Internet of Things: learning and practices. Application to Smart City

Olivier Debauche
Computer Science Unit
University of Mons - FPMs
Mons, Belgium
Email: olivier.debauche@umons.ac.be

Saïd Mahmoudi
Computer Science Unit
University of Mons - FPMs
Mons, Belgium
Email: said.mahmoudi@umons.ac.be

Sidi Ahmed Mahmoudi
Computer Science Unit
University of Mons - FPMs
Mons, Belgium
Email: sidi.mahmoudi@umons.ac.be

***Abstract*—Internet of Things is becoming widely present in our daily life. In fact, more and more devices able to interact together have been recently designed and launched in the market. Learning Internet of Things technologies is becoming unavoidable in education. In this paper, we propose a practical approach allowing to progressively learn, by practice the essential concepts of Internet of Things applied to Smart Cities. From basic knowledge of python language and the use of microcontrollers Pycom such as LoPy, students can develop skills and also smart applications in the field of Internet of Things.**

I. Introduction

Internet of Things is considered as the next information revolution, and we estimate the number of devices connected at horizon of 2020 at 50 billion [1]. The Internet of objects is fundamentally different from the Internet of machines. The Internet networks of objects are dynamic, distributed, low throughput and composed of large quantities of heterogeneous objects from a functional and technical point of view [2], [3].

The Internet of Things includes a wide range of various smart interconnected devices such as sensors, effectors, machines, autonomous devices, drones, intelligent cameras, etc [4]. Smart objects are generally compact devices with communication capabilities, identified by name and address, with computing capabilities, powered by batteries or solar panels [2], [5]. Connected objects are dynamic and have self-adaptive capabilities and very variables amount of data that can go up to 1,000,000 data by second in the case of a GPS for example. They can also be self-configurable and support interoperable communication protocols. These objects can be producers of data, data-consumers or endorse both previous roles, perform control or monitoring actions. The produced data are mainly eventual and temporal [3], connected objects can take the roles of sensors, actuators, controllers, or activity monitors [6].

Nowadays, learning activities related to Internet of Things technologies present a great challenge for educational institutes. Furthermore, all Master specialization are not familiar with the required knowledges of Internet of Things, electronics and related computing. Teaching Internet of Things for non-initiated public presents an important challenge. That is why; we propose in this paper to design a new protocol of practical works which lead progressively students from basic knowledge in Python programming and Low-Power Wide-Area communication protocol to the interaction of connected objects together without human intervention.

The learning protocol dedicated to Internet of Things, presented in this paper, is developed for Architect Engineer Master Student at the University of Mons (Belgium). We propose to explore different sensing methods in order to gradually lead students to understand different concepts and methodologies used in the Internet of Things.

This paper complete the first learning protocol dedicated to Smart Home [7].

II. Related Work

Others learning dispositive have already been proposed such as Gonzalez et al. in 2008, which proposed an architecture based on using a NOKIA 6131 NFC mobile phone and some RFID tags [8] in the aim to interact with physical space. Gómez et al. 2013 proposed an architecture composed of integrates NFC, QR code technology identication and communication protocols (3G or 4G) to interact with physical objects [9]. Fernandez et al. 2015 used led RGB, Arduino and Raspberry PI for teaching Internet of Things to children from 10 to 12 years [10]. More recently, He et al. [11] developed a learning dispositive for undergraduate students composed of Raspberry Pi Arduino and a set of sensors supporting Zigbee. We also previously proposed a learning and practice dispositive for smart home [7].

III. Material & Software

Our learning hardware is composed of three material thematic packages:

- The first one is distributed individually or for groups of two students and contains LoPy material;
- The second package contains PyCom material where the objective is to extend capabilities of LoPy in order to use sensors;
- The third thematic package is composed of a common connected material using Ethernet connections to actuate related devices.

The main devices that compose our material package are:

1) **Nano LoRa Gateway:** A LoPy implanted on an extension board with one antenna WiFi and one antenna LoRa

that ensures the transmissions of package LoRaWan on Internet via the Wireless network. We use the source code provided by PyCom on GitHub[1] to achieve the gateway (Fig. 1). This LoRa / Wi-Fi gateway is used in case of a The Things Network gateway is not available in close proximity. In this case, the Nano Gateway can be deployed to a gateway that is then connected to The Things Networks. The Things Networks is a free community LoRaWan network.

