SEVENTH FRAMEWORK PROGRAMME

IP CREW

Cognitive Radio Experimentation World

Usage of cognitive components in w-iLab.t

Wei Liu
15th Janurary 2014

iMinds imec ctvr berlin TECHNISCHE UNIVERSITÄT DRESDEN THALES EADS IJS

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 258301 (CREW project).

Outline

- **Cognitive components in w-iLab.t**
- **Deployment**
- **Usage**
  - General guideline
  - Customized sensing solution
    - USRP sensing engine
    - imec sensing engine
    - WARP sensing engine
  - Thalis API
- **Hands on**

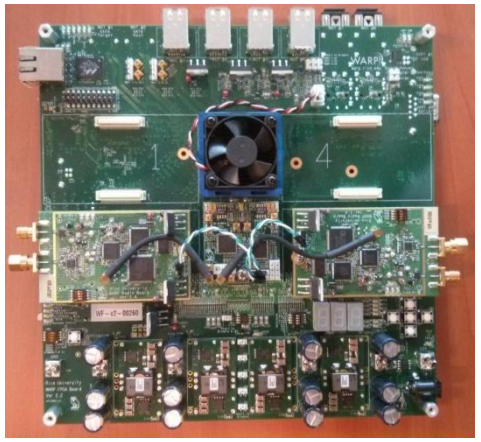
Cognitive components in w-iLab.t

USRP

- USRP N210 + XCVR 2450
- Xilinx® Spartan® 3A-DSP 3400 FPGA
- Gigabit Ethernet connection to host

WARP

- Wireless open-Access Research Platform
- Xilinx® Virtex ® 4 FPGA
- Power PC processor
- Ethernet connection to host

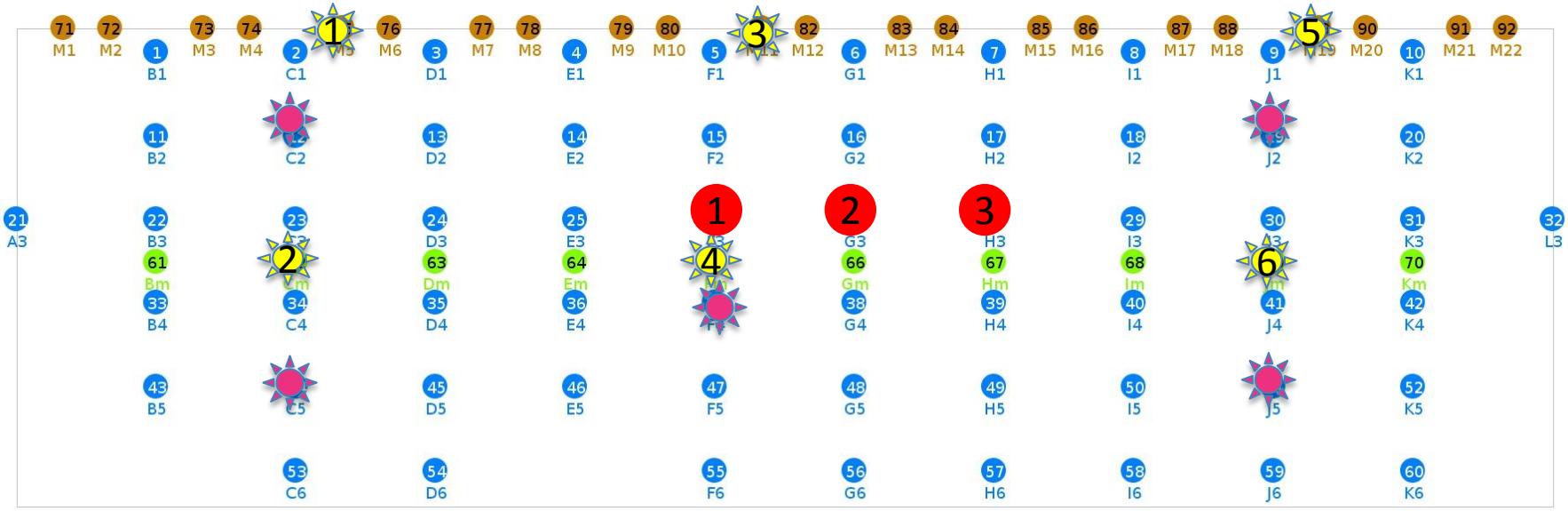


imec sensing engine

- Dedicated sensing engine
- NOT a complete SDR solution
- Scaldio / WARP frontend
- Powered and configured via USB

Deployment

Where are they installed ?

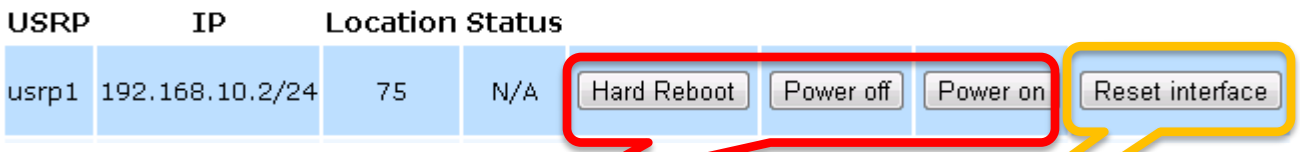


- 6 USRP have different IP addresses 192.168.X0.2 (X=1,2,3,4,5,6)
  - location: 75, 62, 81, 65, 89, 69
- WARP's IP address depends on the bitstream
  - Location: 26, 27, 28
- Imec sensing engine
  - Location: 12, 19, 37, 44, 51

Deployment

How are they connected

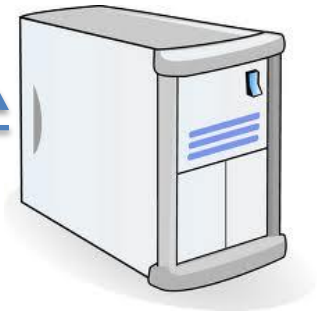
- USRP
  - Topology defined in Emulab experiment
  - Power control on the wilab2 status page



Hexa-core server
UHD, GNU Radio installed
Can also use Iris
Thalis API ...

