

## 1. Data Flow Explanation -

Broker used - Mosquitto

Step	Component	Action
1	MQTT Listener	Subscribes to telemetry topics
2	Queue	Buffers messages for parallel processing
3	Worker	Pulls message
4	DB	Stores latest sensor value
5	Rule Engine	Evaluates alarm conditions
6	State Store	Maintains duration tracking
7	Publisher	Sends alarm notification
8	History DB	Saves alarm log

## 2. Third Party Services/Library -

### Paho MQTT Python Client

#### Reason to choose

- Mature
- Actively maintained
- Low resource usage
- To connect with MQTT broker

## 3. Assumptions

- Telemetry messages arrive in JSON format
- MQTT topic format:

`telemetry/<device>/<sensor>`

Example:

`telemetry/sensor1/temperature`

Payload:

```
{"value": 25.3}
```

Tested MQTT -

Publisher Client -

1. Threshold condition type -

```
watch -n 1 "mosquitto_pub -t telemetry/sensor1/temperature -m  
'{\"value\":25}'"
```

2. Conditional condition type

```
mosquitto_pub -t telemetry/sensor2/current -m '{"value":10}'
```

```
watch -n 1 "mosquitto_pub -t telemetry/sensor1/temperature -m  
'{\"value\":25}'"
```

Subscriber client -

Topic - mosquitto\_sub -t alarm/# -v

Threshold Alarm trigger resp -

```
alarm/sensor1/temperature {"rule_id": 1, "device_id": "sensor1", "sensor_id": "temperature",  
"value": 25, "triggered_at": "2026-02-14T15:03:03.184944+00:00", "message": "ALARM  
TRIGGERED"}
```

Conditional Alarm trigger resp -

```
alarm/sensor1/temperature {"rule_id": 2, "device_id": "sensor1", "sensor_id": "temperature",  
"value": 25, "triggered_at": "2026-02-14T15:02:58.142694+00:00", "message": "ALARM  
TRIGGERED"}
```