Outline

- **Introduction**
- **Evaluation Approach**
  - Power Spectral Density Evaluation
  - Power Consumption Evaluation
  - Sensing Delay Evaluation
- **External Use Case Evaluation**
- **Benefits for the experimenter**

Introduction

- ✓ Accuracy of spectrum sensing and efficiency of free spectrum utilization are considered as the primary objectives in Dynamic Spectrum Access Networks.

- ✓ As the focus of researchers is usually on these two major challenges, other aspects have been in part underestimated.

- ✓ In this work, we consider two factors that are rather important for evaluation of cognitive platforms, namely:
  - Energy Consumption
  - Sensing Delay

- ✓ And develop an appropriate Benchmarking Framework to enable evaluation of Cognitive solutions in terms of these metrics.

Outline

- Introduction
- **Evaluation Approach**
  - Power Spectral Density Evaluation
  - Power Consumption Evaluation
  - Sensing Delay Evaluation
- **External Use Case Evaluation**
- **Benefits for the experimenter**

Evaluation Approach

- We investigate the sensing characteristics of different sensing devices under a common experimental scenario.

- The developed framework is used to compare the performance of each device in the common scenario in terms of:
  - Power Consumption
  - Sensing Delay

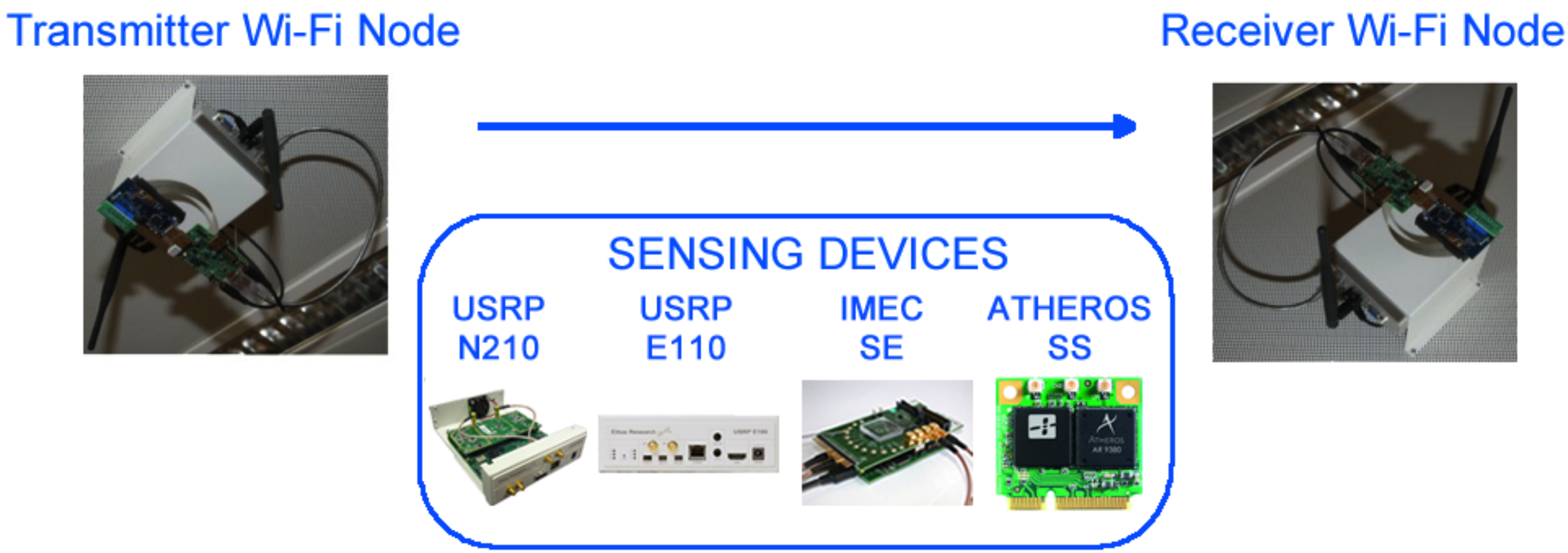
- **Storyline of the Evaluation Approach:**

1. Power Spectral Density (PSD) evaluation through FFT processing.

2. Power Consumption characterization of each device, by using high-end Power Metering devices.

3. Total Sensing Delay distribution between the processes of Sensing, Transferring and Processing of Measurements per device.

Common Experimental Scenario

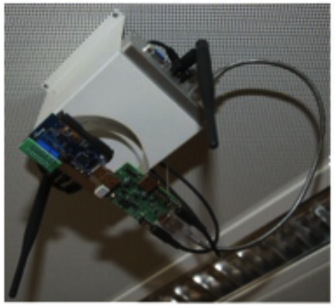


- ✓ The 4 different devices are used in parallel to characterize a signal of **20 MHz bandwidth** that is generated by a pair of **802.11** enabled nodes in the **2.4 GHz** band.
- ✓ All sensing devices are configured to sense the medium for **64 μs**, process the gathered samples and characterize the **PSD**.

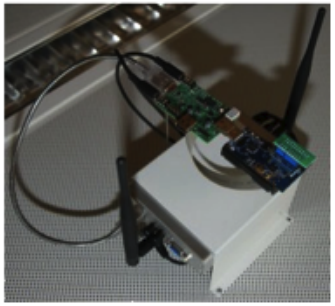
1. PSD Evaluation

Device Characteristics

Transmitter Wi-Fi Node



Receiver Wi-Fi Node



SENSING DEVICES

USRP N210

USRP E110

IMEC SE

ATHEROS SS

✓Varying bandwidth capabilities:

•USRP N210: 25 MHz - USRP E110: 5 MHz

•IMEC SE: 20 MHz - Atheros AR9380: 20 MHz

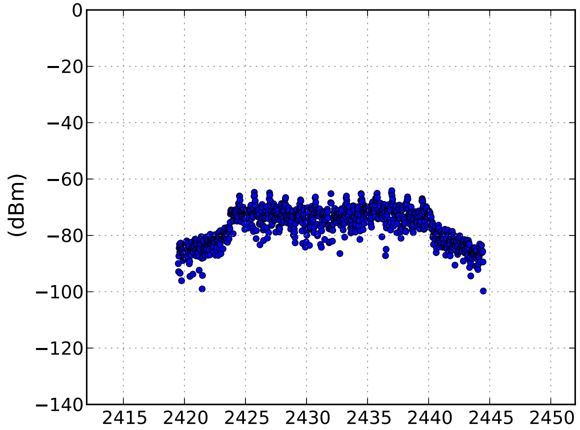
✓Varying FFT resolution:

