3^{RD} Fed4FIRE-GENI SUMMER SCHOOL

FLEX LTE TUTORIAL

Presenter: Virgilios Passas
University of Thessaly

European Commission

FIRE LTE TESTBEDS FOR OPEN EXPERIMENTATION

www.flex-project.eu

Scope of the Tutorial

FIRE LTE TESTBEDS FOR OPEN EXPERIMENTATION

- Use the LTE infrastructure that is available in Fed4FIRE testbeds (through FLEX)
- Familiarize with the Fed4FIRE-FLEX tools
- Use the LTE infrastructure in 2 approaches:
  - Commercial setup
  - Open Source Setup
- Visualize measurements collected from the testbeds

European Commission

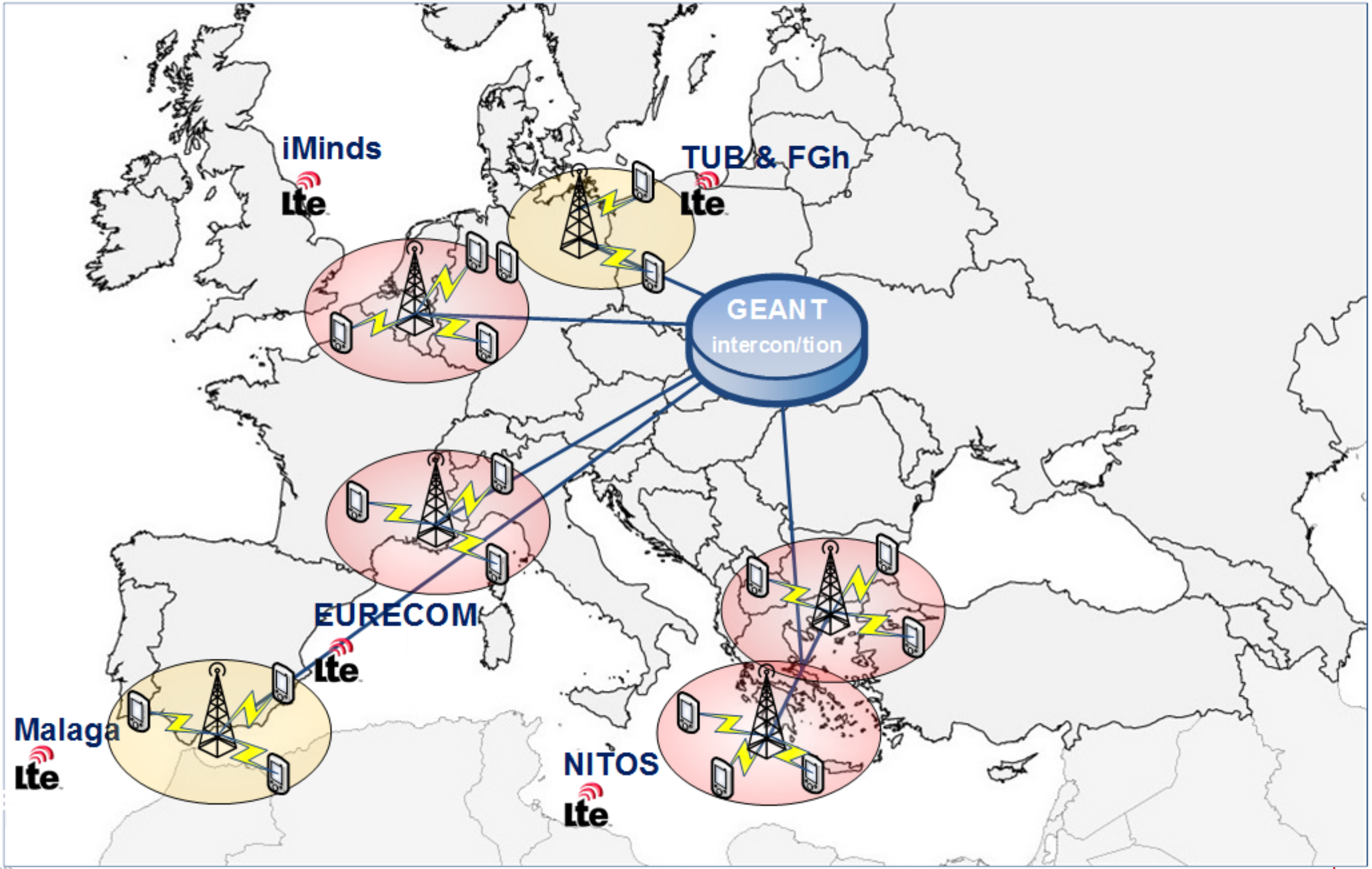
Overview

- □ FLEX project
- □ Tools for this tutorial
- □ Making a Reservation
- □ Experiment using the Commercial Setup
- □ Tutorial with the Open Source Setup (OAI)
- □ Experiment using with LTE-U and Wi-Fi

The FLEX Concept

- Highly Programmable equipment for 4G provided in two manners:
  - ”Commercial” setup, using off-the-shelf BSs, EPCs and UEs
  - “Open Source” setup, using OpenAirInterface
- Evaluating applications and protocols against existing technologies, or completely restructure the existing networking stack.

