Published in the Proceedings of the 2016 *IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, 23-25 March 2016, Crans-Montana, Switzerland, which should be cited to refer to this work.

DOI: https://doi.org/10.1109/AINA.2016.171

Dynamic Street-Parking Optimisation

N. Pazos, M. Müller, M. Favre-Bulle, K. Brandt-Dit-Grieurin, O. Hüsser, M. Aeberli, N. Ouerhani
Applied Science University Western Switzerland (HES-SO)
HE-Arc Engineering School
Switzerland
Nuria.Pazos@he-arc.ch

***Abstract*—An instrument to improve the quality of life in large cities, helping to reduce the car traffic, is presented in this paper. It will result in a mobile guidance software that will help the drivers looking for a parking place to find it efficiently. *SmartPark* relies on available parking information systems, as well as on new sensors or even on social data inputs. A fixed magnetic on-street sensor and a video processing smart camera have been developed and prototypes of both devices were tested. Their data is available through a cloud-based Internet of Things infrastructure and continuously updated every few seconds. Databases will be built over time enabling data mining methods to infer parking availability models over time which will be used, eventually, by the algorithms feeding the mobile application.**

***Keywords*—Smart cities; Internet of Things; Image processing; Smart parking; Incremental algorithm;**

I. Introduction

In almost every big city worldwide, finding a free parking place is a recurrent problem which takes a considerable amount of time and causes irreversible CO2 gas emissions. The main objective of this project is thus to enable a real-time analysis of free parking places in a city and to optimally guide the drivers towards one of these free places.

The huge size of some cities, make it unpractical to draw conclusions at the scale of the whole city. Therefore it becomes necessary to develop a multiscale approach, i.e. an incremental analysis method. This method will be carried out by guiding the driver towards a zone, then to a suburb followed by a precise street and, finally, to a single free place.

The current *SmartPark* project relies on four basic components: (i) Data collection and analysis; (ii) Incremental search algorithm; and (iii) Driver guidance application. For the data collection, four data sources have been applied: (i) Dedicated sensors located on public parking places; (ii) High resolution embedded camera systems placed at strategic locations of public parking areas; (iii) Analysis of other data sources for crowd movements such as social network or planned events; and (iv) The already available PGIS (Park Guidance Information System) information.

These data are collected and pre-analyzed through a Cloud application working permanently and offering different topologies of parking possibilities in the target city. The incremental search algorithm, will be deployed as a server application capable to link the sensed and analyzed data together with the particular context of each user including his needs and preferences.

Finally, the end-user (driver) software is a mobile application, using the smartphone GPS, that will guide the driver towards an adequate parking place with a user-friendly interface compliant to driving modus. Such an application is thought to offer a high tolerance and flexibility to leave the driver adapt himself in real-time to the potential variations of the sensed data.

A. Objectives

This project tackles the development of a smart and efficient solution for assisting the parking in the mid-and big-size cities. Unlike other projects, focusing on a smart management of parking places, *SmartPark* handles the parking issue globally, including all together the street parking, the parking lots and public parking areas, and by applying simultaneously different data sources for estimating the occupancy of the target parking places.

B. Novelties

The key characteristics offered by the *SmartPark* platform are:

1. Data collection and pre-analysis at the edge

A careful analysis has been carried out to select, develop and integrate different relevant sources of information to build an inventory of parking places in a city, as snapshots and probable trends.

2. Real-time reasoning for the optimization of responses to end user requests

A business layer, on a server, implements a routing algorithm that will make use of the distributed intelligence of smart sensors to perform a multiscale incremental analysis. This provides a stable and reliable solution for the driver, guiding him with more precise directions when he comes close to the destination.

3. Guidance to a free suitable parking place

The third and final layer, a user-friendly smartphone application, using the GPS receiver, the business layer as well as traffic information sources, guides the driver towards a free parking place, in proximity of the target location. User preferences are also taken into account (pricing, vehicle dimension, etc.).

C. Structure of the Paper

This paper is structured as follows: Section II summarizes and compares the related work in order to highlight the advantages introduced by *SmartPark*. The proposed platform architecture is then shown in Section III, while Section IV gives the implementation details for each of the target components. Different experimental results are then presented in Section V followed by Section VI which gives the conclusions and sets the perspectives.