•USRP N210: 1024 bins - USRP E110: 256 bins

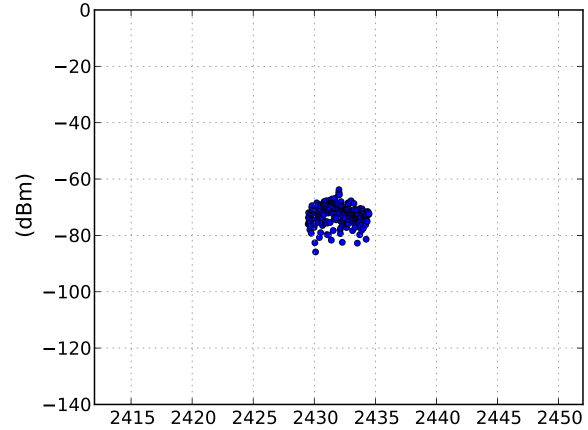
•IMEC SE: 128 bins - Atheros AR9380: 56 bins

1. PSD Evaluation

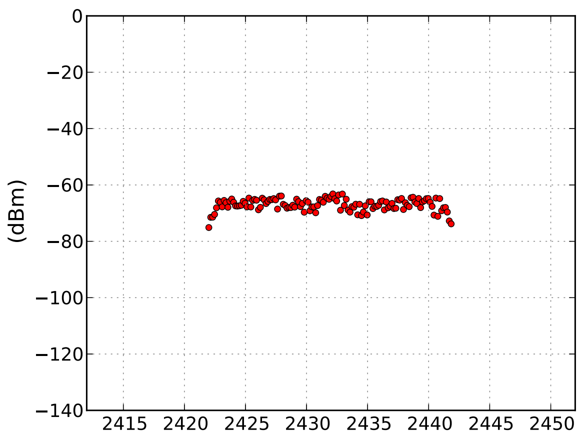
USRP N210



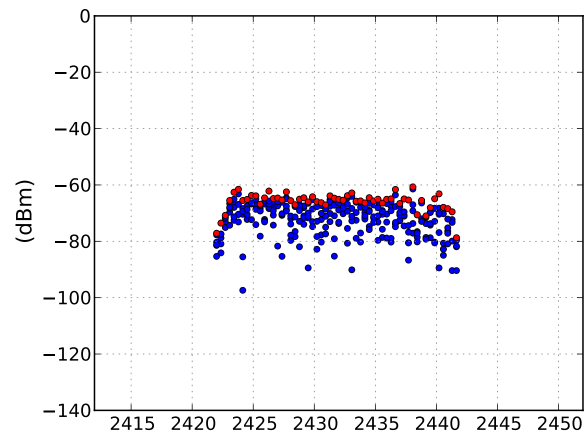
USRP E110



imec SE



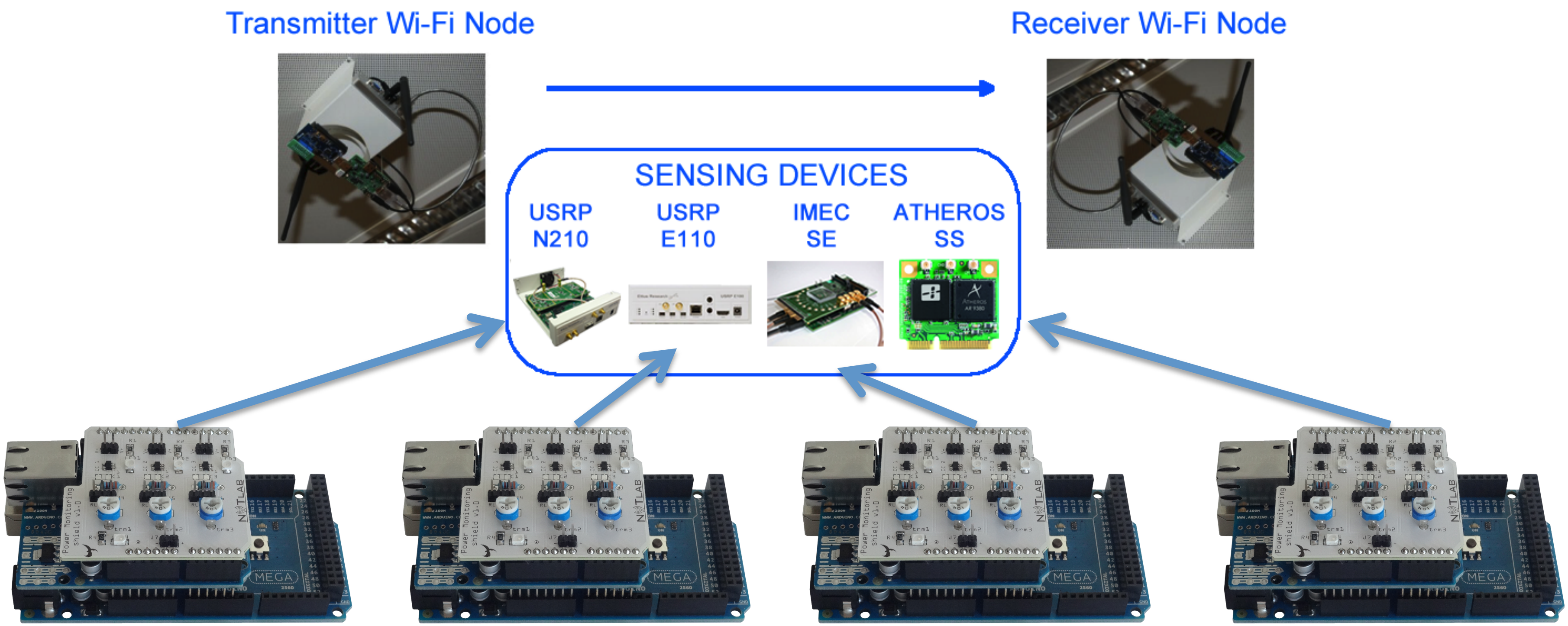
Atheros AR9380



Outline

- Introduction
- **Evaluation Approach**
  - Power Spectral Density Evaluation
  - Power Consumption Evaluation
  - Sensing Delay Evaluation
- **External Use Case Evaluation**
- **Benefits for the experimenter**

2. Power Consumption Evaluation

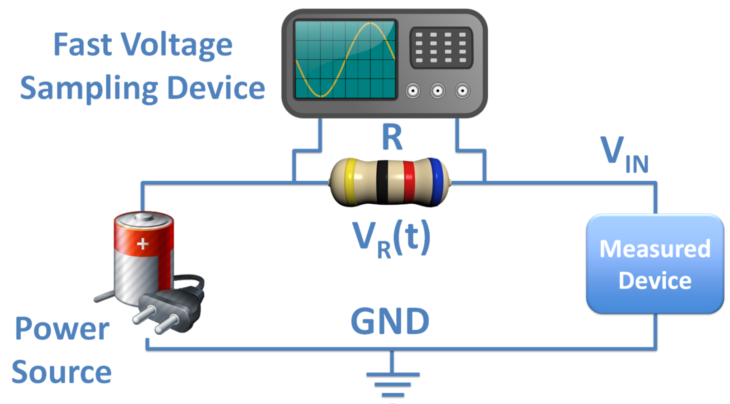


- ✓ The second step is to characterize the **Energy Efficiency** of the **4** different sensing devices

- ✓ In order to accomplish this, we use the developed **NITOS ACM** cards that act as **Network-enabled High-end Power Meters**.

