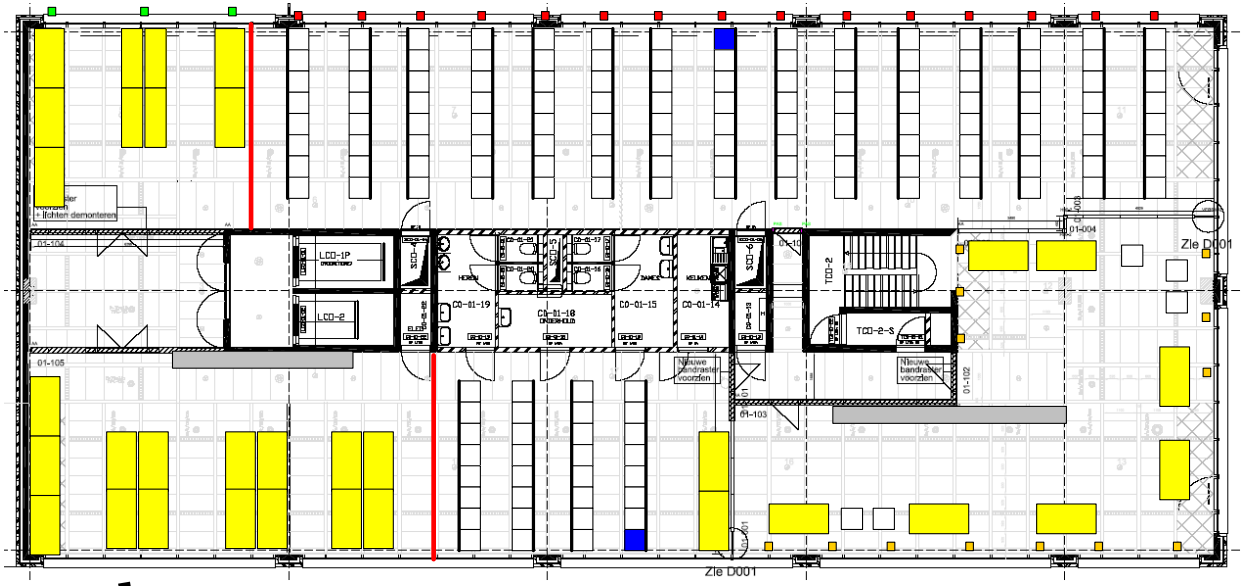
iMinds w-iLab.t

Hands-on: running experiments on w-iLab.t Zwijnaarde

Pieter Becue
Vincent Sercu
Bart Jooris
Stefan Bouckaert

About the iMinds iLab.t

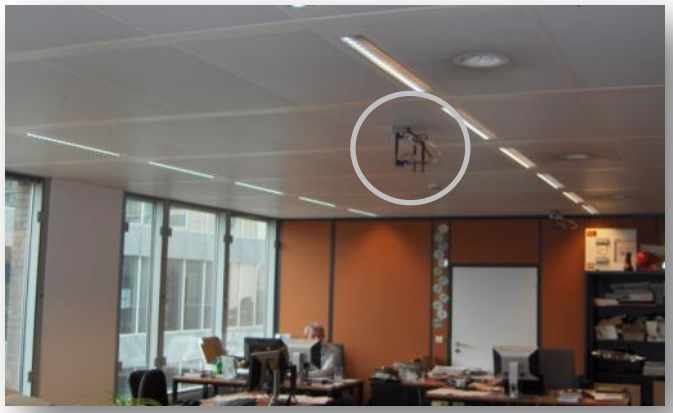
- The iLab.t research centre is located in Ghent



w-iLab.t : Facts and figures

- heterogeneous, generic testbed for wireless networks
  - Sensor networks
  - Wi-Fi based wireless ad-hoc/mesh/vehicular

- 2 testbed locations
  - Office: three office floors of 90m x 18m [200 nodes]
  - “Pseudo-shielded”, Zwijnaarde, 60m x 20m [60 nodes]

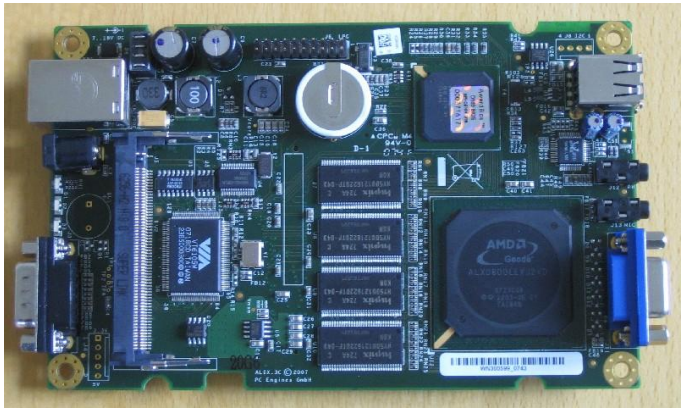


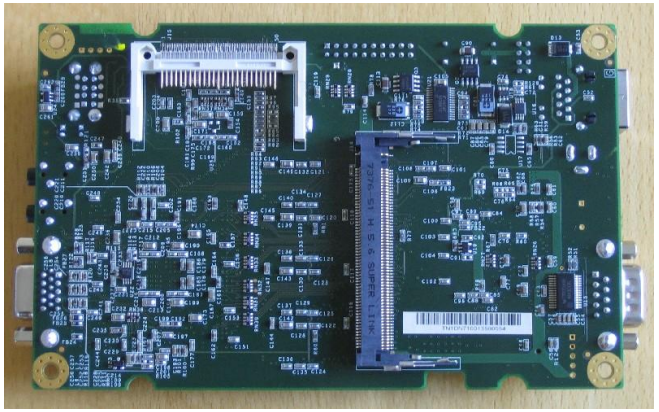
w-iLab.t

Hardware overview

Hardware – Embedded PC

- w-iLab.t Office
  - Alix 3c3 (500 MHz AMD, 256 MB Ram)
    - Ethernet NIC (100Mbit) / Serial port
    - VGA, onboard audio
    - compact flash storage (2 Gb)
    - 2 x mini PCI slot
    - USB

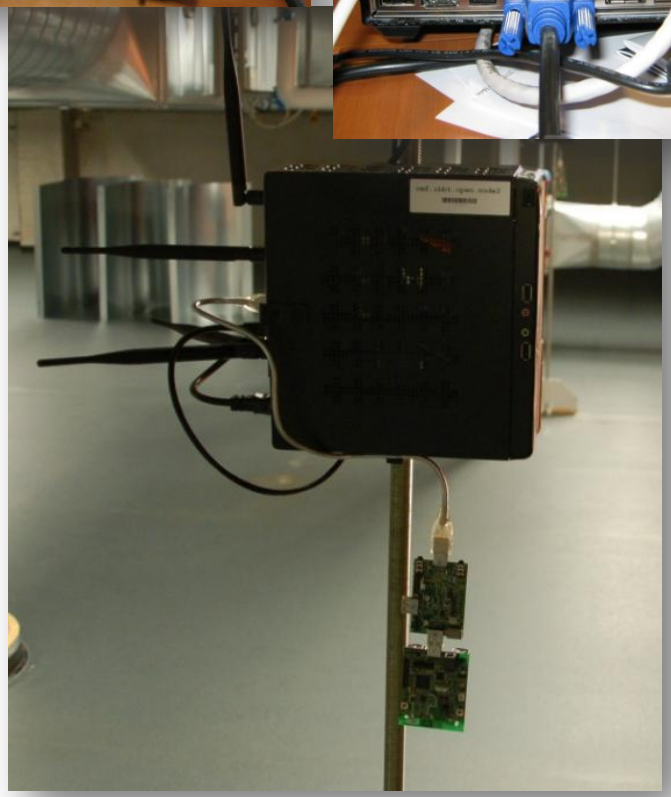




Picture source: www.pcengines.ch

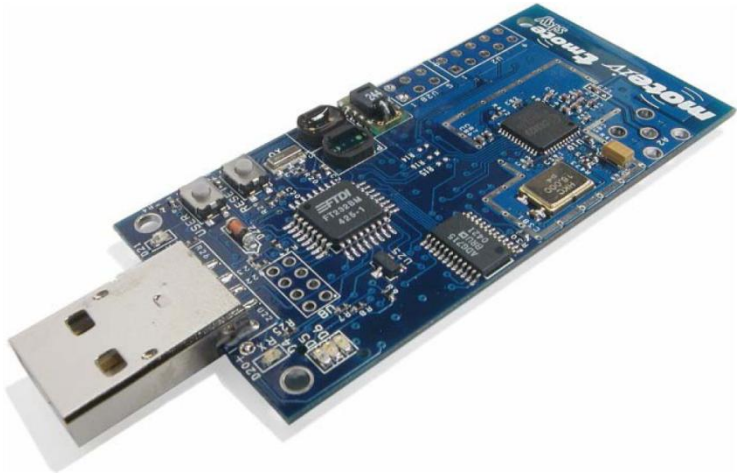
Hardware – Embedded PC

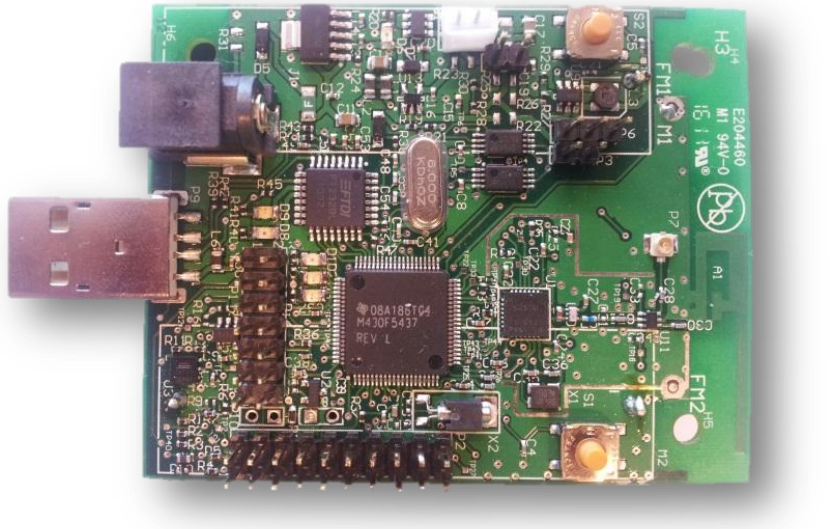
- w-iLab.t Zwijnaarde
  - Zotac
    - 4Gb RAM
    - 160 Gb Hard Drive
    - Intel Atom D510 1.66GHz Dual core
  - Wireless interfaces :
    - 802.11a/b/g/**n** (x2)
    - **Bluetooth**
    - **iMinds Rmoni sensor node** (802.15.4)
  - Environment Emulator
  - **Webcam** (20%)



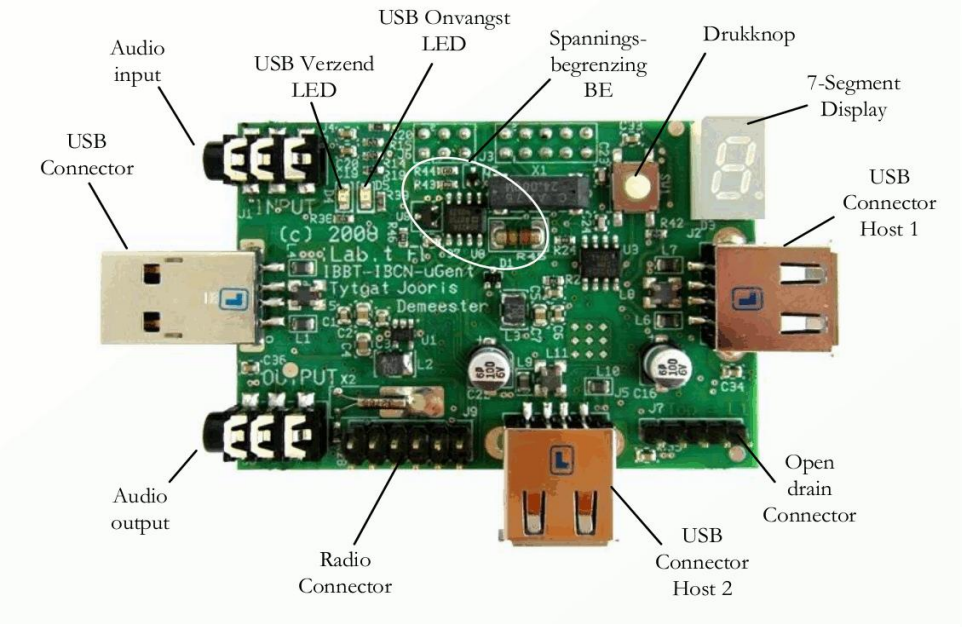
Hardware – sensor devices

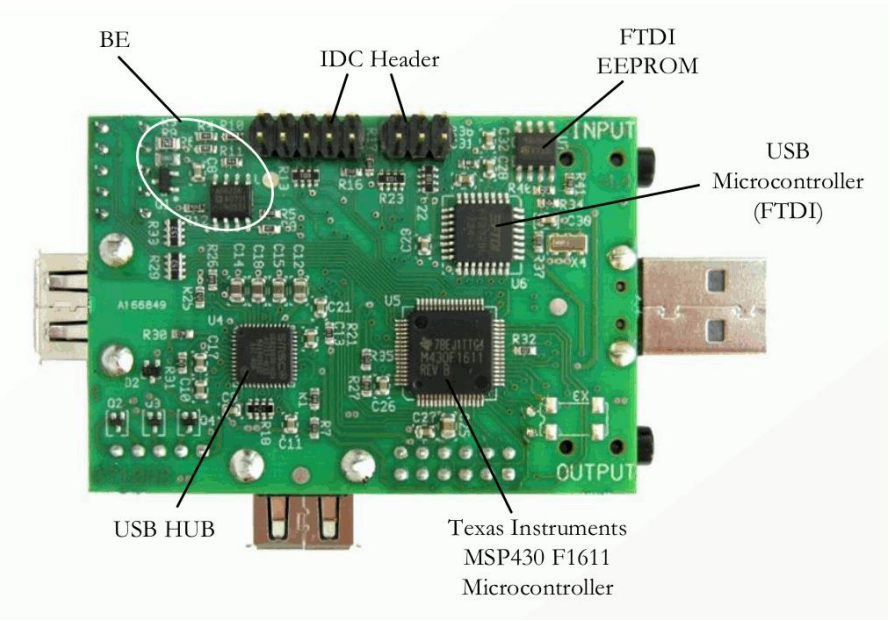
- Tmote Sky (office) & iMinds Rmoni (Zwijnaarde)
  - TI msp430
  - CC2420 or CC2520
  - Sensors for temperature and humidity
- Specifications available on www.crew-project.eu/portal