II. State of the Art

Since the mobility has become one of the key factors of comfort inside a city, numerous projects pursuing diverse smart transport solutions have raised during these last decades. They tackle different aspects directly linked to the mobility, such as:

- Fluidity of traffic circulation
- Security and driving assistance
- Control and regulation of traffic flows
- Driven distance optimization
- Smart management of parking lots
- Optimization of access to public transports.

Within the scope of the smart mobility, the problem related to parking has also been largely studied. Nevertheless, the focus has been mostly on the management of parking lots, private or public. To date, only few projects have considered the problem globally, as the *SmartPark* project does. It handles the optimization of parking places in underground parking lots together with the optimization of street parking places and public parking areas in general.

One of the most emblematic initiative in the smart parking domain is probably the San Francisco *SFPark* project [1] initiated in 2011. It is to date the largest pilot experiment in a big city. More than 8000 wireless sensors were installed on the streets to monitor in real time the parking places availability. This data was made available to drivers through smartphone applications. It was also used to dynamically adjust the pricing of the parking in different areas according to availability and demand. This project is succeeding in reducing the average cycling time of drivers (search of the parking place), improving the overall parking usage by balancing the car load in different areas, and reducing the car citations statistics (drivers respect the law). Furthermore, it helped to improve and asses some sensor technologies readiness to perform reliably over a long time.

Other projects in England (London), Netherlands (Delft) or Spain (Santander) have also demonstrated that the return on investment of deploying sensors for parking measurements was very fast, and could be as low as a couple of years.

Compared to those existing solutions, our project includes different and complementary data sources for sensing the occupancy of parking places which can be either added or merged, depending on the context. Moreover, this physical data will be pre-analyzed according to other data sources, such as social networks or sources of information announcing single events and/or incidents. Another innovation thread introduced by the current project is the management of public parking areas by means of smart cameras.

In this last domain of parking monitoring and management through smart cameras integration, there exist several attempts trying to master this issue. As explained by Ydris in his paper [2], for vehicle and free places counting in a parking lot, the smart cameras are a very effective, non-intrusive, mean of detection. In some cases, the available infrastructure dedicated to safety monitoring could be reused for this purpose, providing a rapid and low-cost solution. A single smart camera, placed at a good stand point, could monitor from 10 up to 100 vehicles with high resolution and lighting conditions. Of course, strategies with complementary sensors should be considered for environmental conditions reducing visibility such as night, fog or snow.

One of the main stakeholder in the smart parking field [3] emphasizes the complementarity between the street fixed sensors and the needs covered by image processing devices in specific parking areas such as open parking lots and temporary or reconfigurable parking places.

Cameras placed on the street- and road-side could also be used to measure the traffic fluidity between parking places [4]. Together with the car's GPS system [5], services like *Here* [6], *GoogleTraffic* or publicly available traffic information OFROU [7], these cameras could help to optimize the routing algorithm with real-time traffic load estimates to guide the driver on the shortest path towards the free parking place.

We distinguish two kinds of smart video processing systems: (i) centralized and (ii) distributed [8]. Centralized systems will transfer all the data to a Cloud platform, where the complete processing and analysis will be performed. On the other side, distributed systems will reduce the load of the communication network, by delegating the image preprocessing and feature extraction phases to the smart camera, transmitting only the meaningful information (i.e., free place number, traffic flow velocity).

III. SMARTPARK PLATFORM

The research of an optimal solution for the parking issue in mid- and big-size cities is a complex topic. Beyond the known factors which influence nowadays the equation, for example, the rotation rate of street parking places, we can expect that new metrics will be added in the close future to the already complex equation. Therefore, in order to remind flexible and scalable towards future enlargements of the initial solution, the proposed *SmartPark* platform is built around a modular architecture capable of:

- Collecting a variable number of data sources related to the parking issue.
- Pre-analyzing and classifying these data sources to generate global trends.
- Drawing conclusions incrementally regarding the user needs according to the available data sources.
- Interacting and guiding the user intuitively thanks to a dedicated mobile application or an extended standard GPS-base application.

Such an architecture has been conceived so that it integrates several different sources of sensed data, it scales with the number of stored data, it applies analysis and classification algorithms to this data in order to improve the incremental guidance algorithm, and, finally, it optimizes the interaction with the end user. A schema of the global architecture of the *SmartPark* platform is depicted in Figure 1.