2. Power Consumption Evaluation

- ✓ Power consumption can be determined by direct measurement of the input voltage and current draw at the device under test.
- ✓ Actual measurements can be gathered using a fast voltage sampling device, as follows:



- ✓ The instantaneous **power consumption** is the product of the input voltage and current draw on the current shunt resistor **R**:

$$P(t) = V_{IN} \frac{V_R(t)}{R}$$

2. Power Consumption Evaluation

Total Energy Consumption over an interval **$\Delta t = t1 - t0$** is calculated as the integral of power consumption:

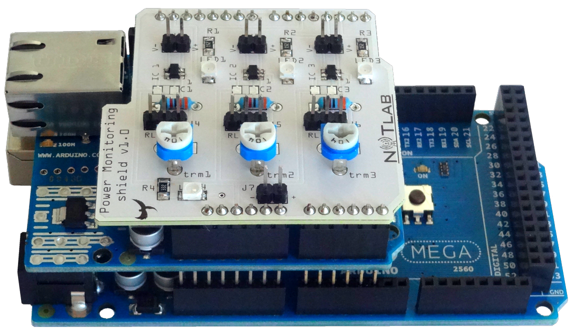
$$E_{t_0 \ldots t_1} = \frac{V_{in}}{R} \int_{t_0}^{t_1} v_r(t)dt$$

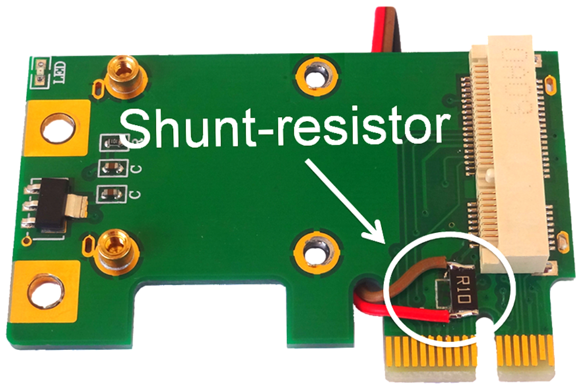
dt: corresponds to the infinitely small observation duration, which equals the inverse of the configured sampling rate

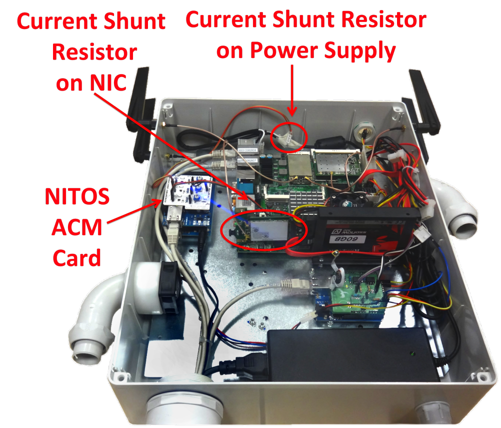
Δt: corresponds to the total duration of each specific **experiment**

In the case of **Spectrum Sensing experiments**,
Δt corresponds to the total duration of the sensing process and needs to be precisely calculated in each scenario.

2. Power Consumption Evaluation



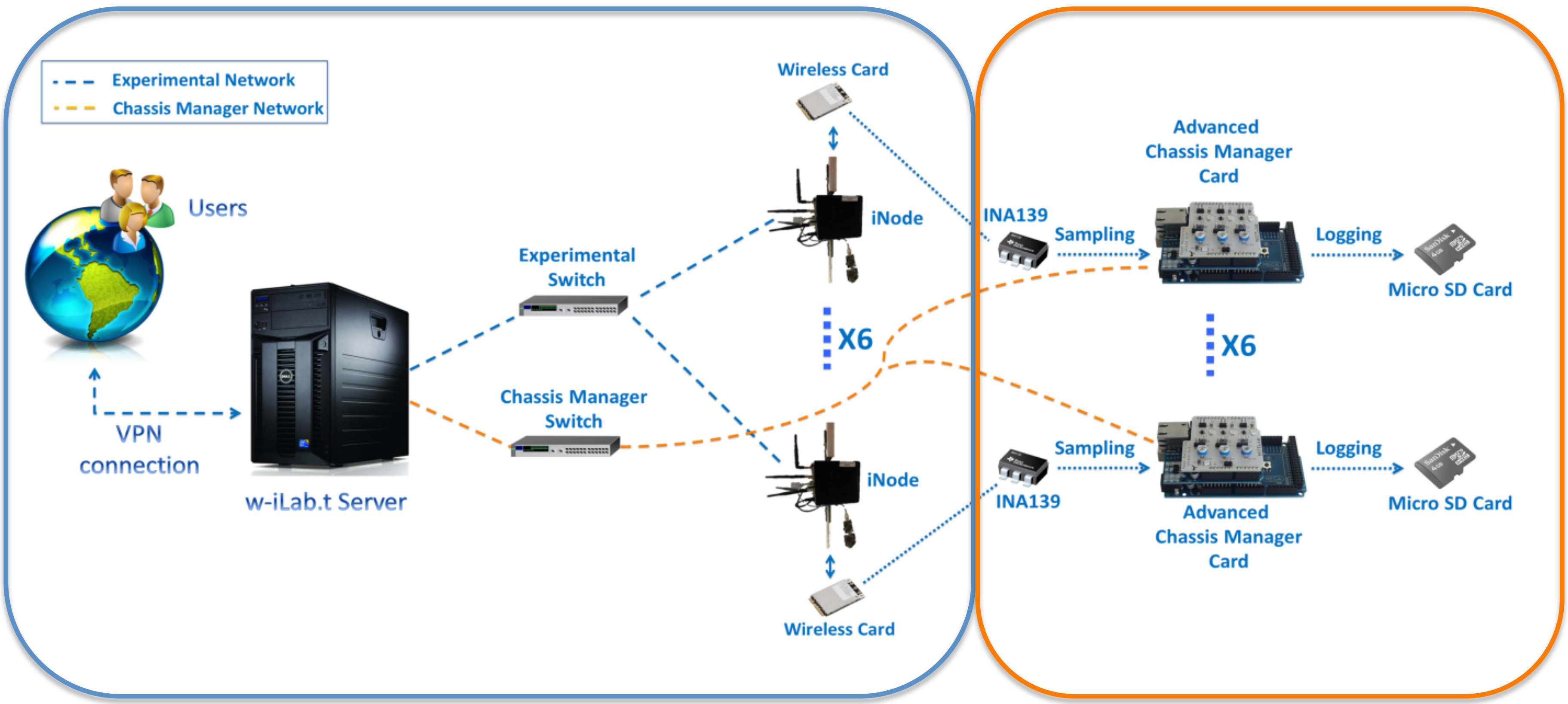




- ✓ **Online Monitoring** of realistic testbed experiments
- ✓ **Distributed** Architecture through Network communication
- ✓ **High Accuracy** (comparable with high-end devices)
- ✓ **High Sampling Rate (63 KHz)**
- ✓ Adaptable to heterogeneous devices (wireless nodes/ cards, spectrum sensing devices, mobile phones, etc.)
- ✓ **Low-cost (less than 80€)**

