

Publish Subscribe Model

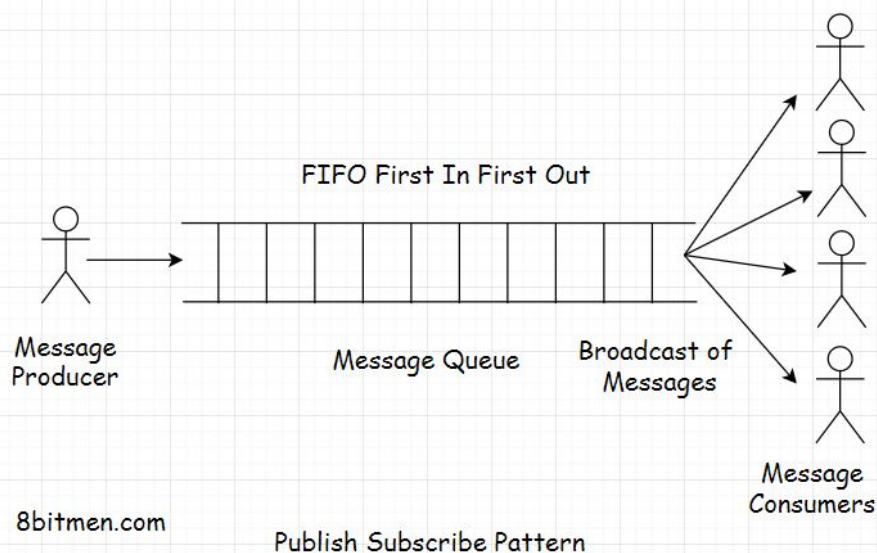
In this lesson, we will learn about the Publish-Subscribe model, when it is used & what are Exchanges in messaging?

We'll cover the following ^

- What Is A Publish Subscribe Model?
- Exchanges

What Is A Publish Subscribe Model?

A *Publish-Subscribe* model is the model where multiple consumers receive the same message sent from a single or multiple producers.

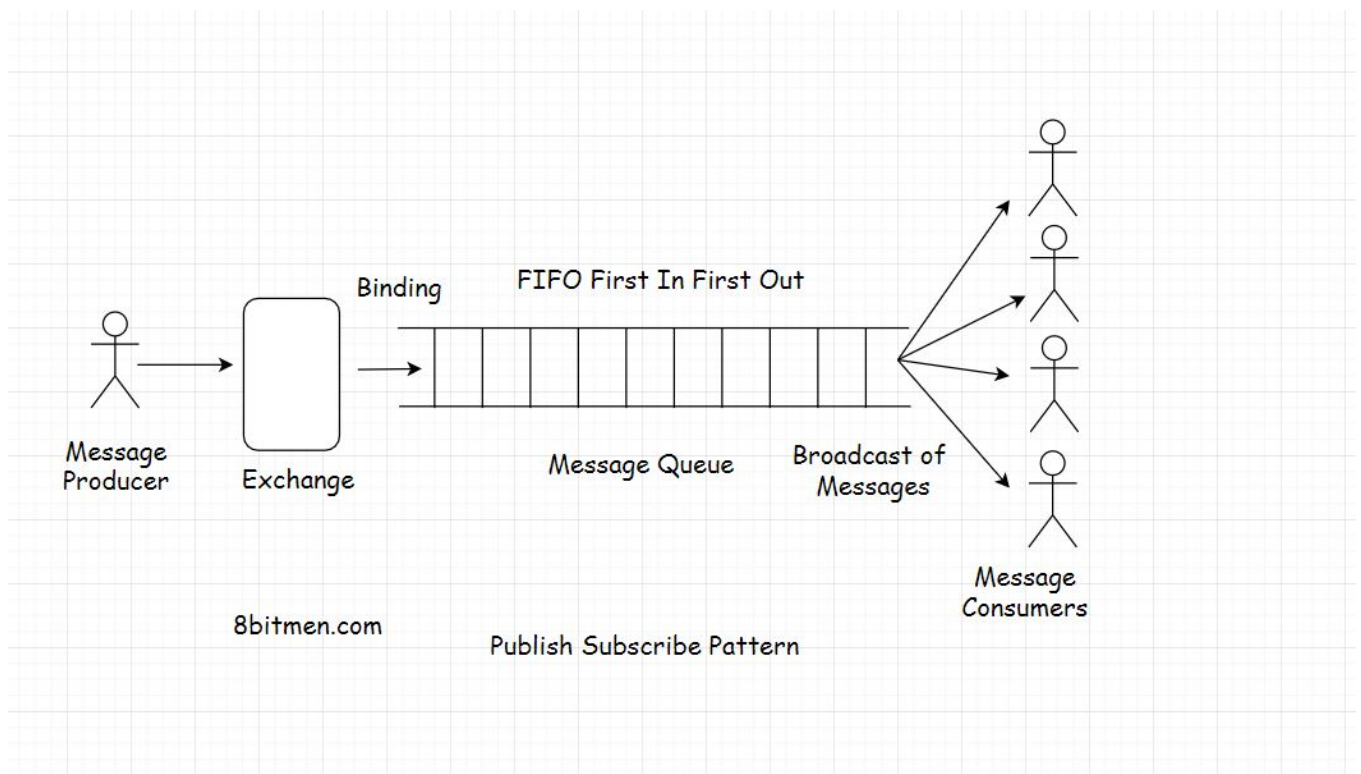


A real-world newspaper service is a good analogy for the *publish-subscribe* pattern, where the consumers subscribe to a newspaper service, and the service delivers the news to the multiple consumers of its service, every single day.

In the online world, we often subscribe to various topics in applications to be continually notified of the new updates on any particular segment. Be it sports, politics, or economics etc.

Exchanges

To implement the *pub-sub* pattern, message queues have *exchanges* which further push the messages to the queues based on the exchange type and the rules which are set. *Exchanges* are just like telephone exchanges which route messages from sender to the receiver through the infrastructure based on a certain logic.



There are different types of exchanges available in message queues, some of which are, *direct*, *topic*, *headers*, *fanout*. To have more insight into how these different exchange types work, [this RabbitMQ article is a good read](#).

There is no certainty that every message queue tech will have the same exchange type. These are just general scenarios I am discussing here. Things can change with technologies. Besides, technology is not important, all you need right now is just an idea of how things work.

So, we would pick a *fanout* exchange type to broadcast the messages from the queue. The exchange will push the message to the queue and the consumers

will receive the message. The relationship between exchange and the queue is known as *Binding*.

This is how we get updates of new content generated in real-time on social apps by businesses or individuals followed by a lot of people.

In the upcoming lessons, I will discuss this in detail how real-time feeds and notification systems work in social networks powered by the message queues.

Let's move on to the point to point messaging model.