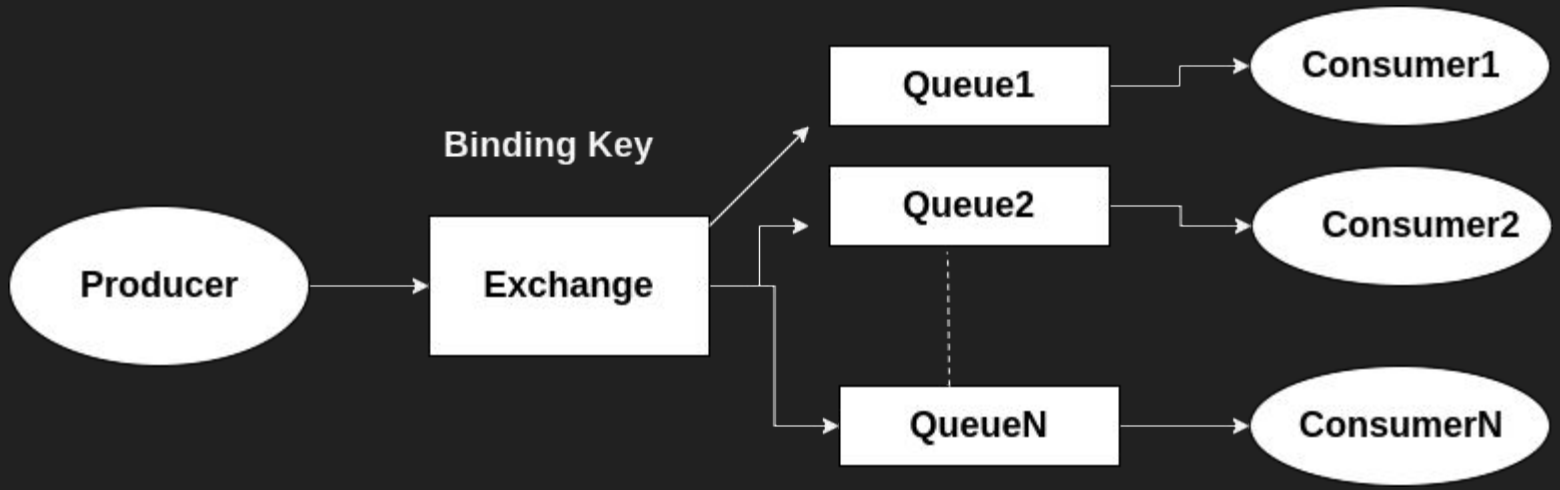
- Flexible for developers
- Acknowledgements for asynchronous messaging
- Lightweight
- Production ready

Who's using Rabbitmq





Understanding simple architecture



Rabbitmq with python - Pika

Producer:

- Establish connection

```
#!/usr/bin/env python
import pika

connection = pika.BlockingConnection(
  pika.ConnectionParameters('localhost')
)
channel = connection.channel()
```