High speed switch

Ethernet



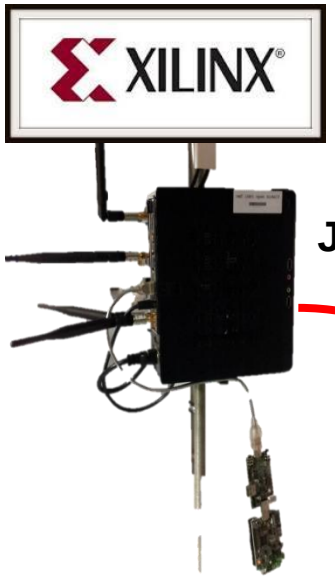
Deployment

How are they connected

- WARP
  - Topology defined by Emulab experiment
  - Power cycle can be controlled on the status page

WARP PDU-port Status

warp1	96	N/A	Hard Reboot	Power off	Power on



JTAG USB

Ethernet

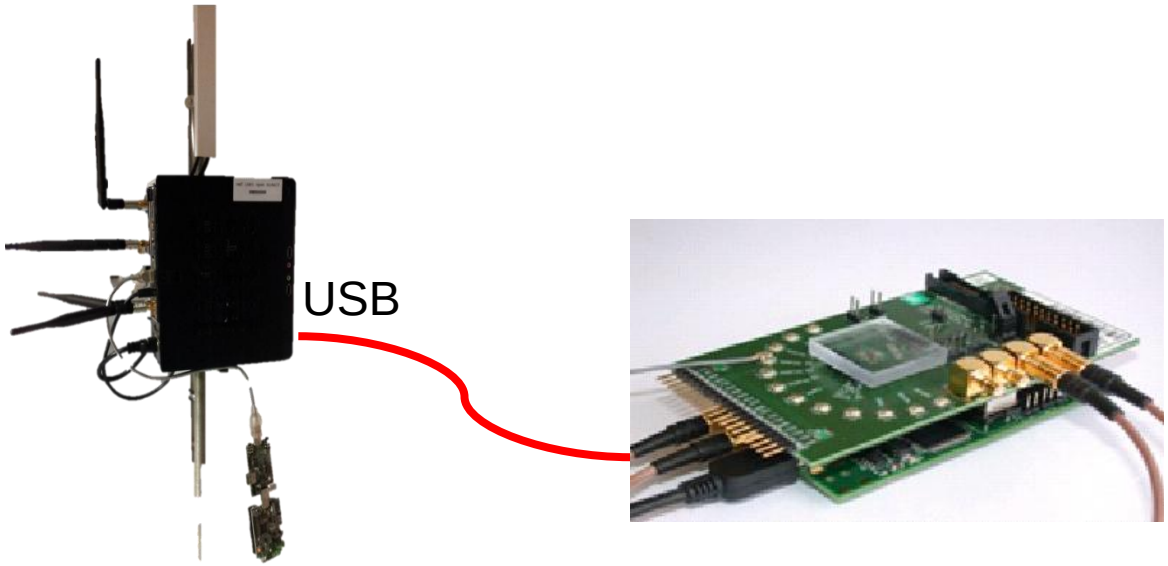
Configure FPGA

Communicate with WARP after the FPGA is configured

Deployment of the sensing engines

How are they connected

- imec sensing engine
  - Attached to dedicated Zotac nodes via USB
  - Include zotac nodes in the “.ns” emulab file
  - Powered and configured over USB



Outline

- **Cognitive components in w-iLab.t**
- **Deployment**
- **Usage**
  - General guideline
  - Customized sensing solution
    - USRP sensing engine
    - imec sensing engine
    - WARP sensing engine
  - Thalis API
- **Hands on**

Usage

General guideline

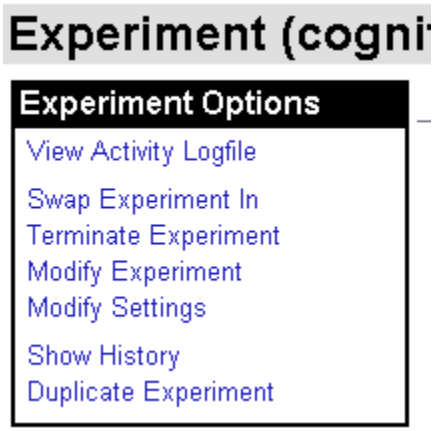
- Create or use existing experiment on Emulab
- Get reservation
  - The experiment will be swapped in when your reservation starts
- Configure FPGA
  - Log on to Zotac node where the imec sensing engine or WARP is attached, and configure FPGA
  - Not needed for USRP, as it boots from the flash
- Configure network
  - For USRP and WARP, log on to the server and make sure the device is in the same subnet of the host interface.
  - Not needed for imec sensing engine, as it only uses USB port
- Reset the switch and the device
  - Reset USRP or the switch interface or both if you can not ping USRP, only needed for USRP

Usage

General guideline

- Create or use existing experiment on Emulab
  - http://wilab2.ilabt.iminds.be/
  - Log in
    - User: crew
    - Password: training@wilab2
- Example ns script : 1 USRP and 1 SERVER 5P

Do not click on:



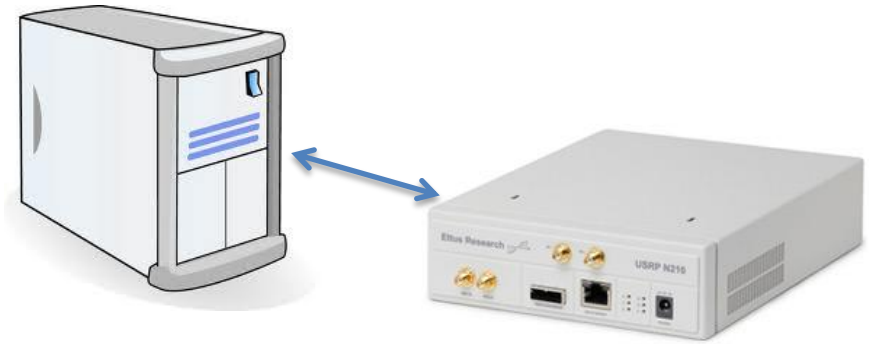

```
# Generated by NetlabClient

set ns [new Simulator]
source tb_compat.tcl

##define serv1 and usrp1 and link0
set serv1 [$ns node]
$serv1 add-desire SERVER5P 1.0
tb-fix-node $serv1 server1

set u1 [$ns node]
tb-fix-node $u1 usrp1
$u1 add-desire USRP 1.0

set link1 [$ns duplex-link $serv1 $u1 1000000.0kb 0.0ms DropTail]
tb-set-ip-link $serv1 $link1 192.168.10.1
tb-set-ip-link $u1 $link1 192.168.10.2
tb-set-netmask $link1 "255.255.255.0"
```



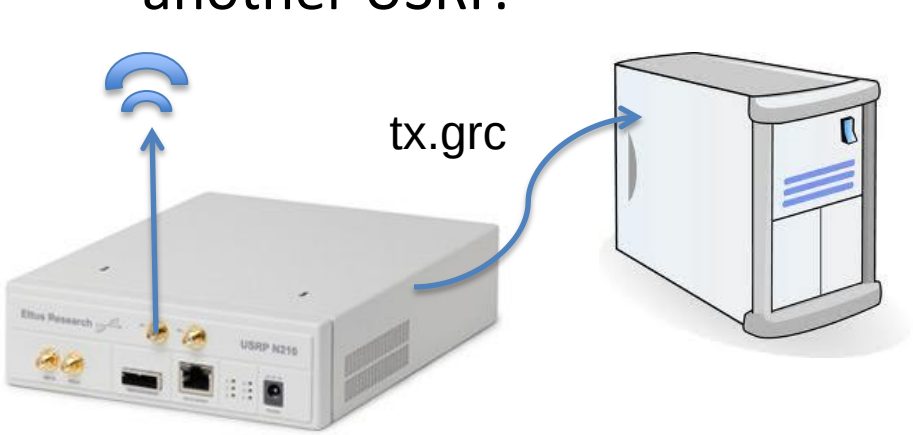
Usage

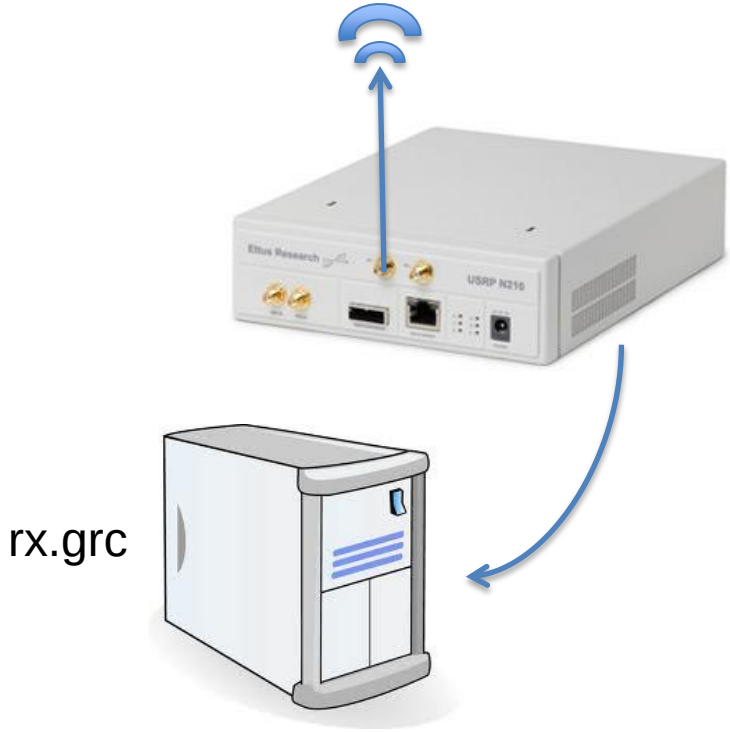
General guideline

Device	Software Environment	Processing	Development
USRP N210	UHD (C++) GNU Radio, Iris, Thalis API, Labview, Simulink…	Host computer	Host computer
WARP	Xilinx System Generator Matlab	FPGA on Board/Host computer	FPGA on Board/Host computer
imec sensing engine	imec sensing engine library (C)	Integrated Circuit (DIFFS)	Host computer

General guideline

- Demo : using Gnuradio on w-iLab.t
- Transmit a signal with one USRP and detect the signal on another USRP.



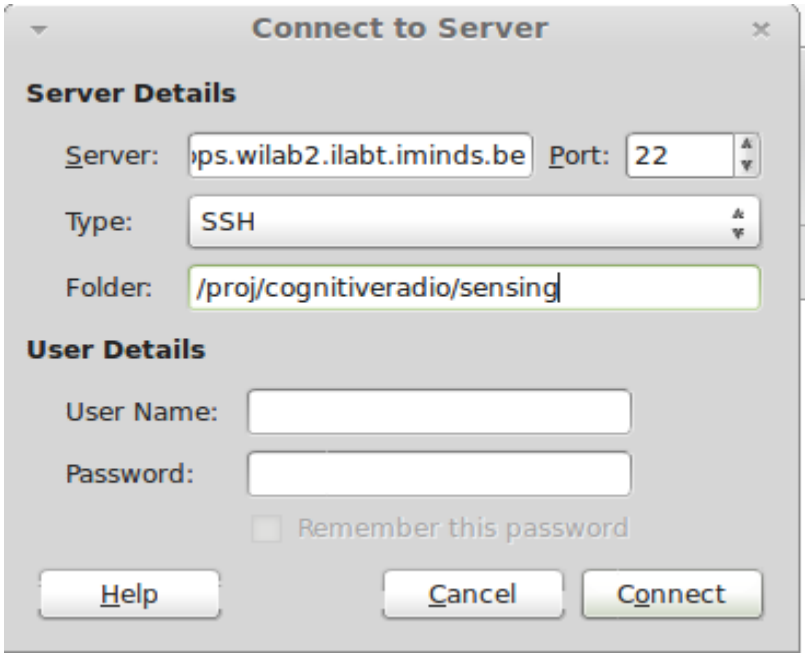


- Log in to the USRPX server:
  - *ssh –X crew@10.11.17.X*
- Start GNURadio Companion:
  - *gnuradio-companion*
- Run the flow graph as if it is running on your own PC