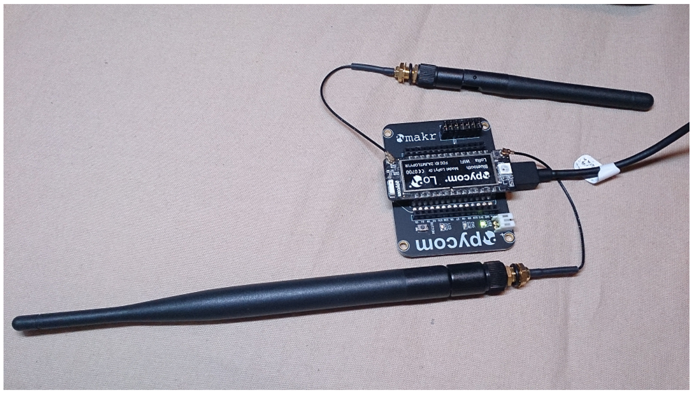

Fig. 1. Nano Gateway LoRaWan

2) **MQTT Server**: A Raspberry Pi 3 with the last release of Raspbian that hosts the last version of Eclipse Mosquitto^{TM}, which is a lightweight of the MQTT Server supporting the MQTT protocol versions 3.1 and 3.1.1. The MQTT Server provides a publish / subscribe messaging service in order to distributes external data to subscribers.
3) **LoPy devices:** Each couple of students received an individual kit composed of an LoPy (Pycom), an USB cable, an expansion board 2.1, a Pysense board (Fig. 2), a bread board, a temperature sensor DS18B20, a set of resistors, breadboard and a set of cables. PyCom microcontrollers are equipped by a Dual Processor Espressif ESP32 chipset, a hardware floating point acceleration, support hash/ encryption SHA, MD5, DES, AES and is provided of 2 UART, SPI, 2 I^{2}C, I^{2}S, micro SD card and a Real Time Controller (RTC) at 32.768Hz. The deep sleep mode consumption is only of 10 μA and the hibernation mode 1 μA. Furthermore, all these microcontrollers weighs only 7g and support SSL/TLS and WPA enterprise. The firmware of each LoPy must be upgraded before beginning to code[2] in version 1.18.0 or higher. The firmware of the Pysense must be upgraded in version 0.0.8 or higher[3] before using it.
4) **Actuation material:** The actuation material that we used contains on one hand a Connected Relay Module, which allows to activate a set of 16 Relay Module Card by mean of a web module WEB_Relay_Con V.1 (GeekTechnesNC1601) (see Fig. 3). The Web module is especially connected on network with Ethernet interface. Each relay can be switched on or off from a simple formatted url[4]. Each relay is individually isolated by an optocoupler, which isolates the relay control circuits in 12Vdc from the voltage actuated by relays, which can go up to 230Vac / 30A. The Web module is powered by 5Vdc.

