PHASE-2: AIR QUALITY MONITORING

INNOVATION

7328-SURYA ENGINEERING COLLEGE,ERODE

TEAM MEMBERS :

DEVANATHAN .V

GEETHA .M

GOKUL .S

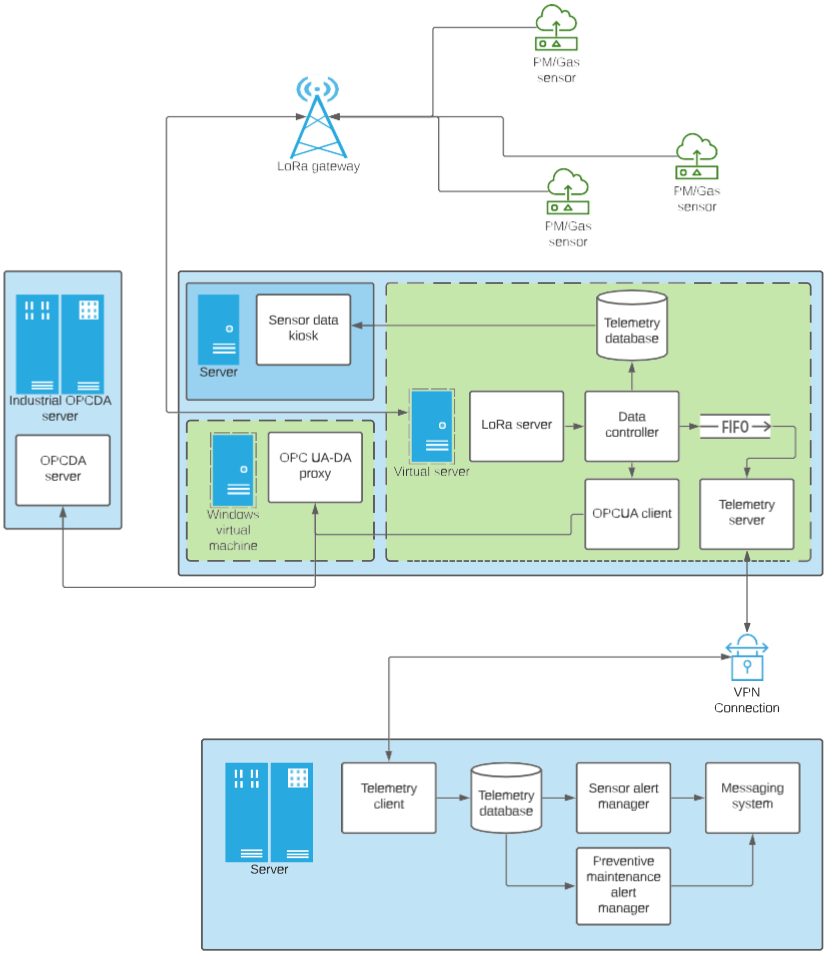
GOKIL .T

GOKULNATH .B

Architecture Description

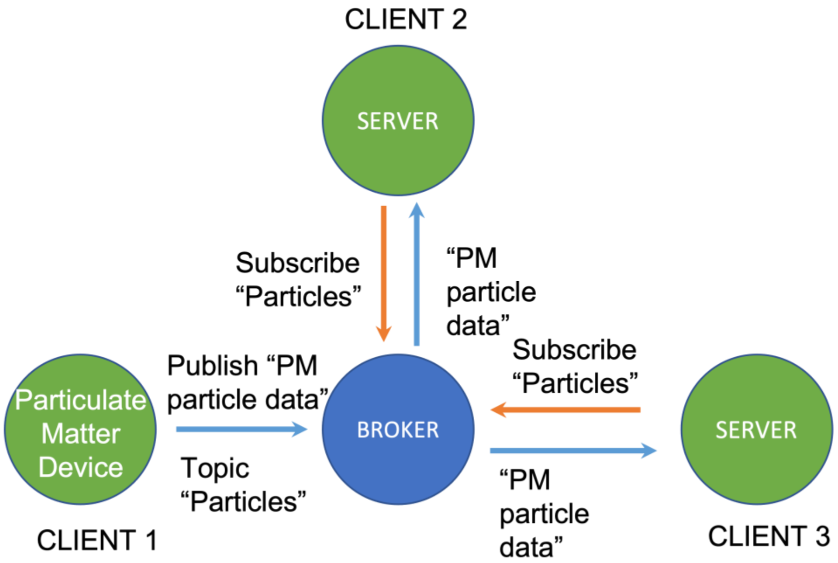
In this section, the proposed system architecture for air quality monitoring in industrial environments is described. shows the general scheme of the developed infrastructure. The goal is to plot air quality for end-clients from the data collected by a series of sensors. The acquired data is forwarded by the devices using LoRa communication technology on the 868 MHz frequency band, as stated in the LoRaWAN specification for deployments in Europe. The server has a data management system in charge of storing data in an InfluxDB database. This type of database stores time series of data and manages the huge amounts of data generated by the devices, applications, and infrastructures, providing a timestamp for each of them when stored. This is useful for their later representation in graphs using the Grafana software. Grafana is a web server that depicts data time series in graph format [26]. This web server represents the evolution (over time) of the particulate matter or polluting gases and the internal parameters of the devices, such as communication link quality levels, allowing the visualization and supervision of the deployment

The data reaching the OPCDA server is then forwarded to an external server through a VPN to facilitate a secure connection. A copy of the data is stored on this server as well, and Grafana can represent the data from any device with an Internet connection since it has its own public IP address that the client can access. This server also has an alert system. As a basic service, Grafana's integrated default warning system can be used to inform the user when abnormal levels of contamination are detected. As an advanced service, the Prometheus Alertmanager tool will notify the network manager of any failure detected in any of the provided services. In the following subsections, all these components are described in detail.