Testbeds Available in FLEX



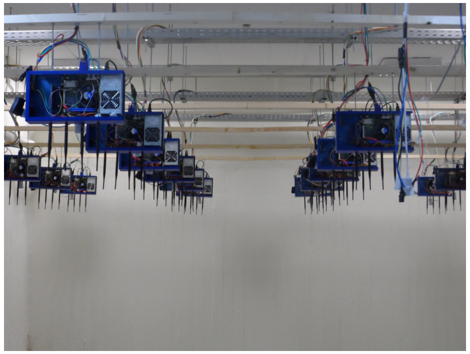
NITOS testbed

http://nitlab.inf.uth.gr

UNIVERSITY OF THESSALY

NITOS testbed

- Three separate deployments
  - An indoor RF-isolated testbed (50 nodes)
  - An outdoor prone to RF interference testbed (50 nodes)
  - An office testbed (10 nodes)

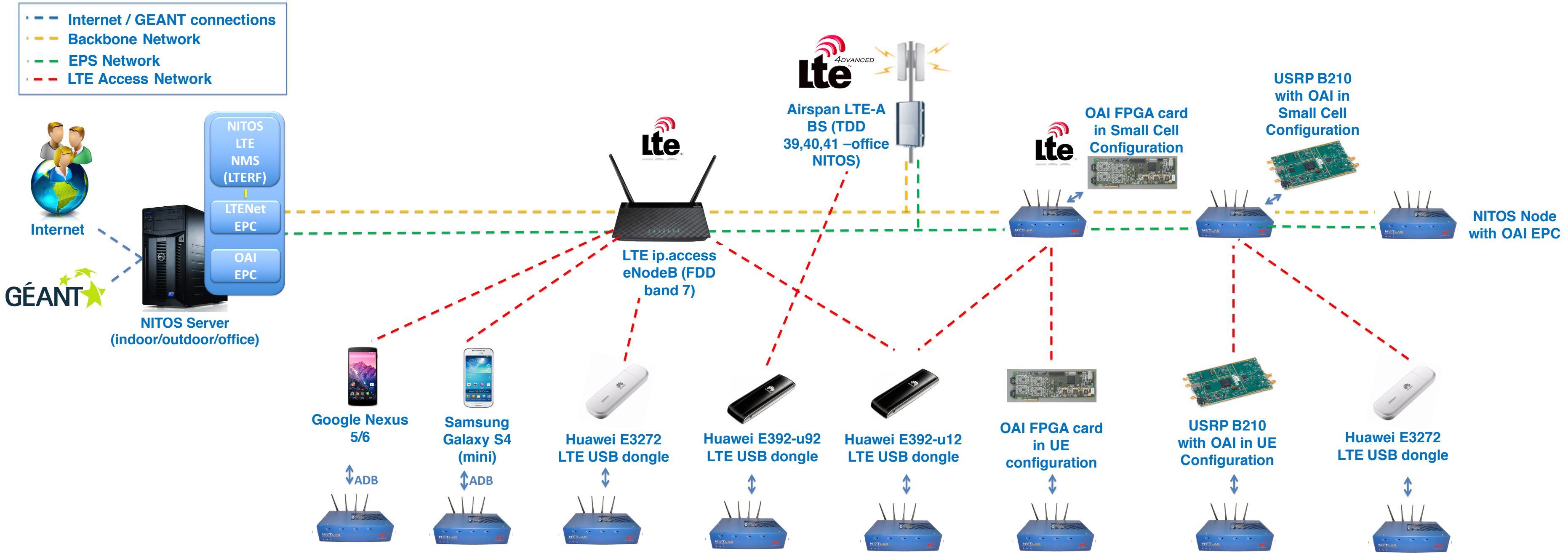


NITOS indoor (left) and outdoor (center) testbed deployments. The macroscale LTE-A base station deployments at the office testbed