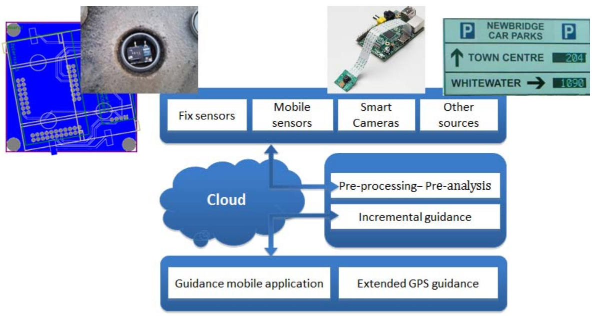

Figure 1. *SmartPark* Global Architecture

In Figure 1, we can thus distinguish three main layers, each of them containing several components:

1. Data sources layer

At this layer, a selection of the required devices for collecting the data required to lead the driver towards the free parking places has been carried out. Different kinds of sensor devices are considered: (i) Fixed sensors; (ii) Mobile sensors; (iii) Smart cameras; (iv) Smart agents on social networks; and (v) Communicating agents on existing systems (parking lots, surveillance cameras, etc.).

One of the main challenge addressed by the *SmartPark* project is that all these sensed sources converge towards a single Cloud application for pre-processing and pre-analysis. Such an application will provide then a real-time monitoring (see Figure 2) of free parking places in the target city. Furthermore, this pre-processing will reduce the response delay to end user request.

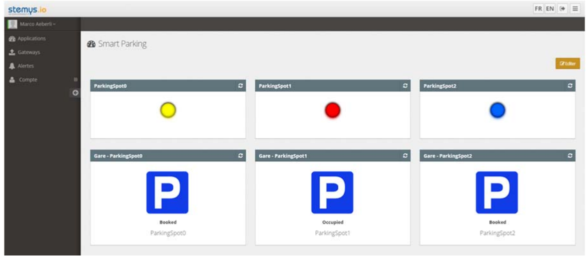

Figure 2. *SmartPark* Real-Time Monitoring

2. Algorithmic layer

This second layer implements a guidance algorithm that will offer to mobile software developers an interface to easily send parking place requests and receive a reliable response in the form of simple guidance instructions. The proposed algorithm is based on two main simplifications: On one hand, the sensing process will be coupled to a pre-analysis algorithm assisting the construction of a reliable and real-time topology of the available parking places in the target city. Such an algorithm is based on a distributed intelligence implemented at the edge capable of pre-processing and pre-analyzing the sensed data. On the other hand, the guidance algorithm for the end user is implemented as an incremental algorithm based on a chain of thoughts, starting from a global suggestion at the beginning and ending with more precise parking propositions when approaching the destination.

3. User interface layer

The third and final layer of the proposed solution covers the end user interaction interface. It is implemented as a powerful and user-friendly mobile application which, taking into consideration the constrained user context (driving modus), is capable of answering to user requests and interact with him. Two versions of this mobile software are proposed: (i) a standalone guidance application assisting the driver by suggesting potential free parking places within a user-selected perimeter and according to user preferences; and (ii) a mobile application (see Figure 3) coupled to a GPS location provider, directing the driver towards a free parking close to a pre-selected address.

Figure 3. *SmartPark* Mobile Application

IV. *SMARTPARK* IMPLEMENTATION

During the implementation of the proposed *SmartPark* platform, three different competences have contributed to the success of the project. The first competence allowed the development of the low-level magnetic-based detection system together with the sensor network to push the data associated to each individual parking place towards the central server. A second competence came up with an embedded image processing solution to identify the number of free parking places in a pre-defined parking area. And the third competence has deployed a universal gateway plus a central IoT platform capable to deal with the heterogeneous data sources and to either integrate the algorithmic layer or provide an interface to third-parties applications.

A. Low-level Occupancy Detection

At the lowest layer, the detection of free parking places is done by applying 3-axes magnetic sensors (HMC5883L) connected to a communicating node based on the MSP430 processor and the CC2420 transceiver. The *Contiki* operating system with the uIPv6 stack is running on the MSP430.

The terrestrial magnetic champ measured by the magnetic sensor is deviated by the car body being thus possible to differentiate when a car is occupying the parking place where the magnetic sensor node is installed. In order to select the required precision of the 3-axes magnetic sensor, a pre-study has been carried out by quantifying the variations of the terrestrial magnetic champ with and without the presence of a car body.

