Open Sensor Networks

Alejandro Andreu Isábal

Sensor boards

Boards comparison

CROSSBOW TELOS B	ARDUINO
Open source	Open source
Unix-based OS	None
High complexity	Simple
Little community	Large and active community
Zigbee supported	Zigbee supported

Data gathering

- Learning Arduino
  - Arquitecture
  - Language (similar to C++)
- Test the sensors
  - GPS
  - Temperature
  - Humidity
  - Light intensity
  - Sound detector

ToDo list

- Measure air quality
  - One sensor vs. several
- Point-to-point communication
  - XBee module

Key aspects and Main factors in NQN

Fernando Gros González

04-10-2012

Table of contents

Introduction

- The NQ area is home to a wide range of SMEs.

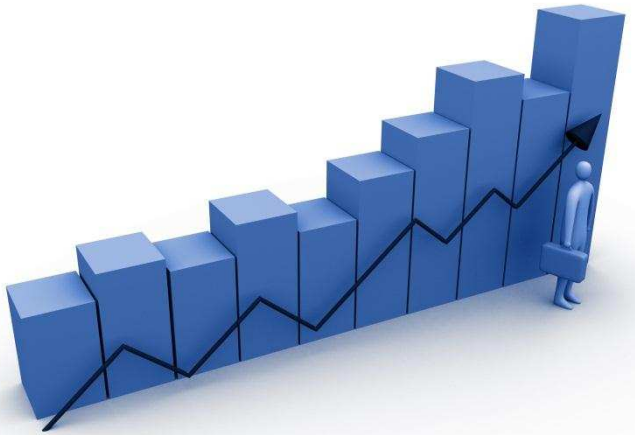
- Low rental and lease costs of buildings
- Physical trading presence in city centre

- GOOD FOR STARTING BUSINESSES!!

Public WiFi Impact in the Area

Customers increase

Revenues Increase



Pricing Definition

- Crucial aspect
- Self-funding and sustainable network after the conclusion of the C4EU project.

Regulatory Issues

- Planning permission
- Radio licensing
- Data protection
- Constitution of the Cooperative
- Health and Safety Public Liability Insurance

Regulatory Issues

- The aspects described have been identified just by reading the pilot proposals document of the C4EU project. It is likely that I have to refocus the point of view I have given to the pilot once I have my first meeting with the MDDA colleagues.

Thank you for your time!

Practicum, Mentor, Thesis, Advisor

Jaume Barcelo

Universitat Pompeu Fabra

2ond C4EU BuB workshop. October 4th, 2012, Barcelona

Training of students in C4EU

Two differentiated aspects:

- Practicum: real-world work. Guided by a mentor.
- Thesis: academic work. Guided by an academic advisor.

Each of them is worth 20 ECTS credits (i.e., 400 hours of work)

We have already assigned two mentors to two of the students.

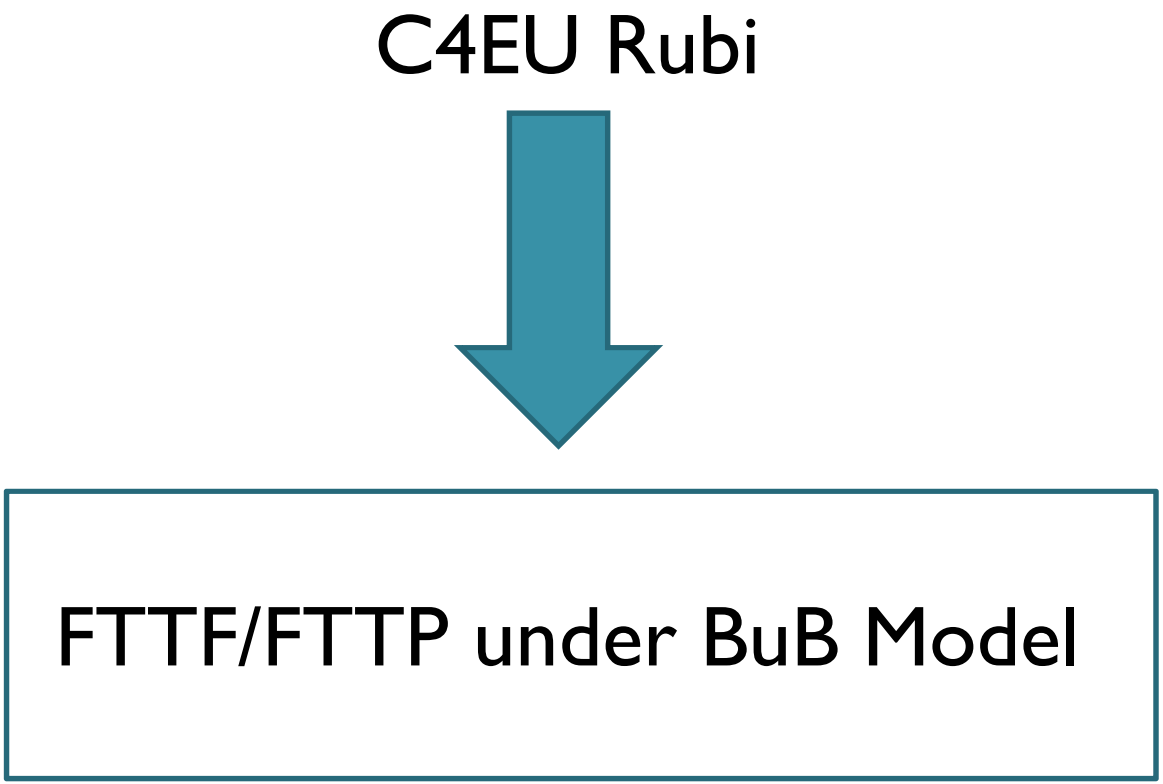
- Giovanni Calcerano mentors Nacho Justel in the FreeEuropeWifi pilot.
- Roger Baig mentors Jorge Beltran in the FFTP pilot.