Current LTE deployment

- 11 eNBs in total (2 ip.access femtocells, 1 Airspan macrocell, 8 OAI front-ends).
- 2 EPC approaches:
  - Commercial approach:
    - SiRRAN EPC (partner of FLEX) installation at a dedicated NITOS server.
  - Open Source approach:
    - Multiple instances of the OAI Core Network available as testbed images for NITOS nodes.
- Multiple UE available across the three testbeds.
  - USB LTE Dongles
  - Android Smartphones

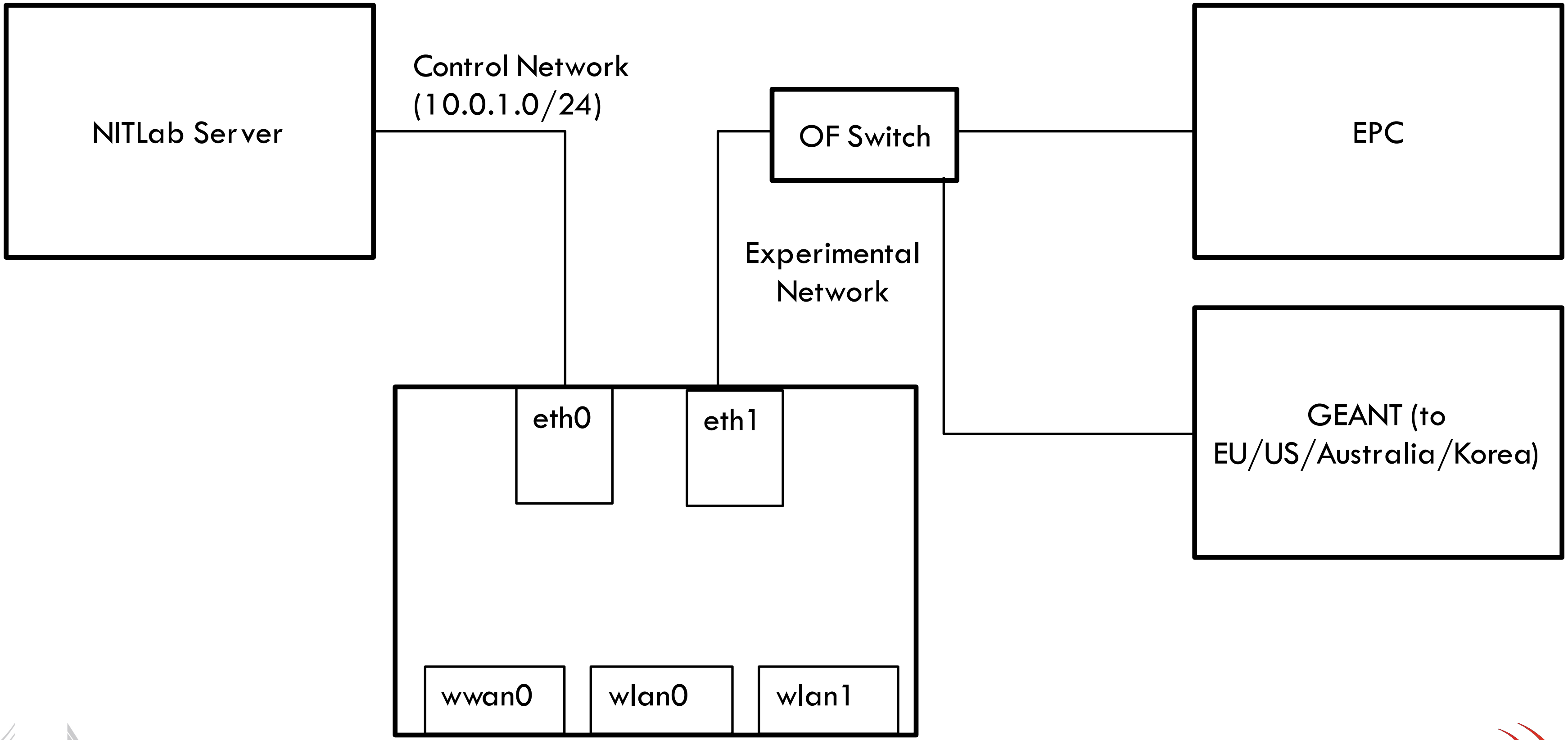
Overall Architecture


Internet / GEANT connections
Backbone Network
EPS Network
LTE Access Network
Internet
GÉANT
NITOS LTE NMS (LTERF)
LTENet EPC
OAI EPC
NITOS Server (indoor/outdoor/office)
LTE ip.access eNodeB (FDD band 7)
4DVANCED
Airspan LTE-A BS (TDD 39,40,41 –office NITOS)
OAI FPGA card in Small Cell Configuration
USRP B210 with OAI in Small Cell Configuration
NITOS Node with OAI EPC
Google Nexus 5/6
ADB
Samsung Galaxy S4 (mini)
ADB
Huawei E3272 LTE USB dongle
Huawei E392-u92 LTE USB dongle
Huawei E392-u12 LTE USB dongle
OAI FPGA card in UE configuration
USRP B210 with OAI in UE Configuration
Huawei E3272 LTE USB dongle

Accessing the nodes

- □ Establishing secure shell to the NITLab portal server of each testbed.
- □ If the reservation is valid, you can load images/ssh on the nodes.
- □ If you have reserved the base station, you get access to the service that allows you to alter configurations to base station/EPC.