Hardware: environment emulator

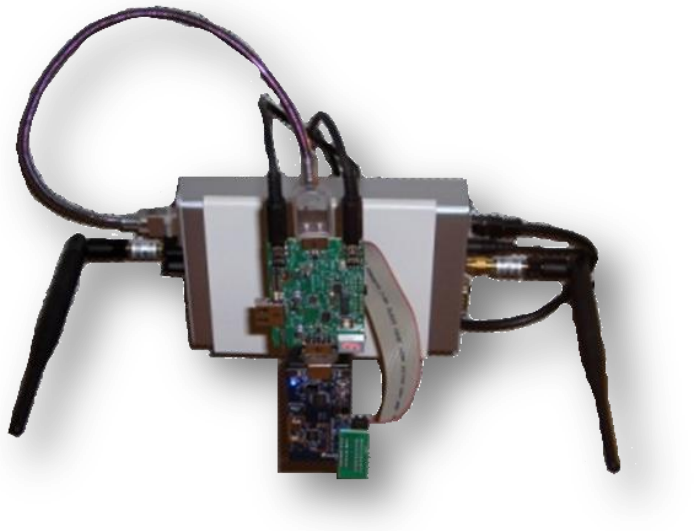


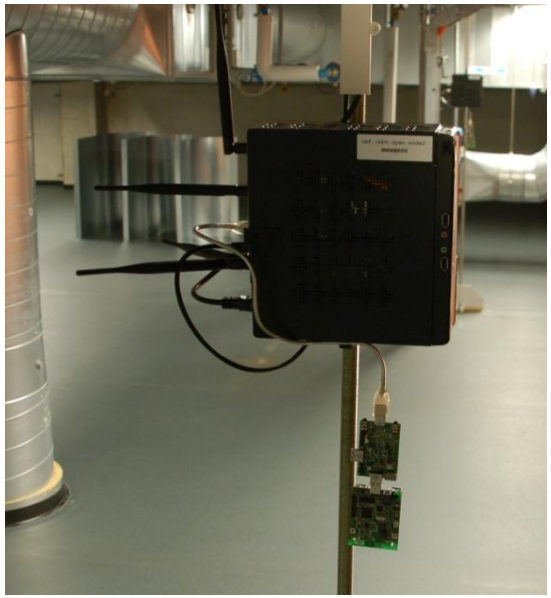


More information: L. Tytgat, B. Jooris, P. De Mil, B. Latré, I. Moerman, P. Demeester, **"Demo abstract: WiLab, a real-life wireless sensor testbed with environment emulation"**, published in European conference on Wireless Sensor Networks, EWSN adjunct poster proceedings (EWSN), Cork, Ireland, 11-13 February 2009

Node locations

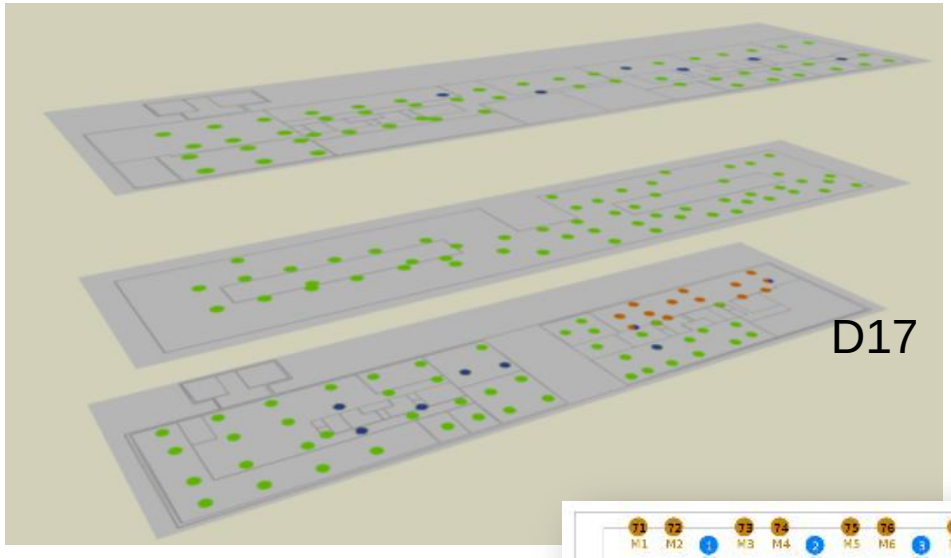
- w-iLab.t Office (x200)
  - Node = Embedded PC + EE + Tmote Sky
- w-iLab.t Zwijnaarde (x60)
  - Node = Embedded PC + EE + RM090





Topology

- w-iLab.t Office



- w-iLab.t Zwijnaarde

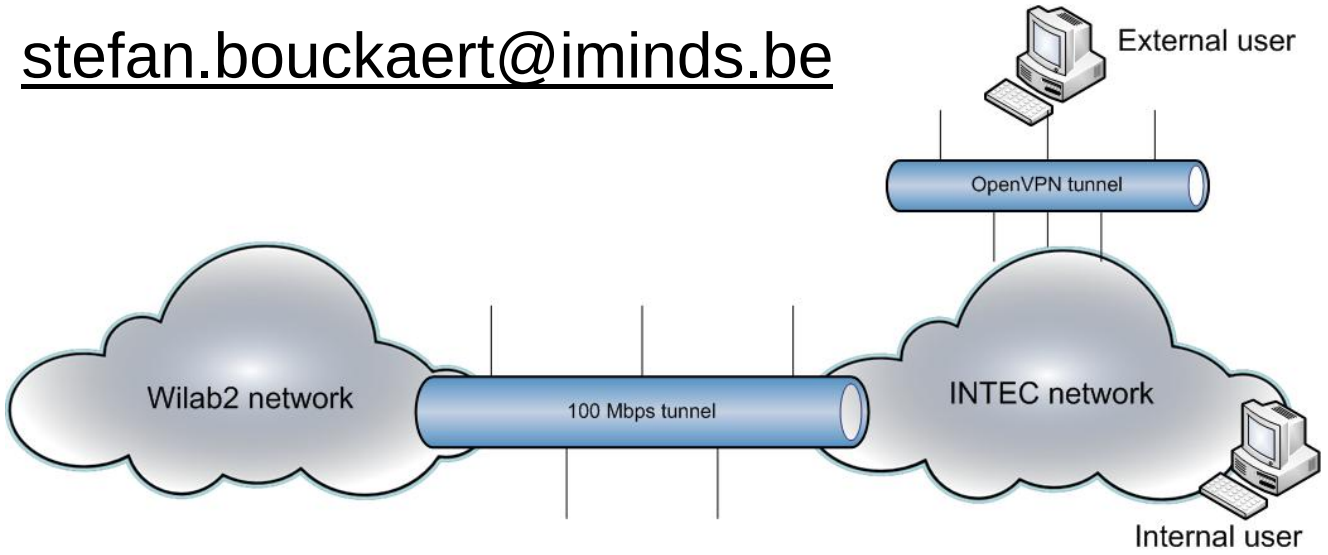
The w-iLab.t Zwijnaarde testbed

w-iLab.t

Testbed Architecture & Management

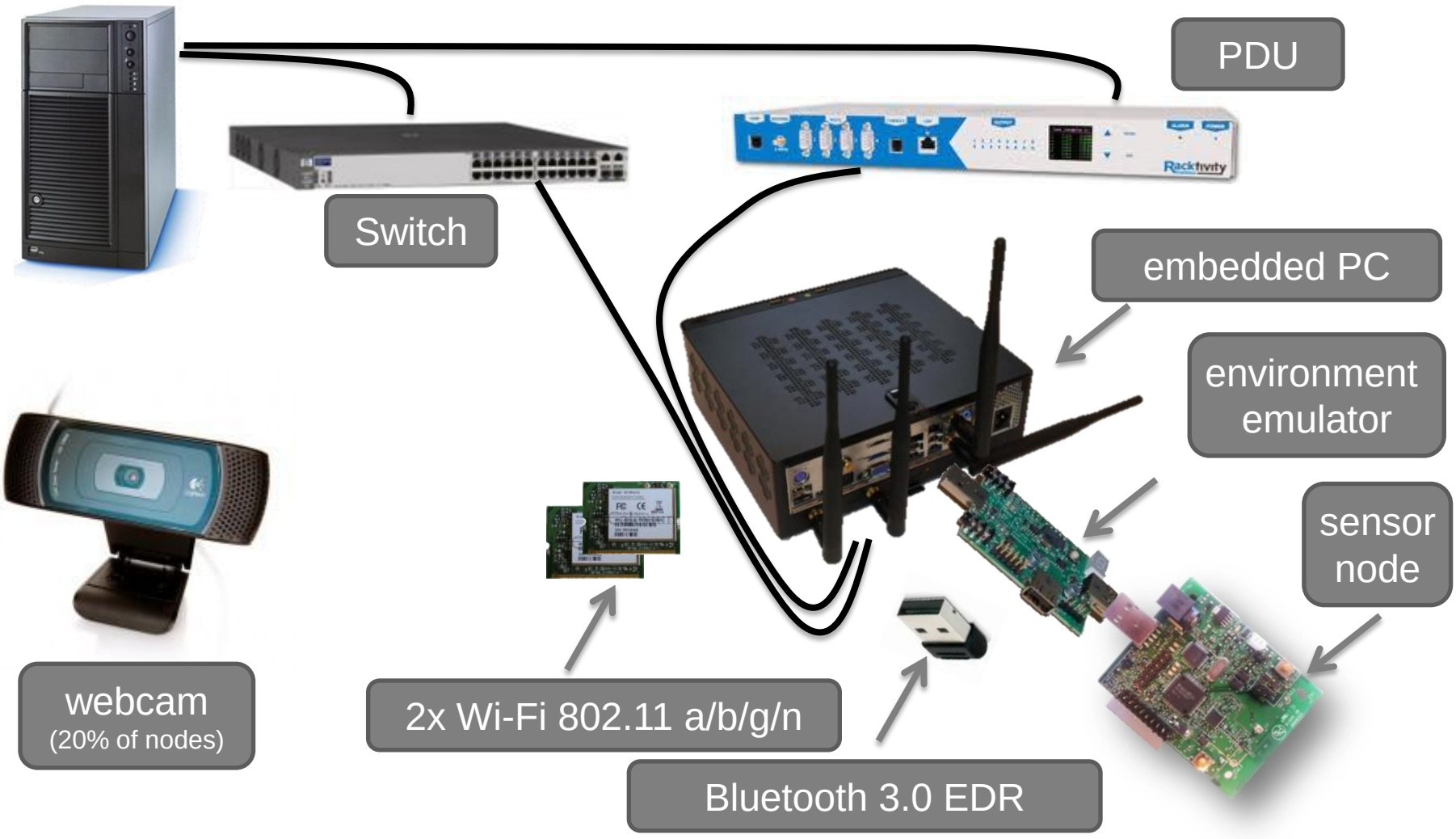
Network

- .wilab2.ilabt.iminds.be (10.11.16.0/20)
- OpenVPN required
  - mailto: stefan.bouckaert@iminds.be



- Outgoing connections only over HTTP_PROXY
  - http://proxy2.intec.ugent.be:8080
- **IPv6 enabled** (no VPN needed)

Testbed Architecture



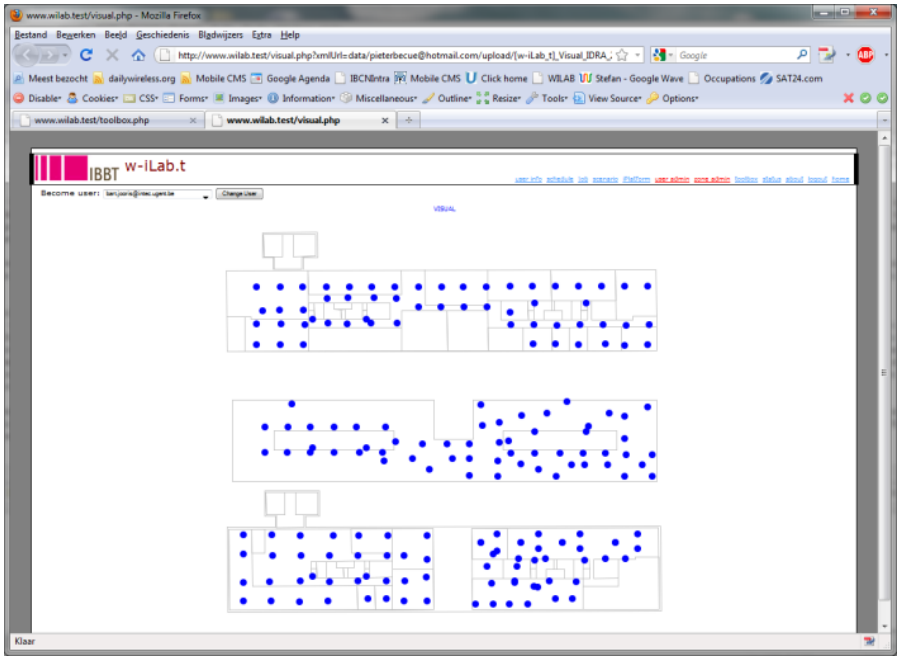
Management Framework & Tools

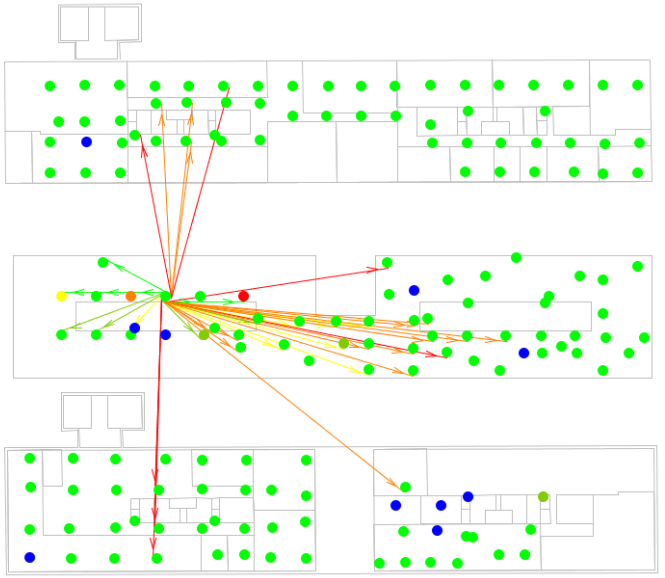
- Emulab (http://emulab.net)

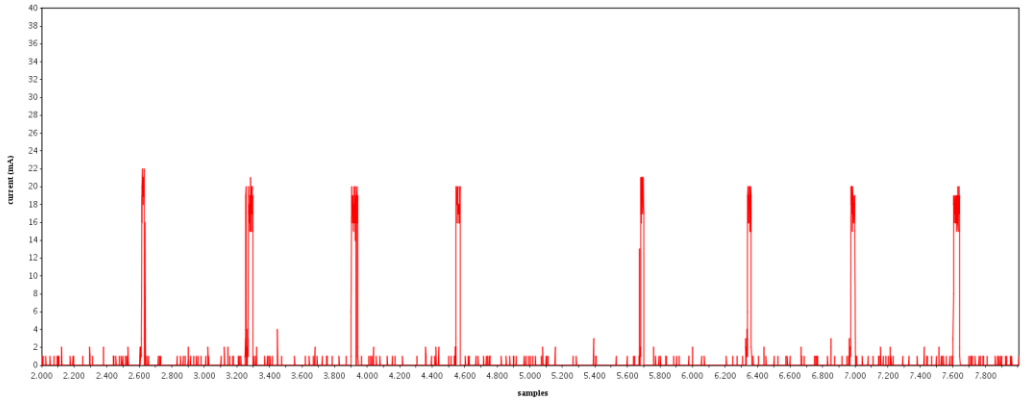
- OMF (http://omf.mytestbed.net)

