

RabbitMQ for Queue based Messaging



Shubham Sinha

Software Engineer | Servicenow

[linkedin@shubham-sinha-psight](https://www.linkedin.com/in/shubham-sinha-psight)



RabbitMQ Architecture

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

channel.queue_declare('user_email')
channel.basic_publish(exchange='', routing_key='user_email', body=json.dumps(message))
```

Pika RMQ code publish

Sample code to declare and publish a message to RMQ using the routing key



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

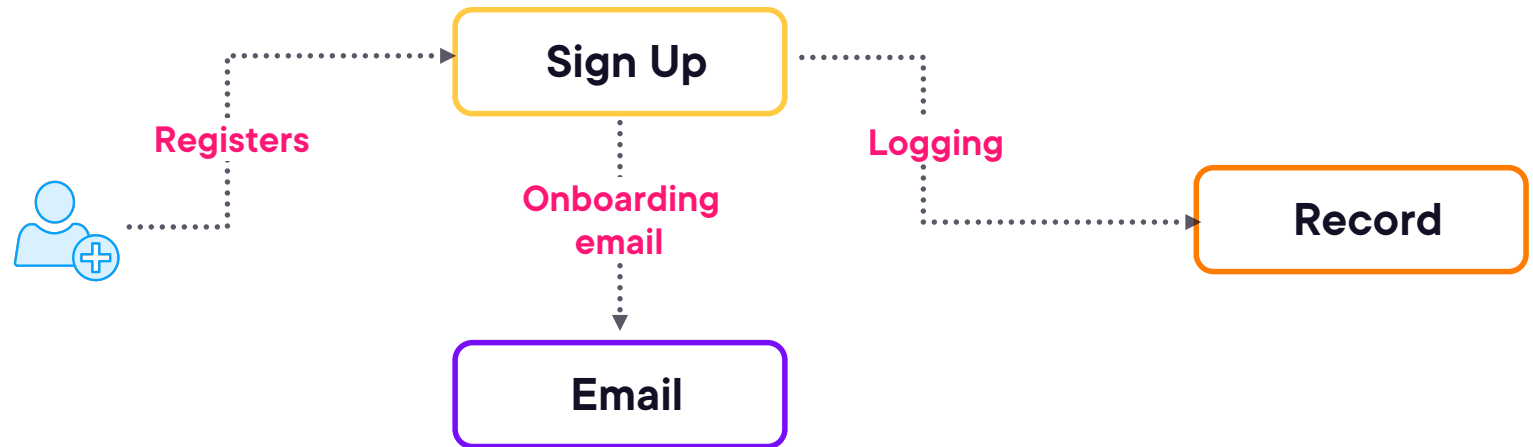
channel.queue_declare('user_email')
channel.basic_consume(queue='user_email', auto_ack=True, on_message_callback=callback)
channel.start_consuming()
```