Node Architecture



Overview

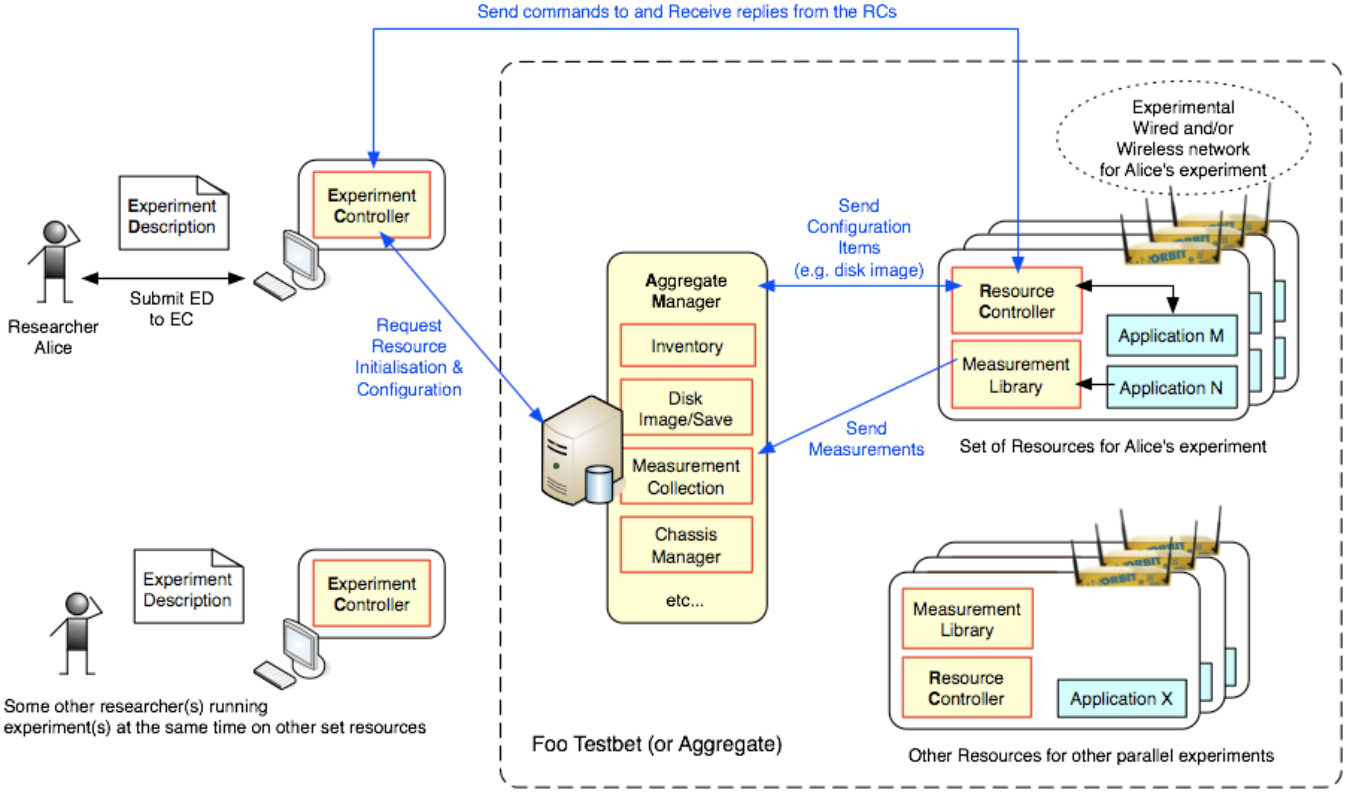
- □ FLEX project
- □ Tools for this tutorial
- □ Making a Reservation
- □ Experiment using the Commercial Setup
- □ Tutorial with the Open Source Setup (OAI)
- □ Experiment using LTE-U and Wi-Fi

OMF Framework (1/2)

