Smart Classroom

Jean-Pierre Gerval, Yann Le Ru

Smart Classroom

Jean-Pierre Gerval, Yann Le Ru

Institut Supérieur de l'Electronique et du Numérique – Brest
20 rue Cuirassé Bretagne – CS 42807 - 29228 BREST cedex 2 – FRANCE
Tel: +33 (0)2 98 03 84 00, Fax: +33 (0)2 98 03 84 10
E-mail: jean-pierre.gerval@isen-bretagne.fr; yann.le-ru@isen-bretagne.fr

Abstract. This paper sets out the methods and the technologies used to design a captive portal to redirect users to the URL (Uniform Resource Locator) of a course taking place in a given room. The captive portal is designed on a Raspberry Pi 2 carrying an Apache HTTP (HyperText Transfer Protocol) server and using iptables for redirections. It has a web configuration interface, developed with AngularJS, which communicates through HTTP request to the server side, developed in PHP, following the principle of a REST (REpresentational State Transfer) architecture. In addition to redirect users to the URL of a course, the interface is configurable in two modes: 1) fixed URL that sets an URL to which the user is redirected automatically, 2) hosting a local website which is used to load a web site in zip format on the device and then redirect users to this web site even when the device is not connected to any Ethernet network.

Keywords: Smart Objects, Internet of Things, Ubiquitous Learning, WiFi network, Captive portal

1 INTRODUCTION

Connected objects (also called smart objects, or Internet of Things) appeared recently. They are connected to the Internet so they can communicate with other systems to obtain or to provide information. This is made possible by the super miniaturization of electronic components.

They can for example:

- Collect and store information according to their environment: heart rate, cellar hygrometry, etc.
- Trigger actions based on the information gathered on the web, such as watering a lawn on the eve of a severe drought day.

To transform an everyday object into a connected object, simply connect it to the Internet and enable the object to react according to available data: weather, stock quotes, and user's action onto a smartphone...

The real "intelligence" of the object lies in the interface, and not in the object itself [1]. For example, a basic lamp, if it is connected to the internet via an interface (i.e. at the power socket level) it can be controlled from a smartphone, and thus becomes a

"connected lamp". To make yourself a connected object, it is now possible to buy at very reasonable price "kits", or using nano-computers such as Raspberry Pi, coupled to relays to operate devices connected to the electrical network.

The main idea of the work presented in this paper concerns the implementation of a system which will automatically download, in a given room, all the data linked to the course taking place in this room.

Main targets of such a work are as follows:

- To save time at the beginning of practical activities, courses...
- To facilitate access to resources, to simplify access to resources: more and more services are available and it becomes increasingly slow to arrive directly on the right page.
- To force access to resources (quiz, homework, etc...) through a specific network that does not give access to the Internet (i.e. to prohibit networks access from 3G or 4G smartphone).
- To provide useful information in the physical location where and when it is needed.

2 System overview

2.1 On the Hardware side

An electronic device with wireless technology enables sending and receiving data by an automated manner. It must be efficient enough to run a Linux distribution and to support an Apache server type. It must be scalable, with at least one Ethernet port and two USB ports (the first for the Bluetooth key and the second for the wireless key).

The device we have chosen is a Raspberry Pi 2 [2]. There are many other devices that are similar in their characteristics either at interfaces, dimensions or performance, often for a price of around € 50.

A criterion which is justifying the choice of such a device is its popularity, which implies a great number of resources that facilitates developments. Raspberry Pi 2 device seems to be the most suitable for the application according to its characteristics (performance, peripherals and consumption) and its popularity (large documentation and community).

In view of the aforementioned criteria, it has four USB ports allowing them to easily connect a Bluetooth dongle, a wireless dongle, a keyboard and a mouse to program easily. There is also the possibility of using a RJ45 port / Ethernet network connection and Bluetooth / Wi-Fi key to share or transmit the data contents.

2.2 On the software side

It is a Client/Server application that will automatically download in a given room all course materials available on Moodle [3], corresponding to the course that takes place there according to the timetable.

Front-End technologies.

On the client side there is the possibility of using different technologies JavaScript, jQuery, frameworks, or generated PHP. To meet standards and new server architectures, the choice is to let PHP on server side and create a REST (REpresentational State Transfer) architecture explained below. It remains the choice between JavaScript and frameworks. Frameworks offer ease of maintenance and quite handy code. New frameworks are quite powerful and structured [4]. Our choice is AngularJS [5] for its ease of learning, maintainability and its large community. The community around a framework is extremely important for the development: slightest problems can be solved according to problems that have already been solved by the community. There are opportunities to interact easily with other people. If the community is active and large then it gets faster to learn. AngularJS well respects the MVC concept (Model View Controller) that separates software components to enable programmers to structure their code and to facilitate the maintenance of software.

Back-End technologies.

