PubSub Message flow and Subscription Mechanisms

Publisher tasks

Publisher application creates the message with the data. It adds metadata to the message if configured.

Makes the call to PubSub server to send the message to the required topic.

Message is base64 encoded by default in transit.

Message Format

- A message consists of the actual data, ordering key and additional attributes if configured in the metadata.

- PubSub service adds the following attributes to the message.

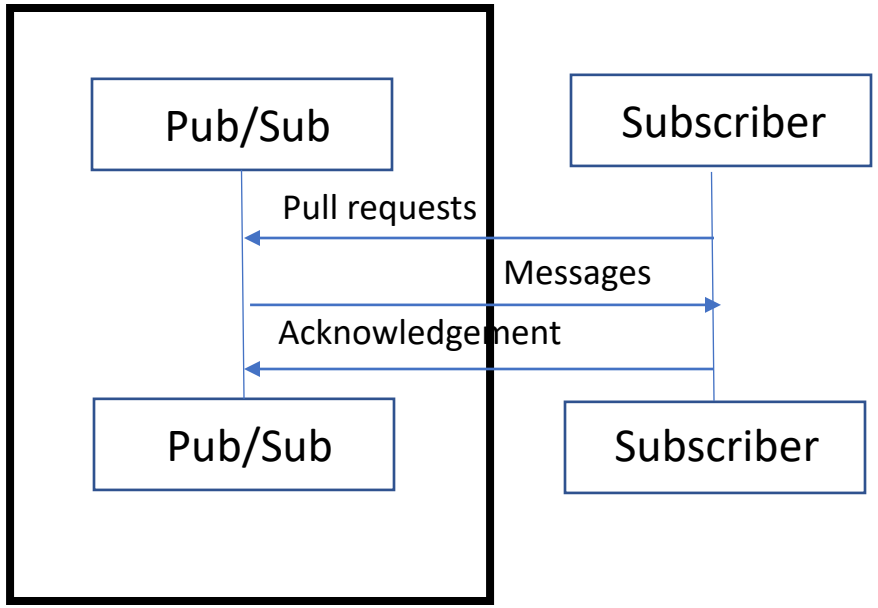
1. Unique message id for the message.
2. Timestamp when the message was received by the service.

Subscriber tasks

Subscriber application subscribes to a topic to receive messages from it. Only messages sent to topic after subscription is created is available to subscribers, all messages sent to topic prior to any subscription cannot be consumed by any subscriber. Pubsub offers at-least-once-delivery gurantee means it ensures all the messages are delivered to all the subscribers at least once before being removed from the topic.

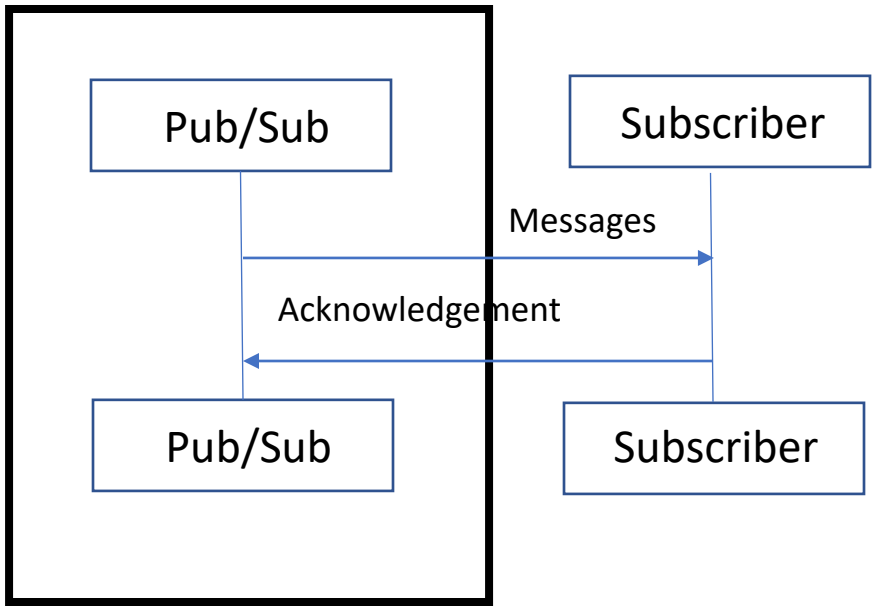
Pubsub sends the message to subscriber and marks the message in outstanding phase and wait for the subscriber to send the acknowledgment of the message back. It waits for predefined time which can be configiured known as ackDeadline before attempting a retry to the subscriber again. If it receives the acknowledgement then it marks the message as acknowledged and checks if it is received by all the subscribers and send it for deletion.