BuB – Commons for Europe

FTTF/FTTP

Jorge A. Beltrán Calderón
joranbel@gmail.com

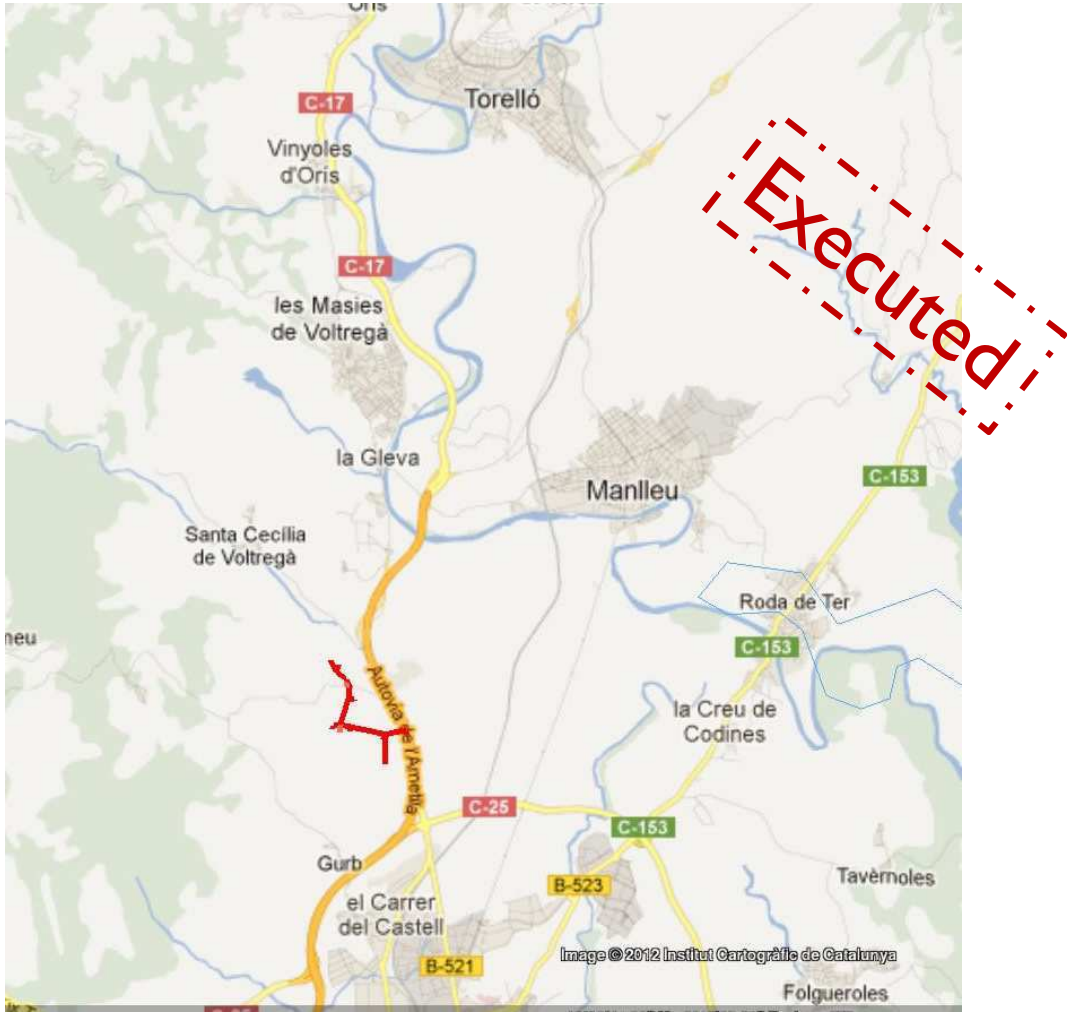
Introduction



Some BuB Projects

Project	Status
Gurb – Osona (Spain) FFTF-FFTH	Completed & Operative 1st Iteration Next iterations to complete the municipality and extend to next municipalities (Vic, Sant Bartomeu...)
Centelles (Spain) FFTH with partnership with Local Electrical Company (Electra)	Going to execution
Rubí (Spain) FFTH to Industrial areas & labour-class neighbourhoods	Preparation
City of Boston (USA) Municipal Fibre backbone for community wi-fi areas	Completed
Barcelona (Spain) Sensor Integration	TBD