2. Power Consumption Evaluation

Integration with w-ilab.t Testbed architecture

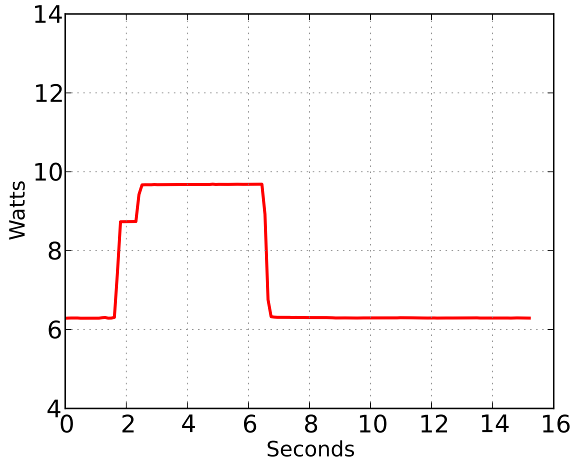


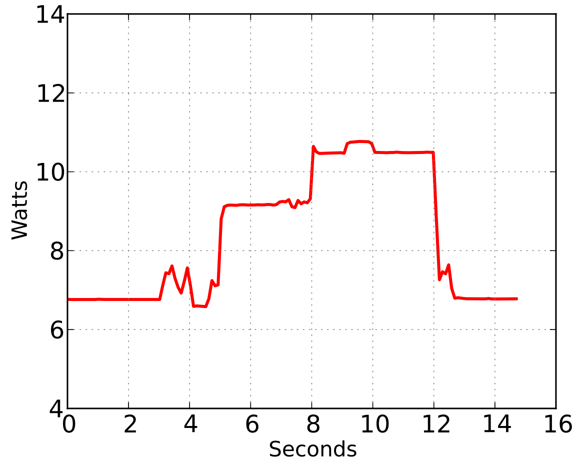
Existing w-ilab.t Architecture

Extension through our approach

2. Power Consumption Evaluation

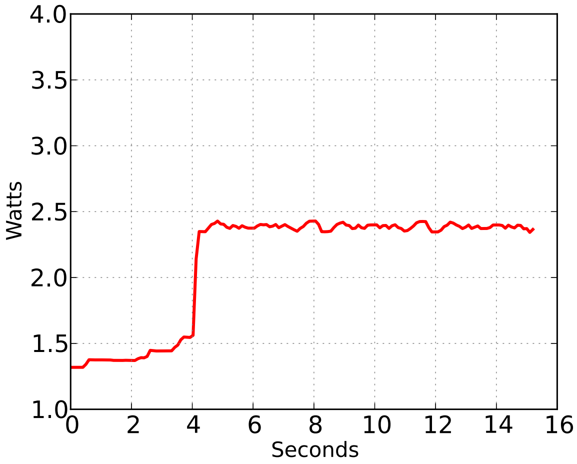
USRP N210

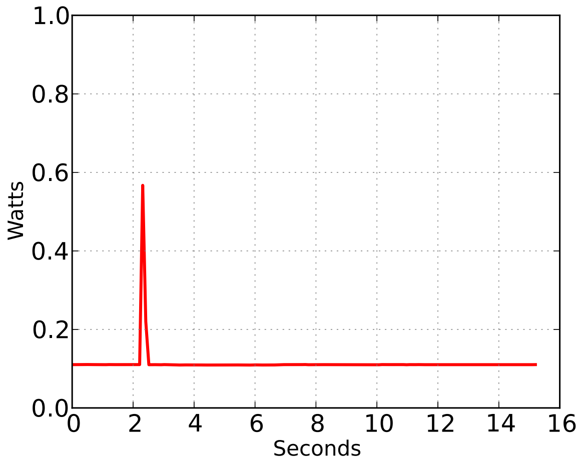




USRP E110

imec SE

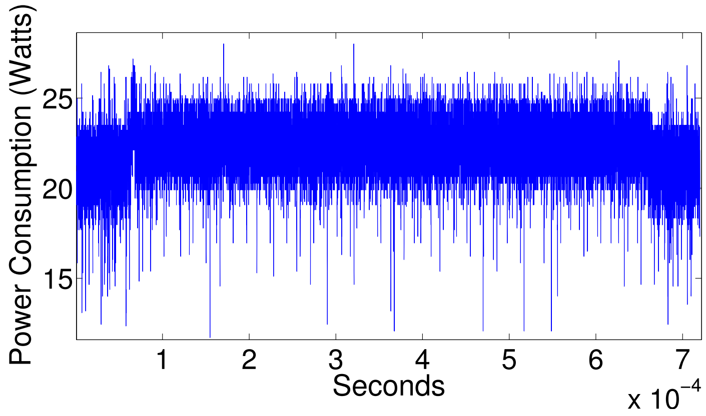




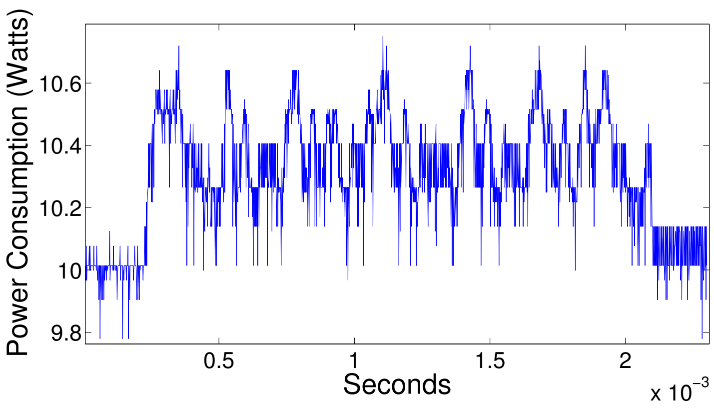
Atheros AR9380

2. Power Consumption Evaluation

Power Consumption of USRP devices
during processing of spectral measurements



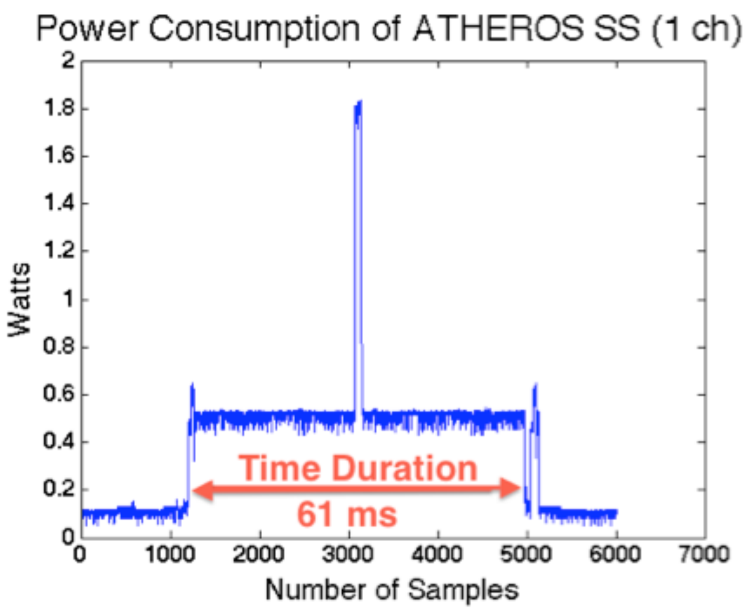
USRP N210 Processing
on ATOM-based setup
1.35 W increase - 701 μs
Energy = 946.35 μJ