- OMF stands for c**O**ntrol and **M**anagement **F**ramework
- It is a software framework used to manage and execute experiments
- With OMF, we write an experiment script and run it using an experiment controller, which tells each of the testbed nodes in an experiment how to configure themselves and what applications to run

http://omf.mytestbed.net

OMF Framework (2/2)


Send commands to and Receive replies from the RCs
Experiment Description
Experiment Controller
Researcher Alice
Submit ED to EC
Request Resource Initialisation & Configuration
Send Configuration Items (e.g. disk image)
Aggregate Manager
Inventory
Disk Image/Save
Measurement Collection
Chassis Manager
etc...
Experimental Wired and/or Wireless network for Alice's experiment
Resource Controller
Application M
Measurement Library
Application N
Send Measurements
Set of Resources for Alice's experiment
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

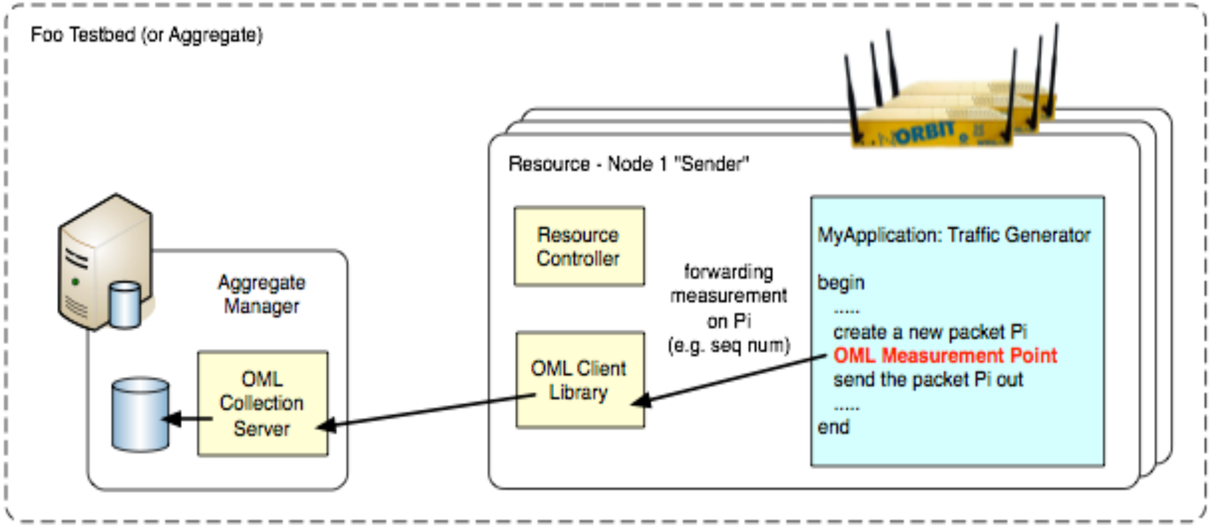
OML Measurement Library (1/2)

- OML stands for **O**MF **M**easurement **L**ibrary
- It is a companion software framework for OMF, focused on supporting the lifecycle of measurement data, i.e.:
  - Generation and capturing
  - Processing – Filtering
  - Collection
  - Storage
- It can also be used independently of OMF, in any environment where devices connected to a network generate measurements

http://oml.mytestbed.net

OML Measurement Library (2/2)

- OML Architecture:



Foo Testbed (or Aggregate)

Aggregate Manager

OML Collection Server

Resource - Node 1 "Sender"