2009: North Gurb → Phase I



2012: North Gurb → Phase 11

2012: North Gurb → Phase 11

Next Steps

- Learn about Gurb Project
  - Where is it ?
    - Terrain, Population, economy ?
  - Users
  - Equipment
  - Deployment
    - Configuration
  - Legal and Compliance

Thank You!

www.guifi.net

www.workspaces.guifi.net/share

Questions?

2nd NeTS mini-Workshop

Wireless Transmission with USRP-E110

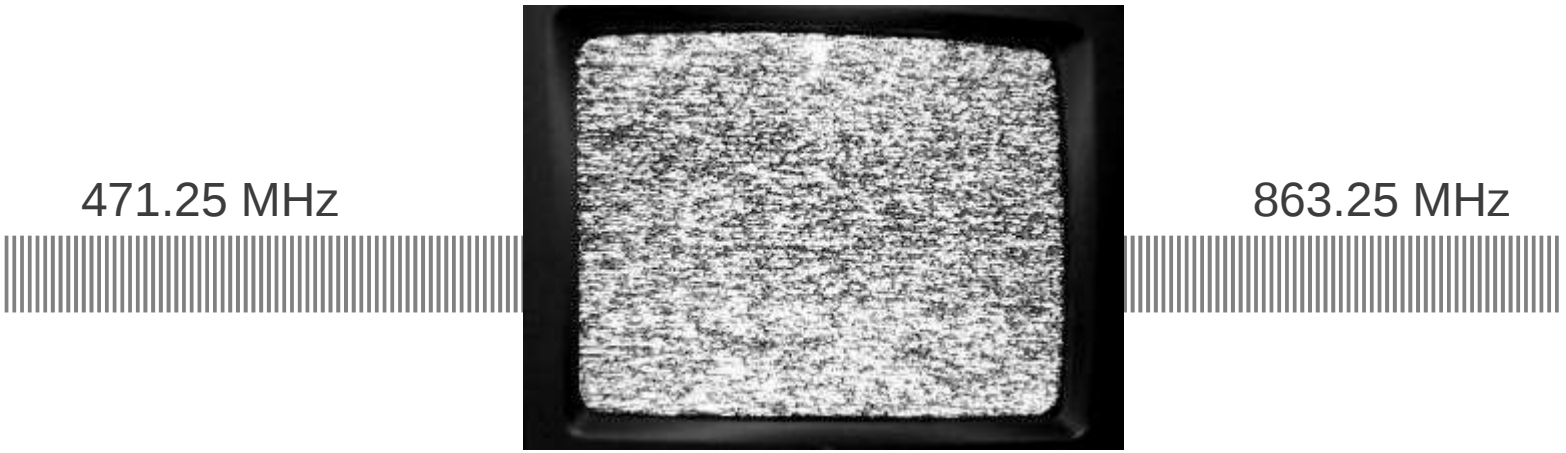
Luis Sanabria-Russo
luis.sanabria@upf.edu

