Ali HAIDAR
Ahmad ALIBRAHIM

CPS2 Project Report

CPS2
December 8, 2019

1 Abstract Description

In this project we provide a data gathering scenario by exposing some measuring stations (Temperature, Pressure, Dust) in different places in Saint Etienne. We configure the sensors to be able to transmit data instantaneously and be accessed by requests to get the current updates. The data is stored maintaining interoperability and it is highly interchangeable in a machine readable manner where it could be useful for any future use-cases such as simple and complex queries or even data analysis.

2 Objective and Expected Results

The expected results are acquiring data from measurement stations that will be found in a time series database (influxDB `https://www.influxdata.com/`) and in RDF format on a triple store (fuseki `https://jena.apache.org/documentation/fuseki2/`). In addition to that, a web-page is used to visualize these stored data from the 2 different sources.

The objectives of this project are the following:

- Every specific time δ, the station sends its current status to the influxDB database of the territory platform of EMSE (`https://www.mines-stetienne.fr/`).
- Every specific time δ, the station sends its current status to a triple store of EMSE.
- One can query the 2 different databases to get the measurements of any time of the stations.
- Get instantaneous measurements from the stations using their things description.
- Comparison of the performance and efficiency of the 2 databases (fuseki and influxDB)

3 Envisioned Use Cases

- **Store data in RDF format for semantic usages:**
  1. Instantaneous data sent from sensors will be converted to RDF format and stored in a triple store.
  2. The triple store will accept simple and complex SPARQL queries to process the data and reply with the correct answer.
- **Push to influxDB using real time data:**
  1. Sensors will send instantaneous data to the influxDB.

2. InfluxDB will accept queries to get current or previous state of the stations.

- **Provide things description to ask questions about stations**
- **Visualization of the data of the stations:**
  1. Get the current state of each station from the 2 database sources.
  2. Get the old states of the stations from the 2 database sources.
  3. Update the web-page with the queried data.

4 Functional and Technical Requirements

- Implementation and deployment of the arduino code on NodeMCU (ESP8266) (`https://www.nodemcu.com/index_en.html`):
  1. SDS011 sensor (`https://aqicn.org/sensor/sds011/`) reads the values of PM2.5 and PM10 (`https://www.irceline.be/fr/documentation/faq/quappelle-t-on-particules`
     which are responsible for measuring the dust in the air.
  2. Read temperature, pressure and altitude using a BMP280 sensor (`https://www.adafruit.com/product/2651`).
- 7 stations consist of the two sensors (BMP280 , SDS011) and an arduino board:
  1. Each station sends data using HTTP to a PHP script.
  2. The PHP script pushes these data to the influxDB and the fuseki triple store.
  3. The script is on the university's host which can be requested from outside.
- Install fuseki triple store on the university's host to be contacted by the PHP script mentioned above.
- Acquire authorization to push/query the influxDB of the university.
- Implementation of a website to visualize the measurements using JavaScript, PHP, and HTML.
- Allow arduino boards to receive HTTP requests and send responses using specific APIs to be provided by the things description.
- Provide a simple ontology to demonstrate the structure of the station that will be followed when pushing data to fuseki triple store.

5 Technical Problems

- Unable to send data to German Website (https://luftdaten.info/en/construction-manual/) due to having different kinds of sensors
- Unable to read data from both sensors on the same arduino board.
- Pins that connect the arduino board to the sensors are not fixed.
- Arduino (ESP8266) restarts from the beggining with built it in timeout (write flash timeout WFT) when the code is blocking.

6 Planning of Realization

We provide the planning of the project but this would be modified later especially the dates and maybe we could add some tasks or even cancel some that wouldn't match the project's goals.

Task	**Start Date**	**End Date**	**Desciption**
Sensors Installation	19-11-2019	22-11-2019	Configure and deploy the air quality sensors then connect them to the network.
Sensors Synchronization	25-11-2019	29-11-2019	Transmit data from sensors to an online database.
Data in RDF	2-12-2019	4-12-2019	Model data in RDF and provide the mechanisim to insert in triple store
Data on influxDB	5-12-2019	9-12-2019	Transmit data from sensors to influxDB instantaneously
Deployment on the territory platform	10-12-2019	20-12-2019	Connect to the territory platform (influxdb, PHP scripts and triple store)
Visualization webpage	21-12-2019	10-01-2020	Visualize data transmitted by sensors in a web-page using the triple store and influx DB as databases.
Things description	13-1-2020	17-1-2020	Provide the things description for the sensors and test it on Qanswer

7 System Design

In figure 1 we show the architecture of the system we are creating. We provide a synchronous communication between the stations deployed and the two kinds of databases we have, on top of those a web-page available for the visualization of the data released acquiring it's content from the databases. Also we provide an API to read directly the current state and this is integrated as things description which tells Qanswer what are those sensors and how they can be accessed for information retrieval.

