Demo: WAZIUP, an Open and Versatile Long-range IoT Framework to Fully Take Advantage of the Cloudification of the IoT

Congduc Pham, Mamour Diop
Univ. Pau, LIUPPA Laboratory
congduc.pham@univ-pau.fr, mamour.diop@univ-pau.fr

***Abstract*—The EU H2020 WAZIUP project builds an open IoT platform tailored to the specific requirements and constraints of developing countries. WAZIUP proposes a 100% DIY IoT approach for very low-cost IoT, yet highly efficient. One of the core component is a flexible, versatile, open-source IoT gateway with extensible customization features to fully take advantage of all data management possibilities offered by the cloudification of the IoT ecosystem. In the proposed demo, we will showcase the WAZIUP solutions for low-cost, long-range IoT and focus on the various possibilities offered by our versatile IoT gateway and how the platform can be used to integrate advanced control mechanisms from research works.**

I. Introduction

The era of IoT can connect a large variety of sensors and devices. In turn, the challenge is about driving business outcomes and creation of new value. However, developing countries are facing many difficulties – lack of infrastructure, high cost of hardware, complexity in deployment, etc – when it comes to real deployment of IoT solutions [1], especially in remote and rural areas. The EU H2020 WAZIUP project contributes by reducing part of this technology gap. WAZIUP is focusing particularly in deploying IoT and Big Data platforms for sub-saharan African countries but many of its core propositions target developing countries is general. In this context, IoT deployment must address 3 major issues: (a) Longer range for rural access, (b) Cost of hardware and services and (c) Limit dependancy to proprietary infrastructures.

For longer range, recent Low-Power Wide Area Networks (LPWAN) technologies for IoT introduced by Sigfox and Semtech's LoRa™offer several kilometers range without relay nodes to reach a central gateway, thus greatly simplifying large-scale deployment of IoT devices as opposed to the need of complex multi-hop routing with short-range radios.

II. The WAZIUP IoT Platform

The WAZIUP IoT framework consists in code templates, libraries and software for LoRa IoT devices and gateways [2].

A. IoT devices

The small form factor Arduino Pro Mini board that is available in the 3.3v & 8MHz version for much lower power consumption can definitely be used to provide a generic low-cost IoT platform as it can be purchased for less than 2 euro from Chinese manufacturers. For more demanding IoT applications (e.g. innovative image sensors) we use the Teensy family boards (LC/31/32) that offer powerful micro-controllers with more memory and advanced power management features at a very reasonable cost (about 10 euro for the LC).

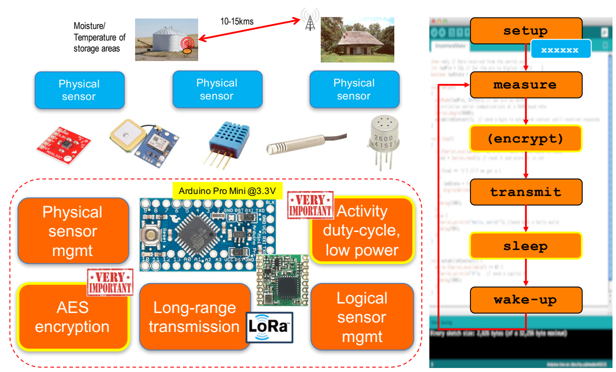

Fig. 1. Generic IoT platform

Fig. 1 also shows data encryption and low-power management building blocks offer security and several years of autonomy with these off-the-shelves components. Fig. 2 shows how the generic platform is customized to build a GPS collar device for cattle localization.

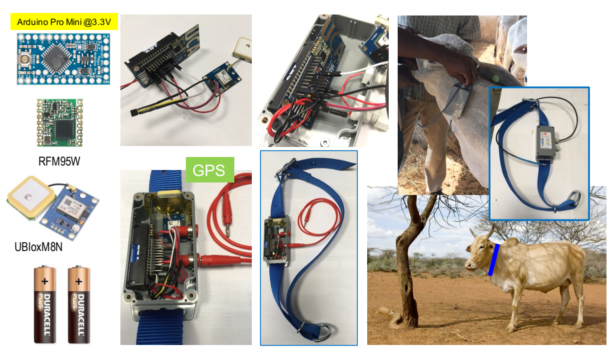

Fig. 2. GPS collar IoT

B. The Versatile Gateway

Our gateway is single-channel but can also receive (and decrypt if necessary) LoRaWAN packets on DR0 channel. Various versions of the gateway is shown in Fig. 3, depending on the intended deployment usage: indoor/outdoor, connectivity options, mobility,. . . . The gateway architecture highly decouples the low-level gateway functionalities from the high-level data post-processing features, privileging high-level languages for the latter stage. For instance, the provided main post-processing stage is written in `Python`.