Outline

- Introduction?
- Our current approach
- Results
- Challenges ahead

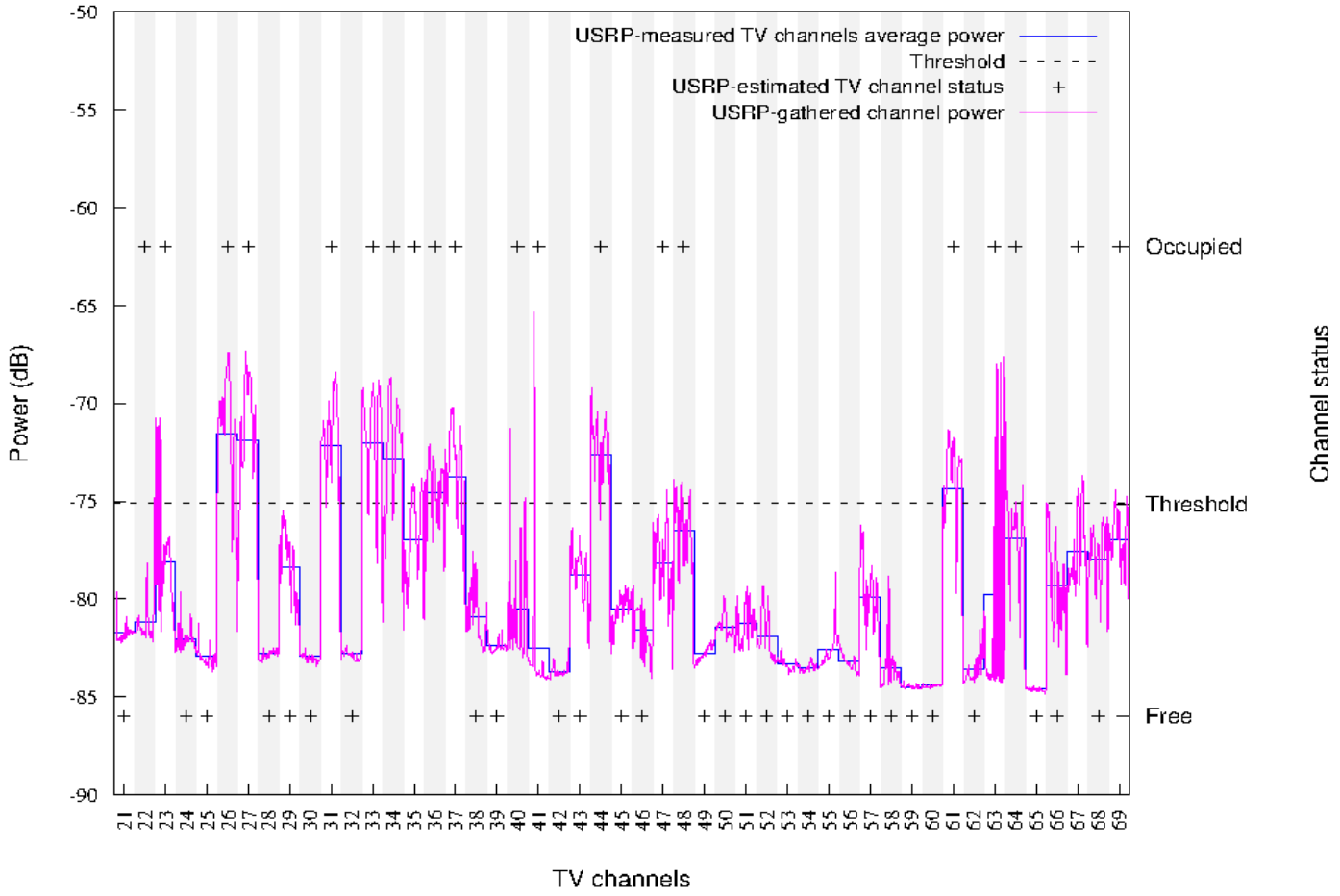
In the identification of TV White Spaces

- TVWS are wholes in the TV band
  - Recently been considered for data transmission
  - Better building penetration, longer distances
  - SuperWiFi/WhiteFi, IEEE 802.22