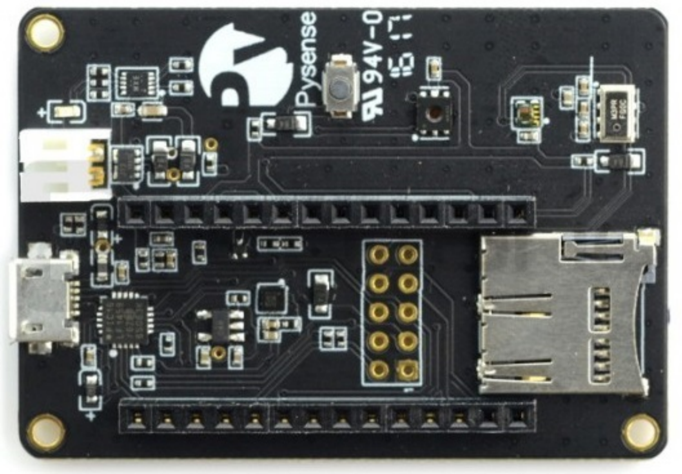

Fig. 2. Pysense

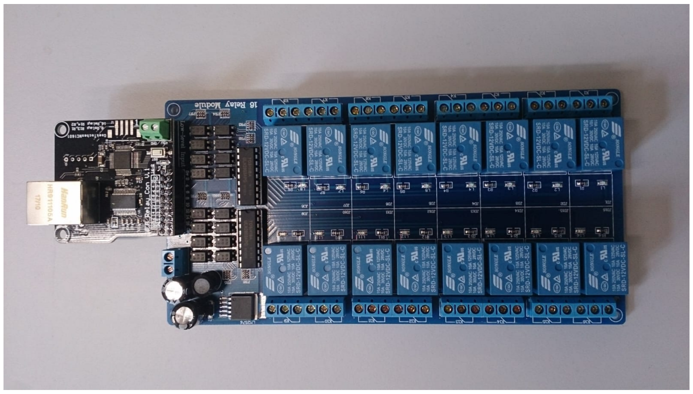

Fig. 3. Relay module with its Ethernet module

On the other hand a M-Duino PLC Arduino ARDBOX PLC 20 I/Os RELAY HF (see Fig. 4) based on an Arduino Leonardo for each couple of students which provides: 1 Serial Peripheral Interface (SPI), 1 Inter-Integrated Circuit (I^{2}C), 1 RS485 serial communications (Half-Full Duplex), 1 USB port, 10 digital inputs (5-24 Vdc) with 6 configurable by software in Analogs input (0-10 Vdc), 8 outputs (Relay 220 Vac).

In terms of software, Atom[5] 1.27.2 and higher with the

[1] https://github.com/pycom/pycom-libraries/tree/master/examples/lorawan-nano-gateway

[2] https://docs.pycom.io/chapter/gettingstarted/installation/firmwaretool.html

[3] https://docs.pycom.io/chapter/pytrackpysense/installation/firmware.html

[4] https://www.banggood.com/Ethernet-Control-Module-With-16-CHs-Relay-For-Arduino-LAN-WAN-WEB-Server-RJ45-Android-iOS-p-1189019.html

[5] https://flight-manual.atom.io/

Fig. 4. M-Duino PLC Arduino ARDBOX PLC 20 I/Os RELAY HF

Pymakr Atom Package[6] 1.3.3 and higher are used to provide a relationship between the different control chains. The installed application allows to program respectively Pycom microcontrollers and upload application on devices by mean of an usb connection between a computer and the expansion board.

Node-Red [12] is used to provide a relationship between different control chains. This application is installed on the top of Node.js [13] LTS 8.x allows to interconnect a system objects.

IV. Methodology

The learning protocol designed in our practical work is aiming to allow students to connect devices like LoPy to a computer and achieve some sensing actions according to the information gathered from the connected devices. In the remaining of this section we will describe the different steps of this process:

A. First connection in LoRaWan

The aim of this first experimentation is to familiarize students with the foundation of programming on LoPy and Atom with Pymakr package manipulation and establish a connection with the LoRaWan Network.

This stage needs a personal computer with Atom with Pymakr package installed, 1 LoPy, 1 pigtail, 1 antenna LoRaWan, 1 USB cable A male - micro B male.

At this stage, students use a LoPy, which is directly connected to a personal computer by a USB cable. The first manipulation consists to create a free account on The Things Network. Then, they create a new application and add the device in this application with an activation method Over-the-Air Activation (*OTAA*). This activation method provides a better security level than Activation by Personalization (*ABP*) method.

The Things Network provides in return on one hand a Application Identification (*AppEui*) and on the other hand an application key (*AppKey*). Students upload a software provided to them on the LoPy, which displays the identification in hexadecimal of the device (*DevEui*). This key must be modified in The Things Network in the corresponding device previously registered in the console. These mandatory parameters allow to create three others (*DevAddr*, *NetSKey*, *AppSKey*), which are need for the authentication and the double encryption of data of the device (see Fig. 5).

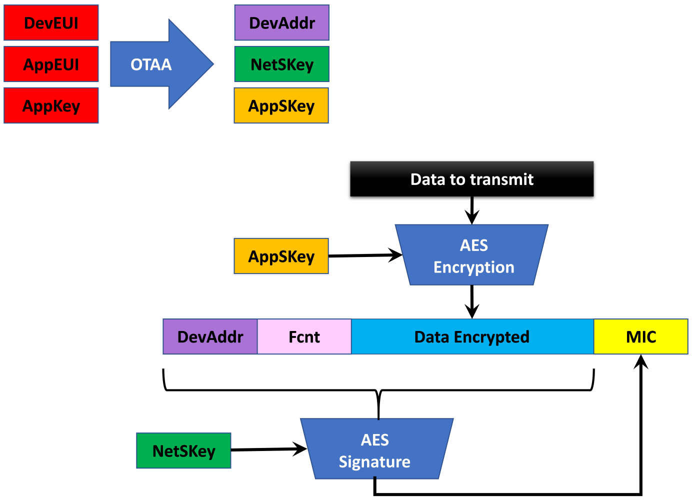

Fig. 5. Schema of authentication and transmission of data

Students received a program where they must add AppKey and AppEui parameters before uploading it in the LoPy.

