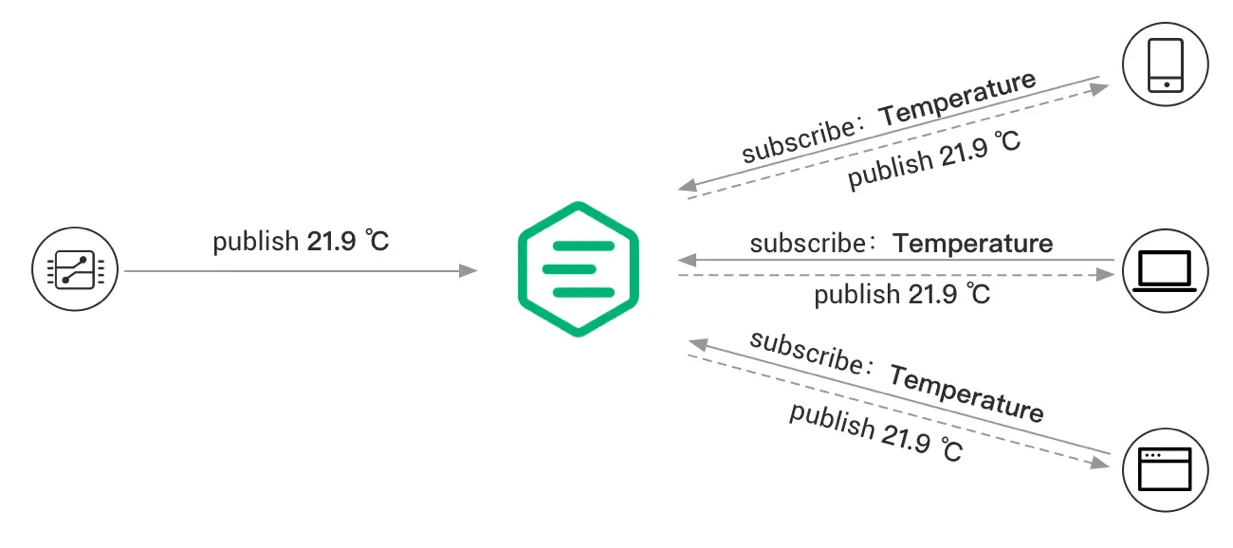
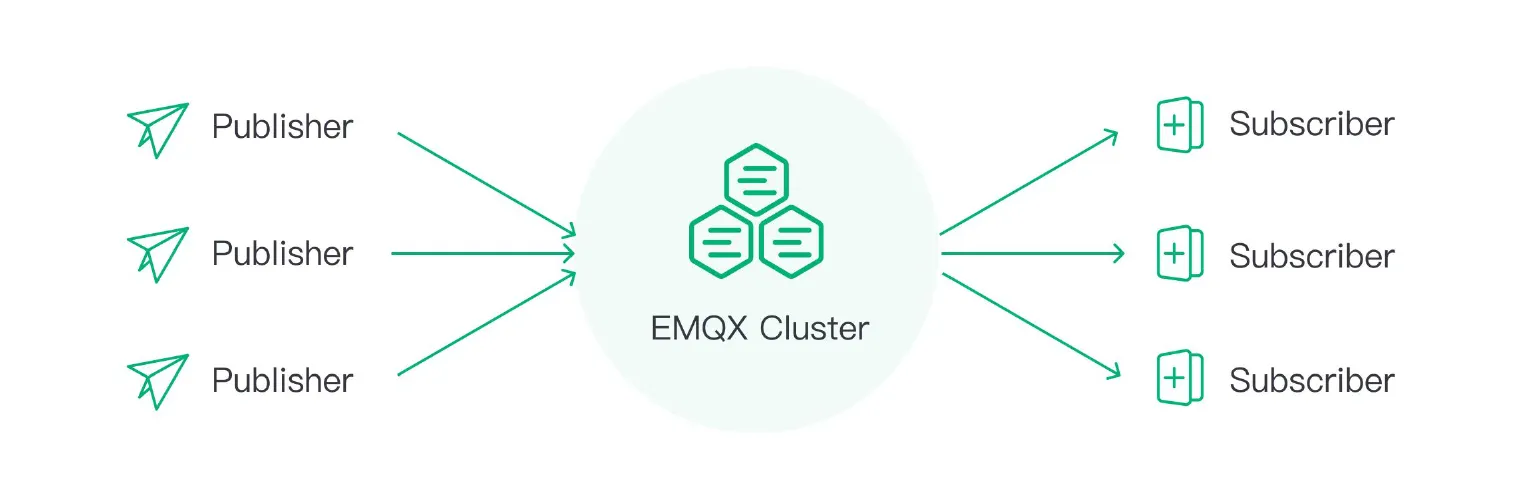
# **MQTT (Message Queuing Telemetry Transport)**

# System Overview:





Ref: <https://www.emqx.com/en/blog/the-ultimate-guide-to-mqtt-broker-comparison>

* **Broker**: Role as an intermediary for message exchange.
* **Producer**: Component that sends messages to the broker.
* **Subscriber**: Component that receives messages from the broker.

## Messages from publisher:

sensor/10/temperature

sensor/+/temperature

From publisher : sensor

Topic : temperature

Value : 10

## Messages from subscriber while registering to a topic:

From subscriber : client1

Topic : temperature

# Detailed Design

### Producer Design

* **Workflow**:
  + Connect → Authenticate → Publish messages.
* **Interface**:
  + APIs or libraries for publishing messages (e.g., publish(topic, message, qos)).

### Subscriber Design

* **Workflow**:
  + Connect → Authenticate → Subscribe → Receive messages.
* **Interface**:
  + APIs or libraries for subscribing to topics (e.g., subscribe(topic, qos)).
  1. Class diagram of broker
     + Start TCP server
     + Accept the connections from clients in infinite loop
     + If client is producer:
       - Start separate thread to receive the messages
       - Push the data received from producer (topic name as key)
     + If client is consumer:
       - Add connection to map (topic wise)
     + When a message gets added to Q, send the same to all its consumer

2.2 Class diagram of subscriber

2.3 Class diagram of producer