Usage

General guideline

- Where to find the documents
  - All documents are located on ops.wilab2.ilabt.iminds.be
  - /proj/cognitiveradio/sensing
  - 3 sub folders: imecse, usrpse, warpse
  - Starting point: README and manual in each folder
- On linux:
  - Open a file browser
  - Click on “File”
  - Select “Connect to Server”
  - Server:
    - ops.wilab2.ilabt.iminds.be
  - Folder
    - /proj/cognitiveradio/sensing
  - User
    - crew
    - training@wilab2

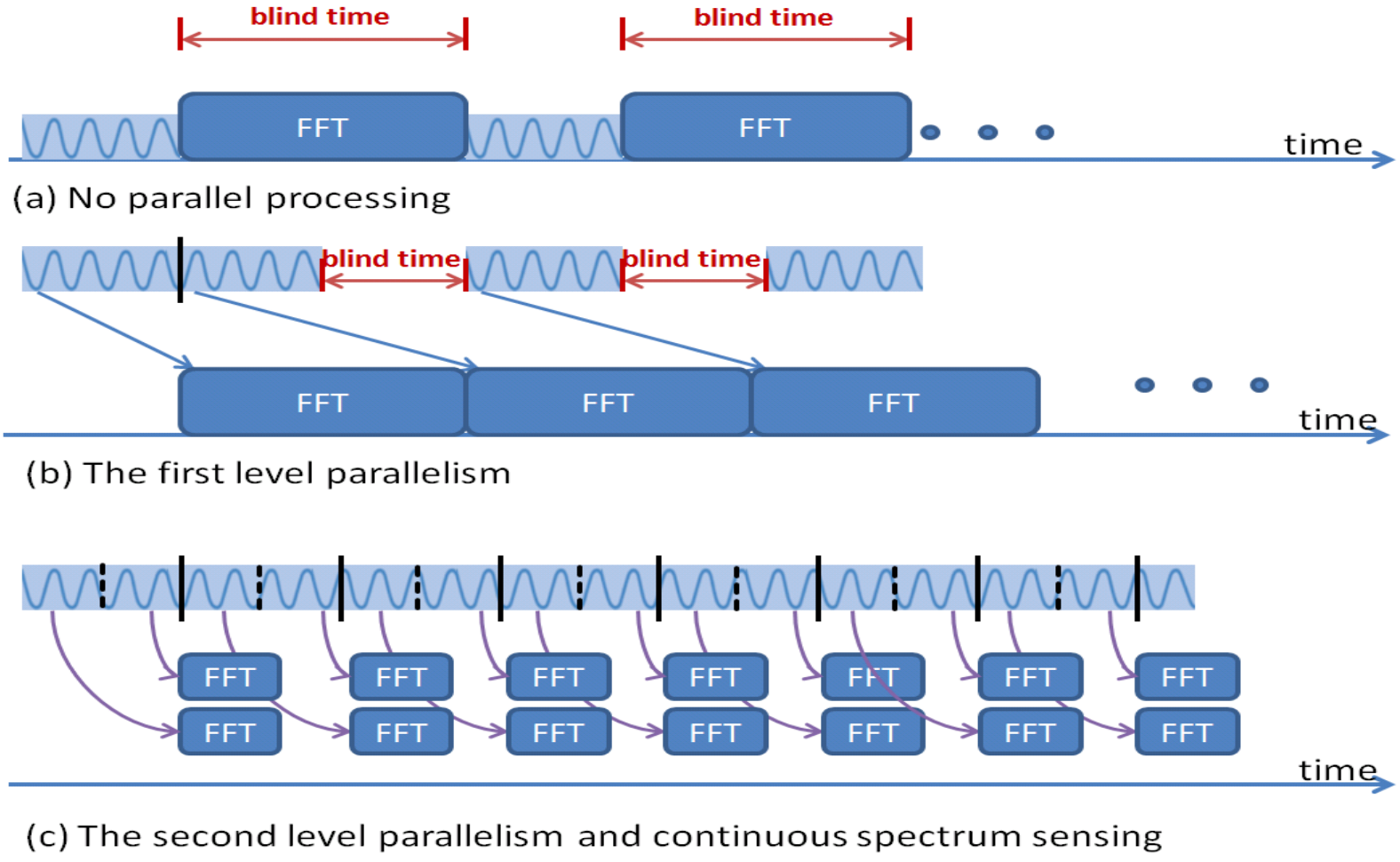


Outline

- **Cognitive components in w-iLab.t**
- **Deployment**
- **Usage**
  - General guideline
  - Customized sensing solution
    - USRP sensing engine
    - imec sensing engine
    - WARP sensing engine
  - Thalis API
- **Hands on**