Resource Controller

OML Client Library

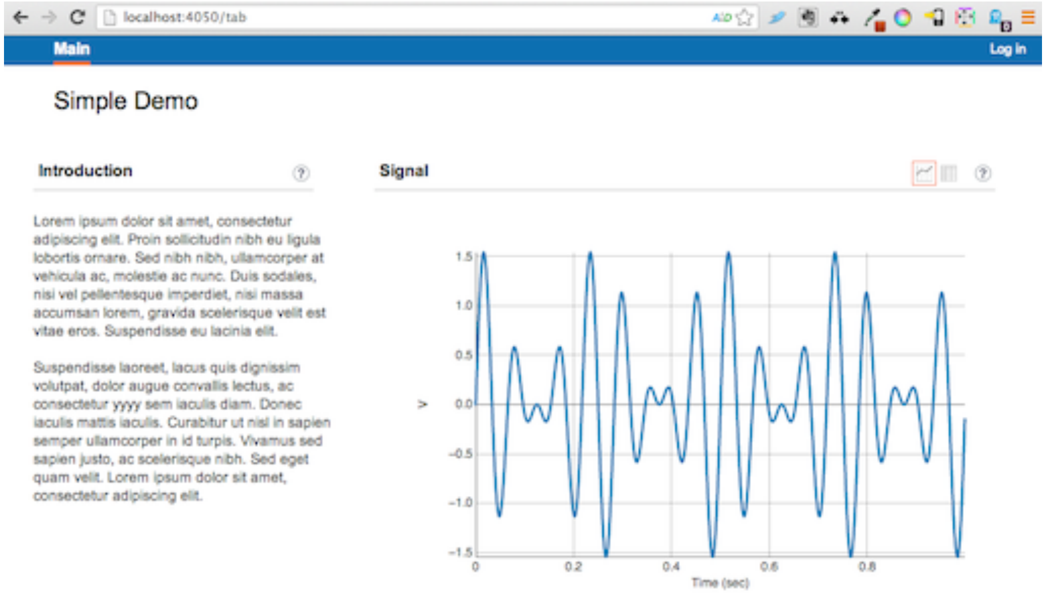
forwarding measurement on Pi (e.g. seq num)

```
MyApplication: Traffic Generator
begin
  .....
  create a new packet Pi
  OML Measurement Point
  send the packet Pi out
  .....
end
```

- Several installations of OML server at NITOS, supporting PostgreSQL/SQLite backend

OMF Web

- OMF Web provides the components for building a web-based data visualization service.
- The experimenter is allowed to investigate a data set stored in a database as well as life data streams.



Simple Demo

Introduction

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin sollicitudin nibh eu ligula lobortis ornare. Sed nibh nibh, ullamcorper at vehicula ac, molestie ac nunc. Duis sodales, nisi vel pellentesque imperdiet, nisi massa accumsan lorem, gravida scelerisque velit est vitae eros. Suspendisse eu lacinia elit.

Suspendisse laoreet, lacus quis dignissim volutpat, dolor augue convallis lectus, ac consectetur yyyy sem iaculis diam. Donec iaculis mattis iaculis. Curabitur ut nisl in sapien semper ullamcorper in id turpis. Vivamus sed sapien justo, ac scelerisque nibh. Sed eget quam velit. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