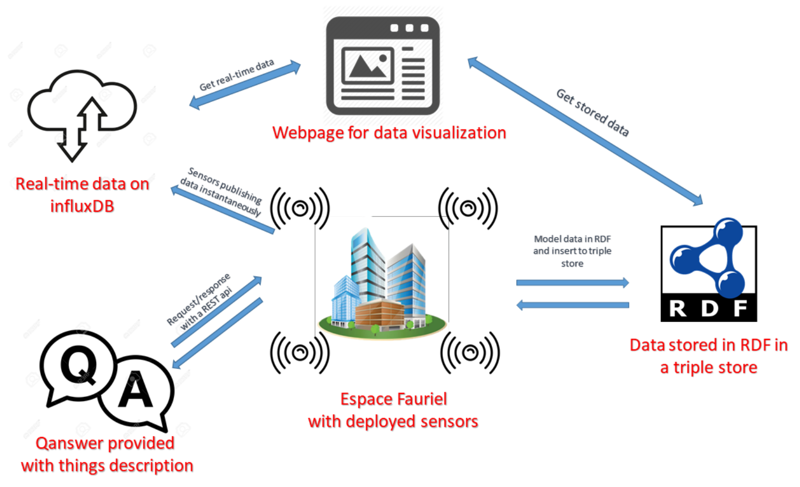
Get real-time data
Webpage for data visualization
Get stored data
Real-time data on influxDB
Sensors publishing data instantaneously
Model data in RDF and insert to triple store
R D F
Data stored in RDF in a triple store
Request/response with a REST api
Q
A
Espace Fauriel with deployed sensors
Qanswer provided with things description

Figure 1: System design and data flow.

The first version of ontology used to define the structure of the stations in our system is defined in figure 2.

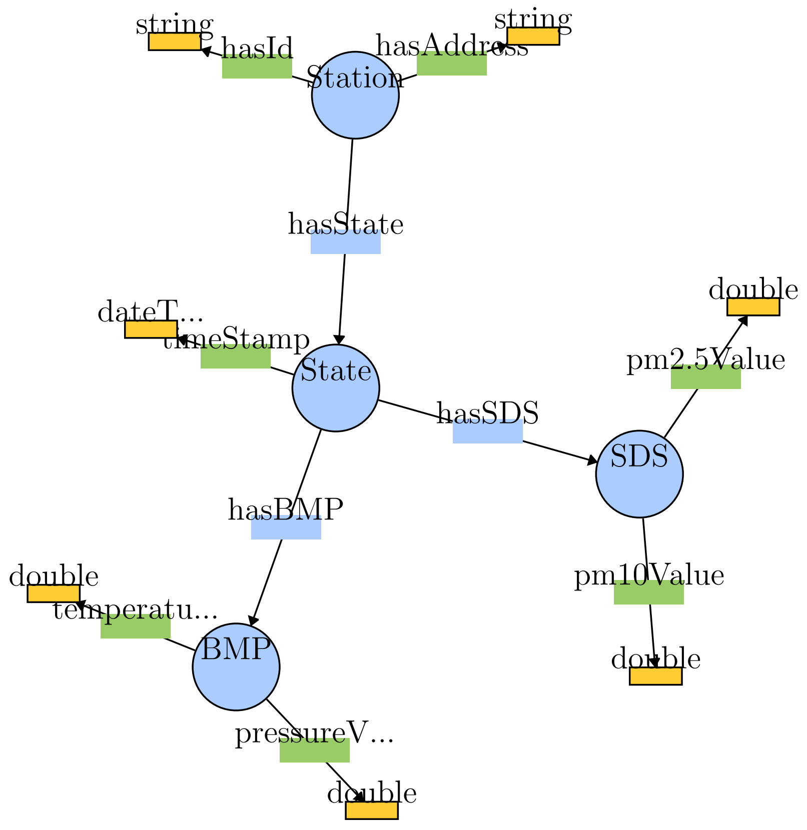

Figure 2: station ontology