Customized sensing solution

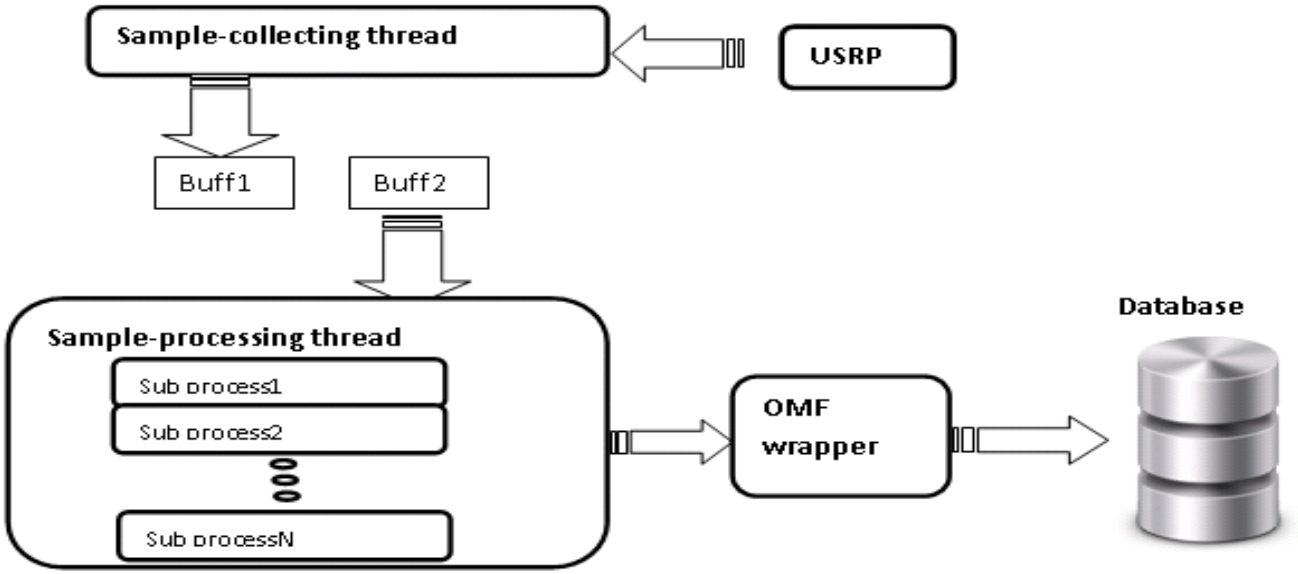
- USRP sensing engine



Usage

Customized sensing solution

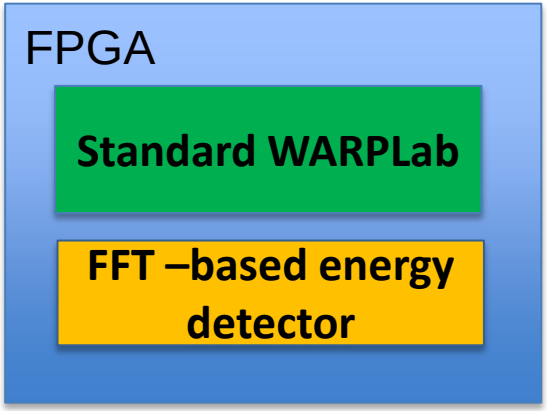
- USRP sensing engine
  - Written in C++ (above UHD driver)
  - Maximum real span 25 MHz
  - Sweeping and none sweeping modes (seamless capturing)
  - Use multi-threading to increase processing speed
  - Different detection mode: maxhold, averaging
  - User interface binary with options



Usage

Customized sensing solution

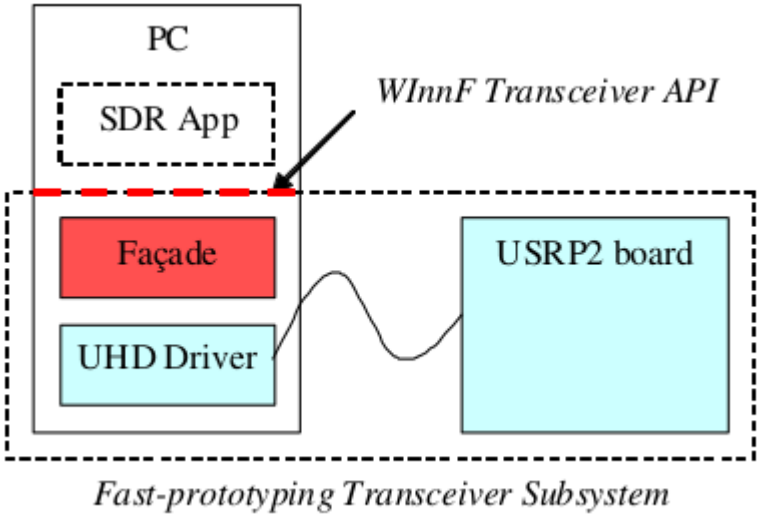
- imec sensing engine (training day)
  - Writen in C, using imec sensing library
  - Very fast sweeping
  - Various detection mode: WLAN_G, ZIGBEE, BLUETOOTH…
  - User interface: binary with options
- WARP sensing engine
  - Custom-designed FPGA bit stream
  - 40 MHz instantaneous frequency span
  - Up to 20 channels estimated in parallel
  - User interface: Matlab



Usage

Thalis API

- Located on the ops.wilab2.ilabt.iminds.be, /proj/cognitiveradio/data/ThalisAPI
- Structure



- More information refers to the presentation:
  - XCVRonUSRP2_Highlights CREW Training Days 190213 TCS presentation 1_2_v1.1.pdf

