IBBT w-iLab.t OMF Workshop

w-iLab.t workshop

June 4th 2012

Bart Jooris – Vincent Sercu – Pieter Becue
Stefan Bouckaert

Agenda

- OMF introduction
- Sample Experiments :
  - Hello World
  - WiFi experiment (ping)
  - (Run OMF sensor experiments on w-iLab.t Zwijnaarde)
- Questions

w-iLab.t OMF workshop

OMF introduction

connect.innovate.create

NICTA

OMF Tutorial

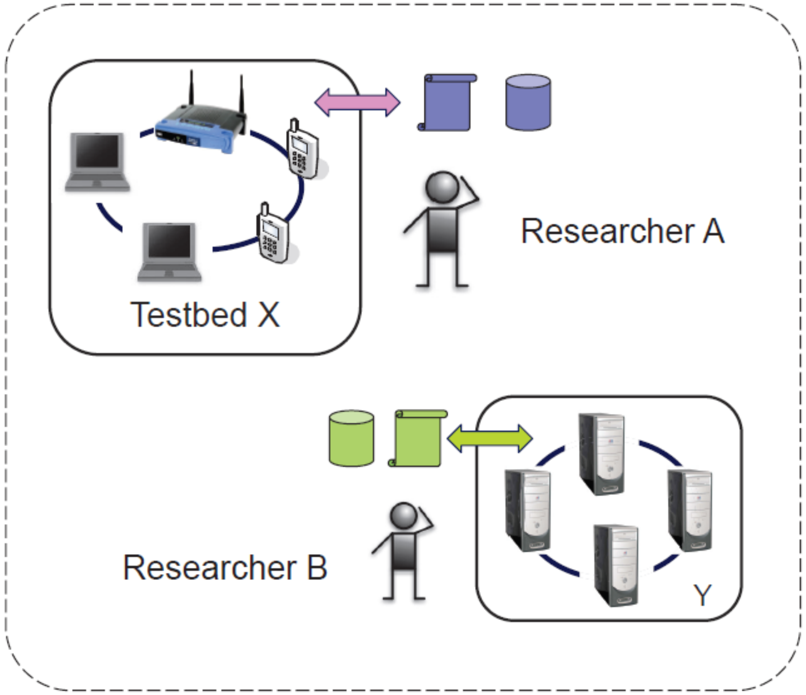
Thierry Rakotoarivelo

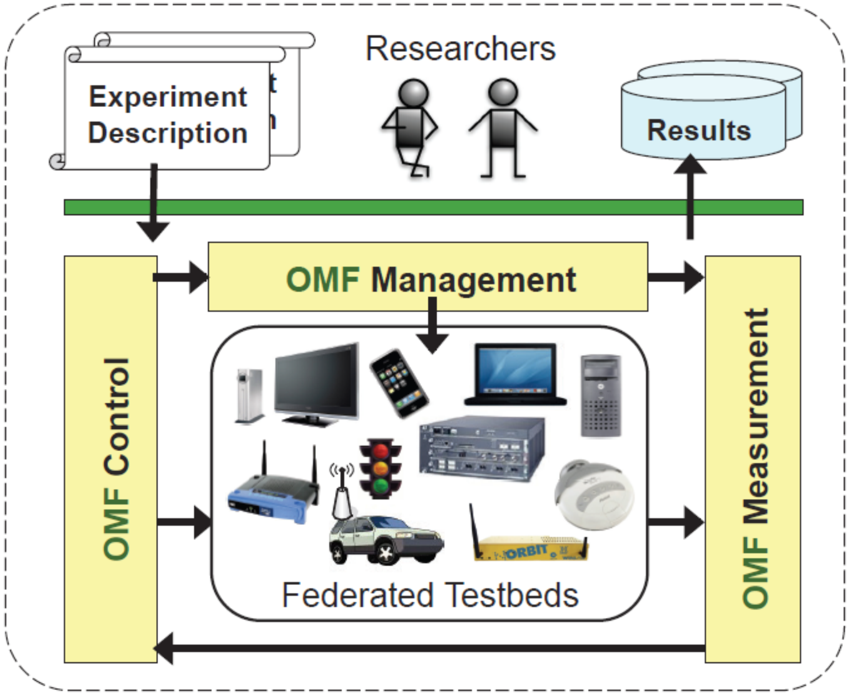
Australian Government
Department of Broadband, Communications and the Digital Economy
Australian Research Council