B. First transmission of data

The aim of this second experimentation is to transmit data from a simple sensor to the LoRaWan Network.

This stage needs a personal computer with Atom with Pymakr package installed, 1 LoPy, 1 pigtail, 1 antenna LoRaWan, 1 expansion board, 1 breadboard, 1 DS18B20, 1 resistor of 4.7 Kohm, 1 set of cables for breadboard male-male, 1 USB cable A male - micro B male.

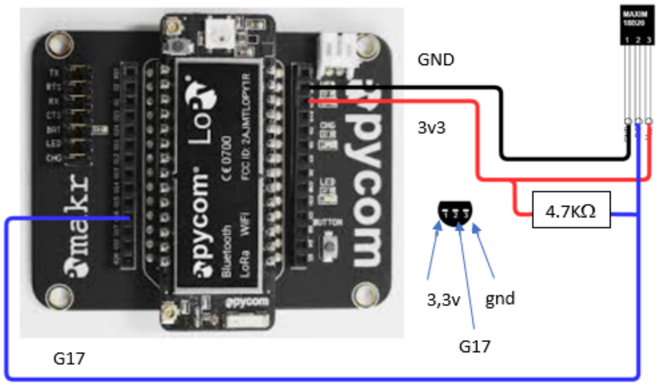

Fig. 6. Implantation schema of the DS18B20

At this stage, students implement one temperature sensor with a resistor in serial on a LoPy and then create a basic

[6] https://atom.io/packages/pymakr

program to connect to the LoRaWan gateway and send the value measured by the temperature sensor (see Fig. 6).

C. Using of several sensors

The aim of this third experimentation is to acquire data from several sensors, send data on The Thing Network, uses integration with Canneye database in order to finally elaborate a dashboard on myDevices.com.

This stage needs in addition of the material used in the previous exercise, a Pysense board. This board is used in replacement of the DS18B20 to acquire temperature, ambient light, barometric pressure, humidity data.

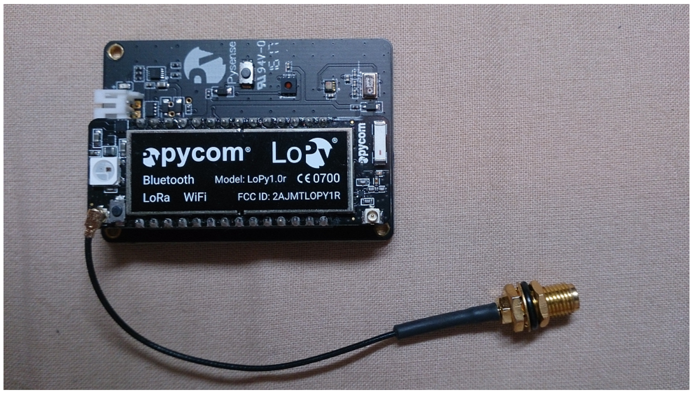

Fig. 7. LoPy with Pysense

The LoPy is extracted from the Expansion Board and plug in the Pysense Board (see Fig. 7). Students must add an integration of Cayenne type in the The Things Network and then create a free account on myDevices[7]. Then, choose LoRa to start a project. In the left column, The Things Network can be selected and a list of devices is displayed. In this one, choose "Cayenne LPP (Cayenne Low Power Payload)" and encode the *DevEui* previously encode in The Things Network during the first stage.

D. Chain sensor to actuator

In lasts stages, students have learned to send data on the LoRaWan Network (The Thing Network - TTN), store data in a Cayenne Database and then retrieve data from Cayenne Database to create a dashboard on myDevices. In the Internet of Things, actuators generally controlled remotely are used to perform actions on the external environment in reaction of stimuli.

The aim of this experimentation is to use Node-Red to control an actuator connected with an usb cable to personal computer and turn on or off a relay which opens or closes a circuit powered according to the value of the sensor transmitted via the LoRaWan protocol to TTN.