Outline

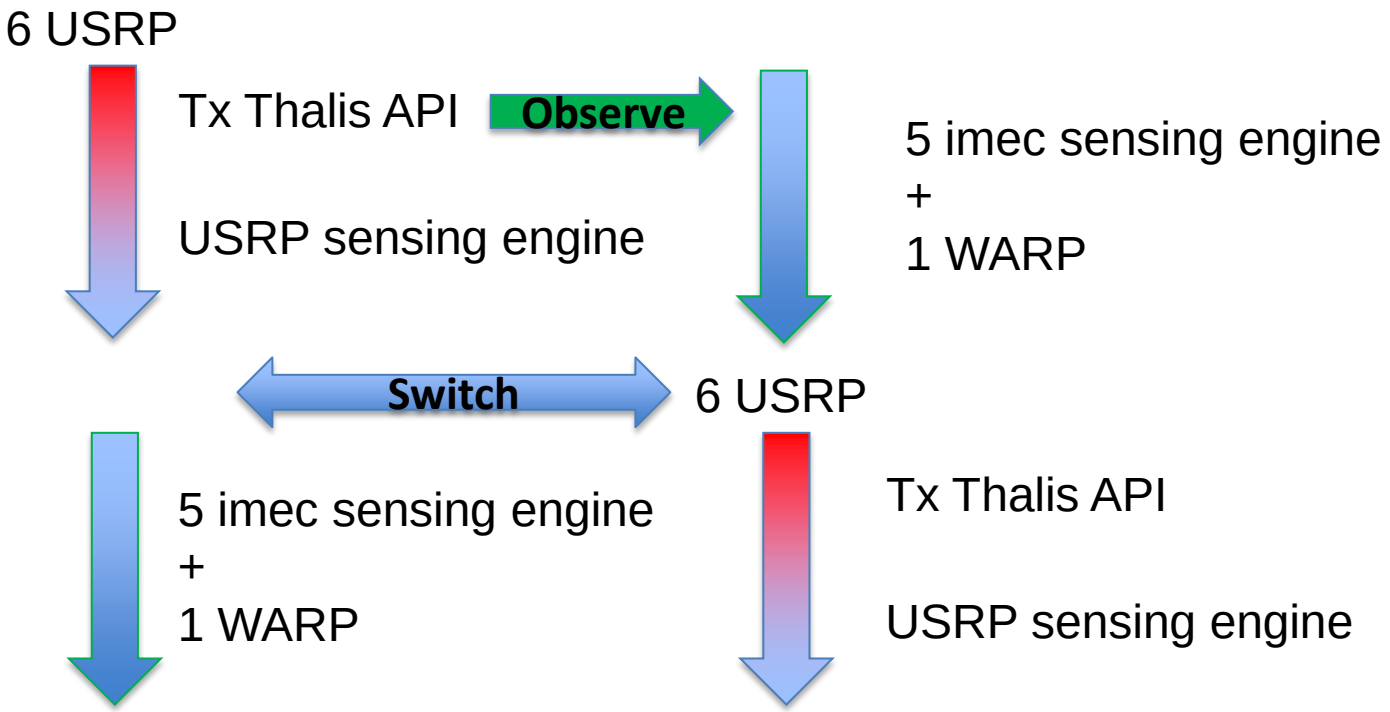
- **Cognitive components in w-iLab.t**
- **Deployment**
- **Usage**
  - General guideline
  - Customized sensing solution
    - USRP sensing engine
    - imec sensing engine
    - WARP sensing engine
  - Thalis API
- **Hands on**

Hands on

3 Types of lab instructions

- USRP N210: Thalis API + USRP sensing engine
- imec sensing engine
- WARP sensing engine