- OML for collecting measurements (http://oml.mytestbed.net)

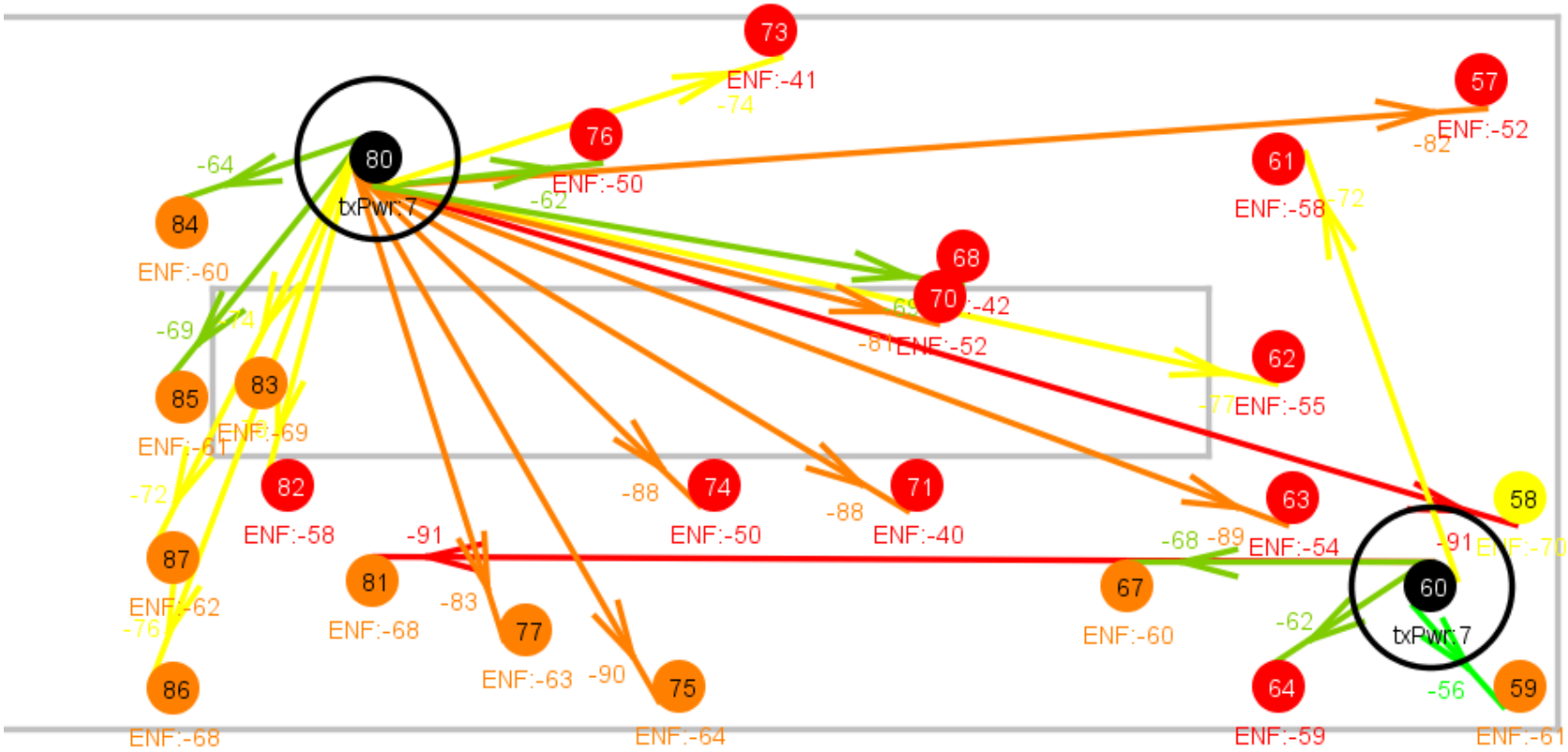
Visualization toolbox







Tx power – RSSI - ENF



w-iLab.t

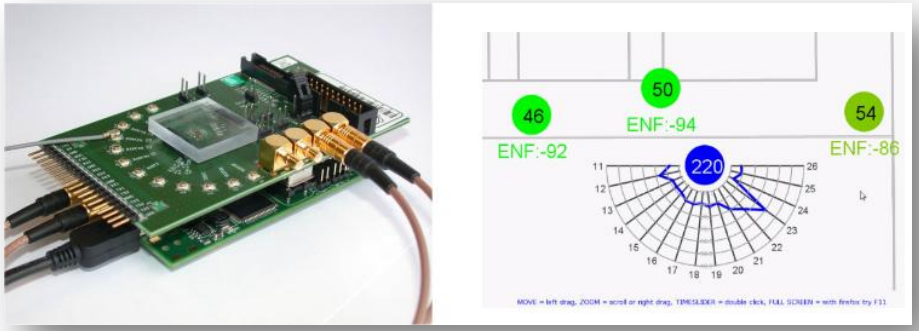
Testbed Hardware Extensions

Shielded testing environment

- shielded from outside interference
- coax connected
- variable attenuators
- emulate mobility

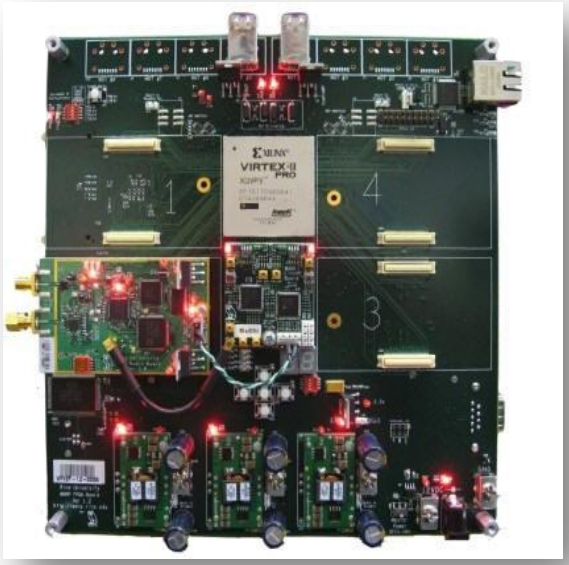
Cognitive components

- imec sensing engine



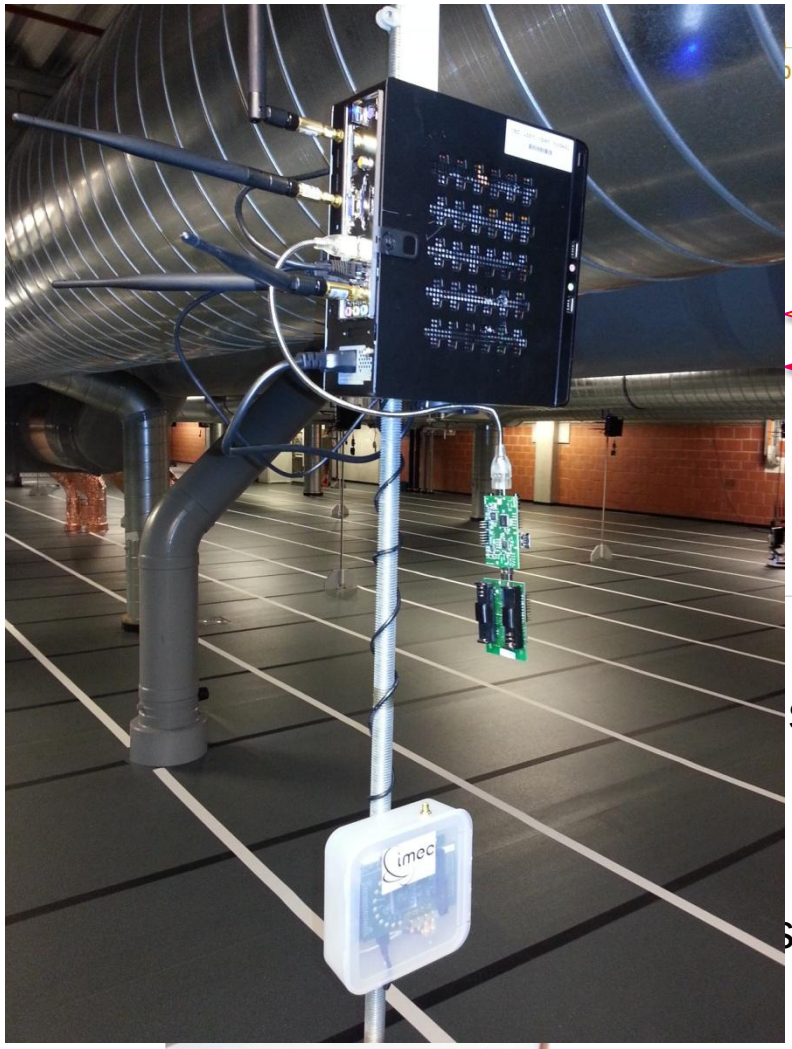
- USRP2 (SDR radios)

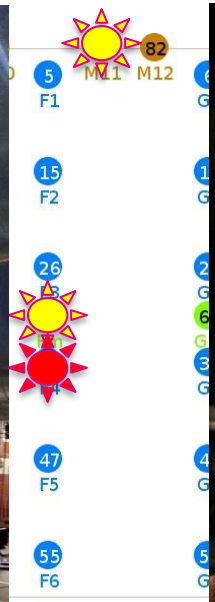
- Rice WARP



Specifications available on
www.crew-project.eu/portal

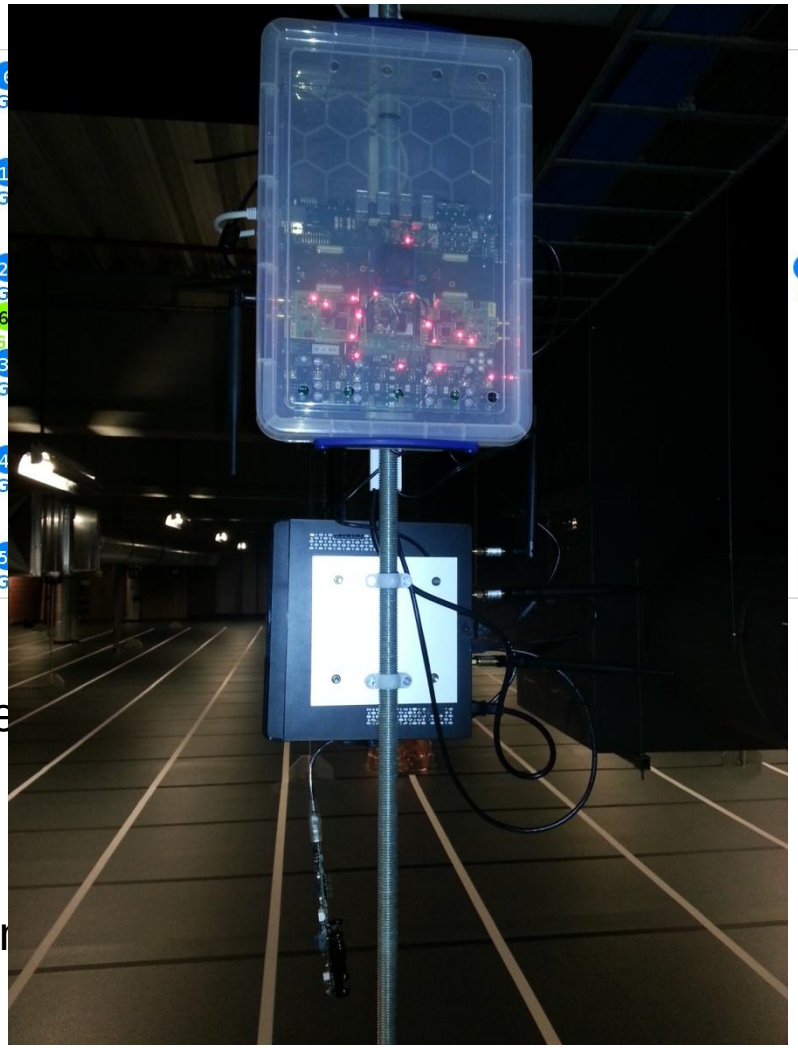
Cognitive radio extions in w-iLab.t Zwijnaarde testbed





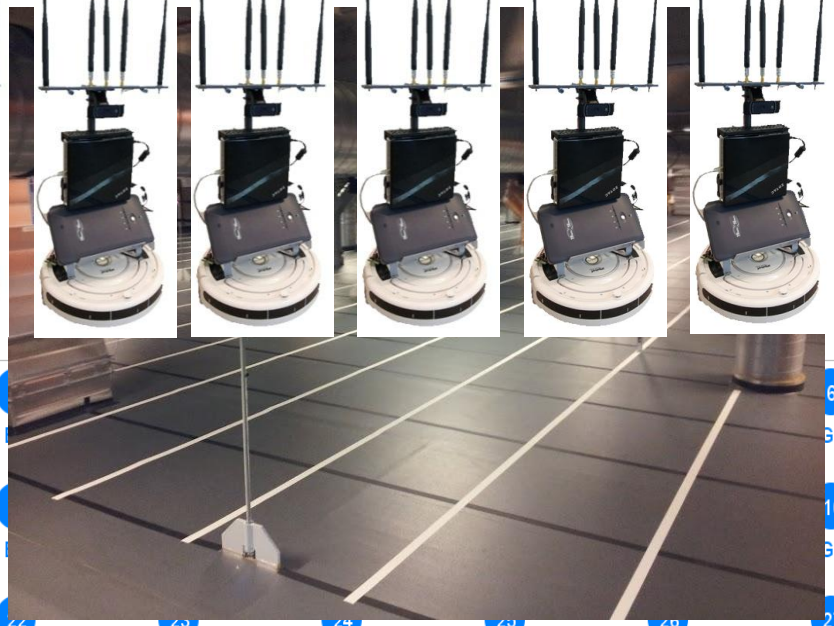
sensing e

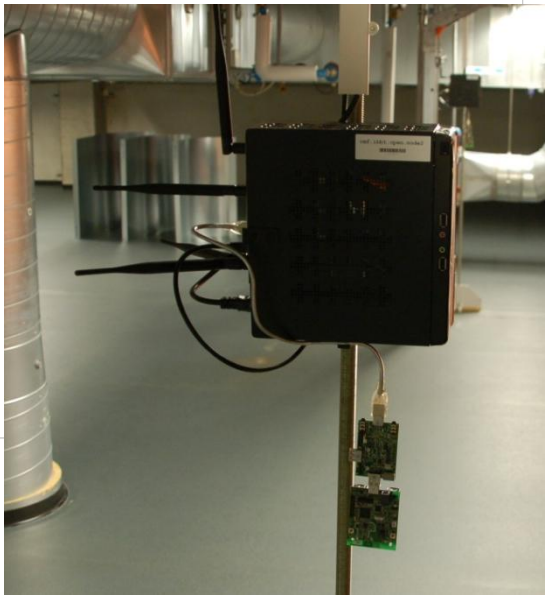
sensing er



Mobility @ w-iLab.t

Easy-to-use
Repeatable mobility!