In this step, we will use for each group: 1 programmed M-Duino with its 12 Vdc power supply, 2 USB cable A male - micro B male, 1 LoPy mounted on a Pysense board, 1 pigtail, 1 antenna LoRaWan, 1 red led, 1 breadboard, 1 resistor of 4.7 Kohm, 1 set of cables for breadboard male-male, 1 Power supply 4 x AA batteries, 1 personal computer. Each couple of students makes a circuit with the red led, the resistance and the 6V power supply on a breadboard and verify that the led light. This circuit is then interrupted and connected to the relay R1 of M-Duino. Node-Red and its extension module TTN[8] will be used to retrieve data from TTN. The exceeding of a threshold value is verified and the relay is activate or not in function of the parameter value. Both string are provided to students which must send them to M-Duino via Node-Red to activate or deactivate the relay R1. Students must implement schema shown in the Fig. 8

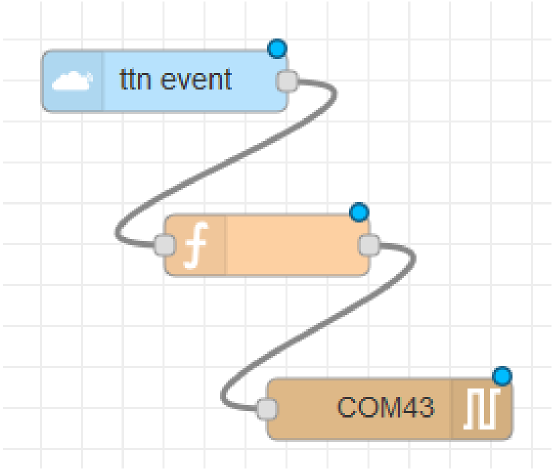

Fig. 8. Node-Red schema to turn on/off the Ethernet device

E. Multi sources of data

In the previous stage, we have seen the implementation of a chain from sensor to actuator. In the Internet of Things data are not only coming from sensors of the application but others sources must be considerate to take a better decision.

The aim of this stage is to use at same time on one hand data from sensors transmitted by a LoRaWan network and on the other hand an open data external source accessible via an MQTT server.

In this step, each couple of students will use: 1 USB cable A male - micro B male, 1 LoPy mounted on a Pysense board, 1 pigtail, 1 antenna LoRaWan, 1 personal computer and in common for all groups a Connected Relay Module connected to the local network by means of an Ethernet cable. Each group of students received two url associated to a relay of the Connected Relay Module to respectively turn on or off the relay. The url must be send using Node-Red to the Ethernet device. The Fig. 9 shows the scheme, which must be implemented in Node-Red to retrieve date from a device. The App and the DeviceId must be customized in the TTN node. External data are available by subscription to a topic of the MQTT Server hosted by the Raspberry Pi 3.

[7] https://mydevices.com

[8] https://www.npmjs.com/package/node-red-contrib-ttn

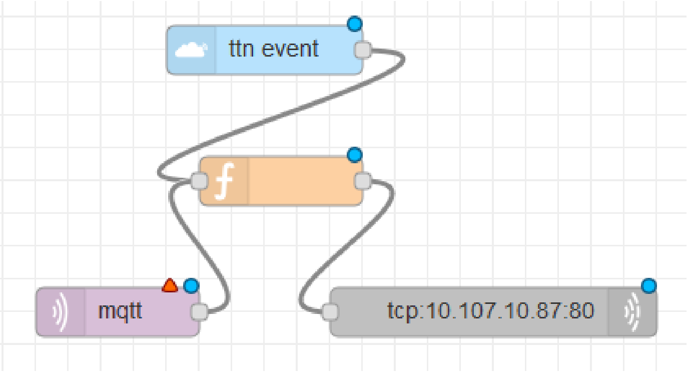

Fig. 9. Node-Red schema to turn on/off the Ethernet device

V. Conclusion

The learning protocol proposed in this paper lead students in a first time to be able to manipulate an LoPy directly connected to their own computer. This learning protocol use successively individual and integrated sensors on a board to send data by LoRa modulation to a gateway of The Things Network. Then, an integration with a Cayenne database is used to store data and finally create a dashboard. The next stage use Node-Red to interface The Thing Network with a local USB actuator. Finally, Node-Red is used to interface TTN network with an Ethernet device and subscribe to external data by mean of MQTT protocol. The proposed protocol has been experienced with 2^{nd} Master Engineer students specialized in Architecture of the Faculty of Engineering at the University of Mons. Actually, a weakness of our approach is to be dependent of LoPy Pycom with the risk of unavailability, disappearance or incompatibility with future versions of the hardware. The disappearance of LoPy microcontrollers is less likely. Therefore, it is necessary to be independent of the manufacturers in order to perpetuate practical works.