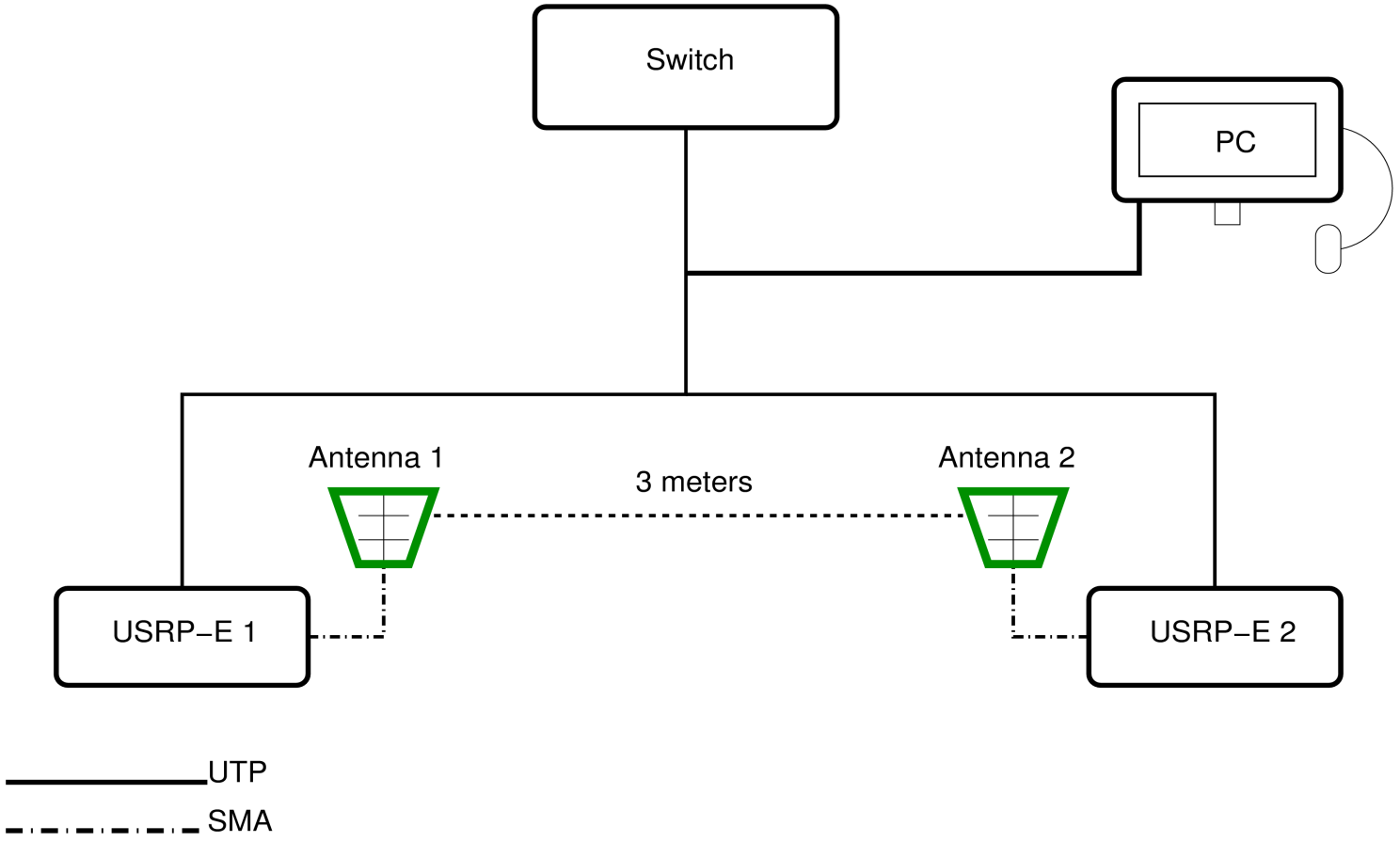
The spectrum of the band

- Help us identify the presence of primary signal



Data transmission:

- GNURadio provides examples of PHY implementation
- Data transmission with **benchmark_tx.py** and **benchmark_rx.py**



Successful transmission and reception:

- Construction of packets from binary file
- Correct reception and reconstruction

After customizing the example codes...

Combine: sensing + communication

- Build a sensing + communication GNURadio example
  - Detection, Selection and Exchange of TV WS
- Implement retransmission and carrier sense mechanisms
- Range test: outdoors

Note: check out our code at https://github.com/SanabriaRusso/USRP_transmissions

Free Europe WiFi

OpenWISP Modules

C4EU - BuB

Ignacio A. Justel Pizarro

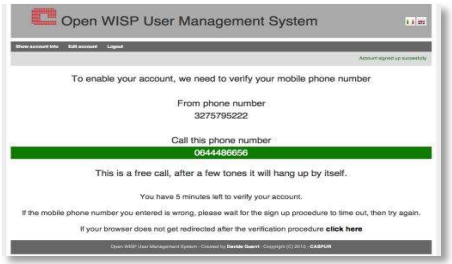
What is OpenWISP?

- OpenWISP is a software platform that can be used to implement a complete Wi-Fi service.
- OpenWISP project is actually divided in five important modules, but there are also another items strictly related with this project.

1- OpenWISP User Management System (I)

- It is a Ruby on Rails application, that interacts directly with the users. By using this module, the user can get access to the OpenWISP global system to get internet access, password recovery if needed and manage his account.

- It is possible to integrate it with MySQL databases and FreeRADIUS 2.1 server, in order to sign-up/sign-out and save user information.



OpenWISP User Management System (II)