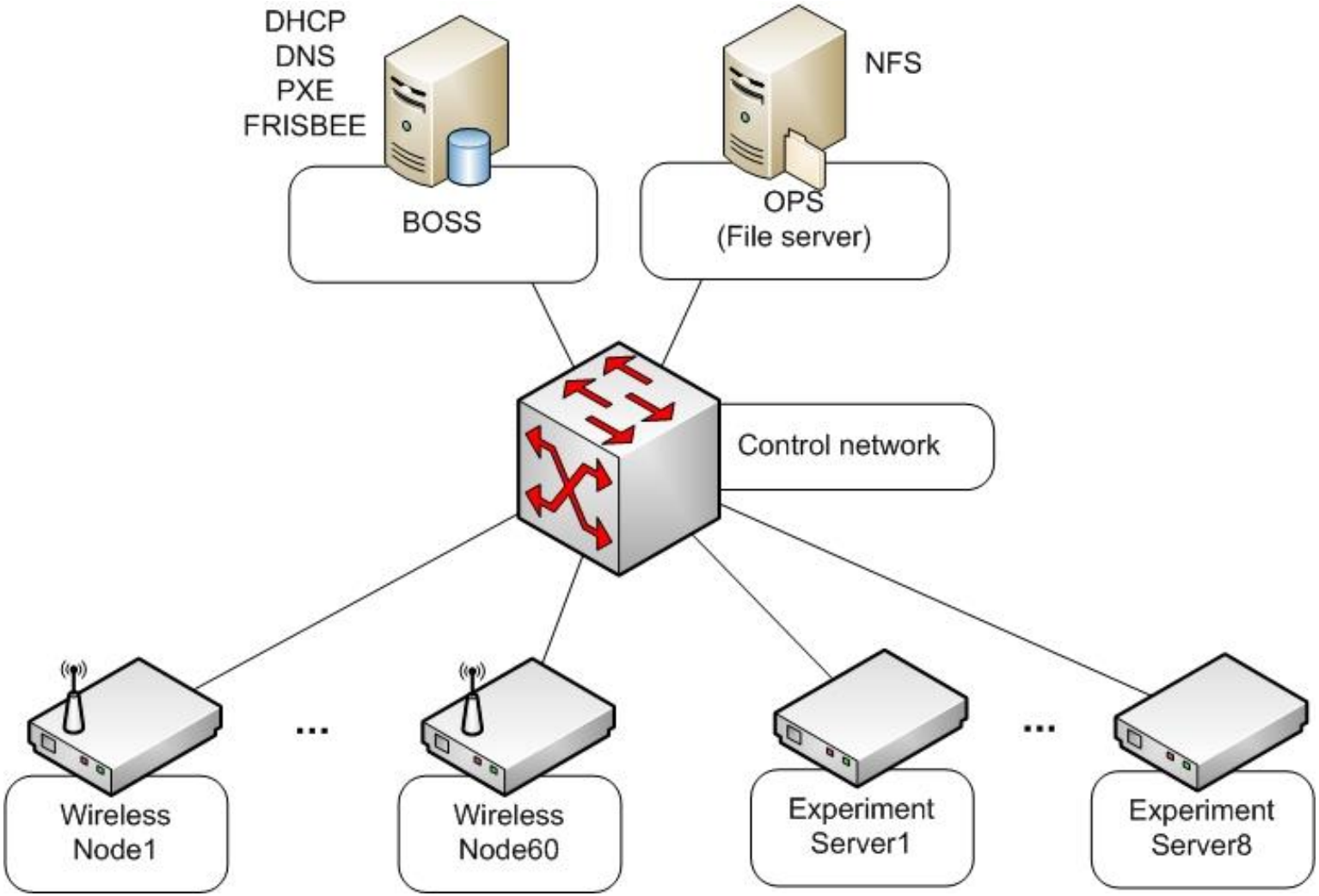




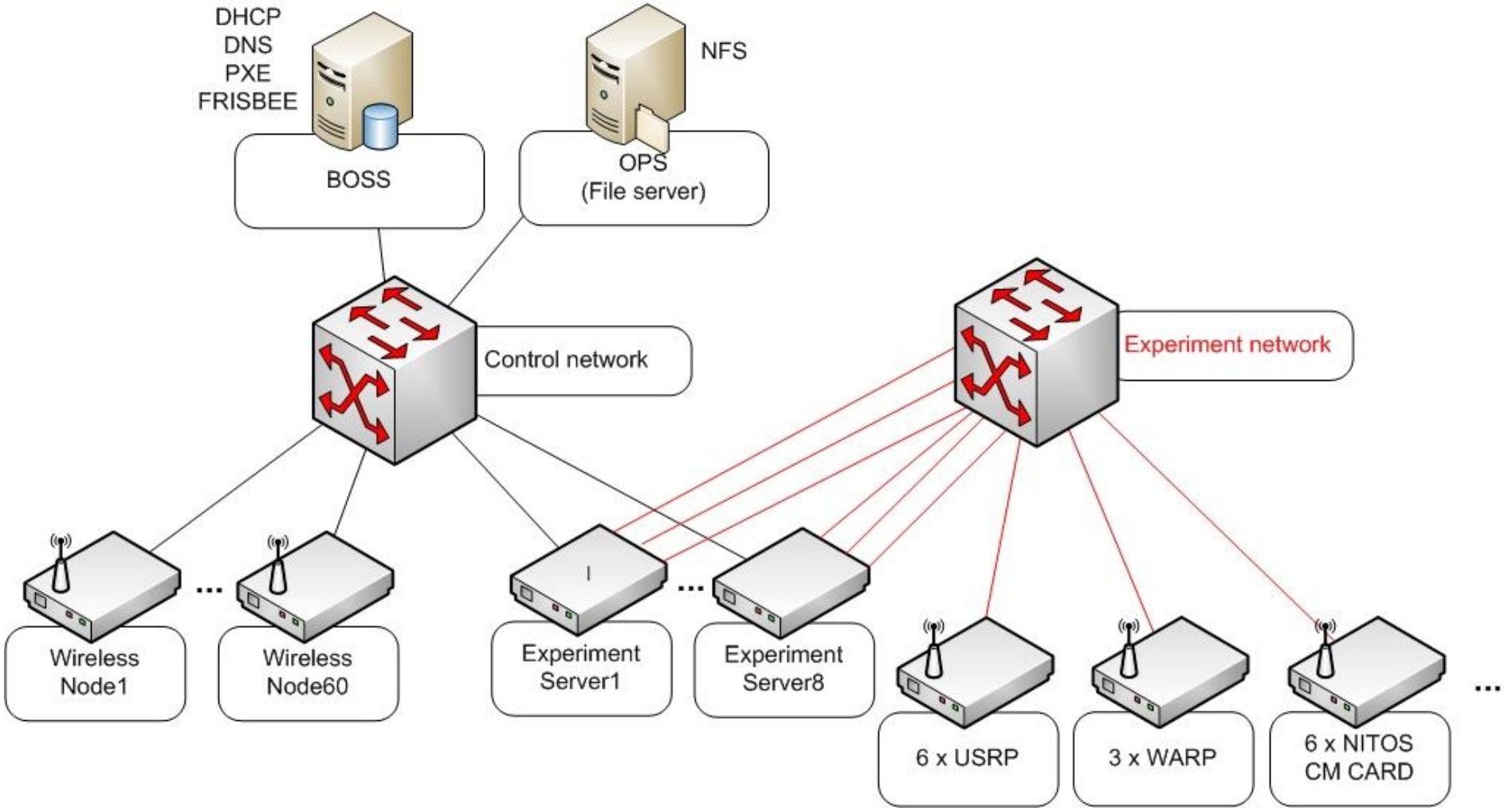
w-iLab.t

Emulab Introduction

Emulab install



Experiment Network



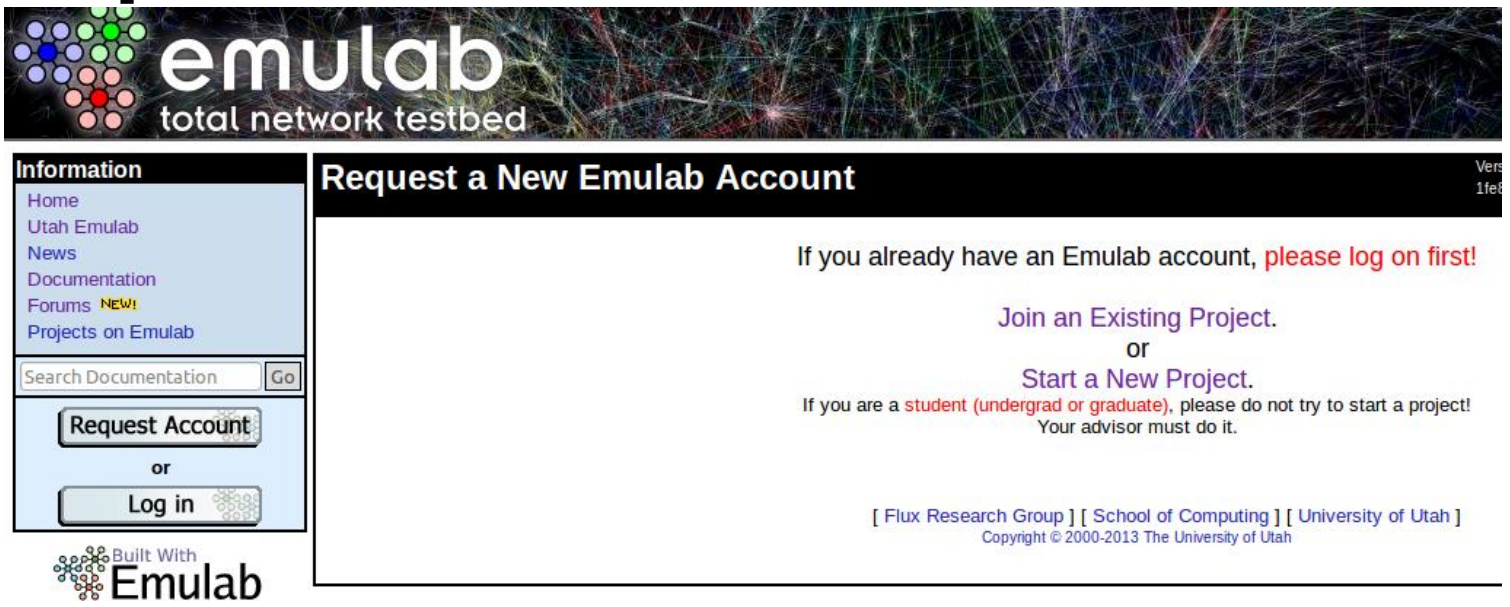
Emulab user-server (OPS)

- ops.wilab2.ilabt.iminds.be
  - Login with web-credentials
    - Or upload SSH pub key to your profile
  - NFS directories
    - /users/username (home-dir)
    - /proj/projectname (project home-dir)
    - **Accessible on all your nodes** during your experiment
  - Emulab projects are like UNIX groups
    - Everybody in same project can modify /proj/ directory

Swap in some Zotacs with the Emulab framework

First Emulab experiment

https://wilab2.ilabt.iminds.be



- Request Account
  - Start your own project or
  - Join Existing project :
    - cognitiveradio
    - GreenWeCan
    - QoCON
    - Wings