NICTA Funding and Supporting Members and Partners

ibbt
connect.innovate.create

The Problem and Our approach





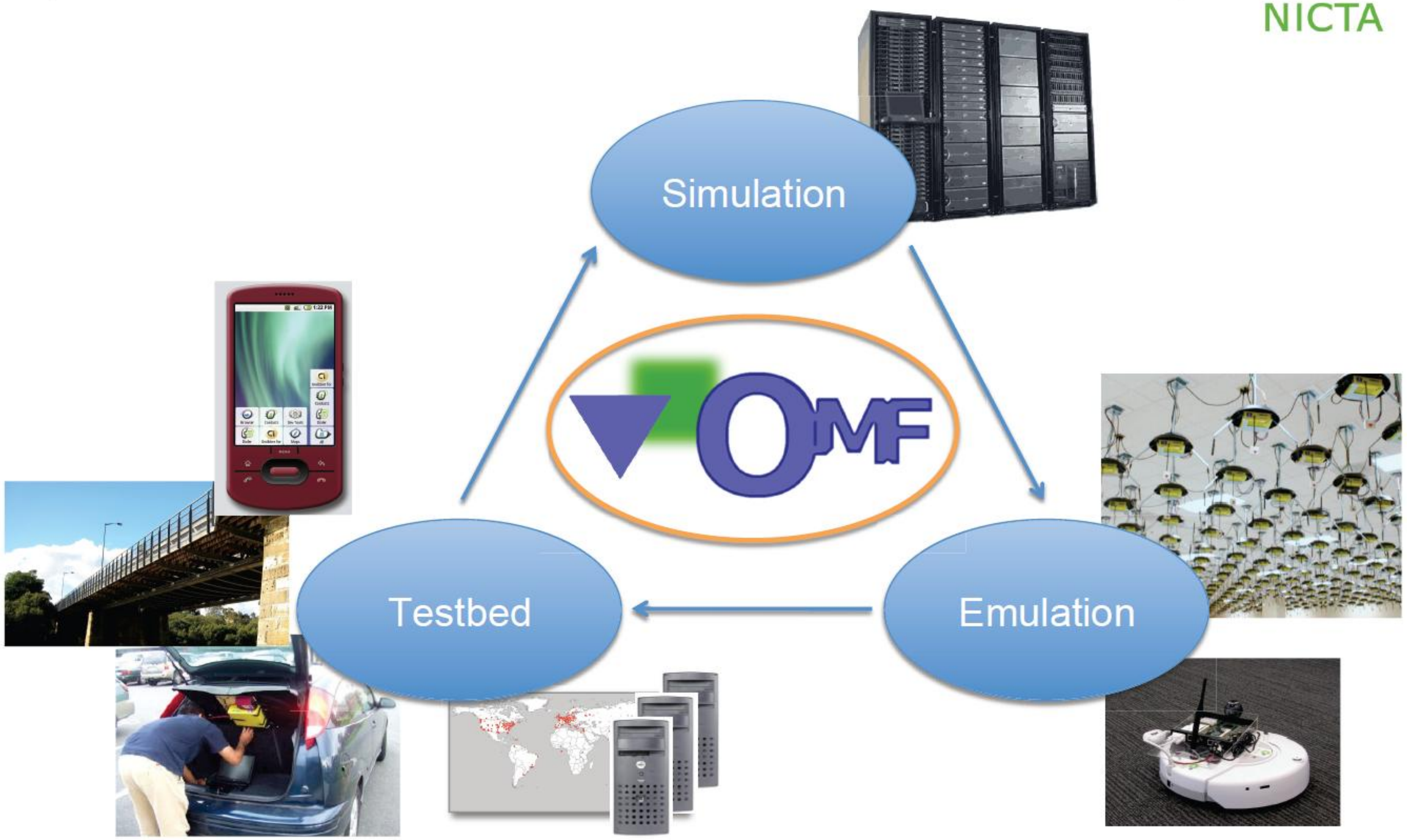
Software & Management	Resources

OMF	Resources

Support & share different resources

Federation of different testbeds

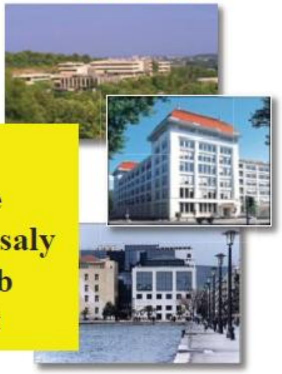
Full Experiment Cycle



OMF deployment worldwide

Rutgers University, New Jersey

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

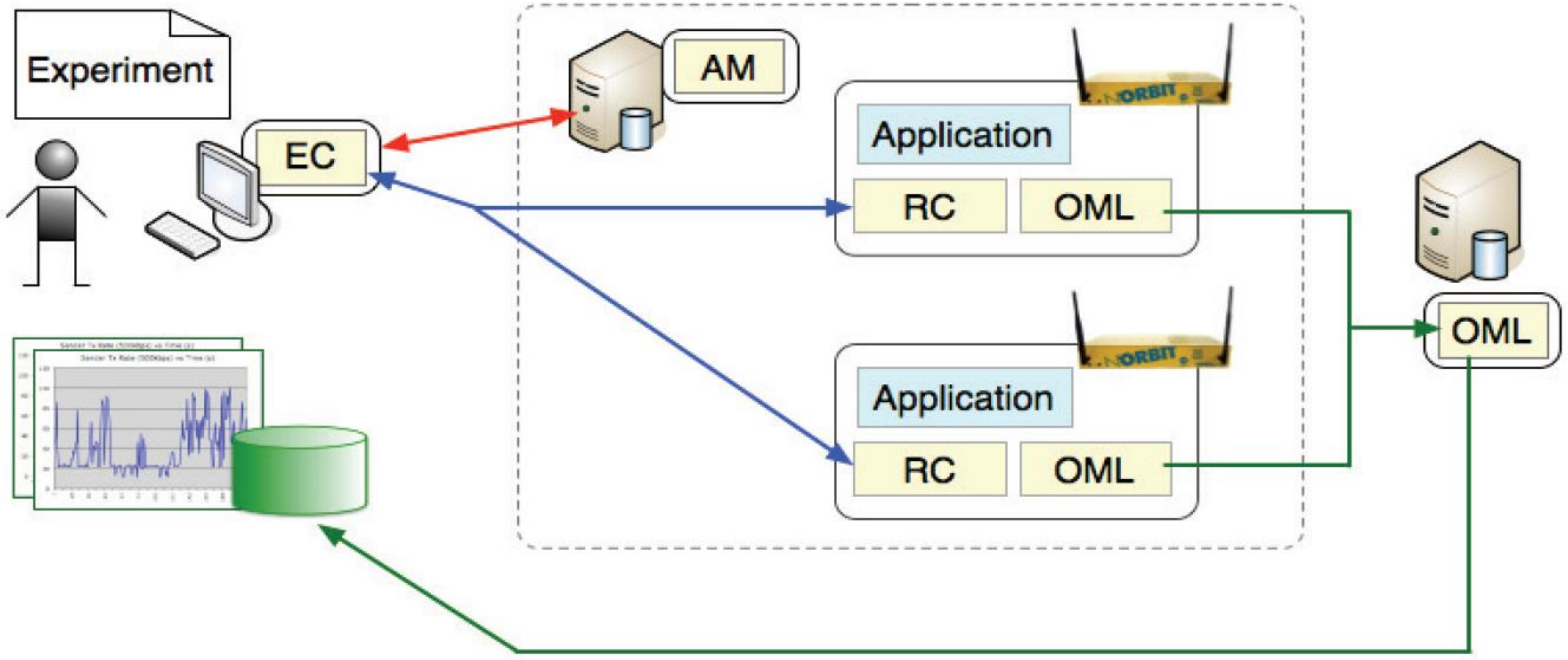


PlanetLab

NICTA, Sydney
Bridge Deployment

rry Rakotoarivelo

How it works from a user's perspective?