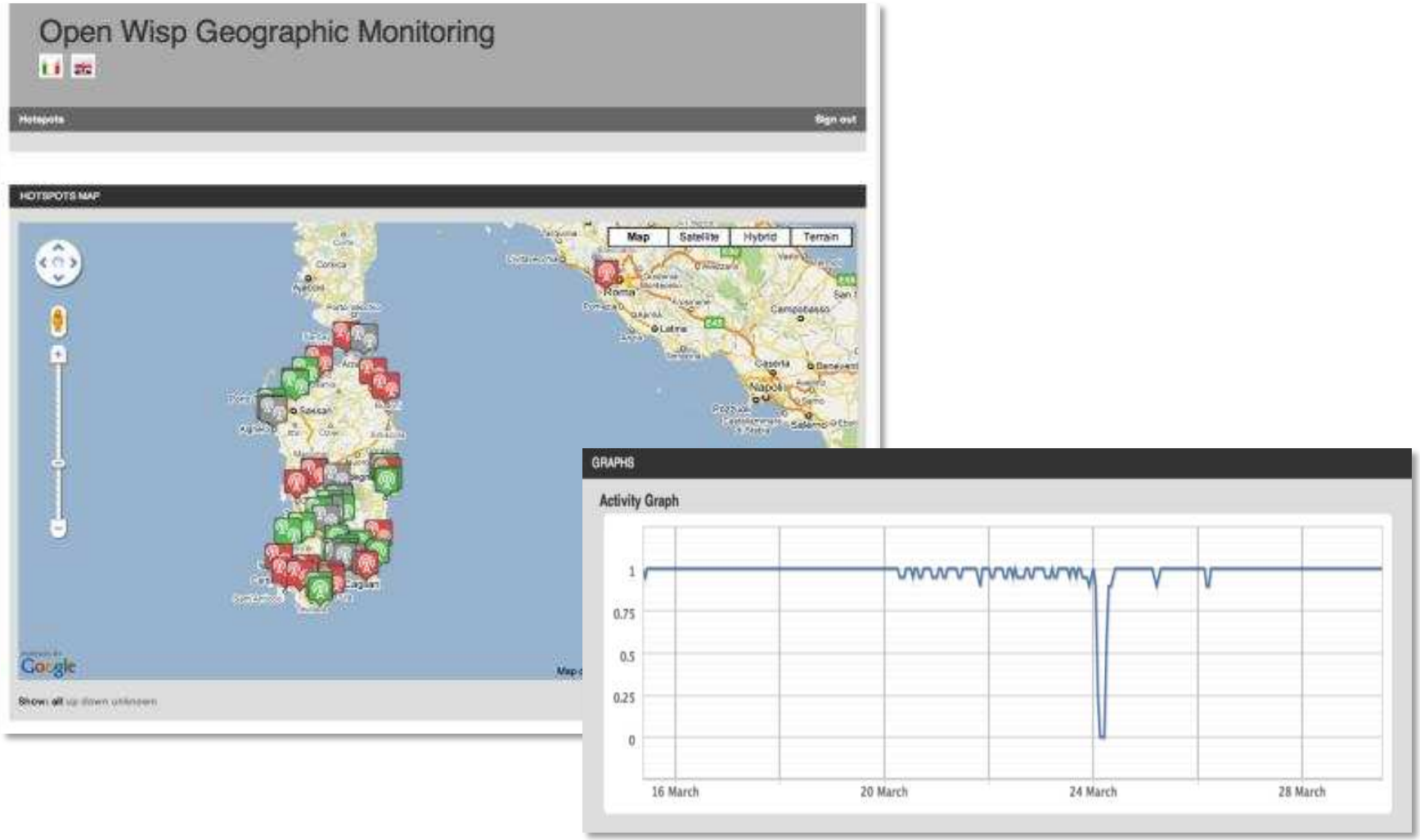
- Some of its features are:
  - User registration via mobile phone, ID card or credit card (Paypal IPN).
  - User interface supporting most common browsers including mobile web browsers.
  - Password recovery via mobile phone or email.
  - Statistics of generated traffic per user.
  - User administration via dedicated admin interface.
  - Admin interface supports role based administration (operator, admin, superadmin).
  - Via admin interface, new users can be added, modified, enabled/disabled or monitoring traffic per user/nationality and logins/registration.

2 - OpenWISP Geographic Monitoring (I)

- It is a Ruby on Rails - HTML 5.0 based, web GUI that allows the management staff to get information about the geographic information of the deployed access points.

- It renders a geographic map of the status your networks: access point up/down/unknown (if an access point has an "unknown" status, then it would not be able to download the configuration to connect to the system from the OpenWISP Access Point Manager.

OpenWISP Geographic Monitoring (II)



3 - OpenWISP Captive Portal Manager

- It is a captive portal written from Scratch with Ruby on Rails. It allows the user surfing the Internet by enabling rules in the server firewall. This module is the operation center of all the system, so all the data that is generated or has as destination the system, will pass through this module.