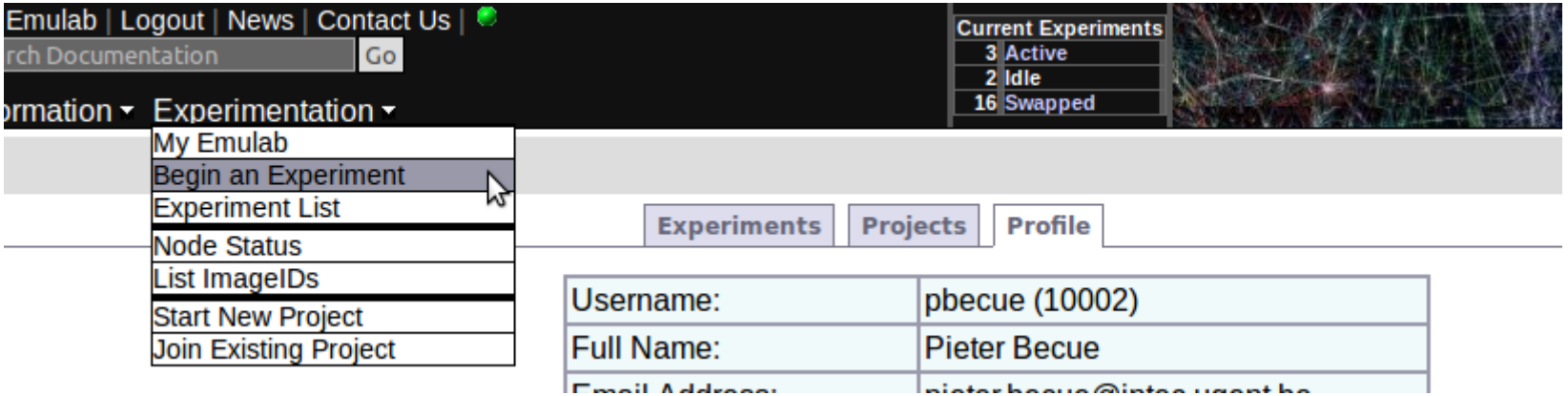
Log in

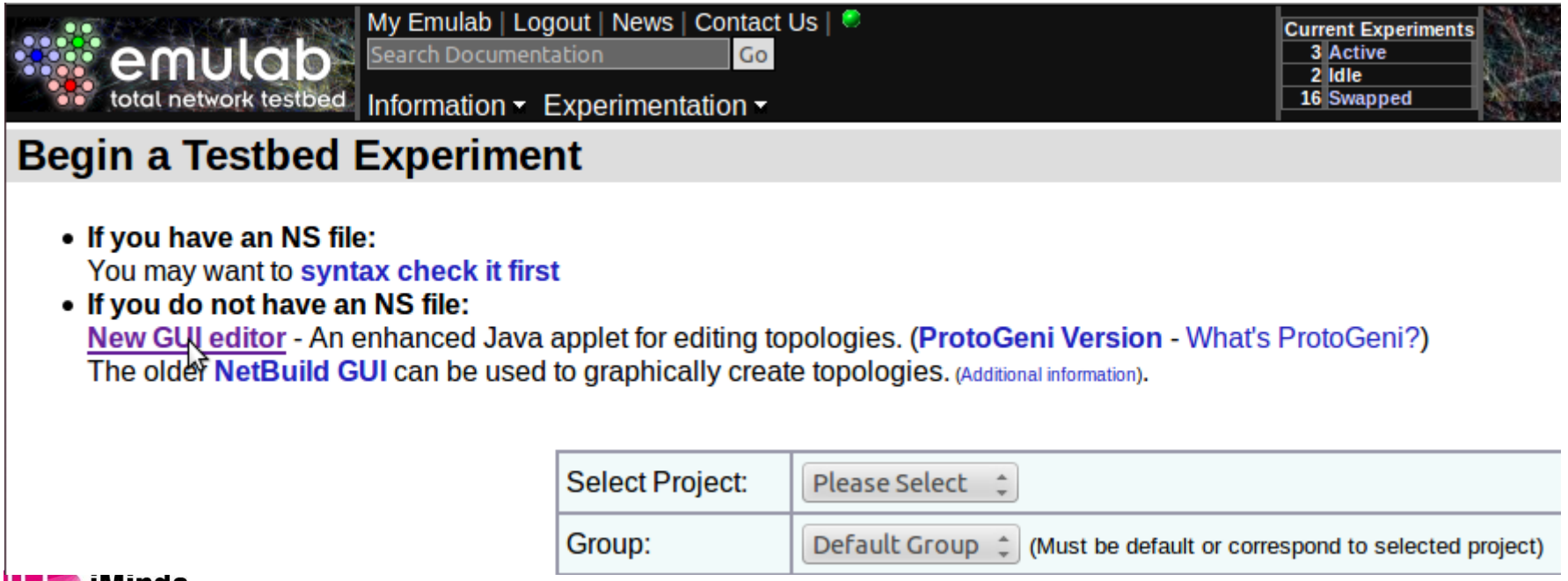
- User : crew
- Pass : training@wilab2

Create experiment

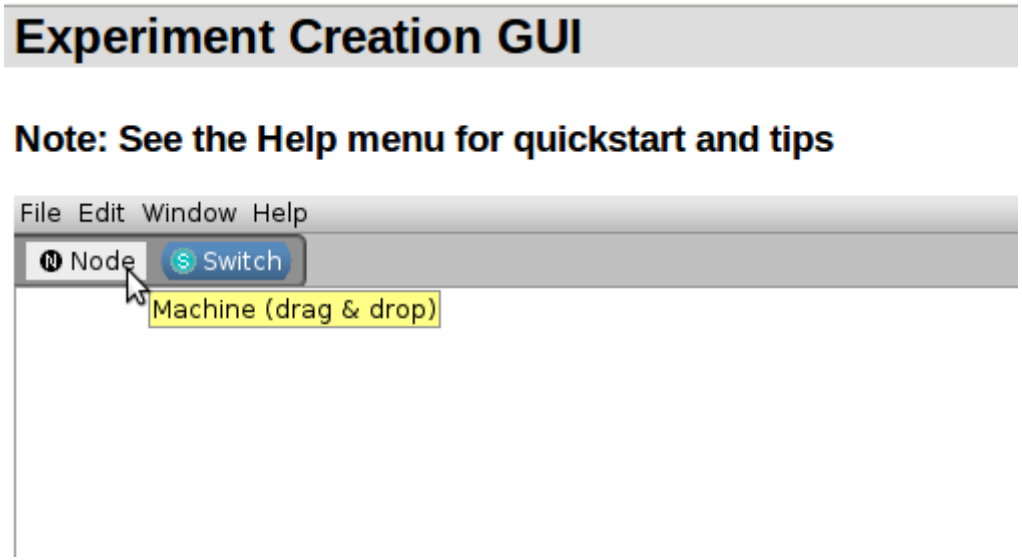
- iperf_group**X**
- X : 1 .. 10

Create a new Experiment (New GUI editor)

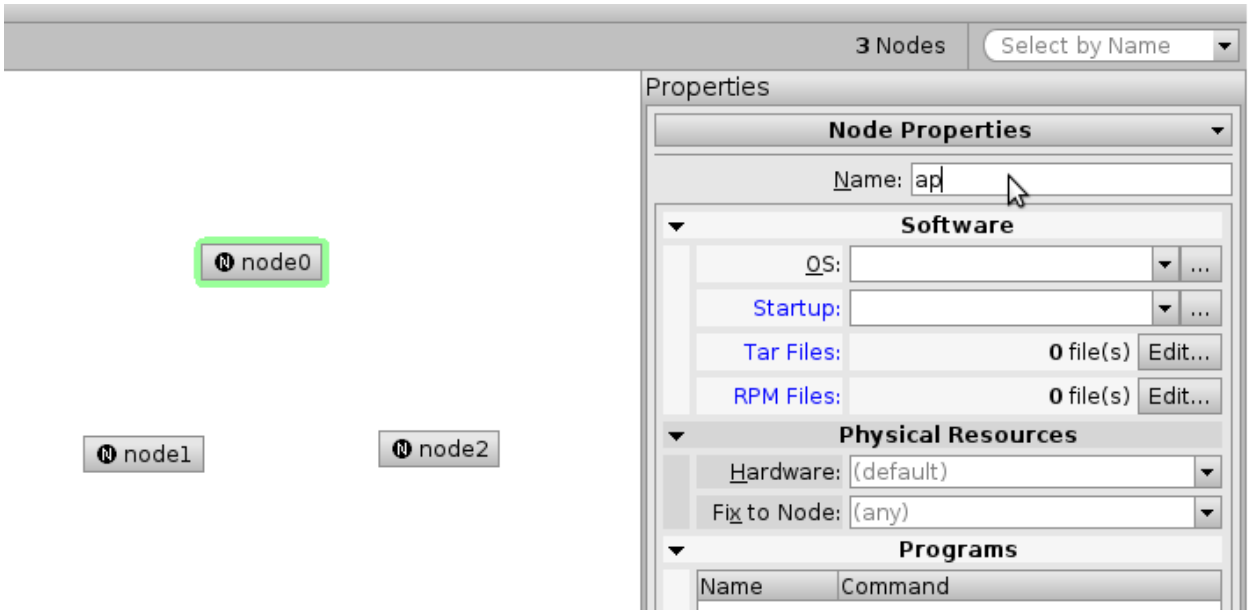




Click & drag some nodes into the white field

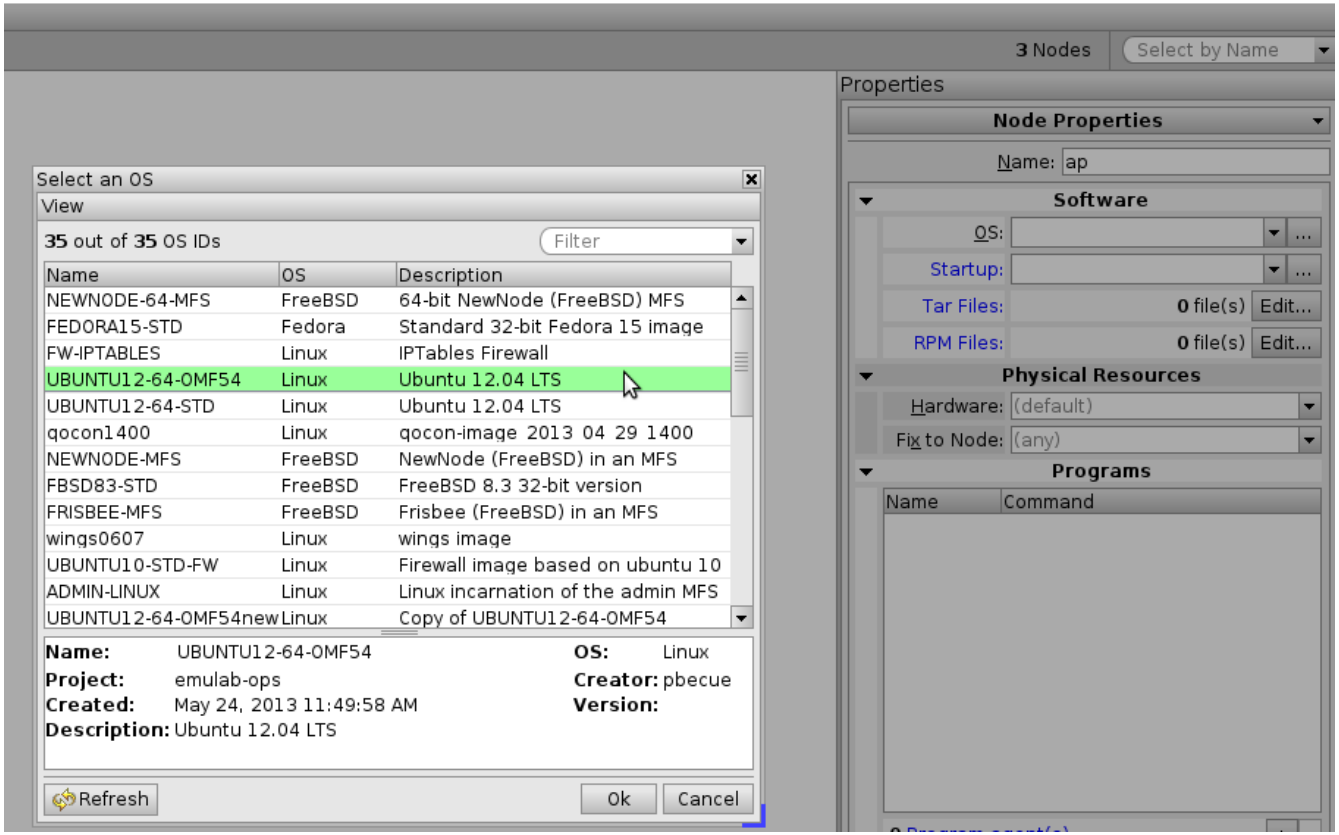


Select a node and give it a name



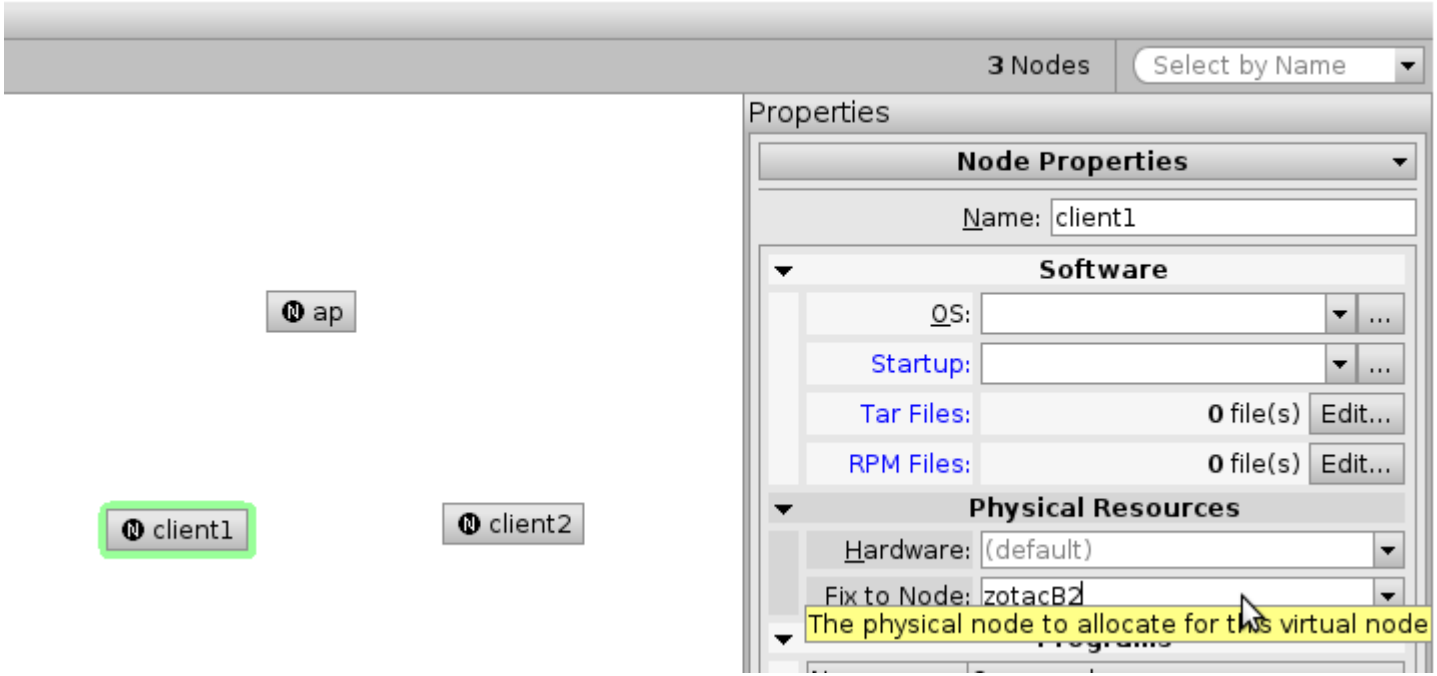
- This will be the DNS name of your node :
  - nodeName.experiment.project.wilab2.ilabt.iminds.be
- Name should be **different** from physical node ID (like zotacB2)
- **Do NOT draw any link between the nodes**

Select an OS (click on …)



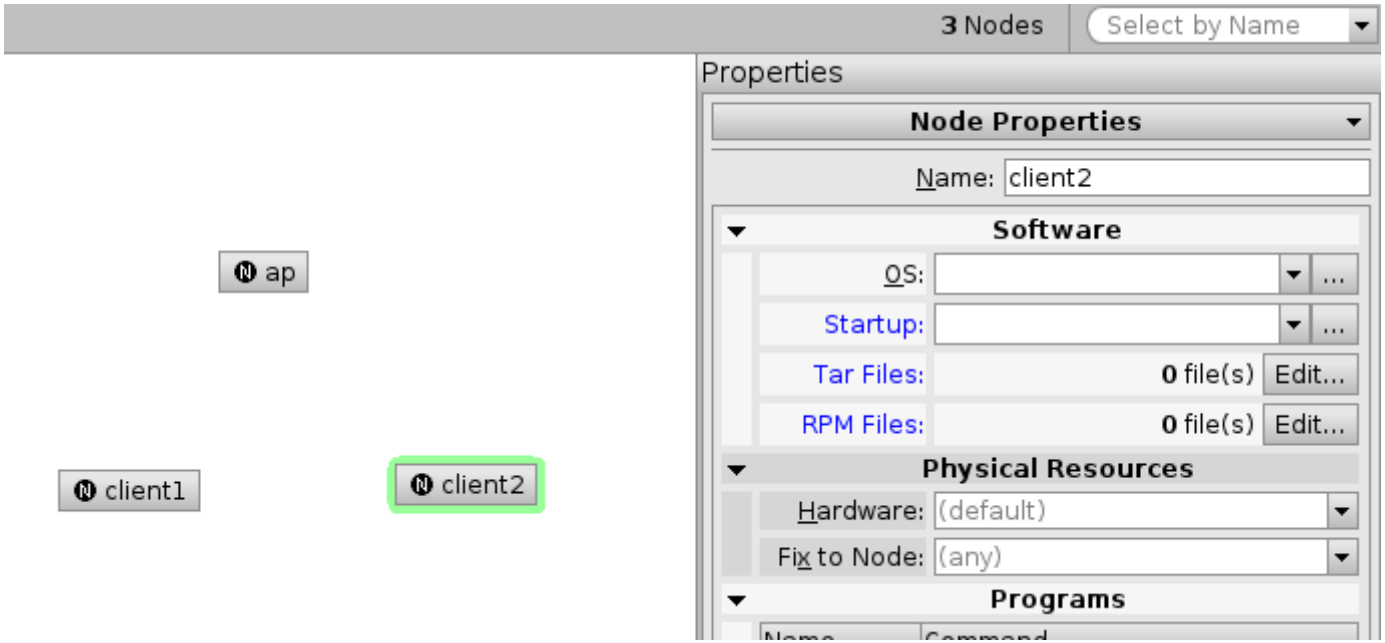
- UBUNTU12-64-OMF54 is the **default** image
- Leave the OS field **blank** if you want to use this default image, else choose one from the list.