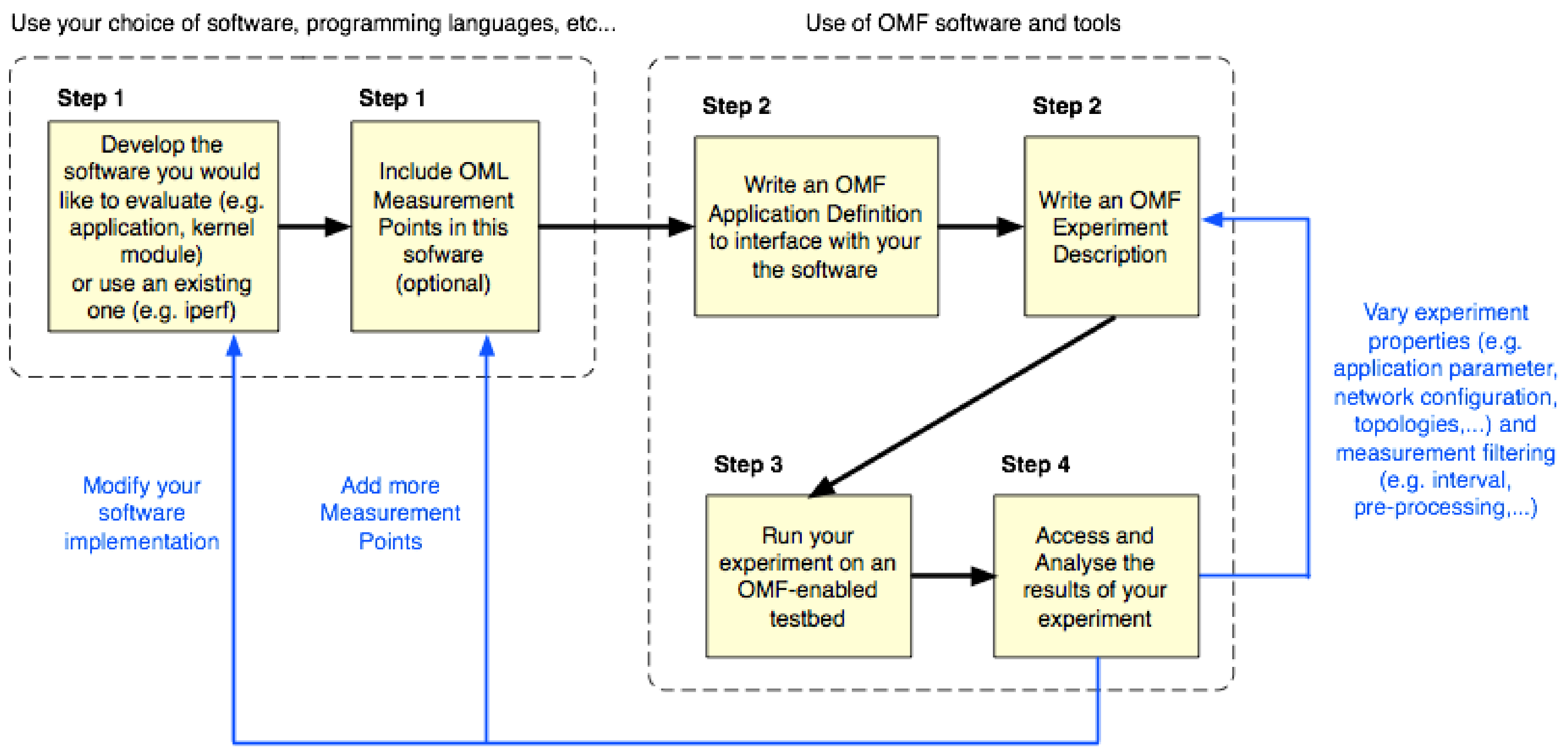


OMF Introduction

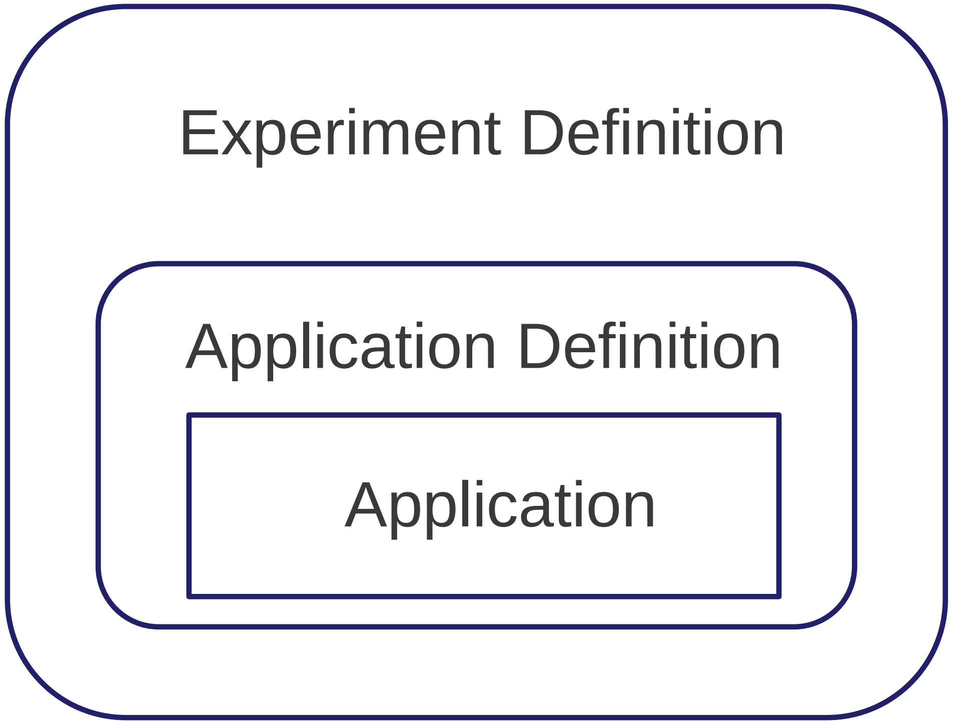
- http://omf.mytestbed.net (Version 5.3)
  - Tutorials
  - Ruby - OEDL Reference

- OML Measurement Library
  - Framework for measurement collection
  - Basic architecture:
    - Client: perform measurements inside the experiment
    - Server: collect and store measurement in sqlite3 databases

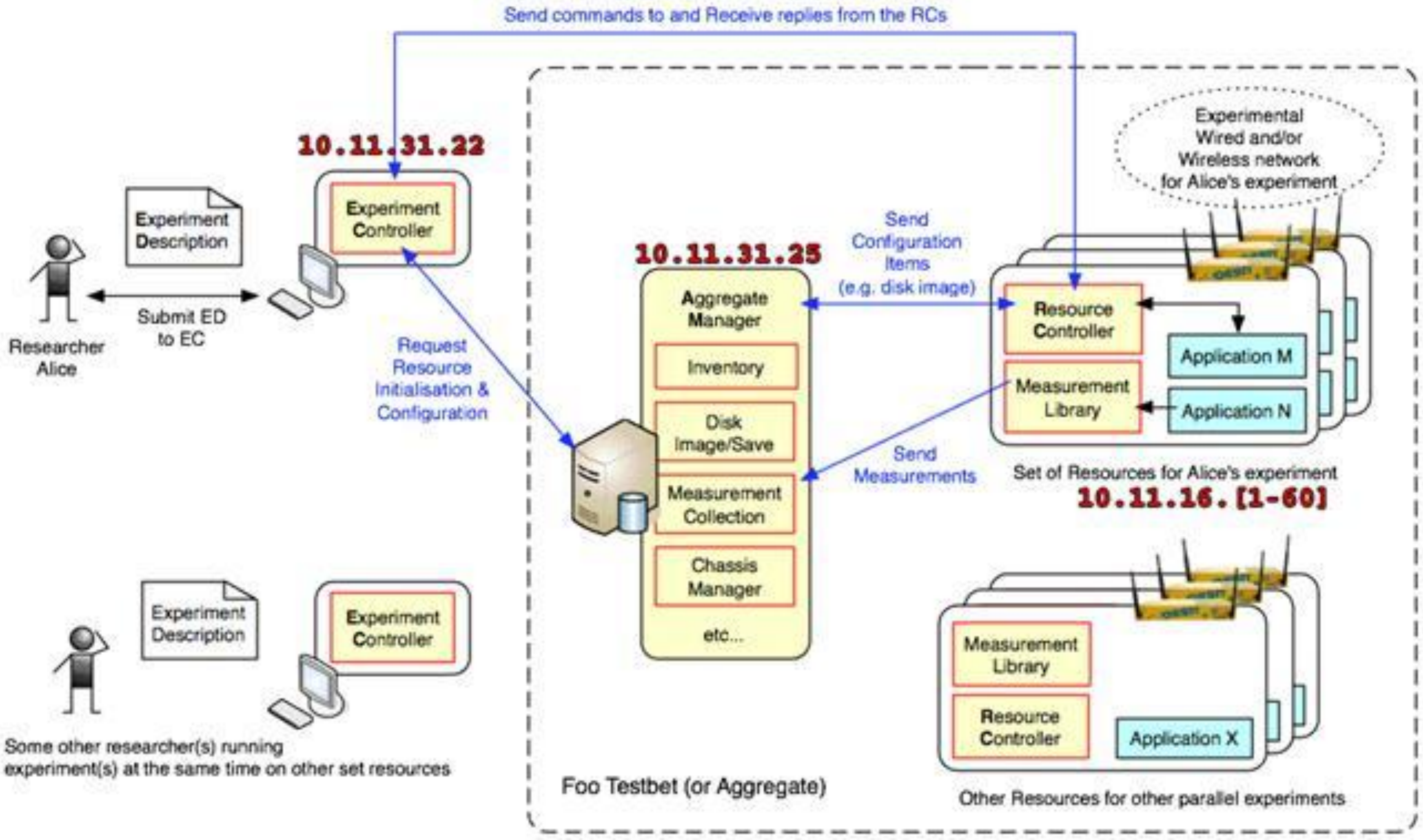
Running OMF Experiments


Use your choice of software, programming languages, etc...
Use of OMF software and tools
Step 1
Develop the software you would like to evaluate (e.g. application, kernel module) or use an existing one (e.g. iperf)
Step 1
Include OML Measurement Points in this sofware (optional)
Step 2
Write an OMF Application Definition to interface with your the software
Step 2
Write an OMF Experiment Description
Step 3
Run your experiment on an OMF-enabled testbed
Step 4
Access and Analyse the results of your experiment
Modify your software implementation
Add more Measurement Points
Vary experiment properties (e.g. application parameter, network configuration, topologies,...) and measurement filtering (e.g. interval, pre-processing,...)