Pika RMQ code consume

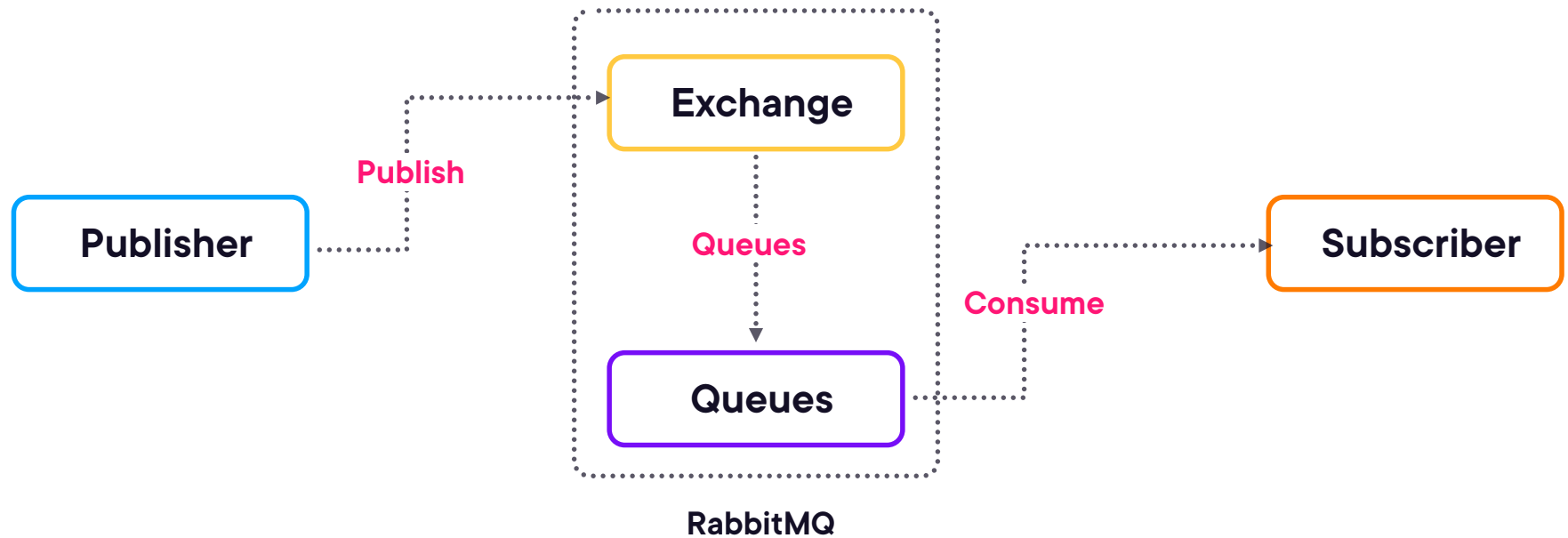
Sample code to declare and consume a message to RMQ using the routing key