JAVA and PHP are two different programming languages that would fit us for this application. PHP seems to be suitable. This scripting language is most often used server side. It is linked with an Apache server which is the software we have selected server side. This couple enables easy data retrieval from databases. We do not have chosen technology like Node.js on server side because speed and flexibility with respect to the number of users is not an important criterion for this application. The system architecture follows the diagram below (Figure 1). The client simply communicates via HTTP (HyperText Transfer Protocol) requests which are redirected to the web server that provides information depending on the type of method which is used. The server communicates with the local network and sends requests (Figure 2) to the Aurion [6] database (School Enterprise Resource Planning) and the Moodle database (Learning Management System).

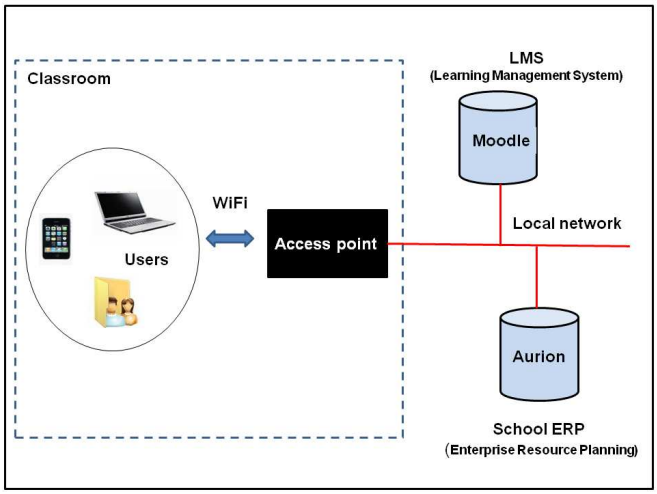

Fig. 1. System architecture

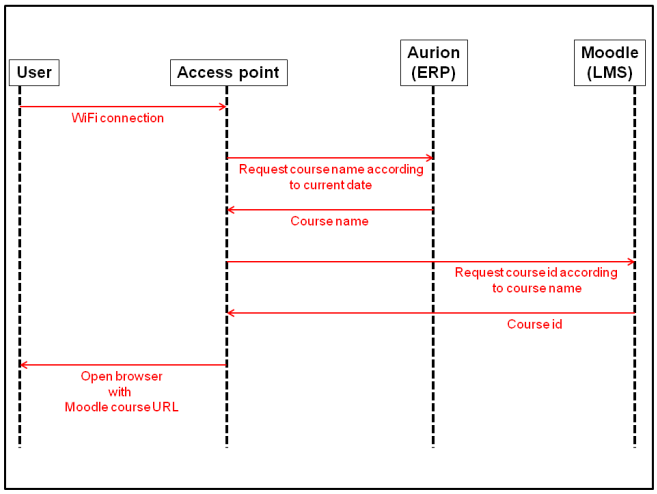

Fig. 2. Sequence diagram

The kernel.

The system installed on the device (Figure 3) is a recent version of Raspbian which is an adapted version of Linux Debian to the ARM (Advanced Risc Machine) architecture of Raspberry.

The web application consists of two parts. The first one is the main page showing the smart content for the room. This is the visible part for users. The second one is the administration of the device allowing administrators to choose the mode, to upload local site and to define the Service Set Identifier (SSID) of the wireless network. The administration access appears in the bottom right of the page with a small icon (Figure 4).

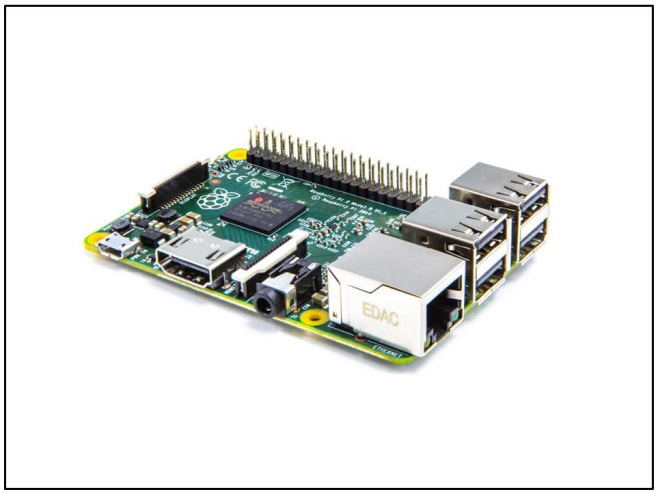

Fig. 3. Raspberry Pi 2

The device communicates with the client by means of a WiFi wireless Internet connection and with our databases on the local network by means of a RJ45 Ethernet link. A DHCP (Dynamic Host Configuration Protocol) server configures clients with IP (Internet Protocol) addresses and the device as a gateway. The administration application creates iptables rules redirecting HTTP flow to the smart content delivered by the server. Nevertheless according to the mode selected by the administrator for the device, contents available are not the same. We will detail these modes in the following chapter.

