Internet of Things demonstrator based on the OpenMote platform

Arturo Medina

Universitat Oberta de Catalunya

June 17, 2018

Outline

Internet of Things

The ever-growing network of Things (entities) that feature Internet connectivity and the communication that occurs between them and other Internet-enabled devices and systems

- IoT transforms ordinary objects into smart and cognitive systems.
- Mostly lacks infraestructure: multiple distributed nodes.
- Devices must be inexpensive and energy-efficient.
- Transversal: applies to many vertical markets.
- Crucial area of application: **smart homes**.

Objective

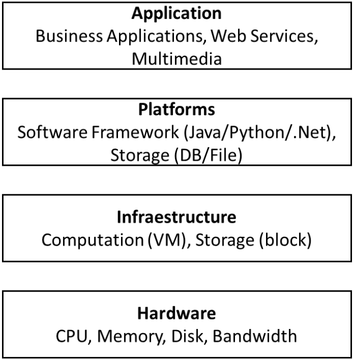
The main objective can be summarized as:
Develop an IoT demonstrator which makes use of currently relevant technologies to demonstrate the applicability of open technologies as key enablers of real-world IoT applications in the area of Smart Homes.
Other objectives include:

- Compare different approaches to IoT: the Edge Computing and Cloud Computing paradigms.
- Investigate the current ecosystem of open-source utilities, operating systems and web platforms in IoT.
- Investigate the status of open-hardware platforms in IoT.
- Make use of OpenWSN and its related tools for a demonstrator.

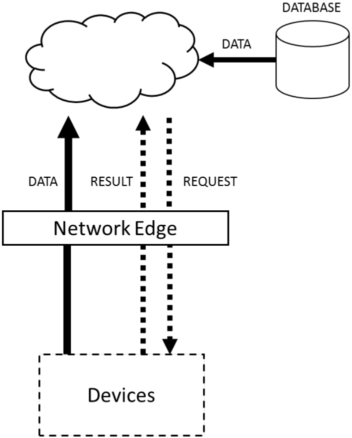
Cloud Computing

Data is stored and processed in the cloud.

- **On-demand self-service**: computing capabilities are automatically assigned.
- **Broad network access**: accessed using standard network protocols.
- **Resource pooling**: resources are shared, but keeping isolation.
- **Rapid elasticity**: adaptable to evolving requirements.
- **Measured service**: monitorable and configurable.



Edge Computing



Data is at least partially stored and processed at edge nodes.

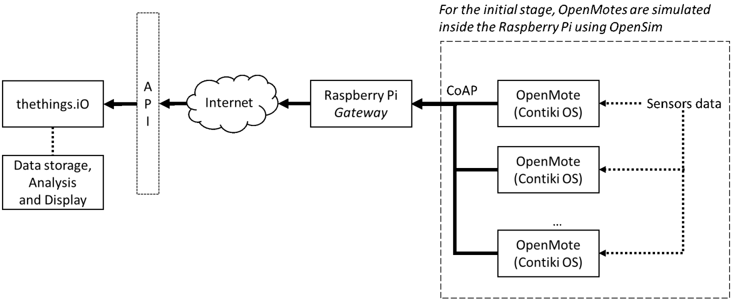
- **Reduced network load**: only computed data is sent to the cloud.
- **Reduced latency**: processing takes place closer to the end devices.
- **Power usage**: no need to send data back and forth.
- **Improved privacy**: sensitive data can be processed before sending it.
- **Greater redundancy**: multiple edge centers in case of failure.

Summary table

Design driver	Choice	Description
Low latency	Edge	Close to zero latency.
Low network traffic	Edge	Only computed data is sent.
High scalability	Edge	Computation power grows adding nodes.
High redundancy	Edge	More distributed computation cores.
Complex data processing	Cloud	Powerful computation nodes.
Centralized management	Cloud	Easier management.
Cost reduction	Cloud	Shared resources.

Table : Choice of computing model based on key design driver.

Proposed design



The following key elements compose the proposed design:

- **OpenMote nodes**: wireless nodes which read sensor data. They communicate using the 802.15.4 and CoAP.
- **Raspberry Pi**: acts as gateway. In the edge model, it also performs computations.
- **thethings.iO**: cloud platform of the system: storage, analysis and visualization of data.

thethings.iO

theThings.iO

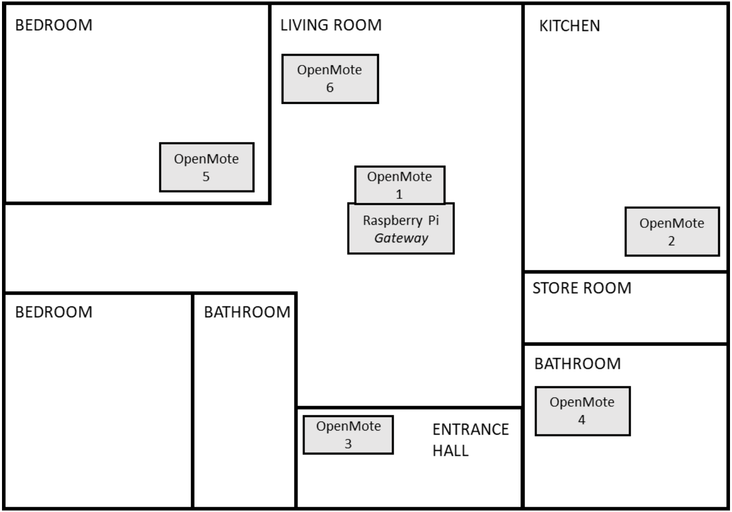
Cloud platform which provides a back-end solution for IoT applications.

