Communication Model in IoT

IoT devices are found everywhere and will enable circulatory intelligence in the future. For

operational perception, it is important and useful to understand how various IoT devices

communicate with each other. Communication models used in IoT have great value. The IoTs

allow people and things to be connected any time, any space, with anything and anyone,

using any network and any service.

Types of Communication Model:

1. Request & Response Model –

This model follows a client-server architecture.

The client, when required, requests the information from the server. This request is usually in

the encoded format.

This model is stateless since the data between the requests is not retained and each request is

independently handled.

The server Categories the request, and fetches the data from the database and its resource

representation. This data is converted to response and is transferred in an encoded format to

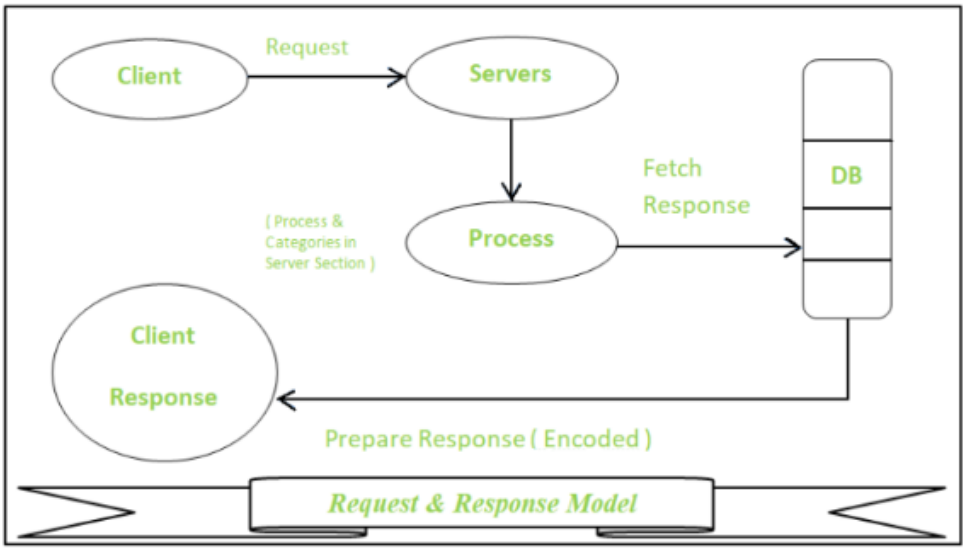
the client. The client, in turn, receives the response.

On the other hand — In Request-Response communication model client sends a request to

the server and the server responds to the request. When the server receives the request it

decides how to respond, fetches the data retrieves resources, and prepares the response, and

sends it to the client.



Publisher Subscriber Model:

This model comprises three entities: Publishers, Brokers, and Consumers.

Publishers are the source of data. It sends the data to the topic which are managed by the

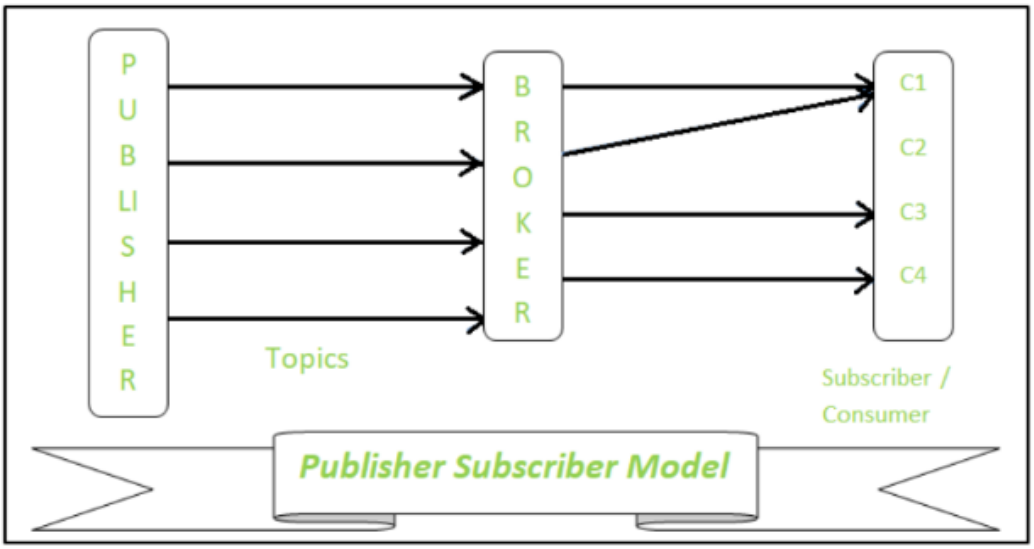
broker. They are not aware of consumers.

Consumers subscribe to the topics which are managed by the broker.

Hence, Brokers responsibility is to accept data from publishers and send it to the appropriate

consumers. The broker only has the information regarding the consumer to which a particular

topic belongs to which the publisher is unaware of.



Push-Pull Model –

The push-pull model constitutes data publishers, data consumers, and data queues.

Publishers and Consumers are not aware of each other.

Publishers publish the message/data and push it into the queue. The consumers, present on

the other side, pull the data out of the queue. Thus, the queue acts as the buffer for the

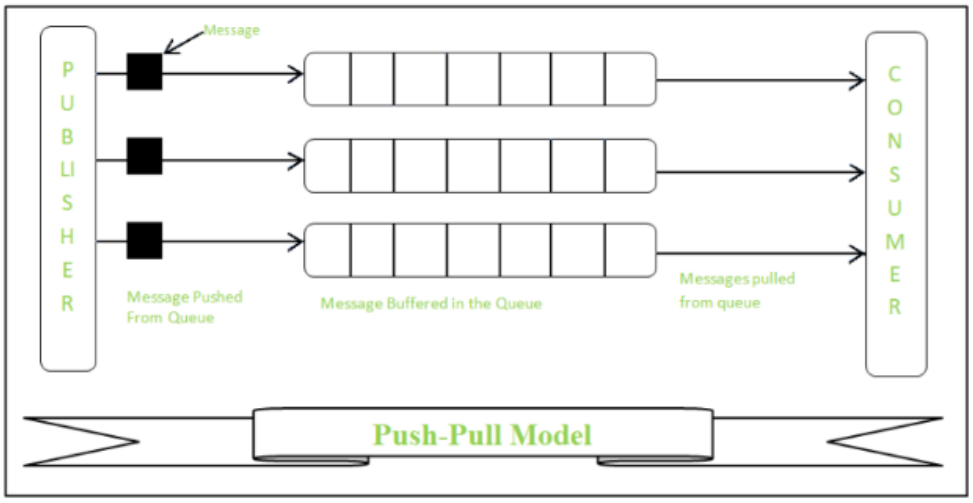
message when the difference occurs in the rate of push or pull of data on the side of a

publisher and consumer.

Queues help in decoupling the messaging between the producer and consumer. Queues also

act as a buffer which helps in situations where there is a mismatch between the rate at which

the producers push the data and consumers pull the data.



Exclusive Pair –

• Exclusive Pair is the bi-directional model, including full-duplex communication among client and server. The connection is constant and remains open till the client sends a request to close the connection.

• The Server has the record of all the connections which has been opened.

• This is a state-full connection model and the server is aware of all open connections.

• WebSocket based communication API