Figure 4. Magnetic Sensor Node Prototype

A prototype of the magnetic sensor node (Figure 4) has been built and tested under different conditions. The detection algorithm and the calibration process have been implemented as follows:

- Detection Algorithm: The 3-axes magnetic sensor measures and represents a magnetic vector in the space. We have thus measured the magnetic vector is two distinct cases: (i) measurement of the terrestrial magnetic field without the presence of a car body; and (ii) measurement of the deviation of the terrestrial magnetic champ when a car is placed above the magnetic sensor node. In Figure 5, we can observe two main bullets which correspond to the two main states "car detected" and "no-car detected" during a parking maneuver.

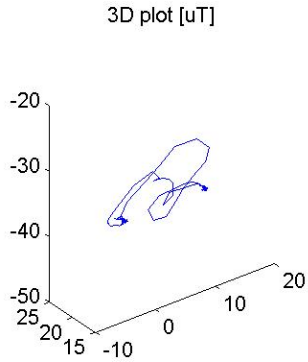

Figure 5. 3D Terrestrial Magnetic Champ Measurements (during a parking maneuver)

- Calibration process: In order to discriminate between the two main states "car detected" and "no-car detected", a calibration of the magnetic sensor node is required. To perform the calibration process, we measure the magnetic vector during several tenths of seconds and we calculate the average value together with the standard deviation. The decision threshold is then fixed to six times the standard deviation.

This decision threshold is further on used to take a decision concerning the occupancy of the parking place according to the Equation (1).

$$N(Vme.-Vmo.)>N(6xStd) \rightarrow \text{"car detected"} \quad (1)$$

The results of a concrete decision point is depicted in Figure 6, where the difference between the measured and the average magnetic vectors is greater than six times the standard deviation and thus we can conclude that the terrestrial magnetic champ has been deviated by a metallic mass and therefore a car is detected.

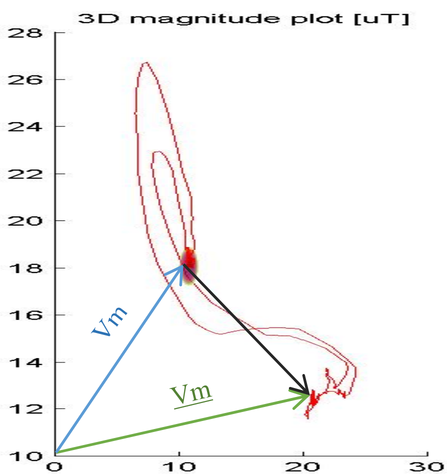

Figure 6. 3D Magnitude Plot [uT]

B. *IPv6 Magnetic Sensor Nodes Network*

As stated before, we pursue a flexible and scalable system and, therefore, the choice of a mesh topology for the magnetic sensor nodes network is a must considering that the selected low-power radio transmission technology (IEEE 802.15.4) is short range (between 10m- 100m). Furthermore, we decided to apply the uIPv6 network stack together with the RPL routing protocol, which are already integrated in the *Contiki* OS and target resource constrained embedded platforms.

In order to route IPv6 packets through an 802.15.4 radio network, the 6LoWPAN adaptation layer is required. The so constructed 6LoWPAN LLN (Low-power and Lossy Network) integrating the magnetic sensor nodes needs an ASBR (Autonomous System Border Router) to connect the local nodes to the data gateway, which will further provide the connection to Internet and so to the Cloud platform (see Figure 7 for the complete system architecture of the implemented 6LoWPAN magnetic sensor node network).