RabbitMQ Use Case



RabbitMQ Architecture



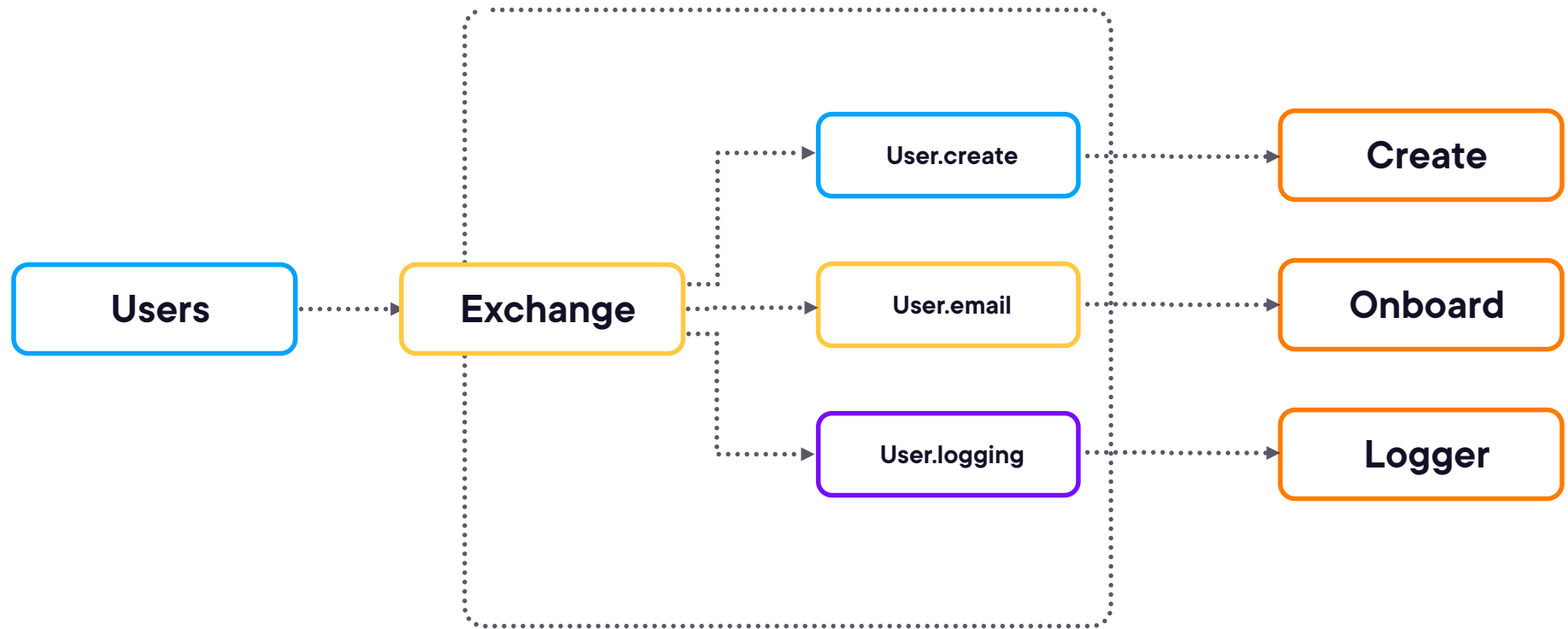
Fanout Exchange Use Cases

Broadcasting Events

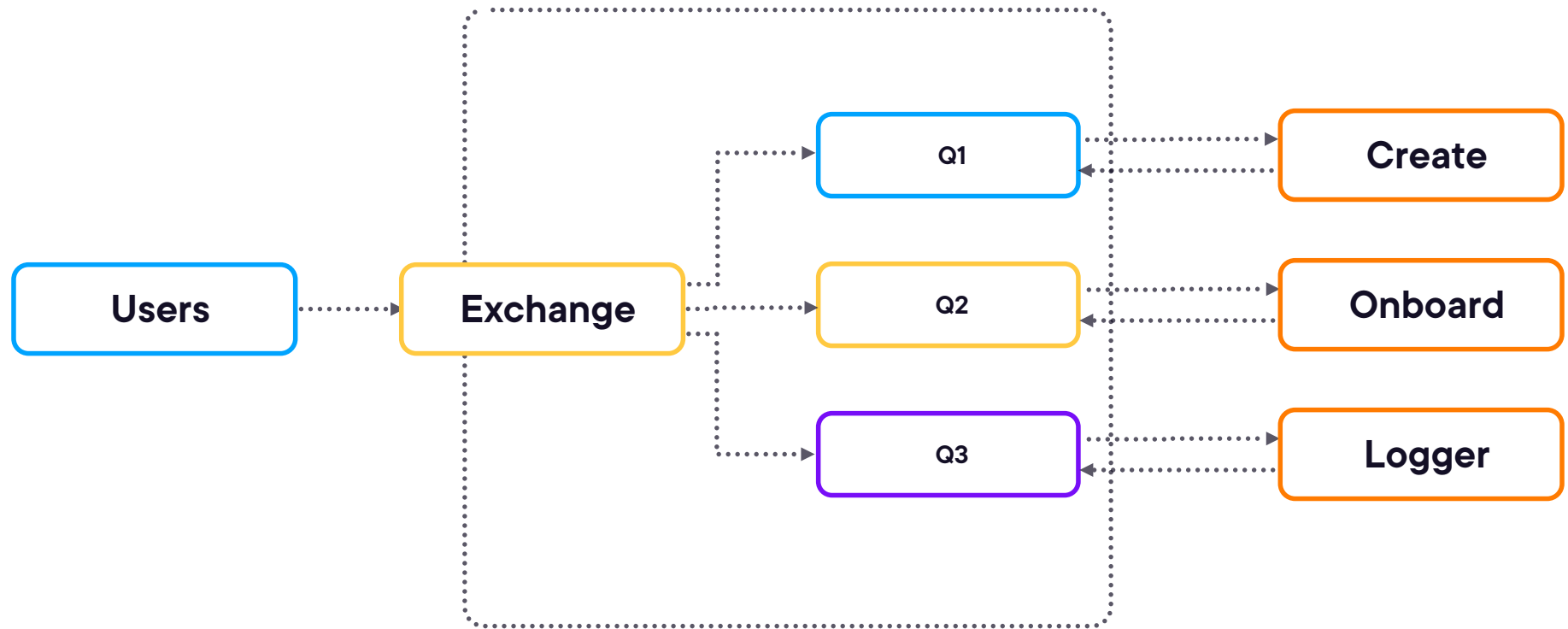
Live Data Streaming

Logging Systems

Direct Exchange (Work Queues)



Fanout Exchange (Publish/Subscribe)





Fanout Exchange



Demo: Integrate RabbitMQ

Integrate RabbitMQ in an application



The background features several light gray rectangular blocks with white wavy line patterns, connected by thin, curved lines. There are also clusters of small dots in the upper left and lower left areas.

Demo: Messaging Queue Models

Direct and Fanout model





Optimisations