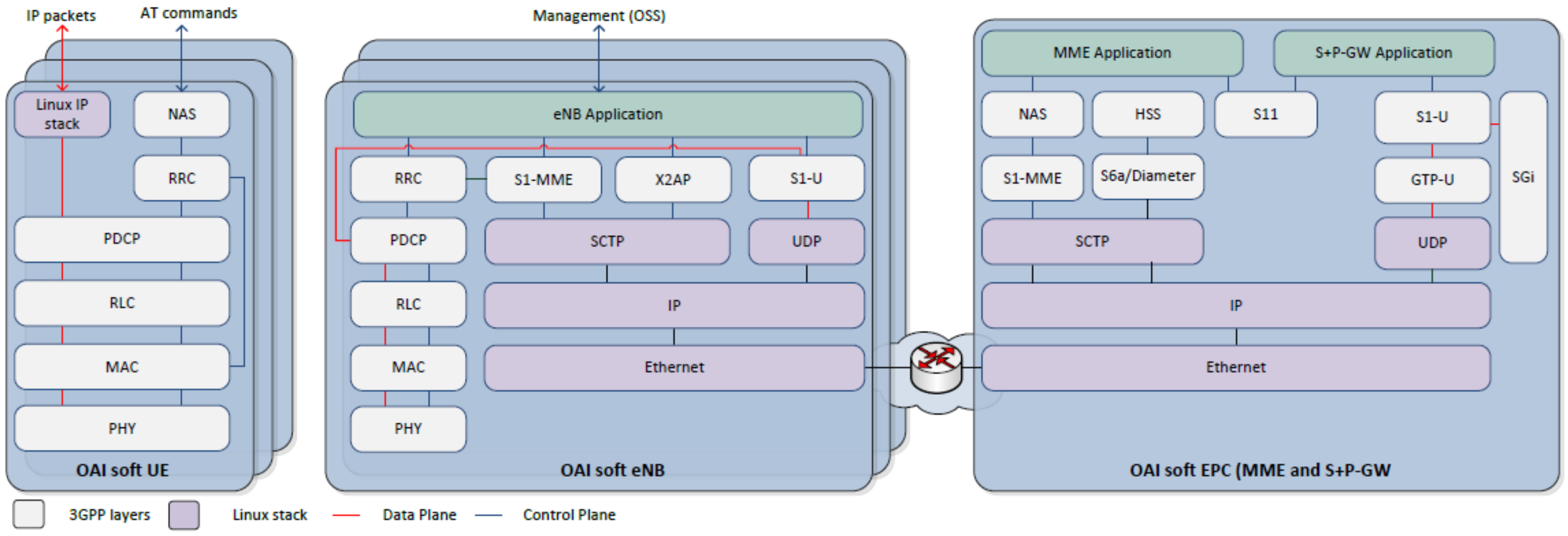
LTERF

- FLEX service allowing the configuration of the eNBs, EPC network and datapaths through a REST based API.
- Service has been built in order to be in-line with the existing tools for the WiMAX BS handling, which are available in GENI testbeds.
- Common API for configuring the resources (regardless of them).
- Separate driver running on the southbound interface for configuring each component accordingly.
- Currently supported:
  - Ip.access LTE245F femtocells
  - SiRRAN EPC network (v1.4 & 1.5)
  - OAI cells
  - OpenBTS cells (2G and 3G)
  - Configuration of the Datapath
- Service for NITOS is available at http://194.177.207.3:5054/lterf

OpenAirInterface (1/2)

- OpenAirInterface platform is a flexible platform towards an open LTE ecosystem.
- The platform offers an open-source software-based implementation of the LTE system spanning the full protocol stack of 3GPP standard both in E-UTRAN and EPC.
- It can be used to build and customized an LTE base station and core network on a PC and connect a commercial UEs to test different configurations and network setups and monitor the network and mobile device in real-time.
- OAI also provides a simulation framework, so you can alter the code and test it in a VM (no need for multiple computers and RF frontends)

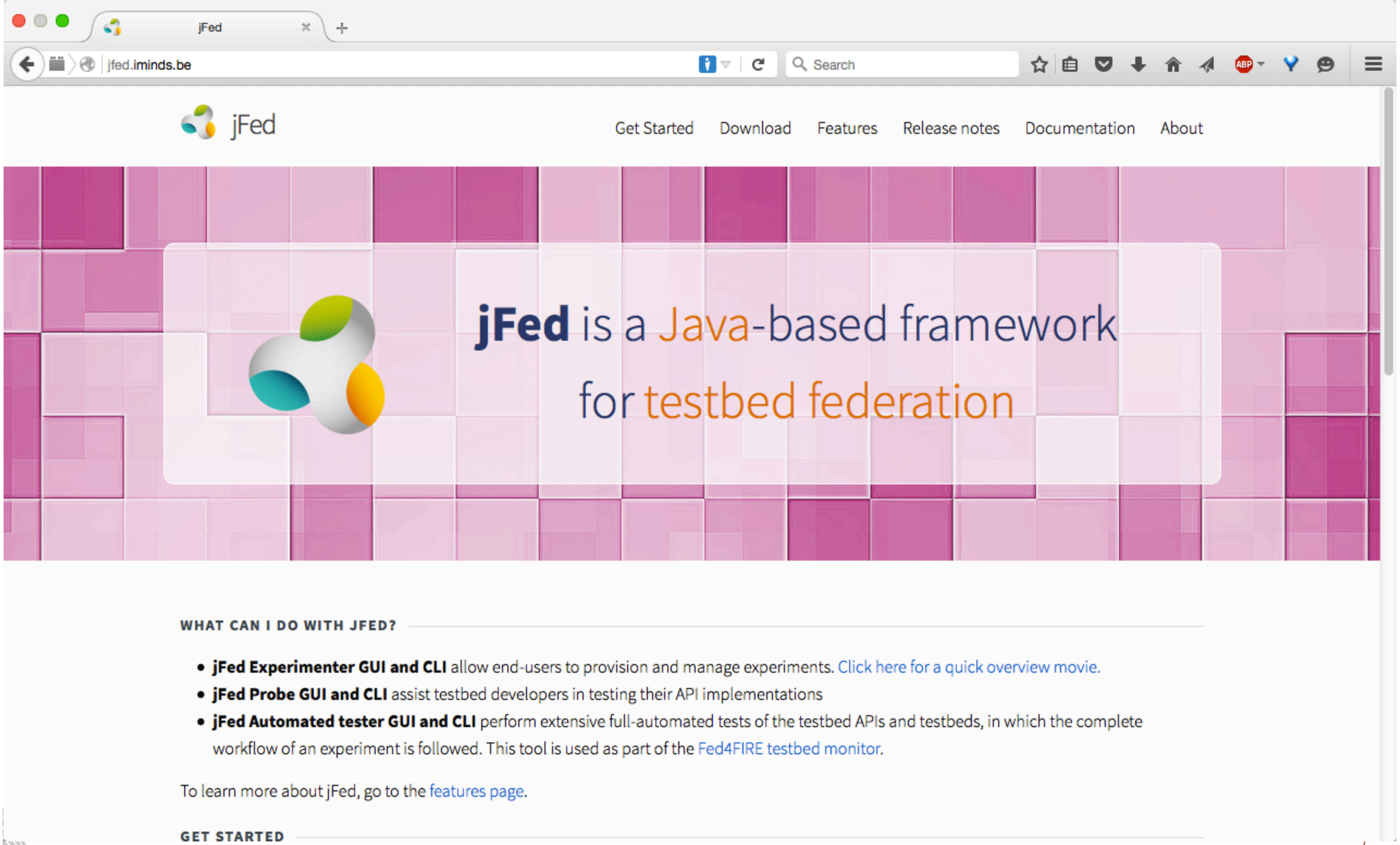
OpenAirInterface (2/2)