2 threads

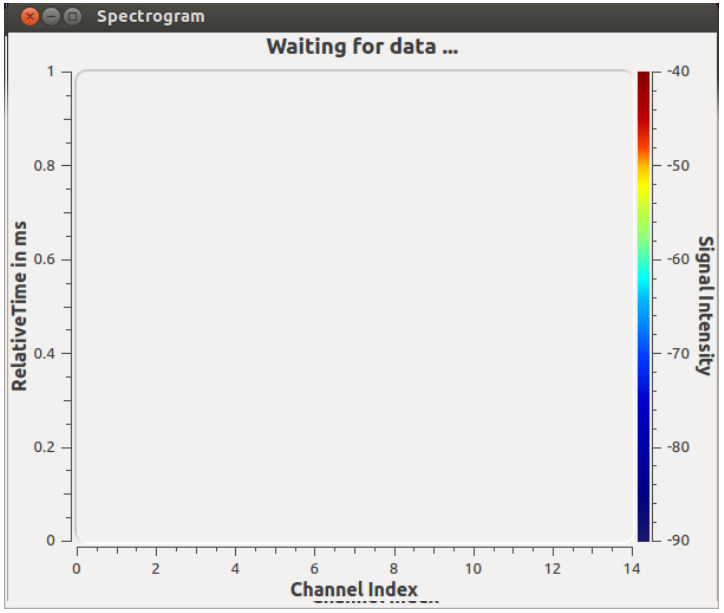


Hands on

USRP and imec sensing engine

- Local GUI application
- Remote sensing engine application

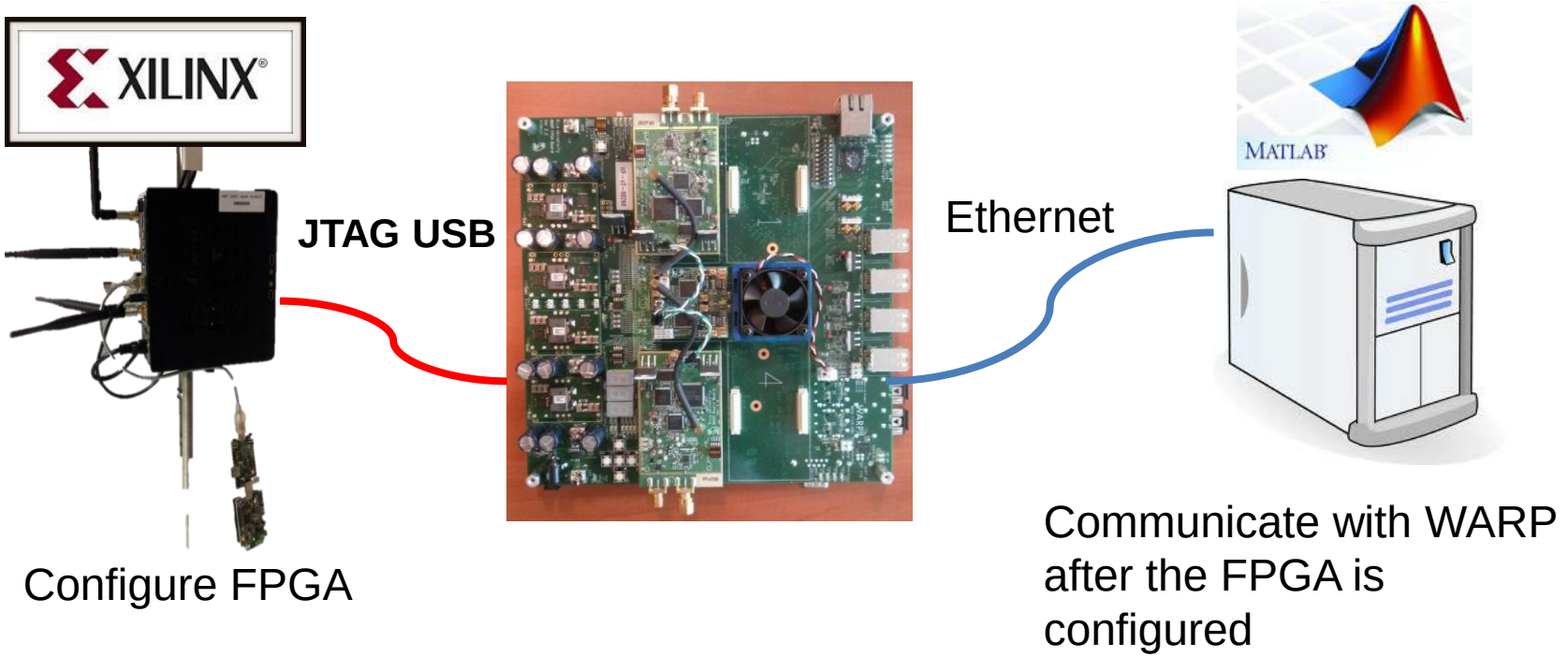
- To kill the apps:
  - First kill the sensing engine
  - Then kill the GUI
- Play with different options
- Take a look at the README
- imec sensing engine:
  - Compile your own application



Hands on

WARP sensing engine

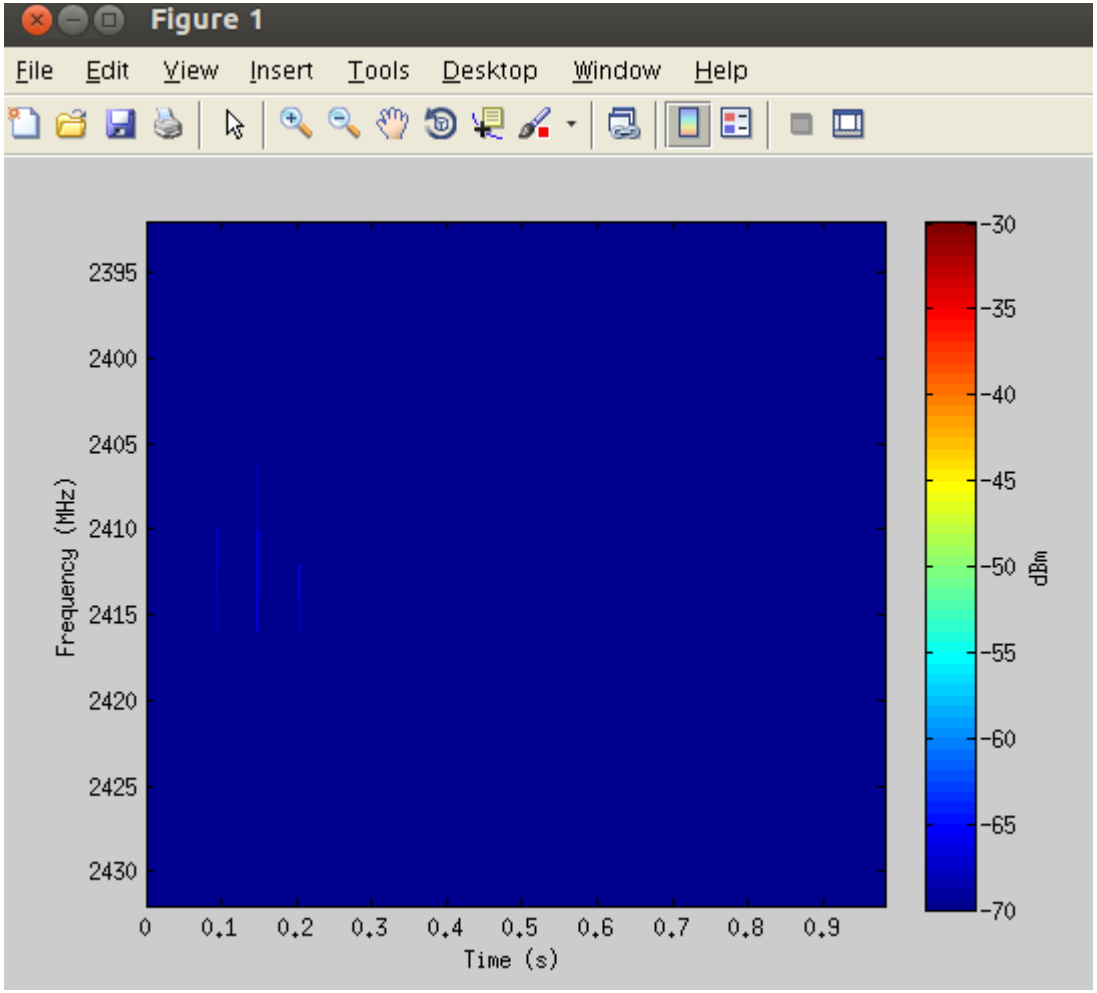
- First configure the FPGA on the Zotac node
- Then run the specified matlab script from the Server