Two iptables rules allow us to filter and redirect the client only to the local web server's home page. This one is just a framework including an "iframe" where the content depends on the configuration.

iptable rule #1

```
-A PREROUTING ! -d $IP_origine/32 -p tcp --dport 80
-j DNAT --to-destination 192.168.8.1:80
```

iptable rule #2

```
-A PREROUTING ! -d $IP_origine/32 -p tcp --dport 443
-j DNAT --to-destination 192.168.8.1:443
```

192.168.8.1 is the address of the interface pointing to the server to which the user is redirected when attempting to access content other than the address defined by the IP address located after "! -d". This address can change dynamically according to the server configuration: IP address or hostname provided by the client URL (Uniform Resource Locator).

We use the "hostapd" software to create a WiFi access point available to the client. When the system gets a connection, a DHCP server "isc-dhcp-server" assigns an IP address to the client. Then as described above iptables rules redirect the client to the interface and the server. The authorized content which is included in the iframe is requested through the Ethernet interface.

3 Experimentation

Two test rooms have been equipped. Users who go into a room where an access point is available can connect to it. Following this connection, a page is automatically opened in the browser of the client showing the course supposed to take place in the room (Figure 4). The system retrieves and sends pages to clients, taking into account schedules and rooms.

In order to increase the flexibility of this system, two additional operating modes have been added:

- Fixed URL mode - The system redirects clients to a website, previously configured.

- Hosting a website locally - The system redirects users to a website hosted locally on the cart.

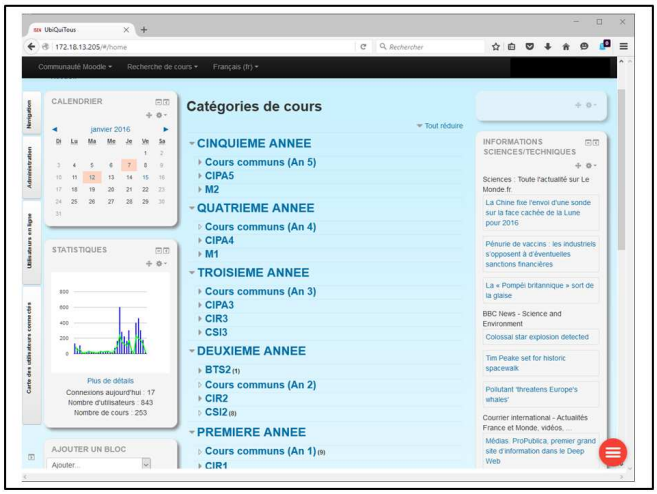

Fig. 4. Captive portal with Moodle inside

A specific interface (Figure 5) enables to configure the device for each operating mode:

- To define the room number and then to set up automatically the WiFi SSID of the device.
- To choose an URL to redirect the client to.
- To upload the web site that will be stored locally on the device.

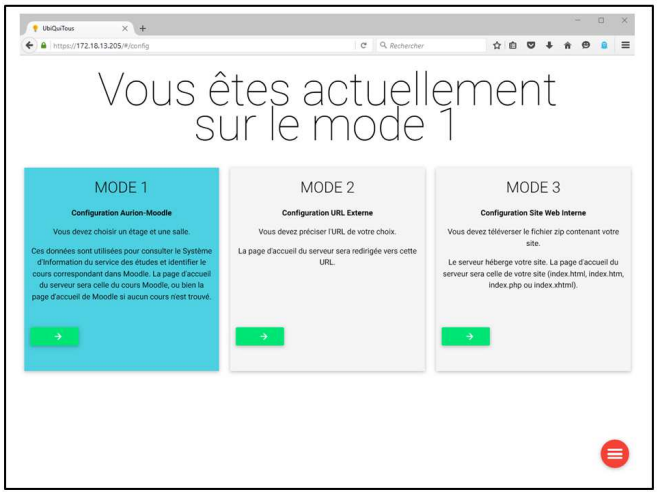

Fig. 5. Configuration interface

4 Conclusion

By the end of the experiments we noted that in order to avoid interference we had to select channels that did not overlap between various rooms and we also had to pay attention to limit the power of devices.

Finally this system could obviously be improved. For instance, a Bluetooth connection or another type of connection could be added to the device. In an era of increasingly digital and interactive, this project should certainly fit with other contexts. Such as museums or exhibitions, where various contents should be delivered taking into account the location of the end user.

References

1. CESER Auvergne (Janvier 2015), Les Usages du Numérique pour la Santé, l'Enseignement Supérieur et la Nouvelle Production Industrielle. http://www.cesdefrance.fr/pdf/13713.pdf (Retrieved: 2015/02/18)
2. Raspberry Pi Foundation. https://www.raspberrypi.org (Retrieved: 2015/04/06)
3. https://moodle.com/ (Retrieved: 2016/03/02)
4. Uri Shake, AngularJS vs. Backbone.js vs. Ember.js. https://www.airpair.com/js/javascript-framework-comparison (Retrieved: 2015/04/27)
5. Google inc. Framework AngularJS. http://www.angularjs.org (Retrieved: 2015/04/27)
6. https://www.auriga.fr/solutions-erp/erp-etablissement-enseignement-superieur.html (Retrieved: 2016/03/02)