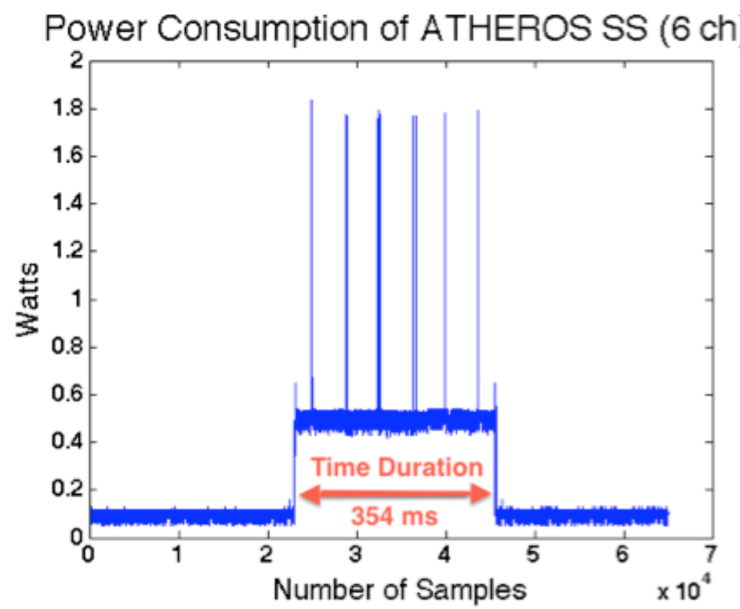
USRP E110 Processing on
embedded ARM processor
0.4 W increase - 1800 μs
Energy = 720 μJ

2. Power Consumption Evaluation

Power Consumption of Atheros AR9380 Spectral Scan
For 1 and 6 channels of 20 MHz width



Energy = 0.035 J



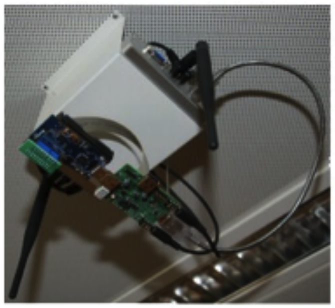
Energy = 0.201 J

Outline

- Introduction
- **Evaluation Approach**
  - Power Spectral Density Evaluation
  - Power Consumption Evaluation
  - Sensing Delay Evaluation
- **External Use Case Evaluation**
- **Benefits for the experimenter**

3. Sensing Delay Evaluation

Transmitter Wi-Fi Node



Receiver Wi-Fi Node

SENSING DEVICES

USRP N210

USRP E110

IMEC SE

ATHEROS SS

✓ In the third step, we aim at characterizing how the total Sensing Delay of each different sensing device is distributed between the processes of:

1. Sensing **2. Transferring** **3. Processing**

4. Channel Switching

3. Sensing Delay Evaluation

```cpp
#include <iostream>
#include "boost/date_time/posix_time/posix_time.hpp"

typedef boost::posix_time::ptime Time;
typedef boost::posix_time::time_duration TimeDuration;

int main (){

    Time t1(boost::posix_time::microsec_clock::local_time());

    Time t2(boost::posix_time::microsec_clock::local_time());

    TimeDuration dt = t2 - t1;

    //print formatted date
    std::cout << dt << std::endl;

    //number of elapsed miliseconds
    long msec = dt.total_milliseconds();

    //print elapsed seconds (with millisecond precision)
    std::cout << msec/1000.0 << std::endl;

    return 0;
}
```

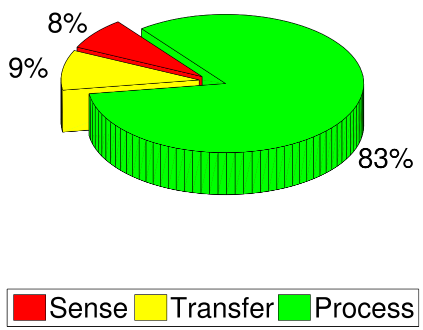
✓ **Software based timers** have been integrated with the driver that controls the operation of each different Sensing device:

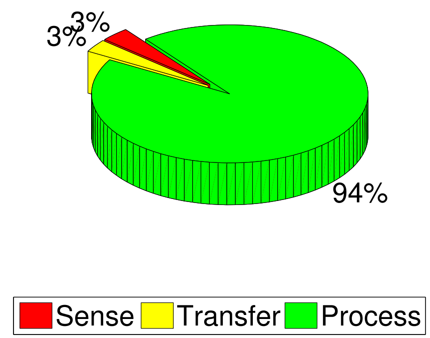
•USRP N210: UHD Driver - USRP E110: UHD Driver

•IMEC SE: imec SE driver - Atheros AR9380: ath9k driver

3. Sensing Delay Evaluation

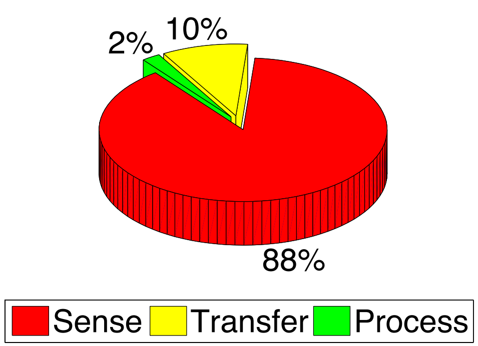
USRP N210

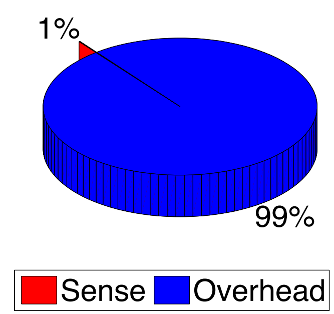




USRP E110

imec SE



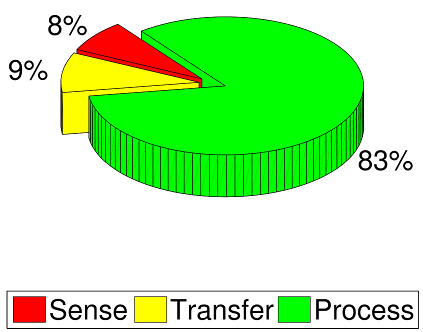


Atheros AR9380

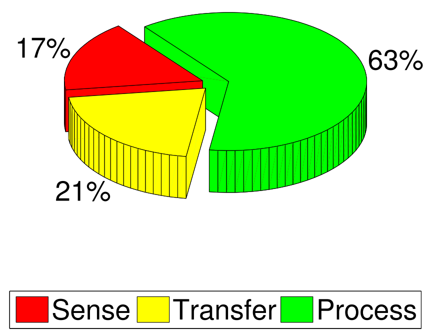
- ✓ Processing dominates the total sensing time for the **USRP devices**.
- ✓ The **imec SE** runs all processes in dedicated hardware and is capable of continuous sensing.
- ✓ The **Atheros card** induces a huge overhead (**55ms**) that makes it incapable of continuous sensing.