Fill in the Fix to Node field if you want to use a specific node



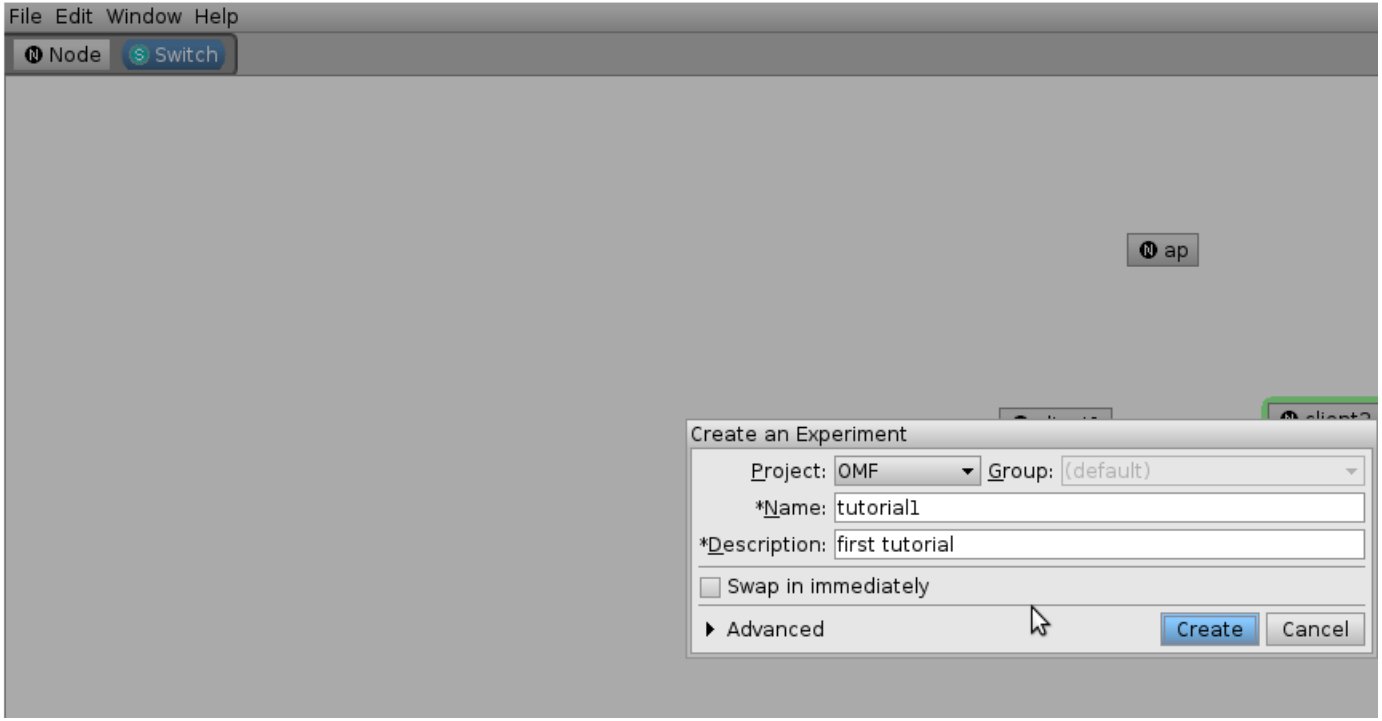
- If this field is left **blank**, Emulab will choose a **random** node for you
- See the w-iLab.t 2 map for the node ID's (e.g. zotacH4)

Leave everything blank if you want :



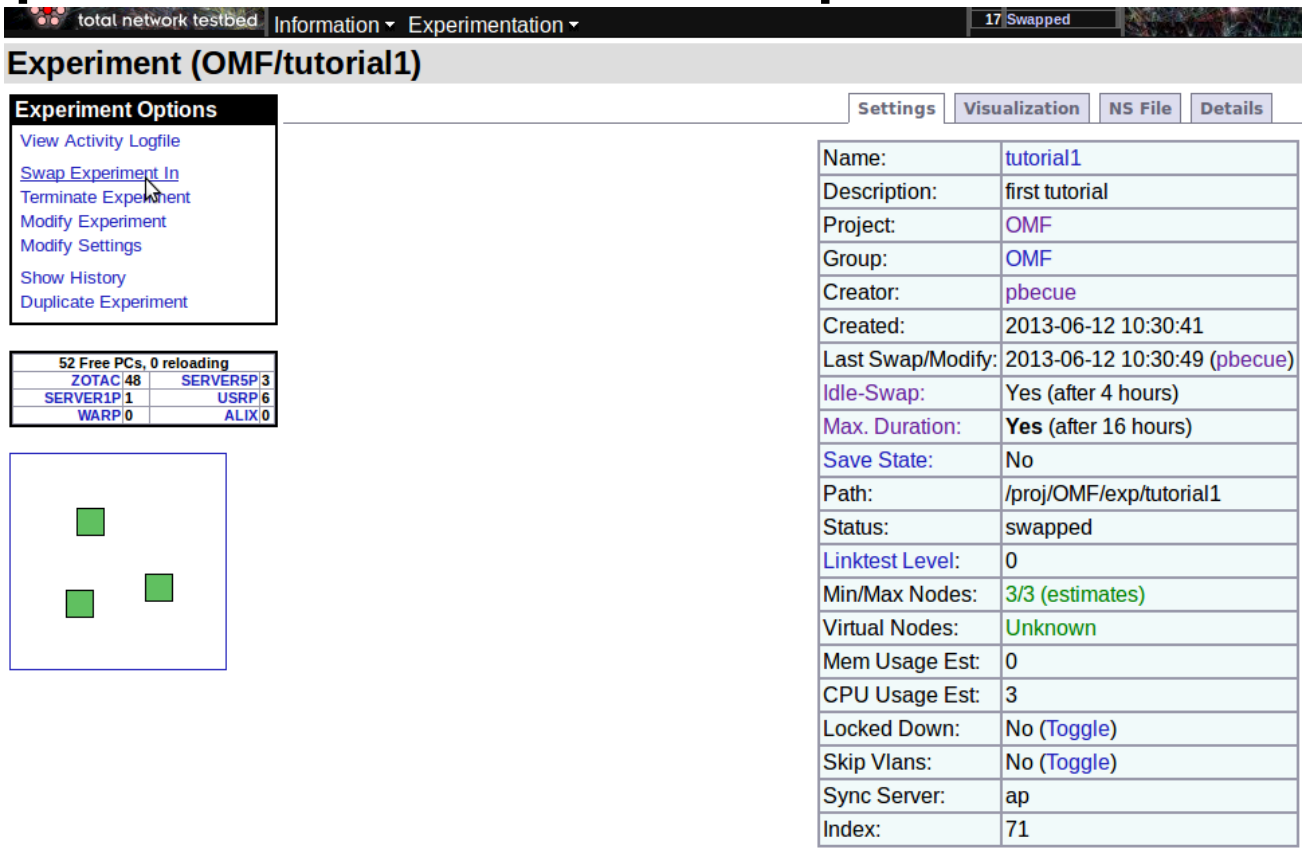
- Default OS will be loaded (UBUNTU12-64-OMF54)
- Emulab will choose random node

Click File > Create New Experiment



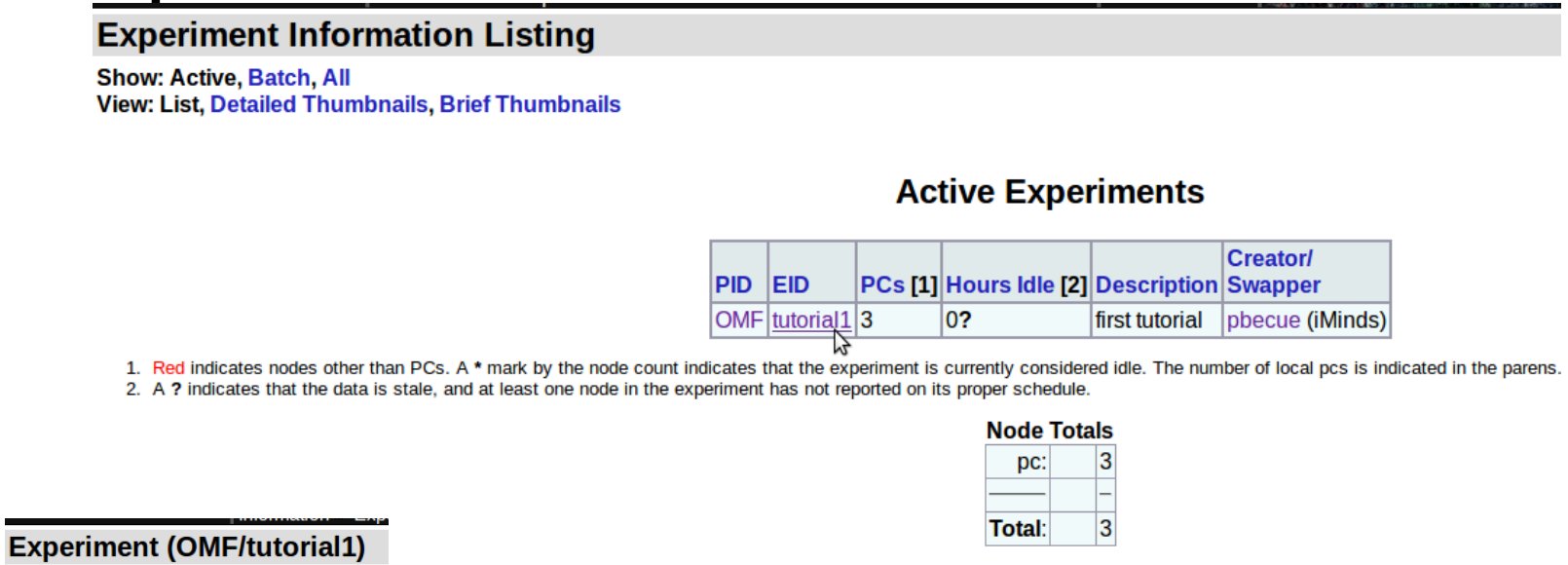
- Choose project , Name and description
- Click **Create**

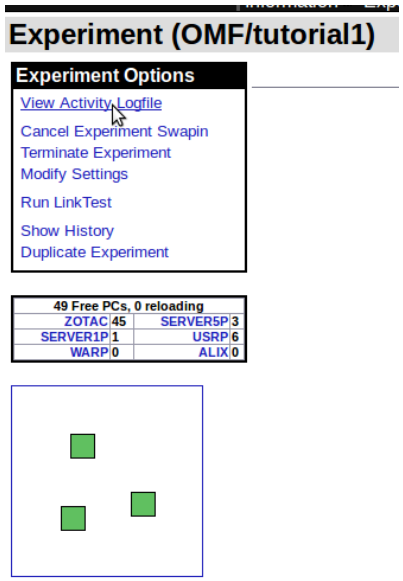
Experimentation > Experiment List



- Select your experiment
- Click **Swap Experiment in**

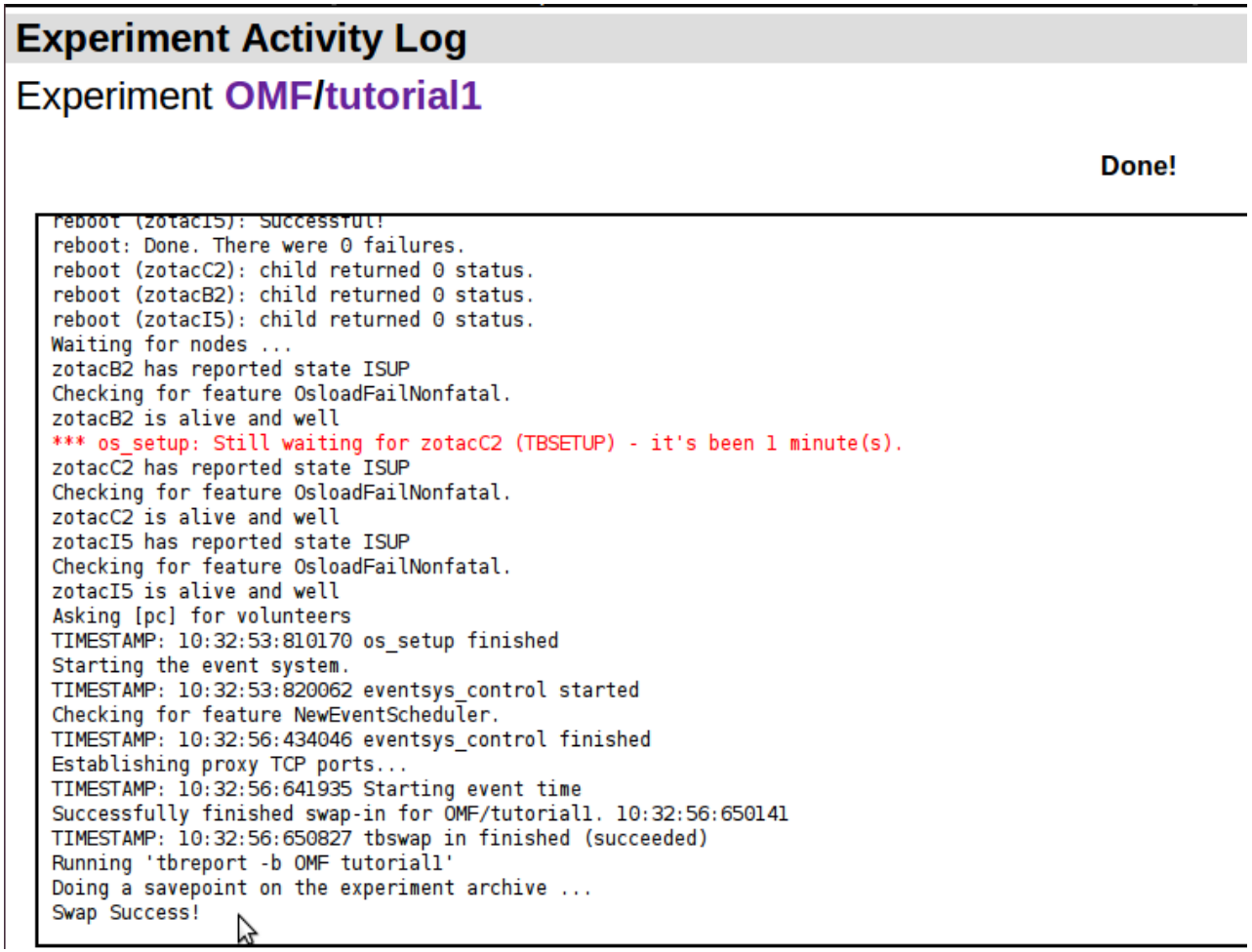
Experiment List





- **View Activity Logfile** to check swap-in/out progress
- http://www.wilab2.ilabt.iminds.be/reservation