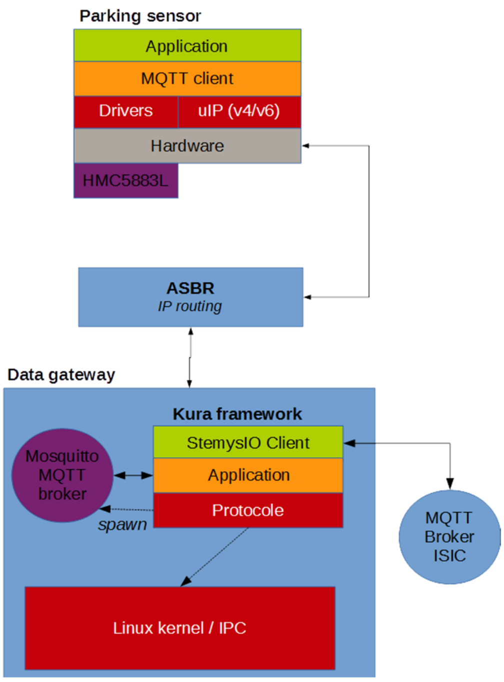

Figure 7. 6LoWPAN Sensor Network

Above the transport layer (TCP), the connection between each magnetic sensor node and the data gateway has been implemented by applying the MQTT (Message Queuing Telemetry Transport) protocol. The decision is taken based on its standardization and its reduced size. For the implementation at the magnetic sensor node side, the MQTT client proposed by *Contiki* 3.0 and developed by Texas Instrument has been applied.

It will be shown later, in Section IV.C, that the connection between the data gateway and the Cloud platform has been also implemented by using a broker MQTT at the server side.

C. *SAPOMS*

The *Semi-Autonomous Parking Occupancy Measurement System* (SAPOMS) was developed as a cost-effective embedded system that could monitor the occupancy of an outdoor parking lot. We believe that it is an outstanding complementary technology to fixed ground sensors in the streets. It can easily be installed or removed. This makes it versatile for temporary outdoor parking lots, commercial malls, entertainment parks, where it could be installed conveniently on a vertical stand (floor lamp, mast).

SAPOMS was developed as a smart camera system that can be configured easily through a secured SSH connection by a desktop configuration tool. In a few clicks, the manager can download an image to outline polygonal zones for contiguous parking places and define a few parameters: number and kind of places (standard, woman, disabled), system ID, latitude, longitude and then start the service. Figure 8 shows the configuration (in red) of a polygon with 3 parking places to be segmented and analyzed by *SAPOMS*.

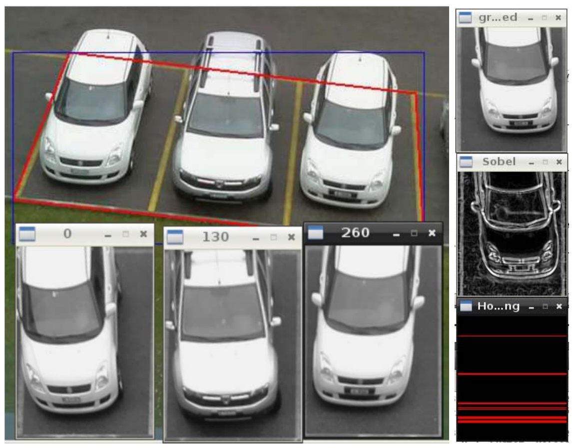

Figure 8. SAPOMS Configuration View

The embedded software of the smart camera takes a high-resolution (2592x1944) visible image every 10 seconds, with a *Raspberry PI/PiCamNOIR* system. It then analyzes each defined parking place with gray level image processing algorithms (filtering, edge detection, Hough detector) written in C++ with the *OpenCV* library [9]. No user interface is available as the system is running as a service (Figure 8 images are only available through the remote desktop configuration tool). *SAPOMS* is using about 3W and it would be possible to integrate an energy harvesting solution (solar cell) for the power supply if needed.

The system can then be used either through HTTP or to the IoT infrastructure described in next section through a gateway. A web request for the metrics of the sensor is available either in JSON or XML.

An important aspect of the smart camera system is its anonymization and privacy since it does not push images to the server. This is important regarding privacy laws that have to be considered when using a camera based system.

Available metrics are the total number of parking places (of each kind), the number of free places, the timestamp of the measurement, the camera ID, and the latitude and longitude of the smart camera placement.

D. IoT Platform

The proposed solution relies on a technological stack inspired from the Internet of Things paradigm [10]. The IoT platform *Stemys.io* (www.stemys.io) [11] built around a set of building blocks for each layer of the IoT technological stack is used to integrate the dynamic parking solution. Figure 9 shows the different components of the platform, from the sensors/actuators layer to the application layer.

The platform is conceived and built to flexibly and seamlessly integrate new communicating objects with minimal effort. Largely addressing the IoT fragmentation challenge, the platform is implemented around the following modules.