As future work, we plan to exploit GPU Tegra Mobile Processors that offers high computation power thanks to the exploitation of GPU processors in parallel [14] [15]. In addition to the high performance, the parallel implementations would offer lower power consumption [16] as result of the fast treatment.

Acknowledgments

We would like to thank our colleagues from Biosystems Dynamics and Exchanges Axis, Biosystem Engineering Department, Gembloux Agro-Bio Tech (ULiège) without whom this work would not have been possible. We would especially like to thank Mr Rudy Schartz and Mr Adriano Guttadauria for their technical support and for setting up all the electronic systems and computing systems necessary for carrying out this research.

References

[1] D. Evans. The internet of things: how the next evolution of the internet is changing everything. Technical report, White paper, CISCO, 2011.

[2] N. Shahid and S. Aneja. Internet of things: Vision, application areas and research challenges. In *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pages 583–587, 2017.

[3] B. Billet. Systme de gestion de flux pour lInternet des objets intelligents. *PhD Thesis, 2015. University of Versailles, Saint-Quentin-En-Yvelinnes (France)*, 2015.

[4] L. Atzori, A. Iera, and G. Morabito. From ”smart objects” to ”social objects”: The next evolutionary step of the internet of things. *IEEE Communications Magazine*, 52(1):97–105, January 2014.

[5] S. Cherrier. Architecture et protocole applicatifs pour la chorgraphie de services dans l’Internet des objets. *PhD Thesis, 2013. University of Paris-East, France*, 2013.

[6] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao. A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. *IEEE Internet of Things Journal*, 4(5):1125–1142, Oct 2017.

[7] O. Debauche, S. Mahmoudi, M.A. Belarbi, M. El Adoui, and S.A. Mahmoudi. Internet of things: learning and practices. application to smart home. In *Proceedings of the International Conference on Advanced Communication Technologies and Networking*, Marrakech, Morocco, 2018.

[8] G. R. Gonzlez, M. M. Organero, and C. D. Kloos. Early infrastructure of an internet of things in spaces for learning. In *2008 Eighth IEEE International Conference on Advanced Learning Technologies*, pages 381–383, July 2008.

[9] J. Gmez, J. F. Huete, O. Hoyos, L. Perez, and D. Grigori. Interaction system based on internet of things as support for education. *Procedia Computer Science*, 21:132 – 139, 2013. The 4th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2013) and the 3rd International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH).

[10] G. C. Fernandez, E. S. Ruiz, M. C. Gil, and F. M. Perez. From rgb led laboratory to servomotor control with websockets and iot as educational tool. In *Proceedings of 2015 12th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, pages 32–36, Feb 2015.

[11] J. He, Dan Chia-Tien Lo, Y. Xie, and J. Lartigue. Integrating internet of things (iot) into stem undergraduate education: Case study of a modern technology infused courseware for embedded system course. In *2016 IEEE Frontiers in Education Conference (FIE)*, pages 1–9, Oct 2016.

[12] M. Blackstock and R. Lea. Toward a distributed data flow platform for the web of things (distributed node-red). In *Proceedings of the 5th International Workshop on Web of Things*, WoT ’14, pages 34–39, New York, NY, USA, 2014. ACM.

[13] S. Tilkov and S. Vinoski. Node.js: Using javascript to

build high-performance network programs. *IEEE Internet Computing*, 14(6):80–83, Nov 2010.

[14] S.A. Mahmoudi and P. Manneback. Multi-gpu based event detection and localization using high definition videos. In *2014 International Conference on Multimedia Computing and Systems (ICMCS)*, pages 81–86, April 2014.

[15] S.A. Mahmoudi and P. Manneback. Multi-cpu/multi-gpu based framework for multimedia processing. In *Computer Science and Its Applications*, pages 54–65, Cham, 2015. Springer International Publishing.

[16] S.A. Mahmoudi, M.A. Belarbi, S. Mahmoudi, and G. Belalem. Towards a smart selection of resources in the cloud for low-energy multimedia processing. *Concurrency and Computation: Practice and Experience*, pages e4372–n/a, 2017. e4372 cpe.4372.