If everything goes well : **Swap Success**



Experiment Activity Log

Experiment **OMF/tutorial1**

Done!

```
reboot (zotacI5): Successful!
reboot: Done. There were 0 failures.
reboot (zotacC2): child returned 0 status.
reboot (zotacB2): child returned 0 status.
reboot (zotacI5): child returned 0 status.
Waiting for nodes ...
zotacB2 has reported state ISUP
Checking for feature OsloadFailNonfatal.
zotacB2 is alive and well
*** os_setup: Still waiting for zotacC2 (TBSETUP) - it's been 1 minute(s).
zotacC2 has reported state ISUP
Checking for feature OsloadFailNonfatal.
zotacC2 is alive and well
zotacI5 has reported state ISUP
Checking for feature OsloadFailNonfatal.
zotacI5 is alive and well
Asking [pc] for volunteers
TIMESTAMP: 10:32:53:810170 os_setup finished
Starting the event system.
TIMESTAMP: 10:32:53:820062 eventsys_control started
Checking for feature NewEventScheduler.
TIMESTAMP: 10:32:56:434046 eventsys_control finished
Establishing proxy TCP ports...
TIMESTAMP: 10:32:56:641935 Starting event time
Successfully finished swap-in for OMF/tutorial1. 10:32:56:650141
TIMESTAMP: 10:32:56:650827 tbswap in finished (succeeded)
Running 'tbreport -b OMF tutorial1'
Doing a savepoint on the experiment archive ...
Swap Success!
```

Click **Details** to see DNS names and physical node mapping

Settings | Visualization | NS File | Details

```
Experiment: OMF/tutorial1
State: active

Virtual Node Info:
ID              Type         OS              Qualified Name
--------------- ------------ --------------- --------------------
ap              pc           UBUNTU12-64-OMF54 ap.tutorial1.OMF.wilab2.ilabt.iminds.be
client1 (zotacB2) ZOTAC        UBUNTU12-64-OMF54 client1.tutorial1.OMF.wilab2.ilabt.iminds.be
client2         pc                           client2.tutorial1.OMF.wilab2.ilabt.iminds.be

Physical Node Mapping:
ID              Type         OS              Physical
--------------- ------------ --------------- ------------
ap              ZOTAC        UBUNTU12-64-OMF54 zotacI5
client1         ZOTAC        UBUNTU12-64-OMF54 zotacB2
client2         ZOTAC        UBUNTU12-64-OMF54 zotacC2
```

Show Events Save to File

Reserved Nodes

Node ID	Name	Type	Default OSID	Node Status	Hours Idle[1]	Startup Status[2]	SSH	Console	Log
zotacB2	client1	ZOTAC	UBUNTU12-64-OMF54	up	0	none			
zotacC2	client2	ZOTAC	UBUNTU12-64-OMF54	up	0	none			

Click on a node you want to check

Reserved Nodes

Node ID	Name	Type	Default OSID	Node Status	Hours Idle[1]	Startup Status[2]	SSH	Console	Log
zotacB2	client1	ZOTAC	UBUNTU12-64-OMF54	up	0	none			
zotacC2	client2	ZOTAC	UBUNTU12-64-OMF54	up	0	none			
zotacI5	ap	ZOTAC	UBUNTU12-64-OMF54	up	0	none			

- Show boot log, create disk image, Reboot node

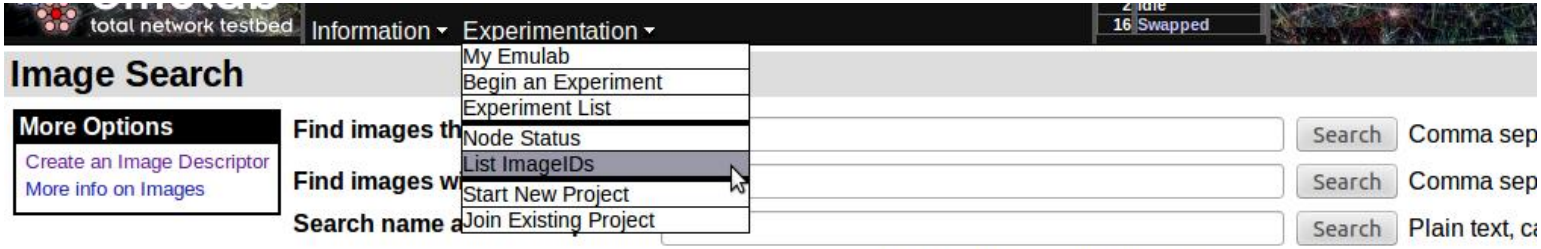
Node zotacI5

Node Options

- SSH to node (howto)
- Edit Node Info
- Update Node
- Reboot Node
- Show Boot Log
- Create a Disk Image
- Show Node Log
- Show Node History
- Set Node Location
- Update Power State
- Modify Node Attributes

Node ID:	zotacI5
Virtual Name:	ap
Project:	OMF
Experiment:	tutorial1
Node Type:	ZOTAC
Def Boot OS:	UBUNTU12-64-OMF54
EventState:	ISUP (2013-06-12 10:32:44)
Operating Mode:	NORMALv2 (2013-06-12 10:31:42)
AllocState:	RES_READY (2013-06-12 10:32:48)
Last Activity:	2013-06-12 10:32:34

List image ID's to see all images you can use in your project



There are 16 matching images.

Image	PID	Description
DEB60-STD	emulab-ops	Debian Squeeze 32bit
DEB60_64-STD	emulab-ops	Debian Squeeze 64bit
FBSD83-64-STD	emulab-ops	FreeBSD 8.3 64-bit version
FBSD83-STD	emulab-ops	FreeBSD 8.3 32-bit version
FEDORA15-64-STD	emulab-ops	Standard 64-bit Fedora 15 image
FEDORA15-OPENVZ-STD	emulab-ops	Fedora15 with OpenVZ
FEDORA15-STD	emulab-ops	Standard 32-bit Fedora 15 image
openWRT	testbed	openWRT transfered from old setup
qocon1400	testbed	qocon-image_2013_04_29_1400
UBUNTU10-STD-FW	emulab-ops	Firewall image based on ubuntu 10
UBUNTU12-64-OMF54	emulab-ops	Ubuntu 12.04 LTS with OMF5.4 support
UBUNTU12-64-OMF54new	testbed	Copy of UBUNTU12-64-OMF54
UBUNTU12-64-STD	emulab-ops	Ubuntu 12.04 LTS
UBUNTU12-OMF54-USRP	emulab-ops	UBUNTU12-64-OMF54 with USRP software
WIN7SP1-STD	emulab-ops	Windows 7 SP1 - 32bit (no activation, 30days)
wings0607	testbed	wings image

w-iLab.t

OMF Introduction

OMF Tutorial

Thierry Rakotoarivelo

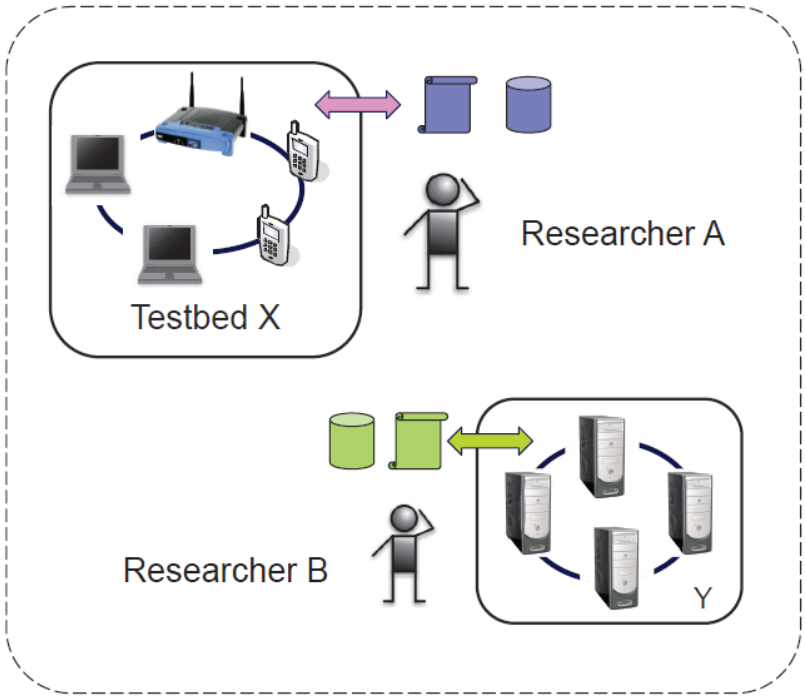
Australian Government
Department of Broadband, Communications and the Digital Economy
Australian Research Council

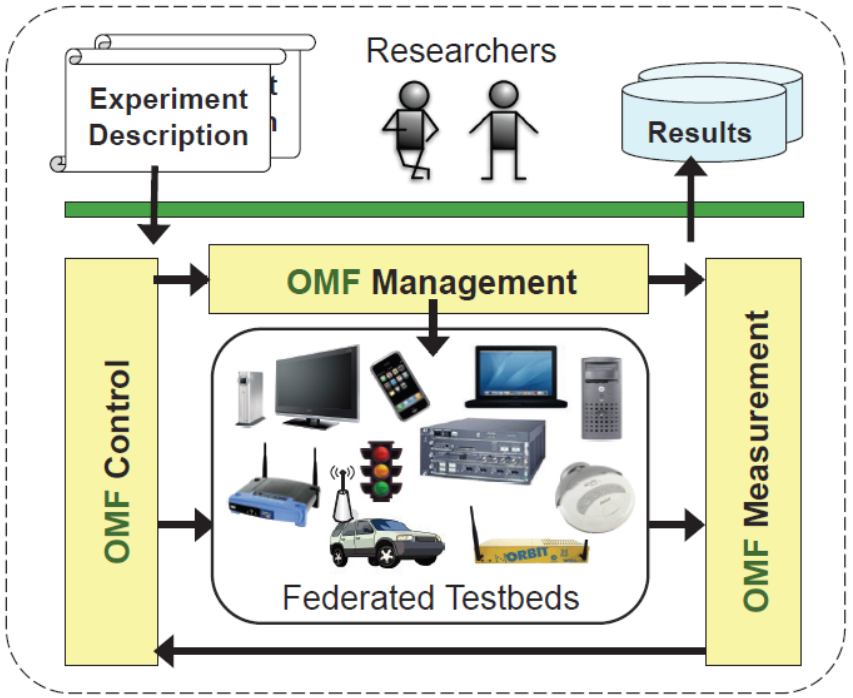
NICTA Funding and Supporting Members and Partners