1) Communicating Devices

The sensors and actuators are integrated on a single embedded device able to forward the measured values and to receive the commands.

2) Gateway

A ubiquitous processing unit capable of communicating with the end devices and to push the measured values to the central platform. Further, the gateway is able to pull the commands to the end devices. A new concept of gateway has been introduced which is based on the so-called communication agents. These communication agents allow the seamless integration of communicating devices. On top of the Java VM, the framework *Open Services Gateway initiative* (OSGi) [12] has been installed. OSGI is about a modular –built up of bundles–and *Service Oriented Architecture* (SOA) for deploying and executing Java services on top of a resources-constrained embedded system. Further, the basic IoT services proposed by Kura [13] have been integrated into the implemented architecture on top of the OSGi framework (see Figure 9). Here are examples of the IoT services provided by Kura:

- Device Abstraction: Serial, USB, Bluetooth;
- Gateway Basic Services: Watchdog, Embedded DB, Clock Services;
- Network Management: Ethernet, WiFi;
- Connectivity and Delivery: Cloud Service, Data Service, MQTT (Paho);
- Field Protocols: ModBUS, CanBUS, ProfiBUS;
- Operation & Management: Remote configuration, remote update, log service.

Thanks to the modularity offered by the framework OSGi, three bundles have been developed which make up the pursued communication agent. Each of the three independent bundles of the OSGi-based agent handles one specific task: (i) Recognize and connect to the end device registered by the IoT application; (ii) Obtain/send the target data from/to the end device; and (iii) Publish/subscribe the target data through the MQTT broker to/from the central platform.

3) Broker

It is message broker capable of forwarding the data between the gateway and the central platform based on the «publish/subscribe» principle. The underlying asynchronous communication mechanism assure a high performance of the data transfer.

4) Central Platform

Group of services executed either on the Cloud or on a dedicated server and capable of processing and exploiting the acquired data. The domain specific applications are integrated at this level.

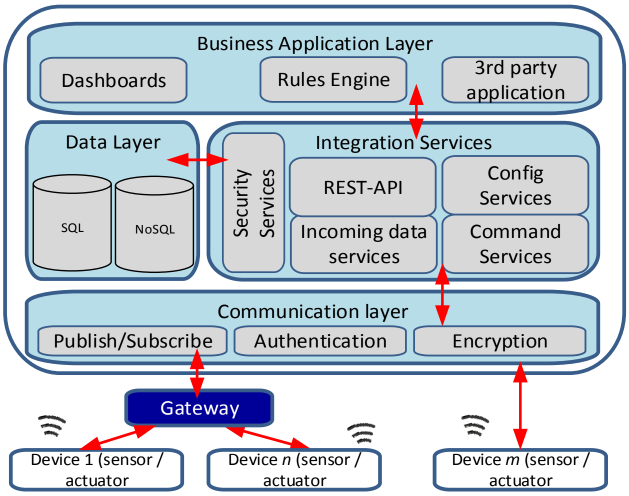

Figure 9. Stemys.io IoT Platform

V. RESULTS

Once the different components of the system have been implemented individually, a complete platform demonstrator has been integrated and tested in a real environment.

A. Magnetic Sensors Nodes Network Test

Figure 10 shows a general schema of the *SmartPark* demonstrator which integrates the network of low level magnetic sensor nodes. The information sensed by each individual magnetic sensor node is pushed to the central IoT platform through the data gateway.

A network of ten magnetic sensor nodes has been installed at our university parking lot for occupancy monitoring during a 6 weeks period. Although some nodes were momentary disconnected due to environmental hazards, the ASBR reacted rapidly by recreating the lost routes and thus automatically integrating the nodes back to the 6LoWPAN sensor network. Nevertheless, an increasing energy consumption has been measured which is mainly due to the network recovery related tasks.