3. Sensing Delay Evaluation

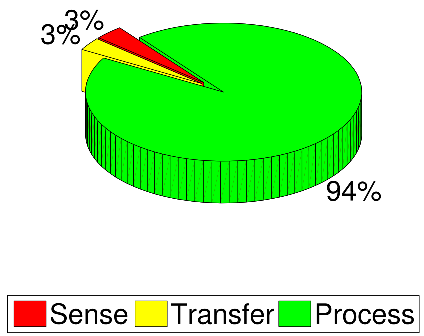
Sensing Delay Distribution for the USRP devices across Different processing platforms.



USRP N210 – ATOM-D525, Dual-core, 1.80 GHz



USRP N210 – i7-2600, Quad-core, 3.40 GHz



USRP E110 – ARM Cortex-A8, Single-core, 1 GHz

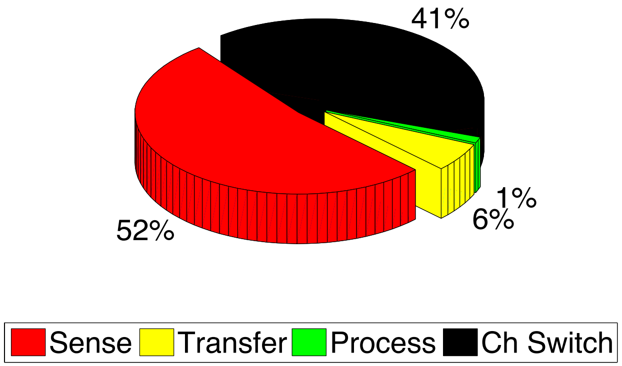
- ✓ The **i7-equipped** setup significantly reduces the Processing time, in comparison with the **ATOM** and **ARM** based setups.
- ✓ While experimenting with longer sensing intervals, we observed that the duration of the Processing process becomes significantly lower.
- ✓ We expect that continuous sensing can be performed in host machines able to achieve **significant amount of parallel processing**.

3. Sensing Delay Evaluation

Sensing Delay Distribution in scenarios requiring Channel Switching.

		Channel Switching Delay
Sensing Device	USRP N210	50 ms
	USRP E110	50 ms
	imec SE	50 μs
	Atheros AR9380	1-2 ms

Channel switching overhead per device



imec SE

Only the **imec SE** results in channel switching overhead values in the order of **μs** that are comparable with the configured sensing interval of **64 μs**.

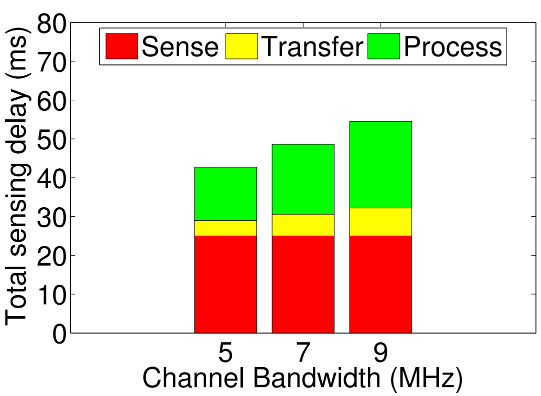
Outline

- Introduction
- Evaluation Approach
  - Power Spectral Density Evaluation
  - Power Consumption Evaluation
  - Sensing Delay Evaluation
- **External Use Case Evaluation**
- **Benefits for the experimenter**

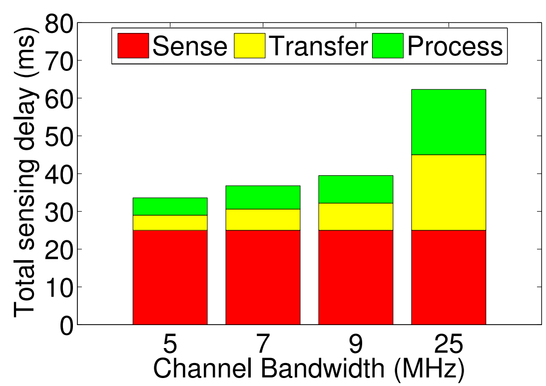
External Use Case Evaluation

- ✓ We assess the performance of a spectrum sensing engine that implements parallel processing on the USRP N210 platform and has been shown to sense in real-time when running on a **hexa-core server** machine.

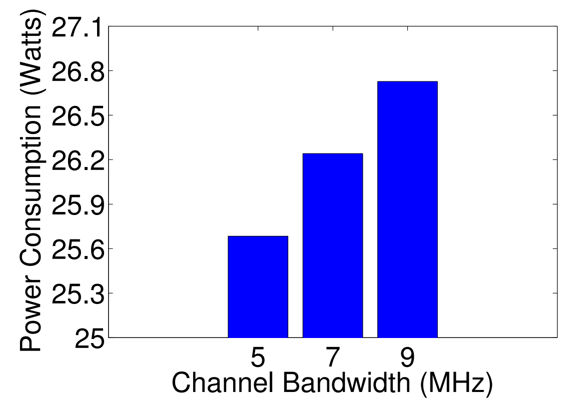
- ✓ We use the developed framework to evaluate performance under the **dual-core ATOM-based** and the **quad-core i7-based** setups and configure the **sensing interval at 25 ms**.



Sensing Delay Distribution - ATOM



Sensing Delay Distribution - i7



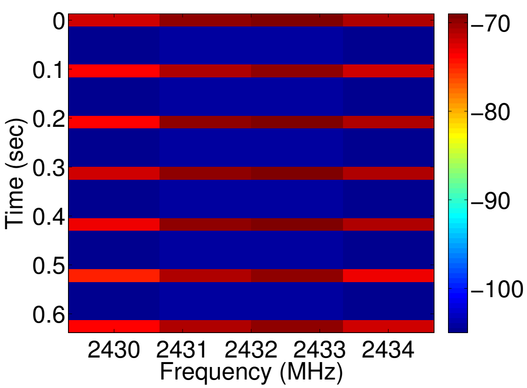
Power Consumption - ATOM

- ✓ The **ATOM-based** setup is able to continuously monitor up to **9 MHz** of bandwidth.
- ✓ The **i7-based** setup is able to sustain even the **25 MHz** bandwidth configuration.

