MSc in CSTE, Cloud Computing Assignment 2023-24

Collecting, Processing and Distributing IoT Data to Clients

Stuart Barnes, Cranfield University

November 21st , 2025
Hand-in date 5/1/2026 (FT), 19/1/2026 (PT), 12noon

1. Introduction

The goal of the assignment is to store the latest air quality data captured from a network of small IoT environmental sensors and make the data available to clients on-demand. The data can be accessed in JSON format from:

- Averaged data from last hour for each sensor: https://data.sensor.community/static/v2/data.1h.json
- Averaged data from last 5 minutes for each sensor (for testing): https://data.sensor.community/static/v2/data.json
- Daily archived data for each sensor: https://archive.sensor.community

An example of the data format is given below but check for changes on the website:


```
{
      "id":10995737600,
      "sampling_rate":null,
      "timestamp":"2022-06-21 12:56:19",
      "location":
          {
                "id":16232,
                "latitude":"48.8",
                "longitude":"9.002",
                "altitude":"365.6",
                "country":"DE",
                "exact_location":0,
                "indoor":0
          },
      "sensor":
          {
                "id":92,
                "pin":"1",
                "sensor_type":
                     {
                          "id":14,
                          "name":"SDS011",
```

```
                        "manufacturer":"Nova Fitness"
                    }
            },
        "sensordatavalues": [
            {
                "id":24428549678,
                "value":"3.68",
                "value_type":"P1"
             },
             {
                "id":24428549711,
                "value":"2.80",
                "value_type":"P2"
             }
        ]
}
```


The data set consists of sensor location, sensor type and sensor values. Note that different sensors may give different types of data, specifically PM_{10} and $PM_{2.5}$ values corresponding to the two main sizes of airborne particulate pollution, as well as temperature, pressure and humidity. For this exercise, the particulate data should be used to calculate the air quality index (AQI).

To calculate the AQI you will need to use the following table to identify the index from values of the P_{10} and $P_{2.5}$ data. For example, if $PM_{2.5}$=39 and PM_{10}=36, the AQI will be 4. Note, as in this example, that if the P_{10} and $P_{2.5}$ measurements provide different AQI values, choose the largest value.

Range	**Air Quality Index**	**$PM_{2.5}$ Particles, 24 hour mean (µg/m^3)**	**PM_{10} Particles, 24 hour mean (µg/m^3)**
Low	1	0-11	0-16
	2	12-23	17-33
	3	24-35	34-50
Medium	4	36-41	51-58
	5	42-47	59-66
	6	48-53	67-75
High	7	54-58	76-83
	8	59-64	84-91
	9	65-70	92-100
Very High	10	>70	>100

Table 1. UK air quality index, "*Review of the UK Air Quality Index",* 2011

2. Tasks

- Accessing on-line data:
    - Develop a method for accessing the environmental sensor data via the on-line sources, processing it and fulfilling a user AQI request. You might store the data in an appropriate Cloud-located database or storage resource, or

you might try to access the data directly from the website as user requests are made.
    - Provide a mechanism for ensuring the data is kept up to date, as it becomes available. This could be on a short timescale (5 minutes) or longer (1 day).
    - Calculate the Air Quality Index (AQI) for each reading, according to the information provided in Table 1. How you execute this calculation will depend on how you access the sensor data. It might take place at the point of bulk data download or on each client request, according to what you think will be the most efficient and/or most cost-effective solution.
- Provide a solution that returns AQI data to clients on demand.
    - Clients should be able to download data according to user-specified criteria, such as geographical location (country, region, etc) and time frame (last hour, day, week and month).
    - Simulate increasing workloads on the front-end system by increasing the number of client requests being made.
    - Measure the current system workload to determine whether a new worker is required. You can decide whether the system load is measured directly from the amount data being downloaded or the number of currently connected clients, or some other criteria.
    - Demonstrate the elasticity of your solution, such that the number of workers increases as the workload increases and decreases as the workload reduces.
    - Consider the cost implications of instantiating new workers and explain the reasoning of your design in the report from a cost perspective.
    - The clients can be served via html requests to a Web server or can simply receive a csv file containing the requested data. Whichever approach you take, think about how you will demonstrate the operation of your solution.
- Additional tasks
    - Present the **experimental methodology** that you used to test and validate your solution
    - Discuss in detail the decisions you made in the design of your solution architecture. Explain how computational requirements, costs or other factors drove your choice of AWS services and how you measured success. If you tried multiple solutions, compare their relative performance and costs.
    - Provide a script that configures the AWS environment. Although providing an automated mechanism for populating AWS services (Lambda, EC2, etc) with your application scripts is preferred, a detailed explanation of the deployment of these on is acceptable.
    - Briefly discuss the implications of data security and sovereignty with respect to the environmental sensor.

.

3. Source Code and Report requirements

Write a report to present and discuss your findings. The report should be no less than 1500 words and must not exceed 3000 words. The report can contain any number of figures/tables, however all figures/tables should be numbered and discussed. All code and

related files should be uploaded to Canvas, in an Appendix along with appropriate documentation.

4. Assignment Submission

The source code and documentation should be submitted electronically via the technical work submission point by 9.30am on the 5th Jan (FT) and 19th Jan (PT). Both the report and the zipped technical work should be submitted electronically via the TurnItIn submission point by the prescribed deadline, for the assignment submission to be considered complete.

This is an individual assessment. Respect the University regulations on plagiarism.

5. Marking

The assignment will be assessed based on the following marking scheme:

- 10% Introduction, conclusions
- 30% Methodology:
    - Description of implementation including system architecture
    - Details of experimental methodology
- 10% Solution: source code, organization, documentation
- 10% Presentation of results of validation/testing
- 30% Discussion and analysis of the results
- 10% Report structure, presentation, clarity, references