- **Connectivity**: easy-to-use API. Multiple communication protocols (HTTP, Websockets, MQTT, CoAP, etc.)
- **Device Management**: devices can be individually monitored and managed.
- **Cloud Code Processing & Action Management**: triggered actions and batch processing.
- **Data Monitoring and Visualization**: customizable dashboards.
- **Interoperability and integrations**: interaction with external services.

Raspberry Pi

- Runs fully-pledged GNU/Linux.
- Large RAM memory.
- Expandable storage of up to 128 GiB.
- Multi-core.
- Connectivity: WiFi, Bluetooth, Ethernet and USB.
- Expandable by means of shields.
- It can be battery-powered.
- High quality documentation.

Distribution of motes



Activities

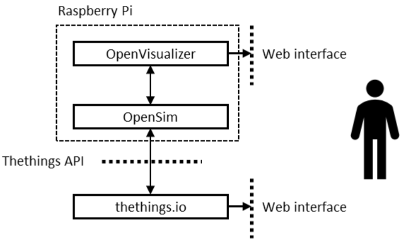
Cloud Computing

- Retrieve sensor data and send it to the cloud.
- At thethings.iO, analyze stored information to compute metrics.
- Data representation.
- Intruder detection.

Edge Computing

- Retrieve sensor data and store it.
- Analyze stored information to compute relevant statistical metrics.
- Actuate over the air conditioning system based on the temperature.
- Data cleanup.

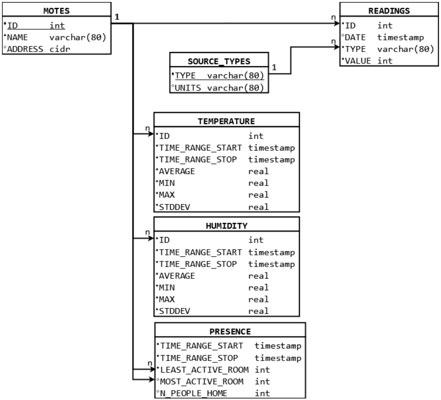
Simulation



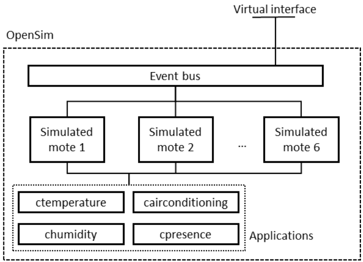
- **OpenSim**: emulate motes to generate data and inject it to the event bus.
- **OpenVisualizer**: provides a management and monitoring interface for the simulated motes.

Database

- In the edge model, data is stored in a PostgreSQL database.
- The PostgreSQL database is accessed from Python using the psycopg2 package.
- The database model ensures data consistency using primart and foreign keys.



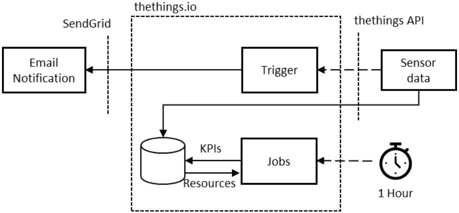
Applications



Applications developed in C as components of OpenWSN to simulate sensor data:

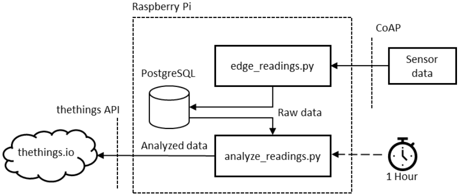
- **ctemperature**: temperature expressed in degrees celsius.
- **chumidity**: humidity expressed in %.
- **cpresence**: presence. '1' for presence detected, '0' otherwise.
- **cairconditioning**: status of the air conditioning system. '1' for enabled and '0' for disabled.

Cloud implementation



- **cloud_readings.py**: reads sensor data from the motes and sends it to the cloud.
- **Cloud-side computations**:
  - *Jobs*: executed every hour. Calculates average, maximum and minimum value of the temperature and humidity of each mote.
  - *Trigger*: executed upon data reception. If a presence is detected during the night, an email alarm is sent using the SendGrid external service.

Edge implementation



- **edge_readings.py**: every minute, reads sensor values and stores them locally. In addition, it determines if it necessary to activate or no the air conditioning system.
- **analyze_readings.py**: every hour, calculates average, maximum, minium and std. dev. of temperature and humidity. It also calculates occupancy metrics, including the most active room, the least active room and the number of people at home.

Summary table

	Cloud	Edge
Latency (ms)	1.05[a]	1
Network Traffic (GiB/year)	1102.7527	27.5688
Power consumption (kWh/year)	[b]	30.73
Cost (/year)	348	50[c]
Security	High by default, less control	Low by default, greater control

Table : [a] Delays of up to 10 seconds have been observed. [b] No estimation has been possible. [c] Including the cost of buying the Raspberry Pi.

Conclusions

- Different approaches exist regarding computation in IoT, namely Cloud Computing and Edge Computing. While both of them are valid and useful, their features make them more suitable for specific applications.
- It is possible to use open-hardware (Raspberry Pi) and open-source (OpenWSN) approaches to develop IoT applications. They become key driving factors of innovation, since they allow for easy and fast creation of prototypes and, therefore, a more efficient transition from test facilities to real deployments.
- IoT is a vibrant research trend, with a huge ecosystem of technologies and areas. Protocols such as 802.15.4 and CoAP and operating systems like RIOT OS and Contiki demonstrate the interest from both industry and acamedic institutions in creating enabling IoT technologies.

Thank you for your attention