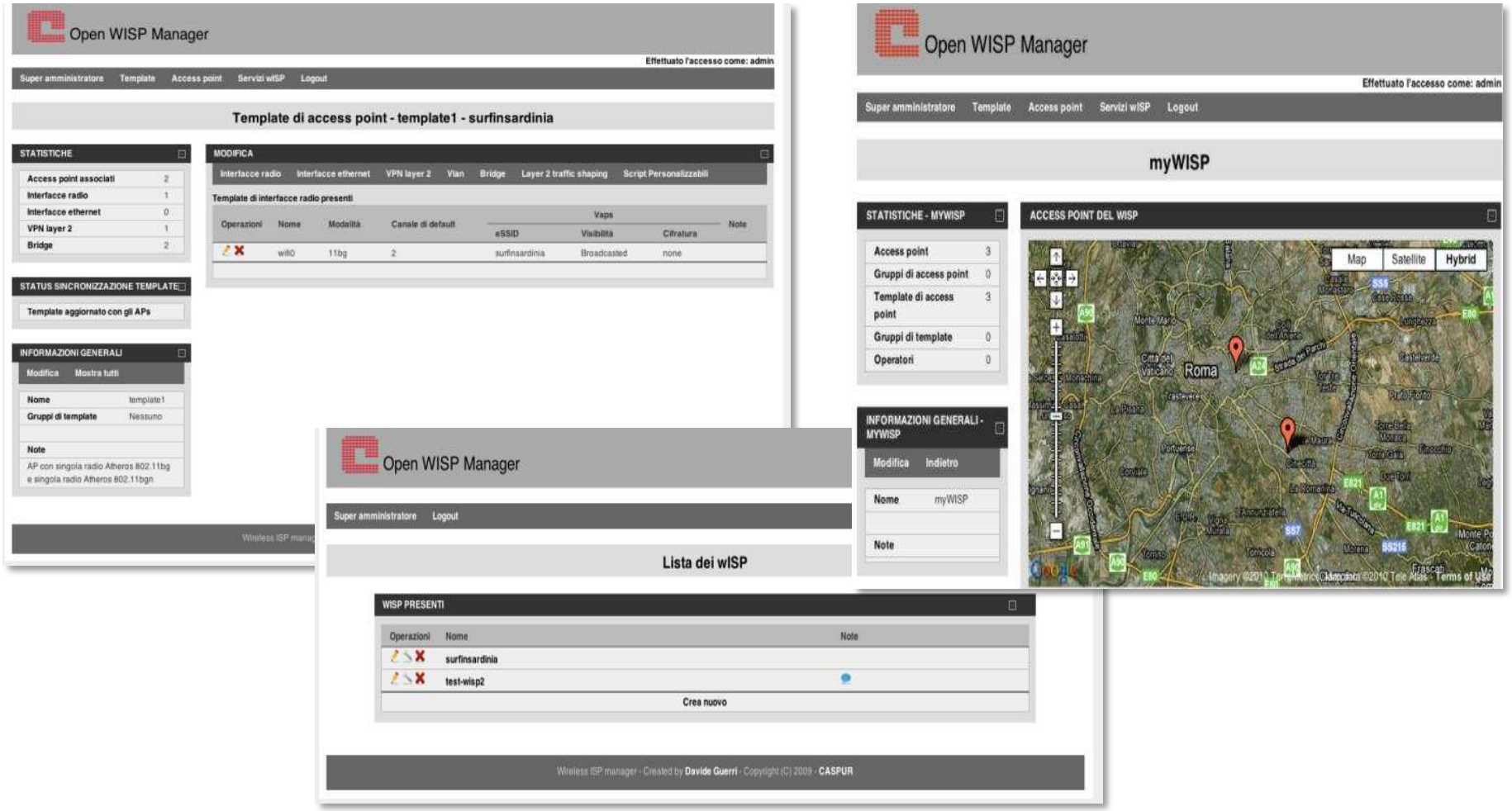
- Its main features are:
  - Multiple captive portals (one per physical or virtual interface).
  - RADIUS / Local authentication.
  - Experimental traffic shaping per user.
  - Multiple OS support.
  - IPv6 support can be easily implemented.

4 - OpenWISP Access Point Manager (I)

- It is a Ruby on Rails based web GUI. This module allows the management staff to configure, monitorization and support of deployed access points.

- It also stores value information about the network, as the amount of traffic each VPN generate, MAC addresses, geographical addresses and network setting, among other data. The access points download the configuration and settings from this module to establish a connection to the gateway.

OpenWISP Access Point Manager (II)



5 - OpenWISP Firmware (I)

- It is a set of scripts (shell and web cgi) that sits on top of OpenWrt*. OWF is a **no visible** module of the system. It is the firmware that every access point has installed to be able to connect to the network.
- Every time a new access point is connected to the network, it will download the settings from the Access Point Manager, as before explained. If an access point is rebooted without network connectivity, will get no configuration until it could establishes a connection with the OpenWISP Manager, so then, the configuration will be sent to it.