Running OMF Experiments



OMF Setup @ w-iLab.t Zwijnaarde



Send commands to and Receive replies from the RCs
10.11.31.22
Experiment Controller
Experiment Description
Submit ED to EC
Researcher Alice
Request Resource Initialisation & Configuration
10.11.31.25
Aggregate Manager
Inventory
Disk Image/Save
Measurement Collection
Chassis Manager
etc...
Send Configuration Items (e.g. disk image)
Send Measurements
Experimental Wired and/or Wireless network for Alice's experiment
Resource Controller
Measurement Library
Application M
Application N
Set of Resources for Alice's experiment
10.11.16.[1-60]
Experiment Description
Experiment Controller
Some other researcher(s) running experiment(s) at the same time on other set resources
Measurement Library
Resource Controller
Application X
Foo Testbet (or Aggregate)
Other Resources for other parallel experiments

w-iLab.t Zwijnaarde workshop

Hello world – step by step

Status & Reservation

- w-iLab.t webinterface
  - http://10.11.31.25/
  - Listed info :
    - Status of nodes
    - Reservation system
    - Tutorials
    - Retrieval of experiment data
    - Topology map
    - Camera’s
- Reservation
  - Create entry in Google Calendar
  - Invite ibbtwilab2@gmail.com
  - E.g. : 10-20,35; hello world
  - Enforced !
  - http://10.11.31.25/status/tutorials/Wical.htm

Experiment Controller

- Request an account
  - pieter.becue@intec.ugent.be , vincent.sercu@intec.ugent.be , bart.jooris@intec.ugent.be
- Home dirs will be NFS-mounted from fileserver (near future)
- Backups every night

- Main tasks
  - Installing image (OS) on resources
    - omf load (omf help load)
    - omf save (omf help save)
  - Execution of experiments
    - omf exec (omf help exec)

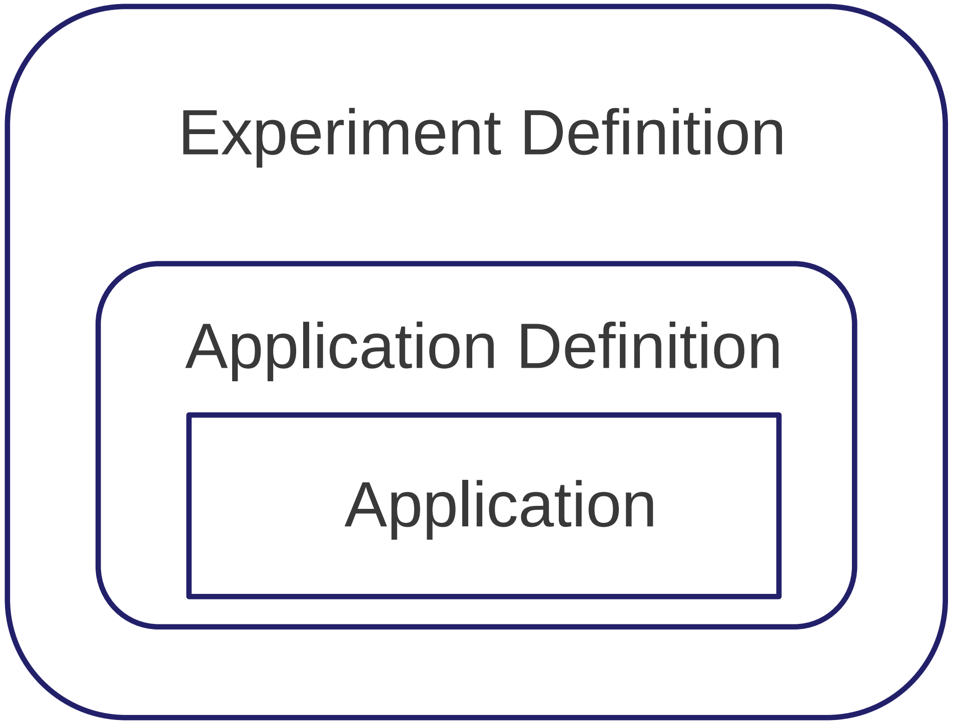
Installation of OS

- Stored in /var/lib/omf-images-5.3
- omf load -t omf.ibbt.open.nodeX,omf.ibbt.open.nodeY -i imagename.ndz
- Image is multicasted
  - Loading time independent of nr of nodes

- baselineIBBTfinal.ndz (root/urbis)
  - Ubuntu 10.04 (LTS)
  - Sensor tools
  - Webcam tools
  - Bluetooth tools

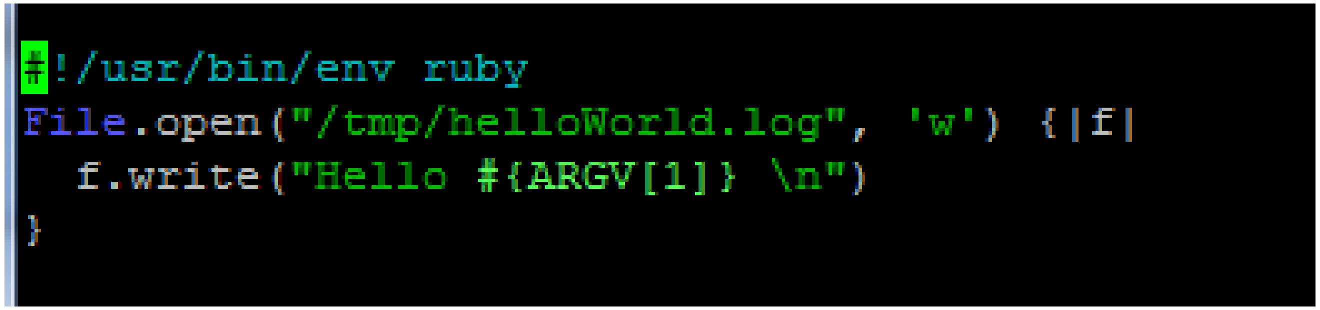
- omf load -t omf.ibbt.open.node1,omf.ibbt.open.node2 -i baselineIBBTfinal.ndz

Running OMF Experiments



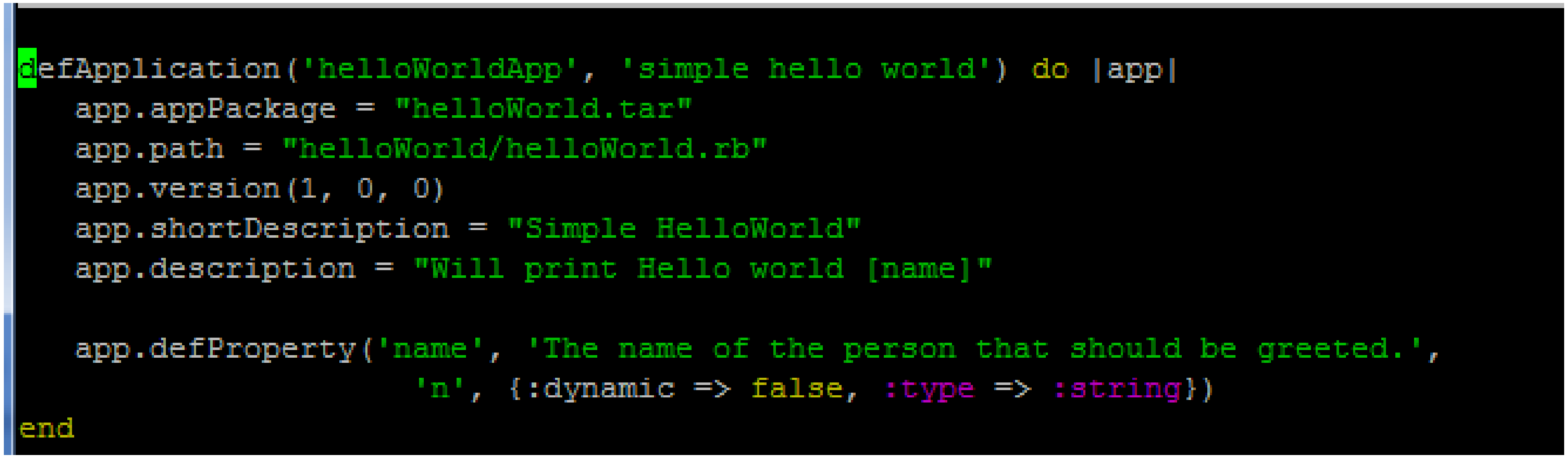
Application

- Ruby Hello World script
- Usage : helloWorld.rb –n “name”