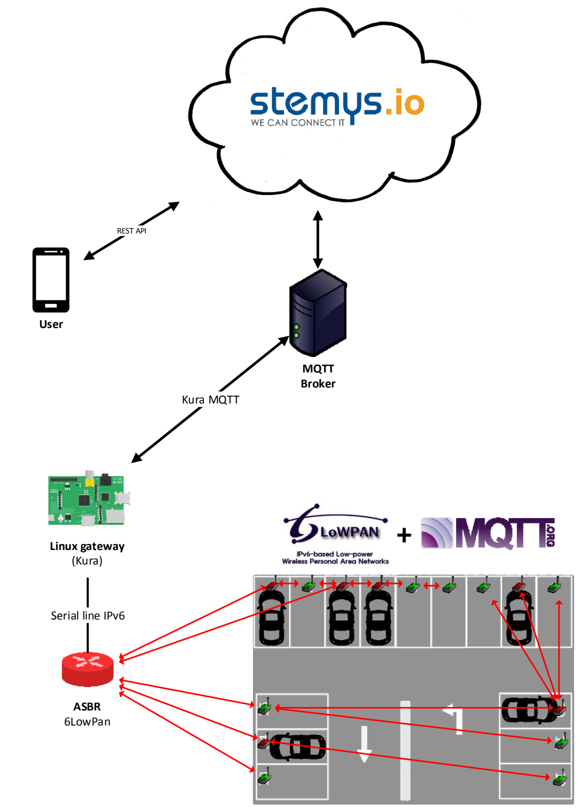

Figure 10. *SmartPark* Demonstrator

B. *Embedded Image System*

The prototype system was installed in winter 2015, during a 6 weeks period atop of our university parking lot, for occupancy monitoring. The solution was running over weeks without any issues, which proves its stability. Some problems with difficult lighting conditions have been identified during the tests, like direct sunshine and sharp shadows. The overall reliability has been measured to over 70% which is not bad considering that much time was spent on infrastructure and securing the data transmission.

Further, it has been shown that the high resolution of the camera could be of an advantage if the system has a large field of view, as it will be possible to monitor more cars with each system.

The most limiting factor for the *SAPOMS* system is its dependency on lighting conditions. With visible sensor, it will always be limited to daytime monitoring unless the parking gets sufficient artificial lighting by night. Infrared cameras being a way too expensive for such a system.

Some tools need to be developed if we want to support efficiently multiple cameras on the same parking with overlapping zones.

VI. CONCLUSIONS AND PERSPECTIVES

An on-street fixed magnetic sensor network and a smart camera system were developed and prototype systems built during the *SmartPark* project. Their integration to the IoT infrastructure was realized and their data made available for algorithm development server-side and mobile guiding applications.

Several tests in real environments were performed with the prototypes demonstrating their functionality compliance and relatively good performance. Nonetheless some work has still to be done to improve robustness to environmental perturbations such as shadows for the embedded image system or connection losses for the magnetic sensor nodes network.

The follow-up of the project will probably take the form of a technology transfer project in collaboration with a Swiss based company and the city of Geneva that is already planning to install fixed street sensors in pilot areas. The focus will be more on the development of the incremental search algorithm and temporal pattern extraction with data mining methods. The goal being to develop a system capable to predict and guide the driver towards the free parking place at arrival time.

ACKNOWLEDGMENT

Part of this work is funded by the University of Applied Sciences and Arts, Western Switzerland (HES-SO) via the research program Inuit (Internet of Things for Urban Innovation).

REFERENCES

[1] SFPark project (http://sfpark.org)

[2] M.Y.I. Idris, “Car Park System: A Review of Smart Parking System and its Techhnology”, Information Technology Journal 8, 2009

[3] http://blogs.cisco.com/government/cisco-and-streetline-innovate-for-smart-parking-introducing-camera-based-detection-and-an-integrated-streetline-iot-gateway-with-cisco-wifi/

[4] Kim S.H., Shi J., et.al, “Real-Time Traffic Video Analysis Using Intel Viewmont Coprocessor”, 2009

[5] http://livetraffic.tomtom.com/

[6] https://maps.here.com/traffic/

[7] OFROU: http://www.astra.admin.ch/index.html?lang=fr

[8] http://www.axis.com/fr/products/video/about_networkvideo/iv/system_design.htm

[9] http://opencv.org/

[10] Internet of Things Architecture: Final architectural reference model for the IoT v3.0, European project (FP7), 2013.

[11] N. Pazos, M. Müller, M. Aeberli and N. Ouerhani. „ConnectOpen – Automatic Integration of IoT Devices”, Proceeding of the 2nd IEEE WF-IOT 2015, Milan, December 2015

[12] OSGi Alliance. Available online at http://www.osgi.org

[13] Kura. Available online at https://eclipse.org/kura/