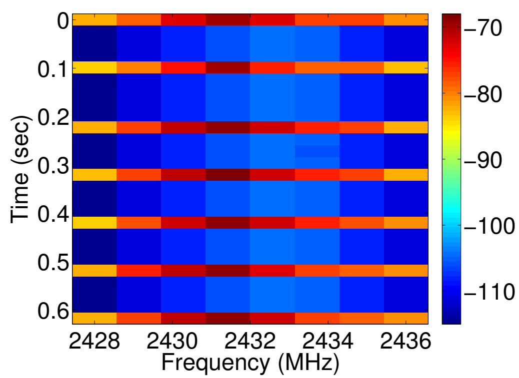
- ✓ We also observe that power consumption is also affected by the bandwidth increase, as **5 MHz, 7 MHz and 9 MHz**, correspond to **25.684 W, 26.2413 W and 26.7276 W** consumption.

External Use Case Evaluation

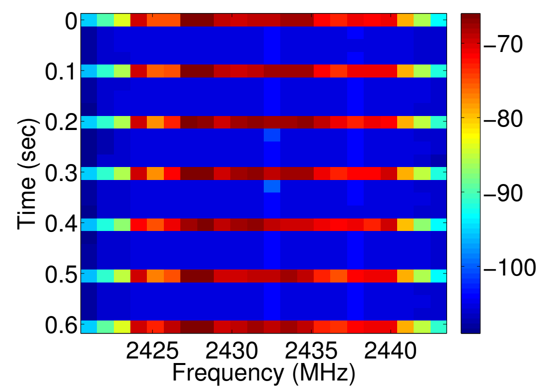
- ✓ We configure the SE to detect the presence of an 802.11 beacon transmitted every 100 ms.
- ✓ We are able to take 4 decisions about channel occupancy within the 100 ms Beacon interval.



USRP N210 – ATOM
5 MHz



USRP N210 – ATOM
9 MHz



USRP N210 – i7
25 MHz

- ✓ We validate that Beacons are detected with **100% success rate** in the 3 different setups.

Outline

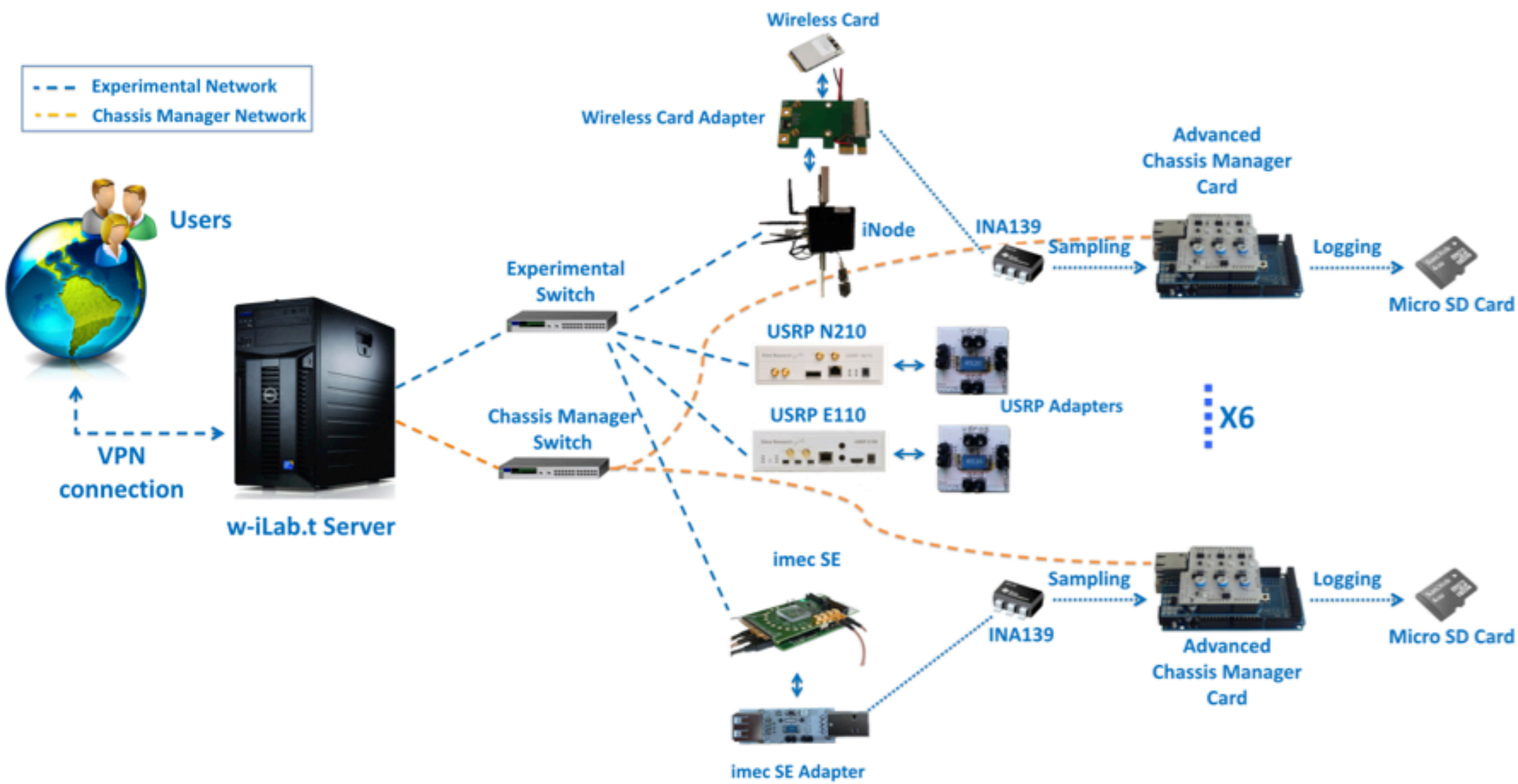
- Introduction
- Evaluation Approach
  - Power Spectral Density Evaluation
  - Power Consumption Evaluation
  - Sensing Delay Evaluation
- External Use Case Evaluation
- **Benefits for the experimenter**

Benefits for the experimenter

- **What functionality can be used by experimenters, e.g.**
  - Power Consumption evaluation procedure
    - Hardware
      - NITOS ACM card
      - Modified adapters (WiFi cards, USRPs, imec SE)
      - Installation and integration with w-ilab.t testbed
    - Software
      - Measurements Processing Software (Python, Matlab)
  - Sensing Delay evaluation procedure
    - Customized software for each device
      - Modified UHD driver
      - IMEC SE driver
      - ath9k driver
  - Automated and transparent use of the framework through OMF
  - Documentation and experience collected through the evaluation of several use case scenarios, including an external use case as well.