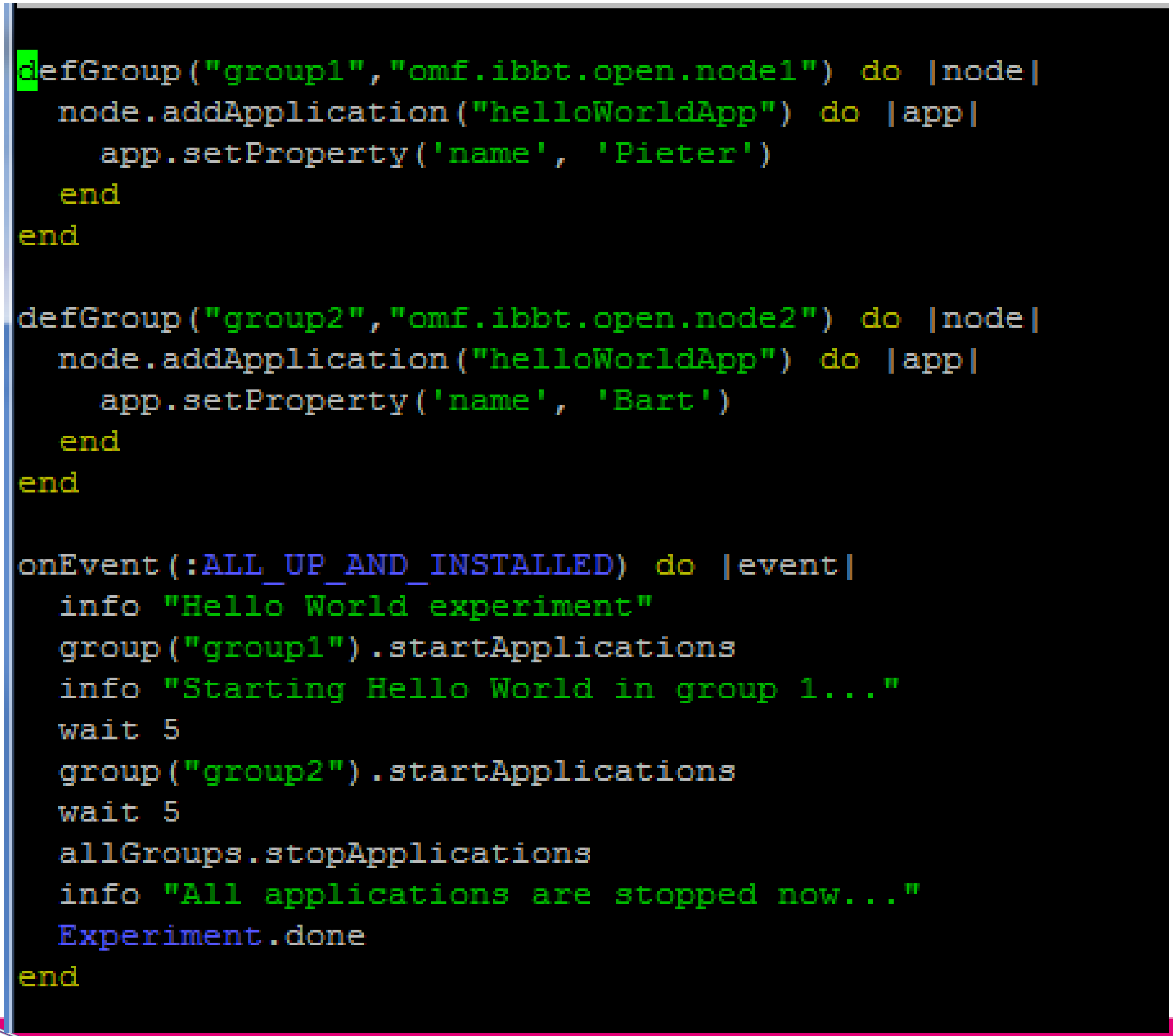
```
#!/usr/bin/env ruby
File.open("/tmp/helloWorld.log", 'w') {|f|
  f.write("Hello #{ARGV[1]} \n")
}
```

Application Definition



```
defApplication('helloWorldApp', 'simple hello world') do |app|
   app.appPackage = "helloWorld.tar"
   app.path = "helloWorld/helloWorld.rb"
   app.version(1, 0, 0)
   app.shortDescription = "Simple HelloWorld"
   app.description = "Will print Hello world [name]"

   app.defProperty('name', 'The name of the person that should be greeted.',
                   'n', {:dynamic => false, :type => :string})
end
```