Pull Subscriber



1. The subscribing application explicitly calls the pull method, which requests messages for delivery.
2. The Pub/Sub server responds with the message (or an error if the queue is empty) , and an ack ID.
3. The subscriber explicitly calls the acknowledge method, using the returned ack ID to acknowledge receipt.

Push Subscriber



1. The Pub/Sub server sends each message as an HTTPS request to the subscriber application at a pre-configured endpoint.
2. The endpoint acknowledges the message by returning an HTTP success status code. A non-success response indicates that the message should be resent.

PubSub Architecture

Pubsub servers run in all GCP regions to allow fast access and low latency. This means single topic messages can be stored in multiple regions and publisher will publish message to region which is close to it and it is replicated internally by google.

Pubsub is divided into two parts i.e. data plane and control plane.

Data plane handles the movement of messages between publishers and subscribers.

Control plane handles the assignment of publishers and subscribers to the servers in the data plane.

Servers in data plane are called forwarders and in control plane are called routers.

Message movement in PubSub Internally

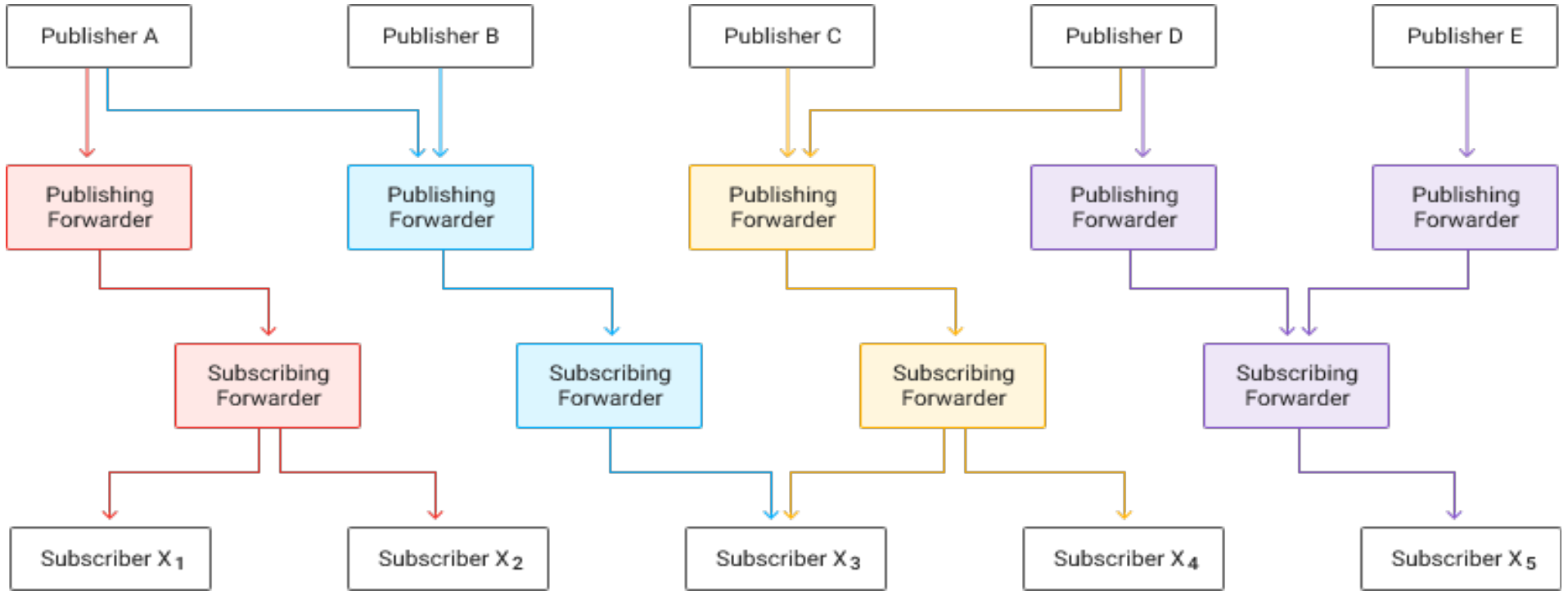


Image Source: Google Official Documentation