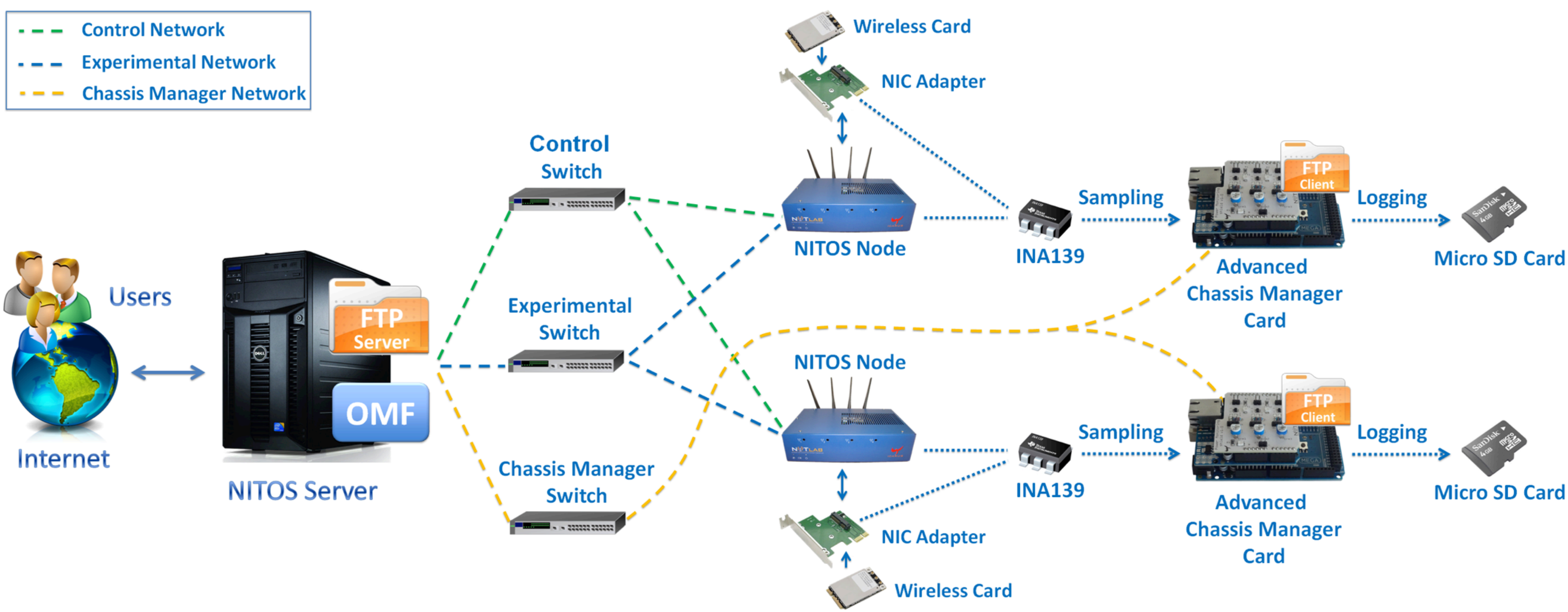
Benefits for the experimenter

The Power Consumption Monitoring framework is currently installed in the iMinds w-ilab.t Testbed and is accessible by CREW experimenters!



Benefits for the experimenter

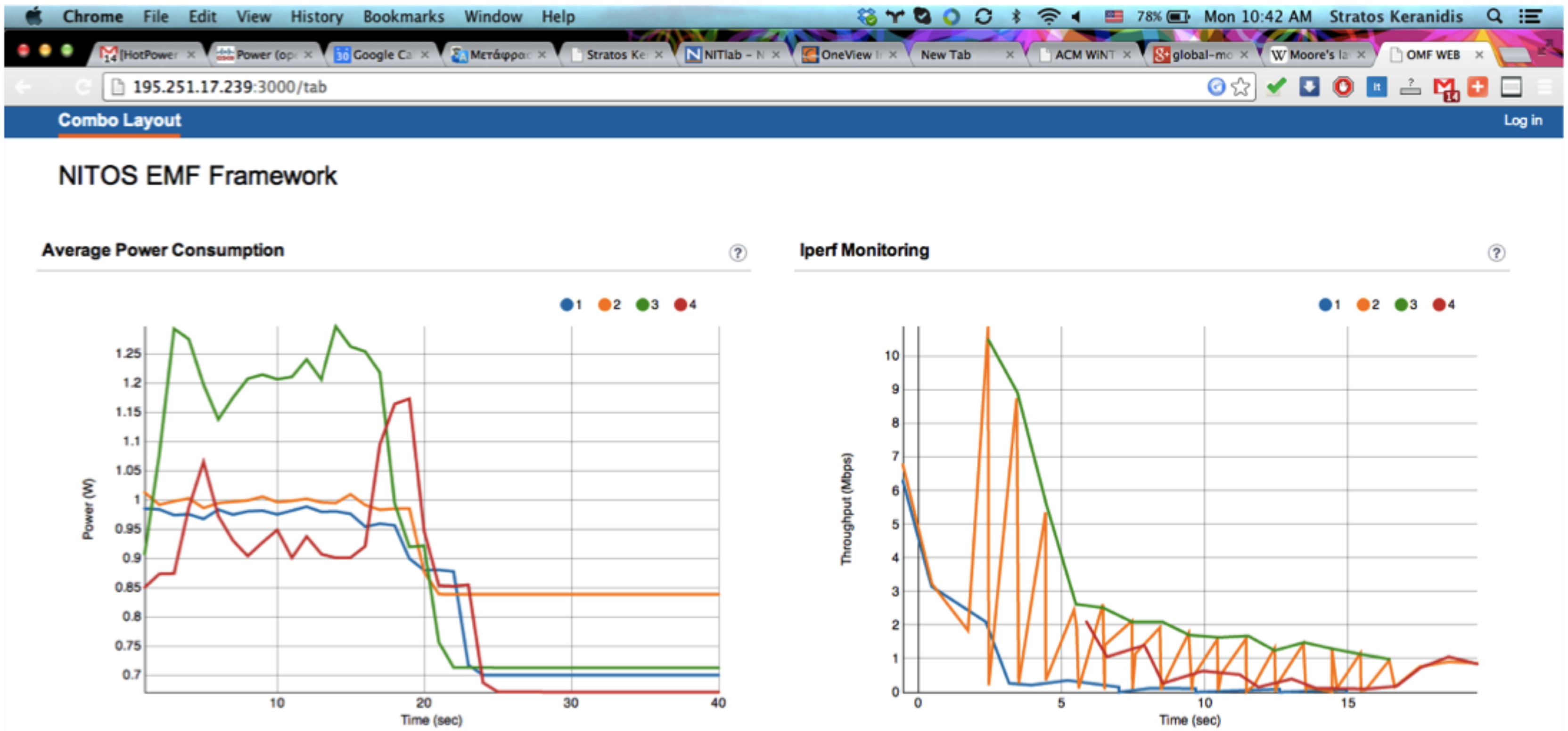
The Power Consumption Monitoring framework is also installed in the UTH NITOS Testbed and is publicly accessible by any interested experimenter!



Benefits for the experimenter

The developed framework is fully integrated with the

OMF Control and Measurement Framework.



Thank You!