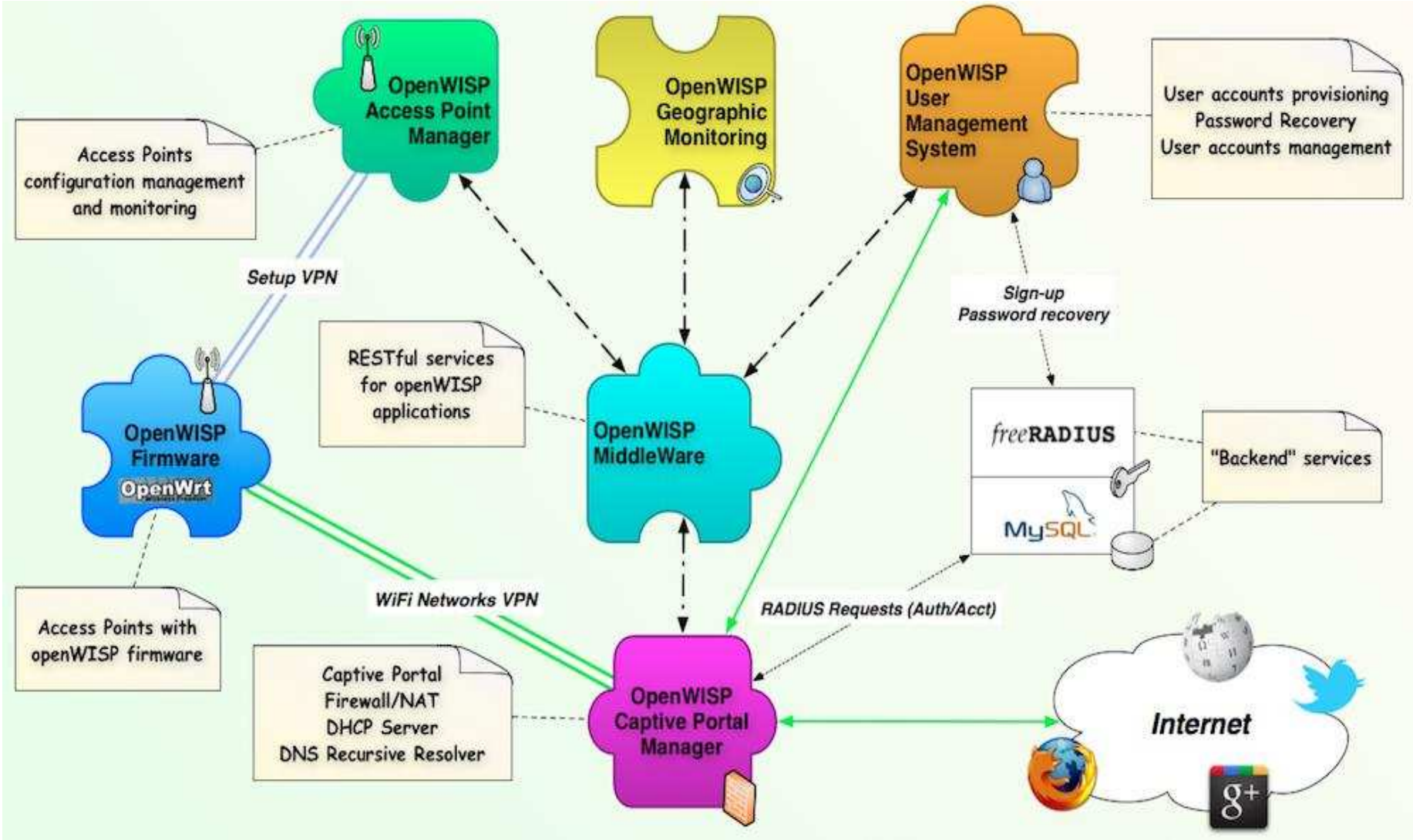
*OpenWrt is a linux distro for embedded devices that provides a fully writable file system with package management. This allows to customize devices keeping freedom from vendor configuration.

OpenWISP Firmware (II)

- OpenWISP firmware provides a daemon for retrieving the configuration of some components at the manager:
  - WiFi (currently only for madwifi-ng and ath9k drivers).
  - Networking.
  - Data layer traffic shaping.
  - OpenVPN (data layer, TAP).
  - Unix cron jobs
- Far from only providing a daemon, it also has a web GUI where:
  - Configuring basic network parameters.
  - Configuring basic OpenWISP server settings.
  - Performing test and technical fails menu.

6 - OpenWISP Middleware

- It is an application based on Ruby/Sinatra that communicates or interacts with the other modules using RESTful services. This module has different tasks to do depending on the other module that is talking with:
  - By communicating with the OpenWISP Captive Portal Manager, it personalizes the captive page for a group of different access points.
  - On the other hand, by interacting with the OpenWISP User Management System, OpenWISP Geographic Monitoring and OpenWISP Access Point Manager, it provides very useful information in a full duplex way.


OpenWISP Access Point Manager
OpenWISP Geographic Monitoring
OpenWISP User Management System
Access Points configuration management and monitoring
User accounts provisioning
Password Recovery
User accounts management
Setup VPN
Sign-up
Password recovery
RESTful services for openWISP applications
OpenWISP Firmware
OpenWrt
OpenWISP MiddleWare
freeRADIUS
MySQL
"Backend" services
Access Points with openWISP firmware
WiFi Networks VPN
RADIUS Requests (Auth/Acct)
Captive Portal
Firewall/NAT
DHCP Server
DNS Recursive Resolver
OpenWISP Captive Portal Manager
Internet

Thanks!