MQTT Server Configuration

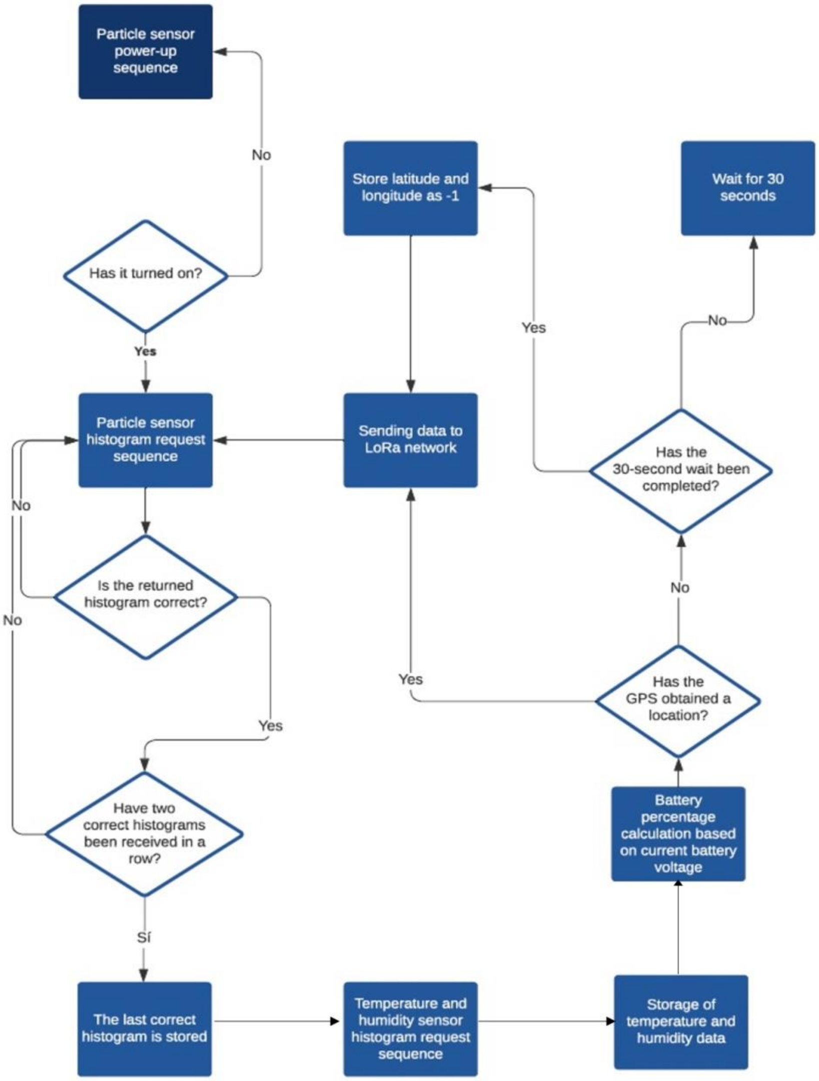
An MQTT server is a lightweight protocol in which messages have a topic. The body of the message



receives the data from the Chirpstack server. The data is then forwarded througha device to a broker that redirects them to the subscribers of the topic of the message. The subscribers can only receive messages with the topic they are subscribed toillustrates an example of an MQTT operation, where the particulate mattermeasuring device publishes a message with the topic "particles" and the broker re-sends it to the two servers subscribed to that topic. In our proposal, the Chirpstack server relays

A series of scripts related to the subscribed topics of interest has been developed in Python. These topics include: (i) link quality data encompassed within a topic bridging information, like RSSI (Received Signal Strength Indicator) or SNR (signal-to-noise ratio), and (ii) environmental pollution data received from the sensors, which is included in the topics of each application configured in the Chirpstack server. When data reaches the MQTT server, these scripts decode, analyze, and store them in different InfluxDB databases. Redundant databases facilitate that the data be stored certainly in our system.

Flow diagram of the operation of the particulate matter measuring device


Particle sensor power-up sequence
No
Has it turned on?
Yes
Particle sensor histogram request sequence
No
Is the returned histogram correct?
No
Yes
Have two correct histograms been received in a row?
Sí
The last correct histogram is stored
Temperature and humidity sensor histogram request sequence
Storage of temperature and humidity data
Battery percentage calculation based on current battery voltage
Has the GPS obtained a location?
Yes
Sending data to LoRa network
No
Has the 30-second wait been completed?
Yes
Store latitude and longitude as -1
No
Wait for 30 seconds