Hands on

WARP sensing engine

- First configure the FPGA on the Zotac node
- Then run the specified matlab script from the Server



Hands on

WARP sensing engine

- First configure the FPGA on the Zotac node
- Then run the specified matlab script from the Server
- Take a look at the Matlab script
  - Configure the RF frequency in initWARPSE_visualization.m

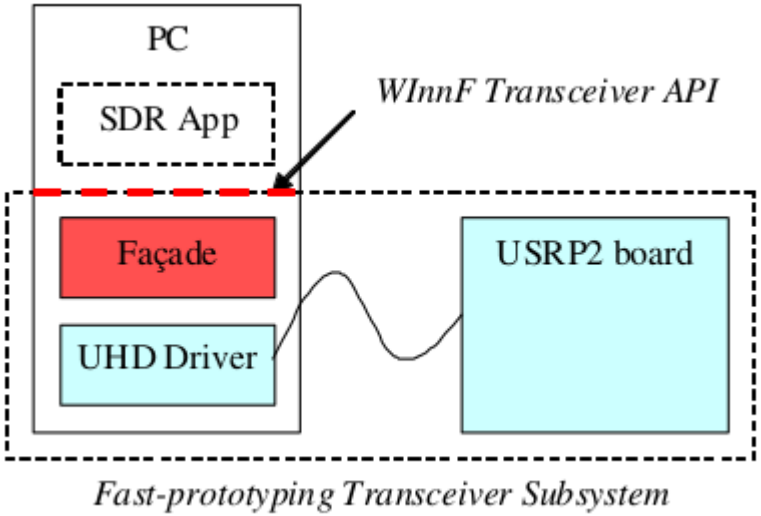
```
%%%%%%%%%%%%%%%%%%%%%%%% USER DEFINED PARAMETERS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CarrierChannel must an integer within 1 to 14, it corresponds the WLAN_G
% channel frequency, the visualizer will plot 40 MHz around the CarrierChannel
CarrierChannel = 1 ; % cover bluetooth 1-20
warplab_setRadioParameter(udp_txrx,CARRIER_CHANNEL,CarrierChannel);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

  - Run the sensing engine again

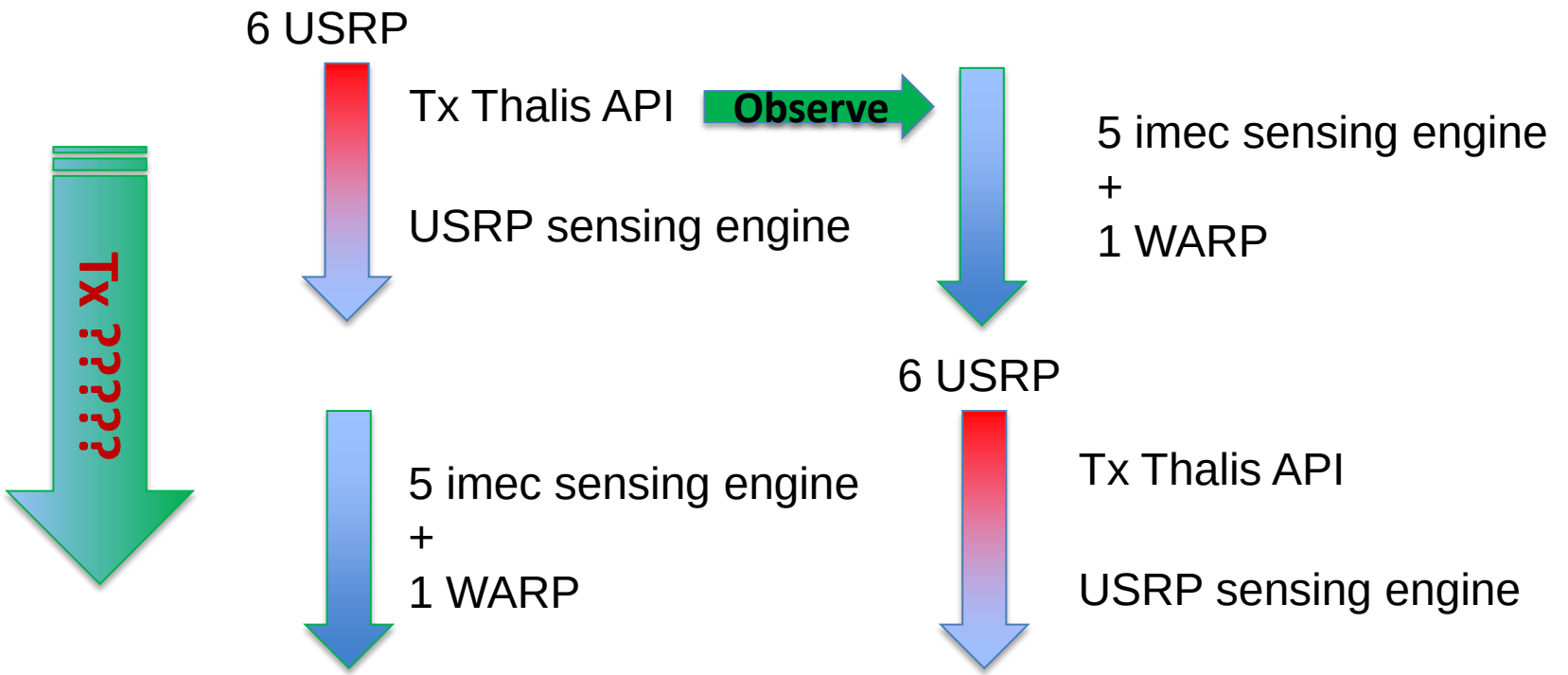
Hands on

Thalis API

- You will compile an application with Thalis API and transmit a signal
- Observe this signal on your neighbor's sensing engine
- Configure parameters and recompile



Hands on



what am I transmitting ?

Eg: Where is the transmitter?

What is the frequency ?

.....