- Declare queue
- Produce

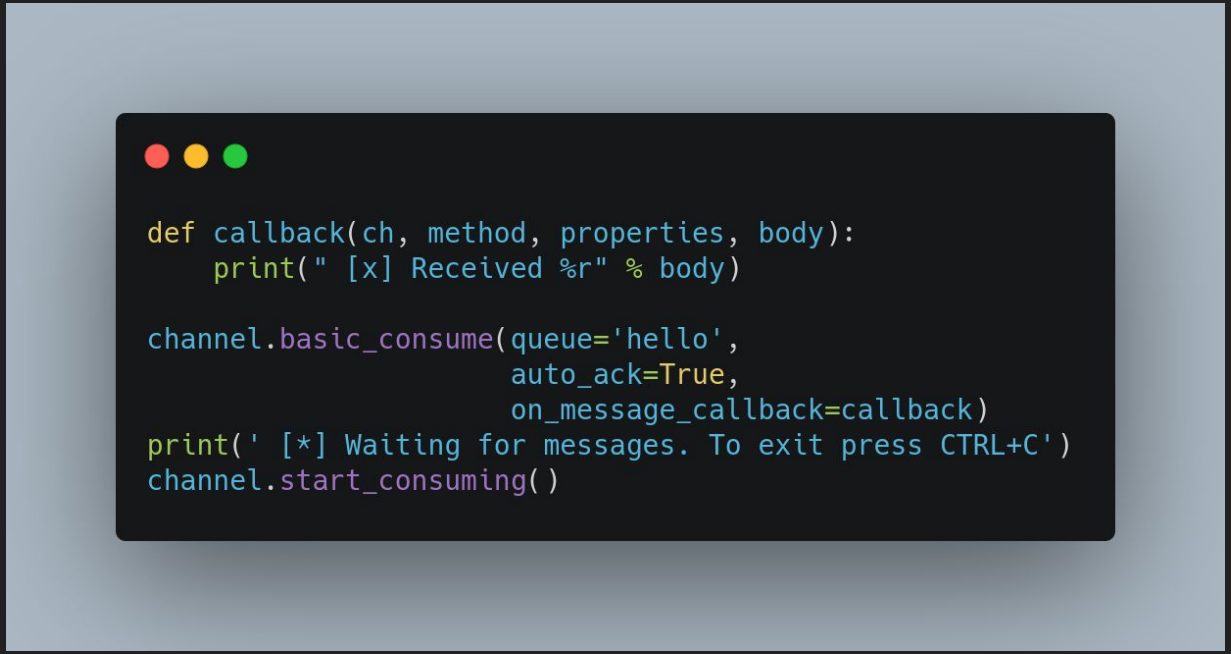
```
channel.queue_declare(queue='hello')

channel.basic_publish(exchange='',
                      routing_key='hello',
                      body='Hello World!')
print(" [x] Sent 'Hello World!'")
```

Consumer:

- Same 1,2 steps

- Call back



```python
def callback(ch, method, properties, body):
    print(" [x] Received %r" % body)

channel.basic_consume(queue='hello',
                      auto_ack=True,
                      on_message_callback=callback)
print(' [*] Waiting for messages. To exit press CTRL+C')
channel.start_consuming()
```

Demo time

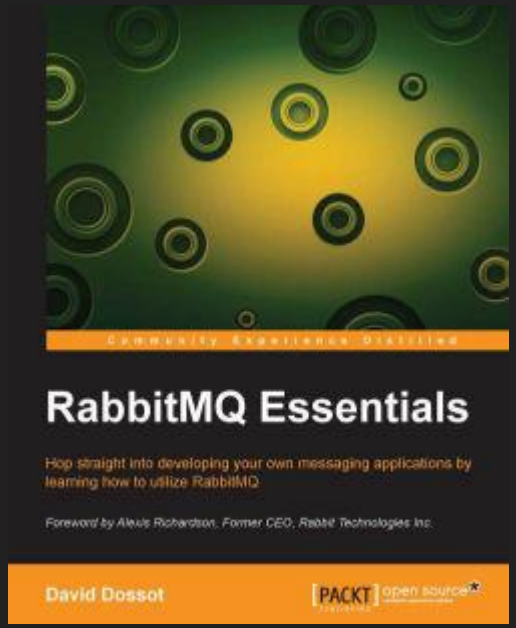
Resources

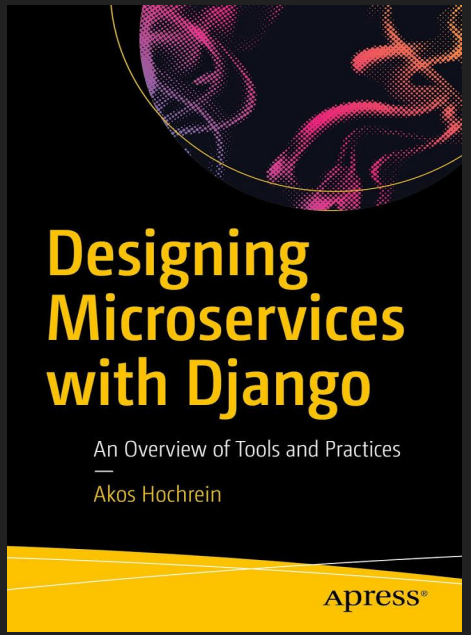
- Official Docs and blogs

- Books

Topics we missed

- Deployment
- Exchange
- Routing
- RPC
- Async





Q&A

Any Questions ?

Thank you for your time