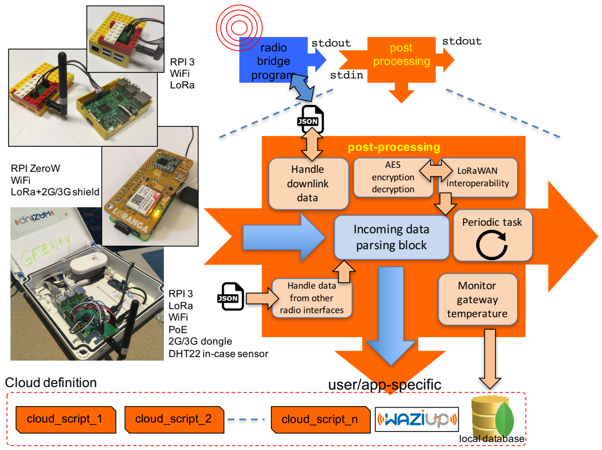

Fig. 3. Modular and versatile low-cost gateway

The gateway can easily handle offline scenarios where Internet connectivity is not available: SMS can be sent if a 2G/3G dongle is attached and received data can be locally stored in a NoSQL MongoDB database. The gateway also embeds several web interfaces for gateway configuration/update, local data visualization (including images from innovative image sensors) and GPS data visualization from remote GPS devices such as the GPS collar shown previously in Fig. 2. New web interfaces can be developed and added for specific usages.

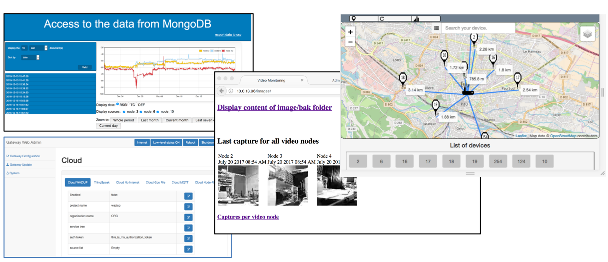

Fig. 4. Embedded web interfaces

The full power of the gateway in the context of the cloudification of the IoT ecosystem is when received data are pushed to IoT clouds. Here, customizing data management tasks can be done in a few minutes, using standard tools, simple REST API interfaces and available public clouds. We provide ready-to-use cloud scripts that already supports a number of publicly available IoT clouds such as `Firebase`™, `ThingSpeak`™, `GroveStream`™ & `FiWare`™ as well as MQTT brokers to name a few. These cloud scripts can be written in various programming/scripting languages such as `Python`, `Go` and shell scripts, to name a few, and can perform from simple to very complex processing tasks, such as performing device address filtering, defining field association schemes, using dedicated cloud API interfaces and authentication mechanisms. On data reception, the post-processing stage will execute all enabled cloud scripts sequentially therefore data can be pushed on various clouds at the same time if necessary.

There are therefore 3 options for customizing the gateway: (i) the "geek way" by modifying/extending the main post-processing script and the "smarter" ways by (ii) adding "cloud" scripts to insert additional processing tasks on packet reception or by (iii) hooking new tasks into the periodic tasks loop which is called independently from packet reception. An example of the periodic tasks loop is to update the gateway's GPS position in fully mobile scenarios. To illustrate the second option, the `CloudNodeRed.py` script easily turns your gateway into a Node-Red node to be integrated in more complex task processing flows developed with the graphical Node-Red environment which provides a large variety of ready-to-use data processing blocks. `CloudNodeRed.py` simply writes into an intermediate file, here `nodered.txt`, that will be monitored by the Node-Red `tail` block. Then further data formatting can be done prior to injection to additional Node-Red processing blocks (Graphics, MQTT, ThingSpeak,…) as illustrated in Fig. 5.

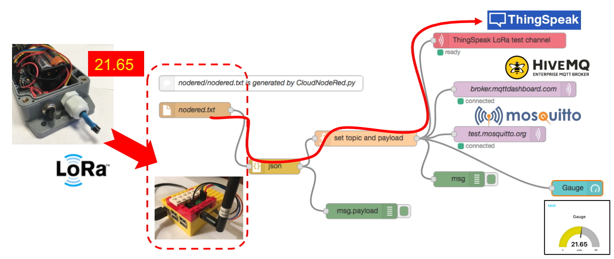

Fig. 5. Node-Red support

C. Integrating advanced features

The platform is particularly well suited for integrating and experimenting advanced control mechanisms from research works. In WAZIUP we propose a robust CSMA channel access mechanism adapted to LoRa networks to better prevent packet collision in dense deployment scenario. In countries under duty-cycle regulations for LoRa transmissions, the framework also proposes an advanced radio activity-sharing mechanism allowing some devices (for instance resource-demanding image sensors) to borrow activity time from other devices in order to globally satisfy the duty-cycle requirements [3].

III. The Proposed Demo

In the proposed demo, we will showcase the WAZIUP solutions for low-cost, long-range IoT and the possibilities offered by the versatile WAZIUP IoT gateway for deploying operational solutions and conducting research experiments.

References

[1] M. Zennaro and A. Bagula, "Iot for development (iot4d). in iot newsletter." July 14, 2015.

[2] C. Pham, "A diy low-cost lora gateway. http://cpham.perso.univ-pau.fr/lora/rpigateway.html and https://github.com/CongducPham/LowCostLoRaGw", accessed April 26, 2018.

[3] C. Pham, "Qos for long-range wireless sensors under duty-cycle regulations with shared activity time usage," *ACM TOSN*, vol. 12(4), 2016.