iMinds
ILAB.T

The Problem and Our approach

NICTA





Software & Management | Resources

OMF | Resources

Support & share different resources

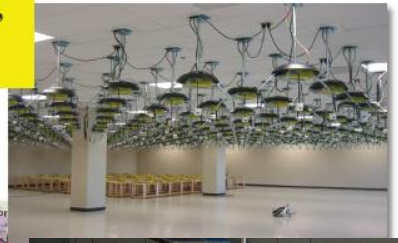
Federation of different testbeds

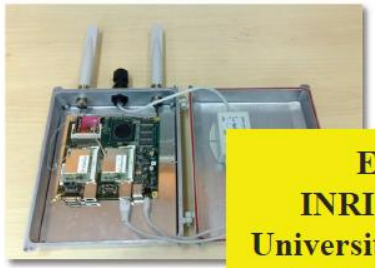
iMinds ILAB.T

OMF deployment worldwide

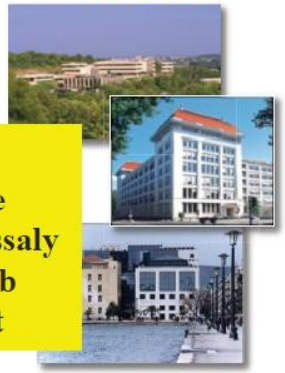
NICTA

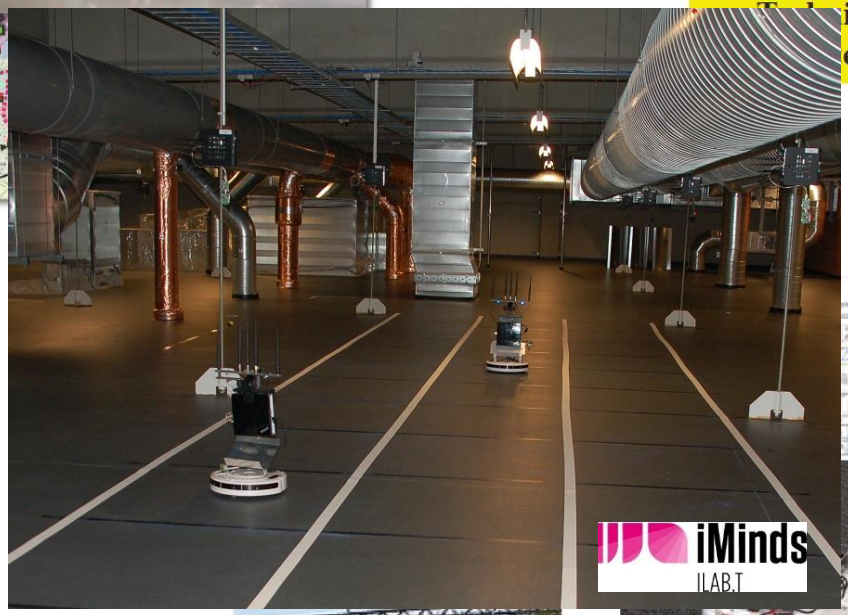
Rutgers University, New Jersey





Europe
INRIA, France
University of Thessaly
Technicolor Lab
Alcatel-Lucent



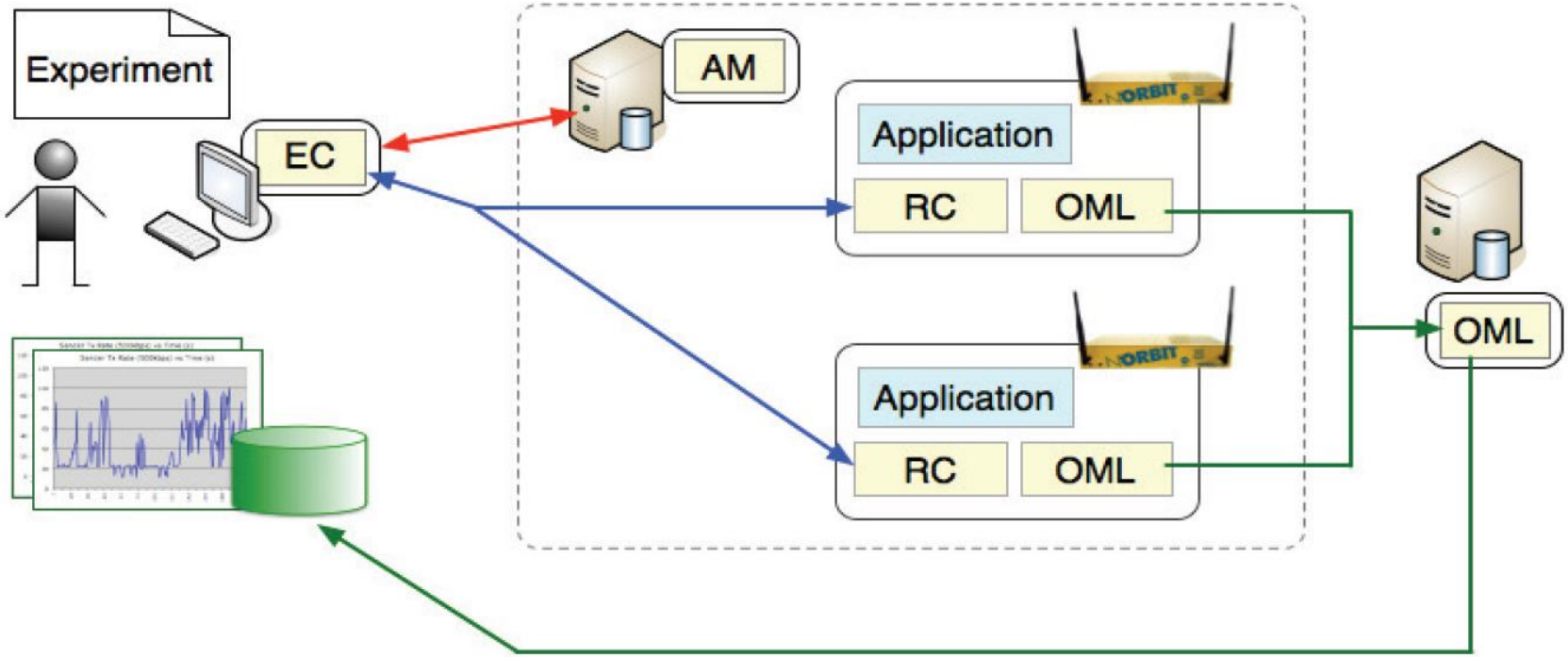


PlanetLab

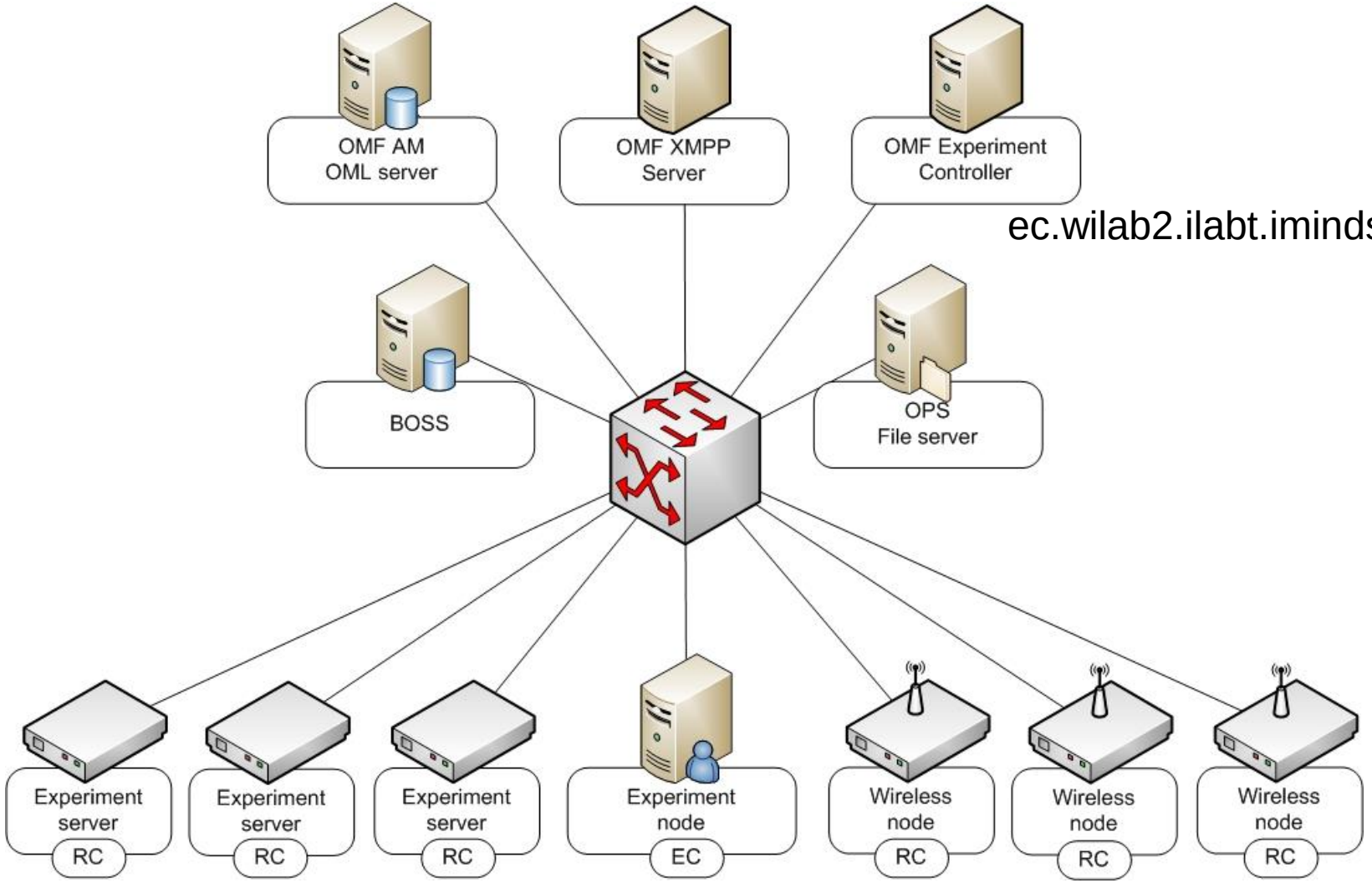
NICTA, Sydney Bridge Deployment

iMinds
ILAB.T

How it works from a user's perspective?



OMF Setup @ w-iLab.t Zwijnaarde

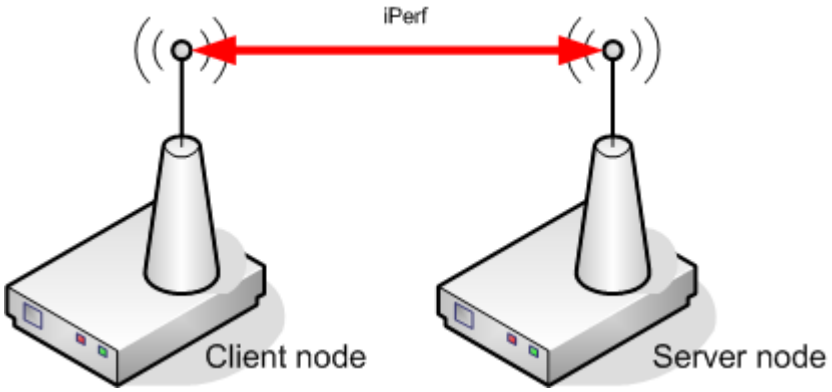


w-iLab.t

OMF Tutorial – Step by step

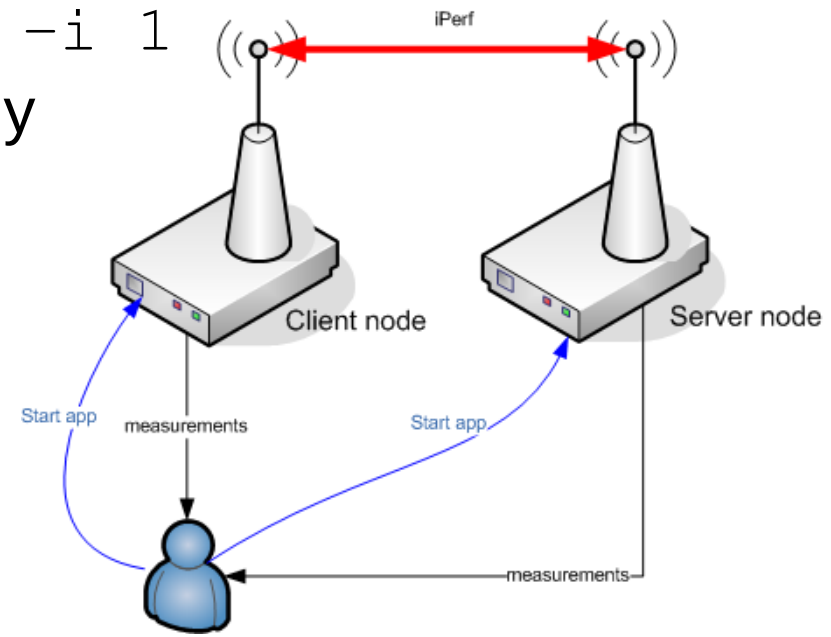
Running basic OMF Experiments: demo case

Goal: Test throughput of Wi-Fi using iPerf



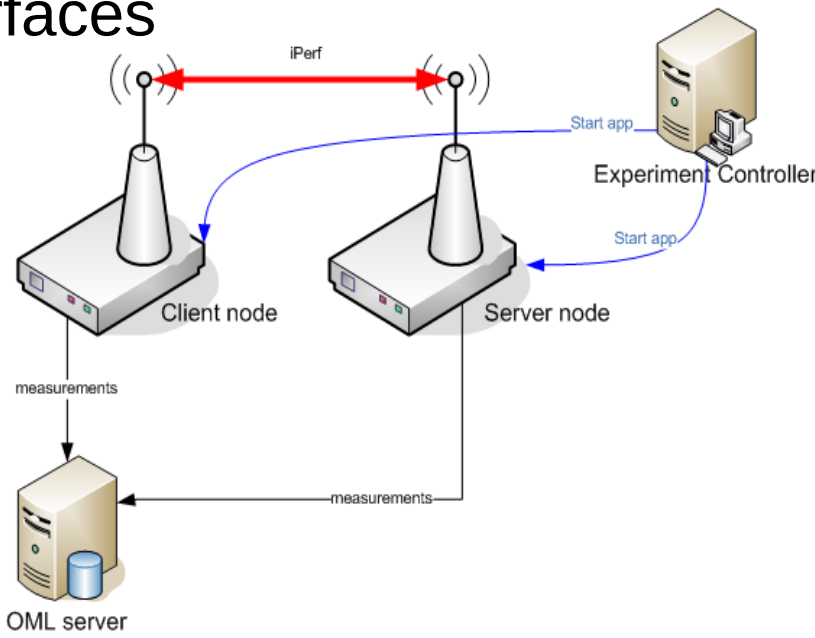
Running basic OMF Experiments: manual approach

1. ssh to the server
2. set up wireless interface
3. start serverside application with `iperf -s -u`
4. ssh to the client
5. set up wireless interface and connect it to the server
6. start clientside application with: `iperf -c [serverip] -u -b 1000M -i 1`
7. collect measurements manually



Running basic OMF Experiments: the OMF way

1. ssh to the experiment controller
2. write an **application definition**
   - where can OMF find iPerf?
   - what are valid commandline arguments?
   - what is the output of iPerf? (wrapper/native)
3. write an **experiment description**
   - configuration of wireless interfaces
   - which nodes?
   - timeline of the experiment
4. execute the experiment
5. read measurements in database



Running basic OMF Experiments

Experiment Definition

Application Definition

Application

`/usr/bin/iperf`

Using a wrapper

Experiment Definition

Application Definition

On the resource

Wrapper script
(parses output of iPerf for measurements)

Application

`/usr/bin/iperf`

Running basic OMF Experiments

Experiment Definition

```
defGroup("ap","ap.exp1.crew.wilab2.ilabt.iminds.be")
    <config wireless interface>
    <add application(s)>
defGroup("client","client.exp1.crew.wilab2...")
    <config wireless interface>
    <add applications(s)>
```

```
defApplication('iperfwrap', 'iperfwrap')
```

```
onEvent(:ALL_UP_AND_INSTALLED) do |event|
  info "iPerf experiment - wait for interfaces to come
up"
  wait 5
  group("ap").startApplications
  info "Starting stream 1 server..."
  wait 2
  info "Starting stream 1 client..."
  group("client").startApplications
  wait 60
  group("client").stopApplications
  group("ap").stopApplications
  Experiment.done
end
```

Application Definition

Wrapper script
(parses output of iPerf for measurements)

Application

/usr/bin/iperf

Running basic OMF Experiments

Experiment Definition

Application Definition

```
defApplication('iperfwrap', 'iperfwrap') do |app|
  app.path = File.join(File.dirname(__FILE__), 'iperfwrap.rb')
  app.appPackage = "iperfwrap.tar"

  app.shortDescription = "iPerf wrapper"
  app.description = "Simple iPerf wrapper with OML logging"

  app.defProperty('server', 'Act as server (true/false)', 's' , …)
<more properties>

  # the measurement definition
  app.defMeasurement('iperfmp') do |mp|
<metrics: i.e. the fields in every record in the database>
  end
end
```

Wrapper script
(parses output of iPerf for measurements)

Application

`/usr/bin/iperf`

Running basic OMF Experiments

Experiment Definition

Application Definition

Wrapper script

(parses output of iPerf for measurements)

```
require "/usr/share/liboml2-dev/oml4r.rb"

APPPATH = "/usr/bin/iperf"

class MPStat < OML4R::MPBase
        name :iperfmp
        param :hostname
        param :interval_begin, :type => :double #sec
        param :interval_end, :type => :double #sec
        <other metric definitions>
end

<logic to pass commandline args to iperf>
<logic to parse output and generate OML>
```

Run it!

- ssh to ec.wilab2.ilabt.iminds.be (login on sheets)

- **`cd/users/crew/`**
- **`mkdir groupX`**
- **`cp iperf/* groupX/`**

- modify resource names !
- modify mode, channel and essid (see sheets)
- run the experiment using
  `omf exec iperfED.rb`

- See measurements on http://am.wilab2.ilabt.iminds.be/phppgadmin

The w-iLab.t testbed

details:
http://ilabt.iminds.be/

vincent.sercu@intec.ugent.be
pieter.becue@intec.ugent.be
bart.jooris@intec.ugent.be
stefan.bouckaert@iminds.be
www.ibcn.intec.ugent.be – www.iminds.be