

gRPC vs RabbitMQ

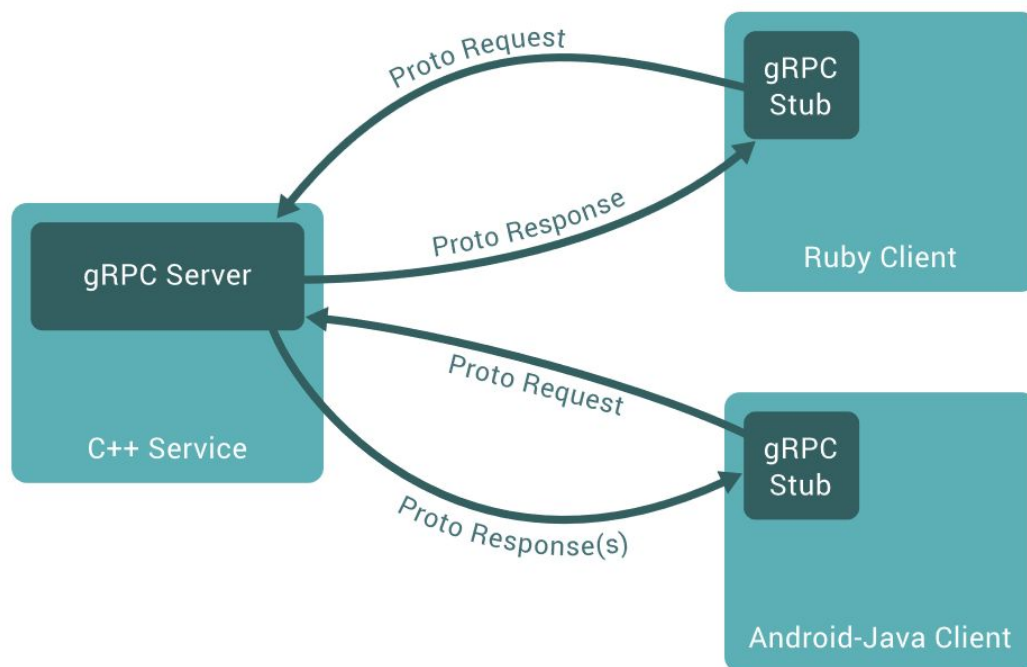
Francisco Abarca 201673552-6

Alexander Ruz 201673613-1

En este informe, se realizará una comparación entre las tecnologías que se utilizaron para la tarea 2, la cual trataba de un chat en donde un servidor era el coordinador de mensajes para una cantidad n de clientes, dichas tecnologías son gRPC y RabbitMQ.

Para comenzar, se detallarán el cómo funciona cada tecnología, terminado esto se mencionarán las diferencia entre ellas para finalizar con las razones de cuál fue la mejor tecnología para esta tarea.

gRPC: Es una framework RPC (llamada a procedimiento remoto) el cual permite que la aplicación del cliente pueda llamar a los métodos que se encuentran en la aplicación del servidor aunque estén en máquinas distintas, simulando que estos métodos se encuentran de forma local en la aplicación del cliente. Este framework, está basado en la idea de definir un servicio, crear los métodos con sus respectivos parámetros y retornos, además es importante destacar que el servidor como los clientes pueden estar basados en cualquier lenguaje de programación, ejemplo; el servidor puede estar escrito en C++, un cliente conectado a este servidor, tiene su aplicación basada en Python y otro cliente en Java.



Simulación de cómo funciona gRPC

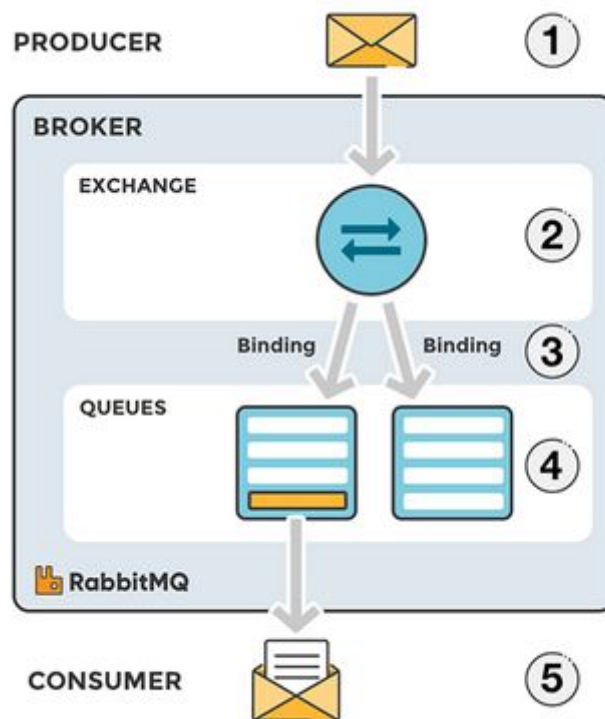
En la imagen¹, se observa que funciona como cualquier comunicación cliente-servidor, con la diferencia de que las solicitudes y respuestas (requests y responses

¹ <https://grpc.io/docs/guides/>

respectivamente), deben estar basadas en protobuf, el cual es un lenguaje en el que se define la estructura de los datos, que pasan a ser los mensajes que gracias a esta estructura pueden ser trabajados en cualquier lenguaje de programación.

RabbitMQ: Es un software llamado broker de mensajería o gestor de colas, ya que en este software se definen colas, en las cuales las aplicaciones (clientes) se pueden conectar a éstas y enviar o recibir mensajes contenidos en las colas, ya que estos mensajes son almacenados en las colas que un cliente (aplicación) pida una solicitud para leer el mensaje.

Para RabbitMQ, se definen dos tipos de clientes (productor y consumidor) y el servidor, los productores son aquellos que envían los mensajes y los consumidores quienes se conectan con alguna cola según donde se encuentre el mensaje que desean.



Simulación de cómo funciona RabbitMQ

La imagen anterior², se observa el comportamiento del productor y consumidor al mandar y recibir mensajes respectivamente, donde hay un servidor que coordina los mensajes.

Conclusión: Al analizar ambas tecnologías, se puede concluir que gRPC es una mensajería casi directa (no es directa porque los mensajes pasan por el servidor antes) y RabbitMQ almacena los mensajes que ha recibido en una cola, por lo que gRPC puede funcionar para chat donde no exista tanta concurrencia de mensajes mientras que

² <https://blog.bi-geek.com/rabbitmq-para-principiantes/>

RabbitMQ, al almacenar los mensajes en una cola, puede guardar mucho de estos. Por lo tanto nuestra recomendación es que para aplicaciones con poca concurrencia de mensajes es mejor utilizar gRPC, en caso contrario RabbitMQ, ya que es una aplicación escalable.