Experiment Definition

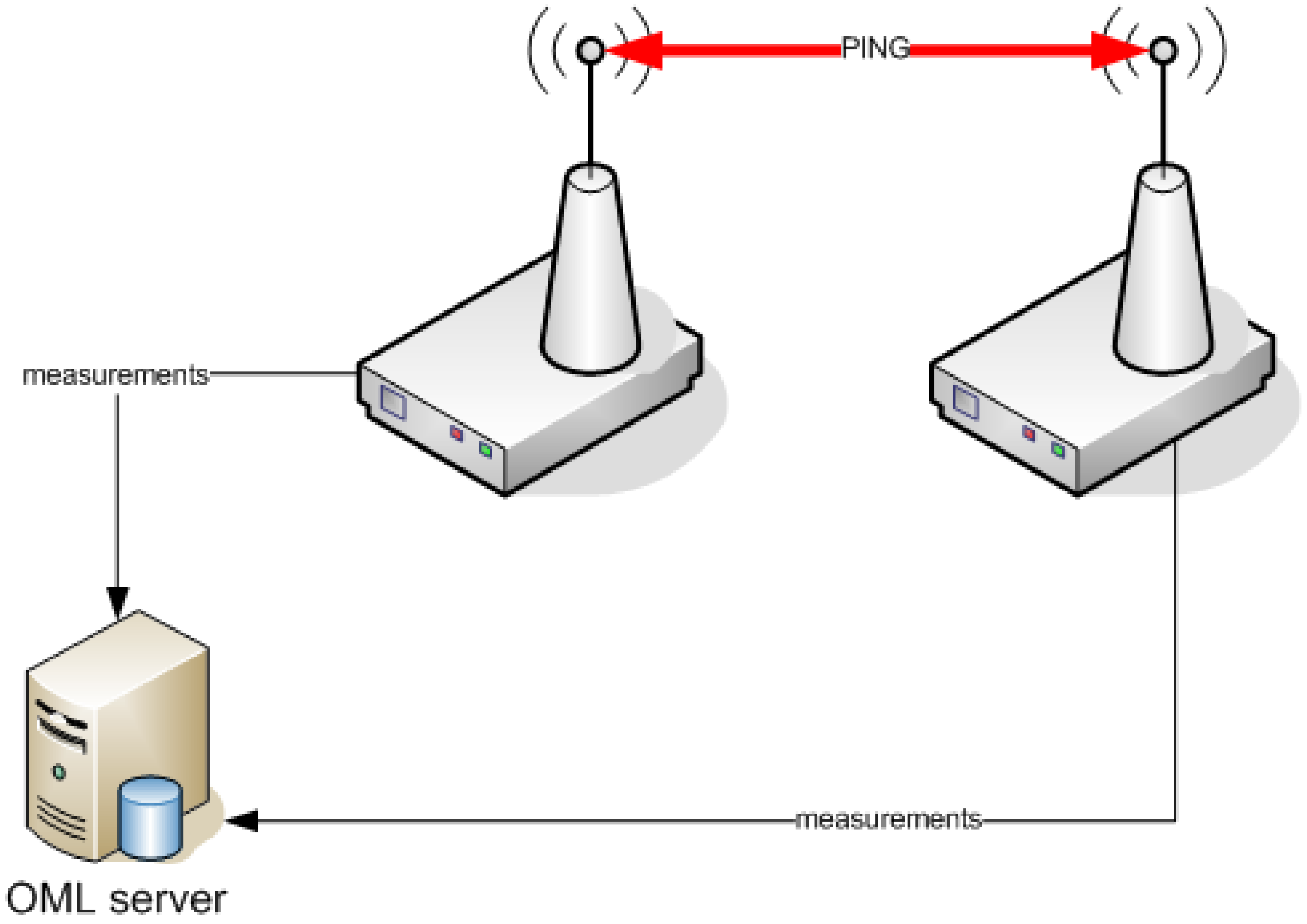


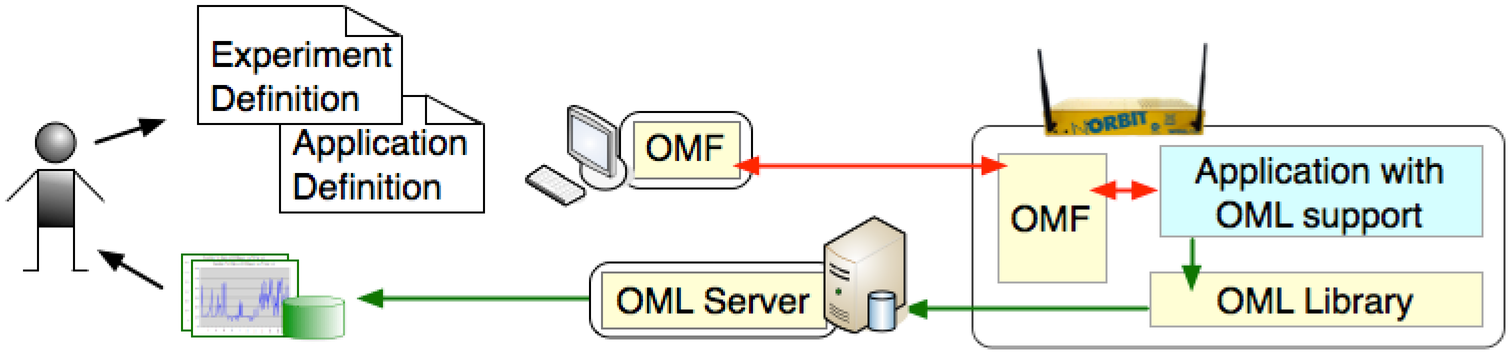
```
defGroup("group1","omf.ibbt.open.node1") do |node|
  node.addApplication("helloWorldApp") do |app|
    app.setProperty('name', 'Pieter')
  end
end

defGroup("group2","omf.ibbt.open.node2") do |node|
  node.addApplication("helloWorldApp") do |app|
    app.setProperty('name', 'Bart')
  end
end

onEvent(:ALL_UP_AND_INSTALLED) do |event|
  info "Hello World experiment"
  group("group1").startApplications
  info "Starting Hello World in group 1..."
  wait 5
  group("group2").startApplications
  wait 5
  allGroups.stopApplications
  info "All applications are stopped now..."
  Experiment.done
end
```

w-iLab.t Zwijnaarde workshop

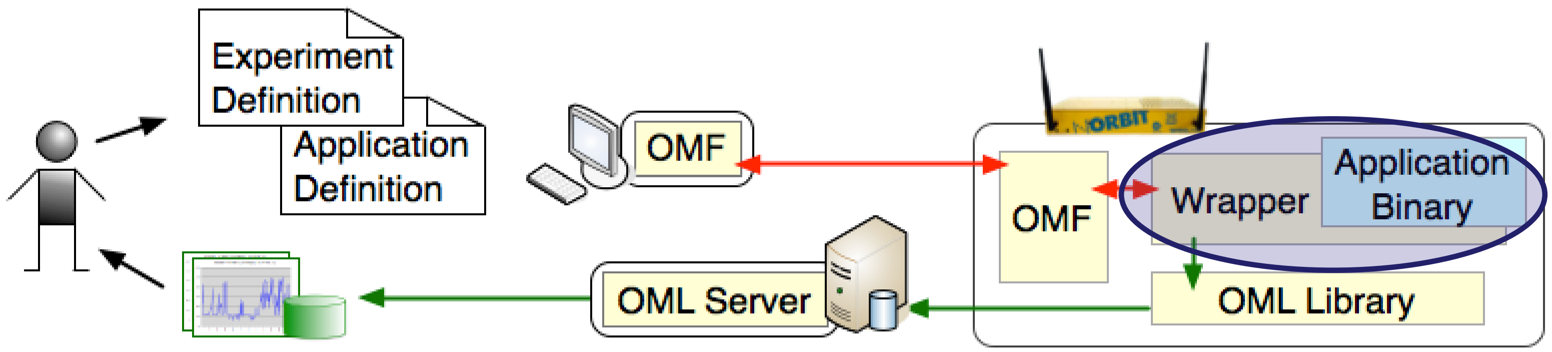
Wireless experiment (Ping)

Measuring



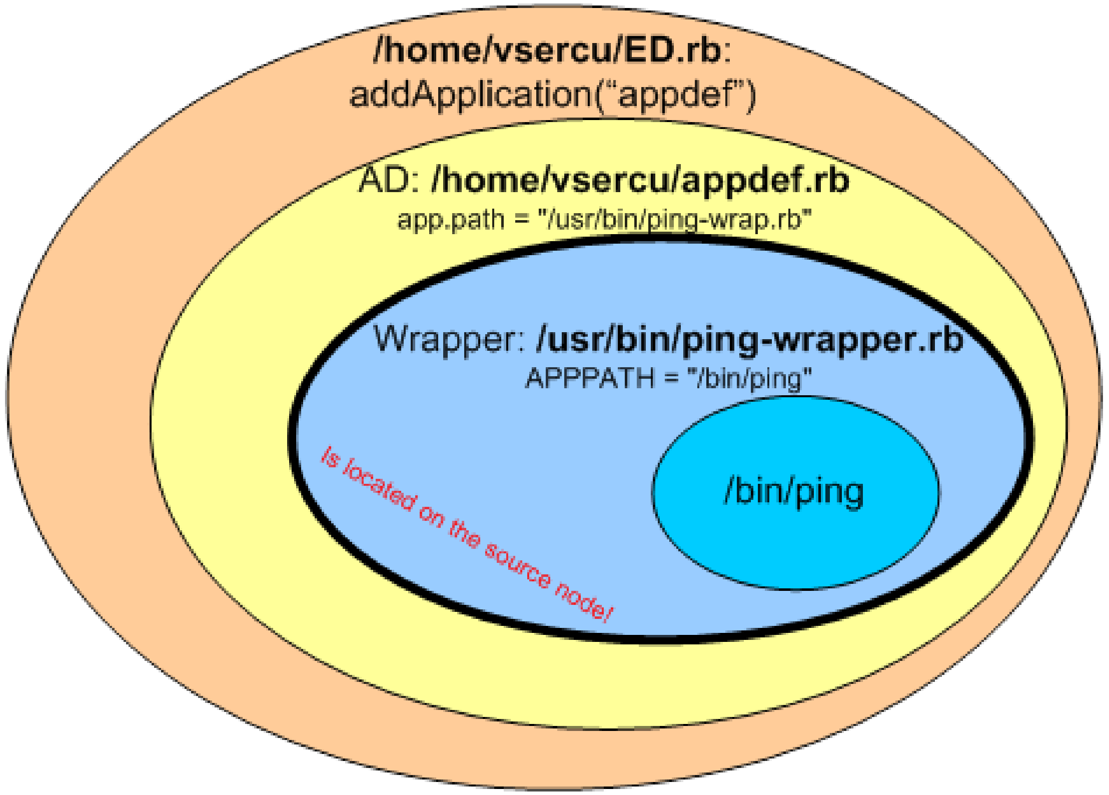


Wrapping the existing binary application



ibbt
connect.innovate.create

Call stack


/home/vsercu/ED.rb:
addApplication("appdef")
AD: /home/vsercu/appdef.rb
app.path = "/usr/bin/ping-wrap.rb"
Wrapper: /usr/bin/ping-wrapper.rb
APPPATH = "/bin/ping"
Is located on the source node!
/bin/ping

