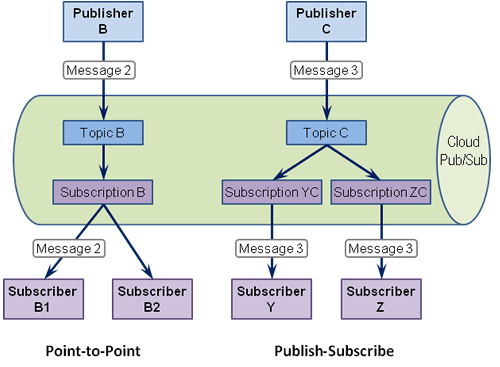
**Publish-Subscribe**

 publish–subscribe is a [messaging pattern](https://en.wikipedia.org/wiki/Messaging_pattern) where senders of [messages](https://en.wikipedia.org/wiki/Message_passing), called publishers, do not program the messages to be sent directly to specific receivers, called subscribers, but instead categorize published messages into classes without knowledge of which subscribers, if any, there may be. Similarly, subscribers express interest in one or more classes and only receive messages that are of interest, without knowledge of which publishers, if any, there are

Publish–subscribe is a sibling of the [message queue](https://en.wikipedia.org/wiki/Message_queue) paradigm, and is typically one part of a larger [message-oriented middleware](https://en.wikipedia.org/wiki/Message-oriented_middleware) system. Most messaging systems support both the pub/sub and message queue models in their [API](https://en.wikipedia.org/wiki/Application_programming_interface), e.g. [Java Message Service](https://en.wikipedia.org/wiki/Java_Message_Service) (JMS).

****

**Configuration**

Open Eclipse:

1. Create New Java Project.
2. Create Two Classes Producer and Consumer.
3. Copy the Above Code.
4. Add This Jar File -> kafka-clients-0.10.2.0.
5. The go to your Kafka Folder in that folder there is a lib folder in that there are some jar file just add those jar files.
6. Then Run the Zookeeper.
7. Then Run the Kafka Server.

In Program you have to set these properties in Producer Program:

1. props.put("bootstrap.servers", "localhost:9092");

//Assign localhost id

1. props.put("acks", "all");

//Set acknowledgements for producer requests.

1. props.put("retries", 0);

//If the request fails, the producer can automatically retry,

1. props.put("batch.size", 16384);

//Specify buffer size in config

1. props.put("linger.ms", 1);

//Reduce the no of requests less than 0

1. props.put("buffer.memory", 33554432);

//The buffer.memory controls the total amount of memory available to the producer for buffering.

1. props.put("key.serializer","org.apache.kafka.common.serialization.

StringSerializer");

1. props.put("value.serializer","org.apache.kafka.common.serialization.StringSerializer");

In Program you have to set these properties in Consumer Program:

1. props.put("bootstrap.servers", "localhost:9092");
2. props.put("group.id", "group-1");
3. props.put("enable.auto.commit", "true");
4. props.put("auto.commit.interval.ms", "100");
5. props.put("auto.offset.reset", "earliest");
6. props.put("auto.commit.interval.ms", "1000");
7. props.put("session.timeout.ms", "30000");
8. props.put("key.deserializer","org.apache.kafka.common.serialization.StringDeserializer");
9. props.put("value.deserializer","org.apache.kafka.common.serialization.StringDeserializer");

Then Run Producer Program and then run the Consumer Program you get the output.

