CPS Humidity Homework

Janos Benjamin Antal

January 11, 2018

Prerequisites

To build the application on Linux mint 18.2 the following steps are needed:

- Install required packages with apt:
  - `libboost-all-dev`
  - `libcurl4-openssl-dev`
  - `libssl-dev` (some help can be found here)
  - `uuid-dev`
- Install CMake and CNU C++ compiler:

  ```
  $ sudo apt install cmake g++
  ```

- Install RapidJSON with the steps described here:

  ```
  $ sudo apt-get remove libcurlpp0
  $ cd [wherever]
  $ git clone https://github.com/jpbarrette/curlpp.git
  $ cd curlpp
  $ cmake .
  $ sudo make install
  ```

- Install AzureIoT C SDK
  - Build and install from source
  - or install with apt
- Download and install RTI Connext DDS 5.3
- Set the `NDDSHOME` environment variable to the root of RTI Connext DDS, e.g.: `\\opt\\rti_connext_dds-5.3.0`

Build

1. Clone git repository

```
$ git clone git@github.com:antaljanosbenjamin/cps_homework
   .git
```

2. Generate source code from .idl files

```
$ cd cps_homework
$ {NDDSHOME}/bin/rtiddsgen -language C++11 -stl -d DDS/
   Config/common -replace idl_files/Config.idl
$ {NDDSHOME}/bin/rtiddsgen -language C++11 -stl -d DDS/
   Decision/common -replace idl_files/Decision.idl
$ {NDDSHOME}/bin/rtiddsgen -language C++11 -stl -d DDS/
   Schedule/common -replace idl_files/Schedule.idl
$ {NDDSHOME}/bin/rtiddsgen -language C++11 -stl -d DDS/
   Humidity/common -replace idl_files/UvegHaz.idl
$ {NDDSHOME}/bin/rtiddsgen -language C++11 -stl -d DDS/
   Weather/common -replace idl_files/Weather.idl
```

3. Build

```
$ mkdir build
$ cd build
$ cmake ..
$ make [-j 8]
```

Run demo

1. Start humidity publisher

   ```
   $ ./cps_main h <humidityDataFilePath>
   ```

   As result of this command the application will read the data file and start to publish a humidity value every 4 minutes. The file shall contains a humidity value per line. Each humidity value is a decimal number. See example file.

2. Start IoTEdge

   ```
   $ ./cps_main e <weatherApiKey> <azureConnectionString> <
     scheduleFilePath>
   ```

   The meaning of parameteres are the following:

   - `weatherApiKey`: an API key for http://api.airvisual.com
   - `azureConnectionString`: the connection string of the device used Azure IoT Hub
   - `scheduleFilePath`: path to a CSV file which stores the schedules time intervals. See example file.

3. Start humidity controller

   ```
   $ ./cps_main c
   ```

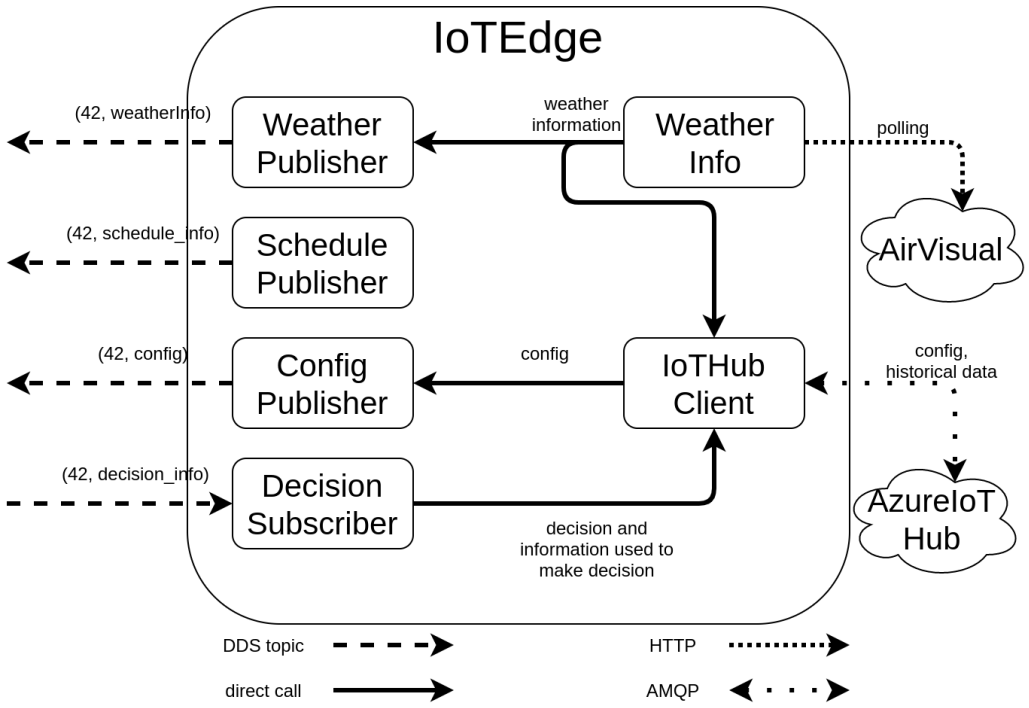

Figure 1: Architecture of IoT Edge

System architecture

IoTEdge

Figure 1 shows a boat.