Exper. Description + App Definition on EC

Wrapper + binary on Resource

Blue / yellow bold border = network

Wrapper (simplified)

```
class MPStat < OML4R::MPBase
    name :pingo
    param :pingout
    # param :a_numeric_metric, :type => :long
end

class Wrapper
  def process_output(output)
    # logic for splitting should come here
    MPStat.inject("#{output}")
  end

  def initialize(args)
     # init oml logic
     # parsing arguments from the commandline and storing them locally for later use
  end

  def start()
     cmd = "#{APPPATH} #{@addr}"
     output = IO.popen(cmd) # execute the actual command
     output.each {|line|  process_output(line) }
  end
end
begin
  app = Wrapper.new(ARGV)
  puts "Executing the prog and inserting measurements"
  app.start()
end
```

- Measurement class, extends the MPBase class
- Name (later used in AD), also tablename
- Parameters, aka name of the database column

CREATE TABLE "ping_**pingo**" (oml_sender_id INTEGER, oml_seq INTEGER, oml_ts_client REAL, oml_ts_server REAL, "**pingout**" TEXT);
INSERT INTO "ping_pingo" VALUES(1,2,22039.019445,3.565378,'From 192.168.0.3 icmp_seq=1 ...');

- Inject the measurements into OMF using previously created MPStat class
- This method executes the actual program with the arguments
- When the script starts it executes the initialize() method and passes the arguments
- Execute the start-method

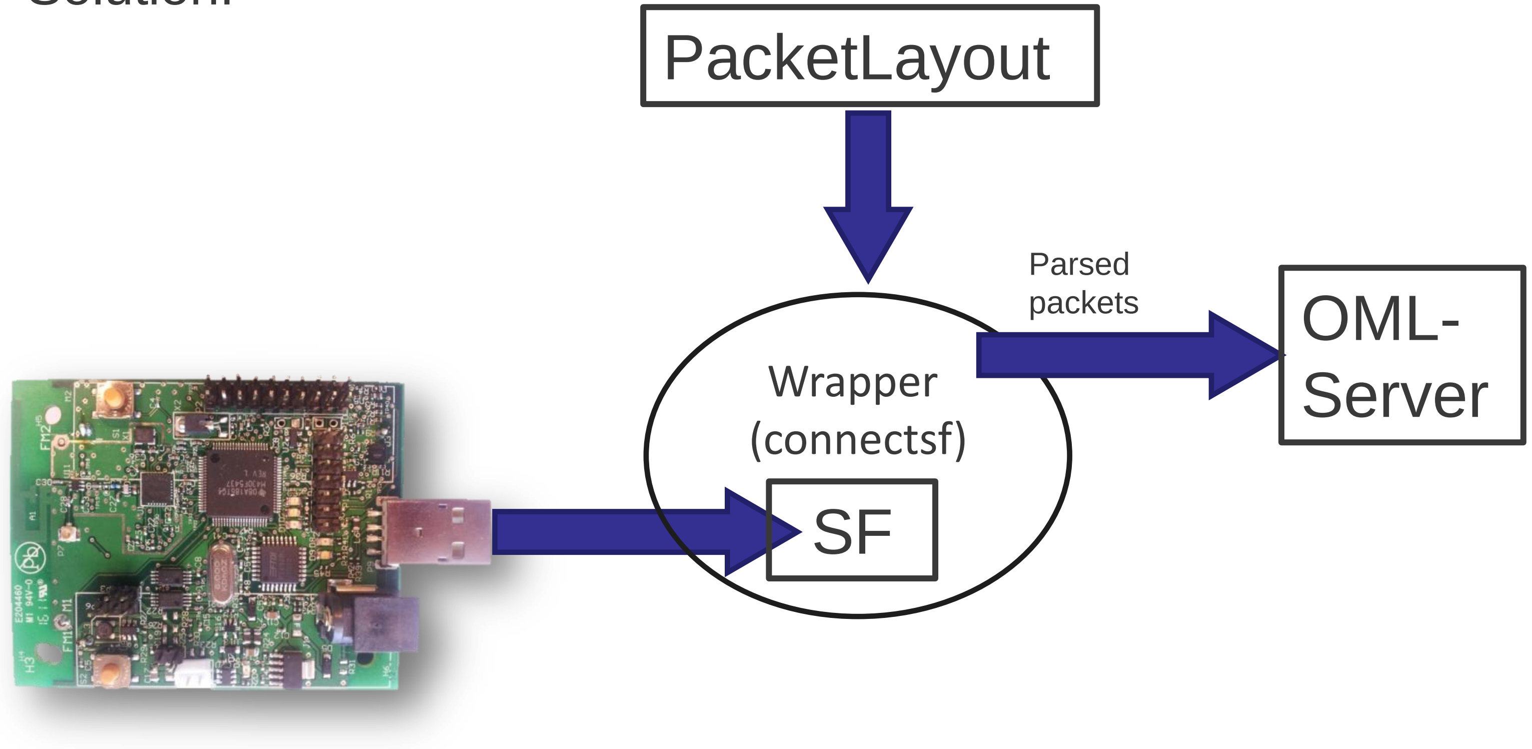
Distribute wrapper amoung nodes with tarball

w-iLab.t Zwijnaarde workshop

Running a sensor experiment on w-iLab.t Zwijnaarde

Serial wrapper & packetlayout

- Standard not implemented by OMF…
- Solution:



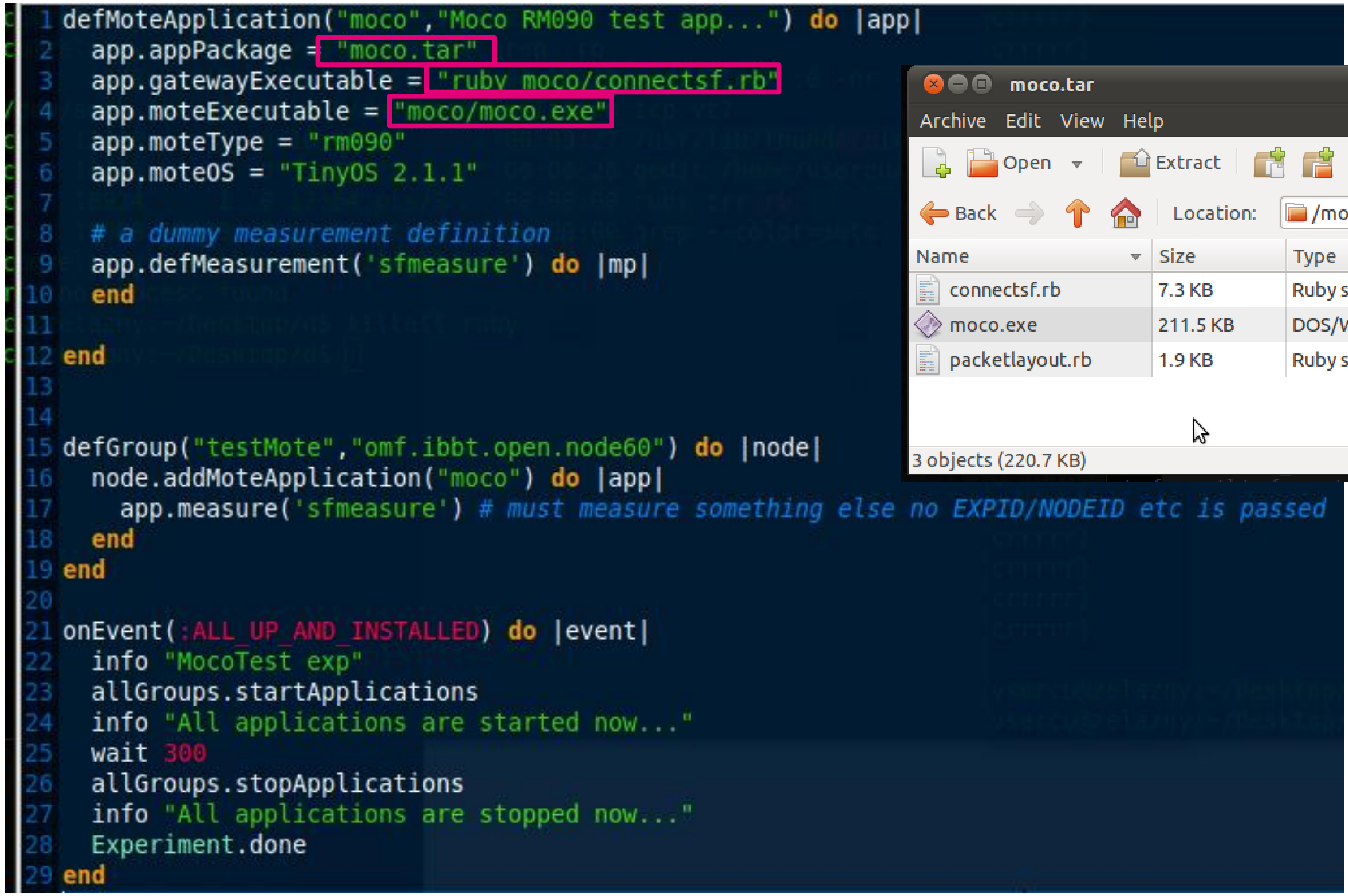
```
#define use_msg_t

typedef 204 DiscoveryPacket;
typedef nx_struct {
    nx_uint16_t len;
    nx_uint16_t type;
    nx_uint16_t sender;
    nx_uint16_t seqno;
} DiscoveryPacket;

typedef 205 Test;
typedef nx_struct {
    nx_uint16_t len;
} Test;

typedef 100 Printf;
typedef nx_struct {
    char[28] msg;
} Printf;

typedef 200 Msgt;
typedef nx_struct {
    // capture empty messaget packet
} Msgt;
```

Packetlayout

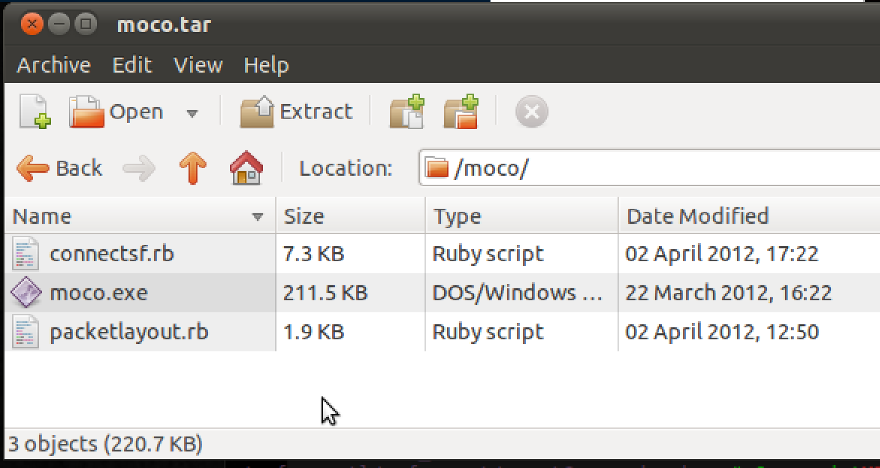
Only for flat structures: only basic types (no structs in structs)

```
#####
# This file is auto generated using parser.rb
# on 2012-03-02T10:30:59+01:00
#####

MSGID = Array[
    [ 'msgt_preamble', 'uint8_t' ],
    [ 'msgt_dst', 'nx_uint16_t' ],
    [ 'msgt_src', 'nx_uint16_t' ],
    [ 'msgt_len', 'uint8_t' ],
    [ 'msgt_group' , 'uint8_t' ],
    [ 'msgt_am' , 'uint8_t' ],
]

Msgid_offset = 7
Msgid_datatype = 'uint8_t'

### User packets:

DiscoveryPacket = Array [
     [ "DiscoveryPacket_len", "nx_uint16_t"],
     [ "DiscoveryPacket_type", "nx_uint16_t"],
     [ "DiscoveryPacket_sender", "nx_uint16_t"],
     [ "DiscoveryPacket_seqno", "nx_uint16_t"],
]
Test = Array [
     [ "Test_len", "nx_uint16_t"],
]
Printf = Array [
     [ "Printf_msg", "char[28]"],
]
Msgt = Array [
]

### AM hash:

AM = Hash[
     204 => MSGID + DiscoveryPacket,
     205 => MSGID + Test,
     100 => MSGID + Printf,
     200 => MSGID + Msgt,
]
```

Application Definition & OEDL



```
defMoteApplication("moco","Moco RM090 test app...") do |app|
  app.appPackage = "moco.tar"
  app.gatewayExecutable = "ruby moco/connectsf.rb"
  app.moteExecutable = "moco/moco.exe"
  app.moteType = "rm090"
  app.moteOS = "TinyOS 2.1.1"

  # a dummy measurement definition
  app.defMeasurement('sfmeasure') do |mp|
  end

end


defGroup("testMote","omf.ibbt.open.node60") do |node|
  node.addMoteApplication("moco") do |app|
    app.measure('sfmeasure') # must measure something else no EXPID/NODEID etc is passed
  end
end

onEvent(:ALL_UP_AND_INSTALLED) do |event|
  info "MocoTest exp"
  allGroups.startApplications
  info "All applications are started now..."
  wait 300
  allGroups.stopApplications
  info "All applications are stopped now..."
  Experiment.done
end
```



moco.tar

Location: /moco/

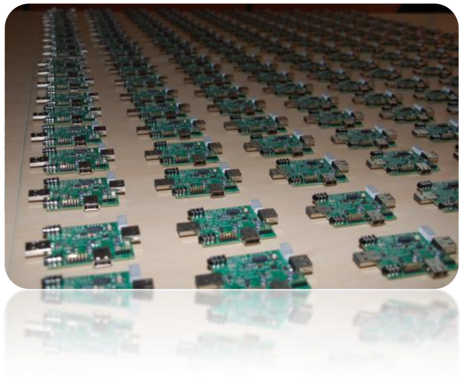
Name	Size	Type	Date Modified
connectsf.rb	7.3 KB	Ruby script	02 April 2012, 17:22
moco.exe	211.5 KB	DOS/Windows ...	22 March 2012, 16:22
packetlayout.rb	1.9 KB	Ruby script	02 April 2012, 12:50

3 objects (220.7 KB)

The w-iLab.t testbed

details: www.crew-project.eu/portal/w-ilabt

Questions?



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