Overview

- □ FLEX project
- □ Tools for this tutorial
- □ Making a Reservation
- □ Experiment using the Commercial Setup
- □ Tutorial with the Open Source Setup (OAI)
- □ Experiment using LTE-U and Wi-Fi

JFED



jFed is a Java-based framework for testbed federation

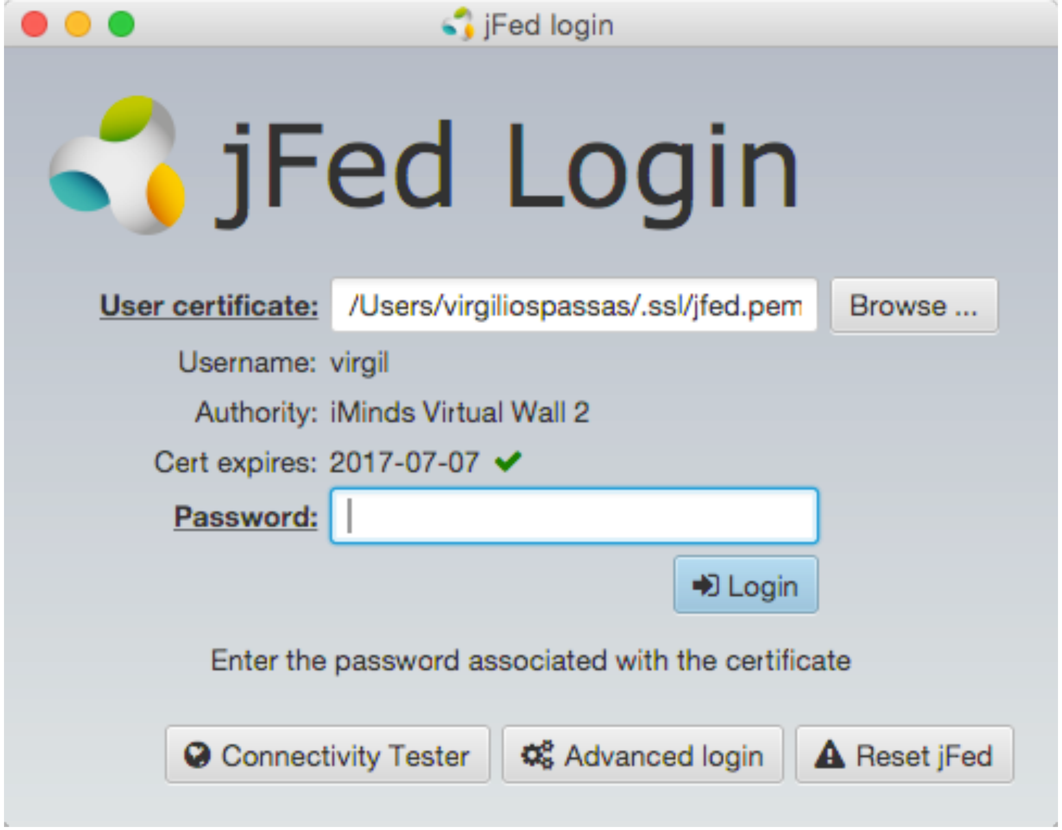
WHAT CAN I DO WITH JFED?

- **jFed Experimenter GUI and CLI** allow end-users to provision and manage experiments. Click here for a quick overview movie.
- **jFed Probe GUI and CLI** assist testbed developers in testing their API implementations
- **jFed Automated tester GUI and CLI** perform extensive full-automated tests of the testbed APIs and testbeds, in which the complete workflow of an experiment is followed. This tool is used as part of the Fed4FIRE testbed monitor.

To learn more about jFed, go to the features page.

GET STARTED

JFED



jFed login

jFed Login

User certificate: /Users/virgiliospassas/.ssl/jfed.pem

Username: virgil

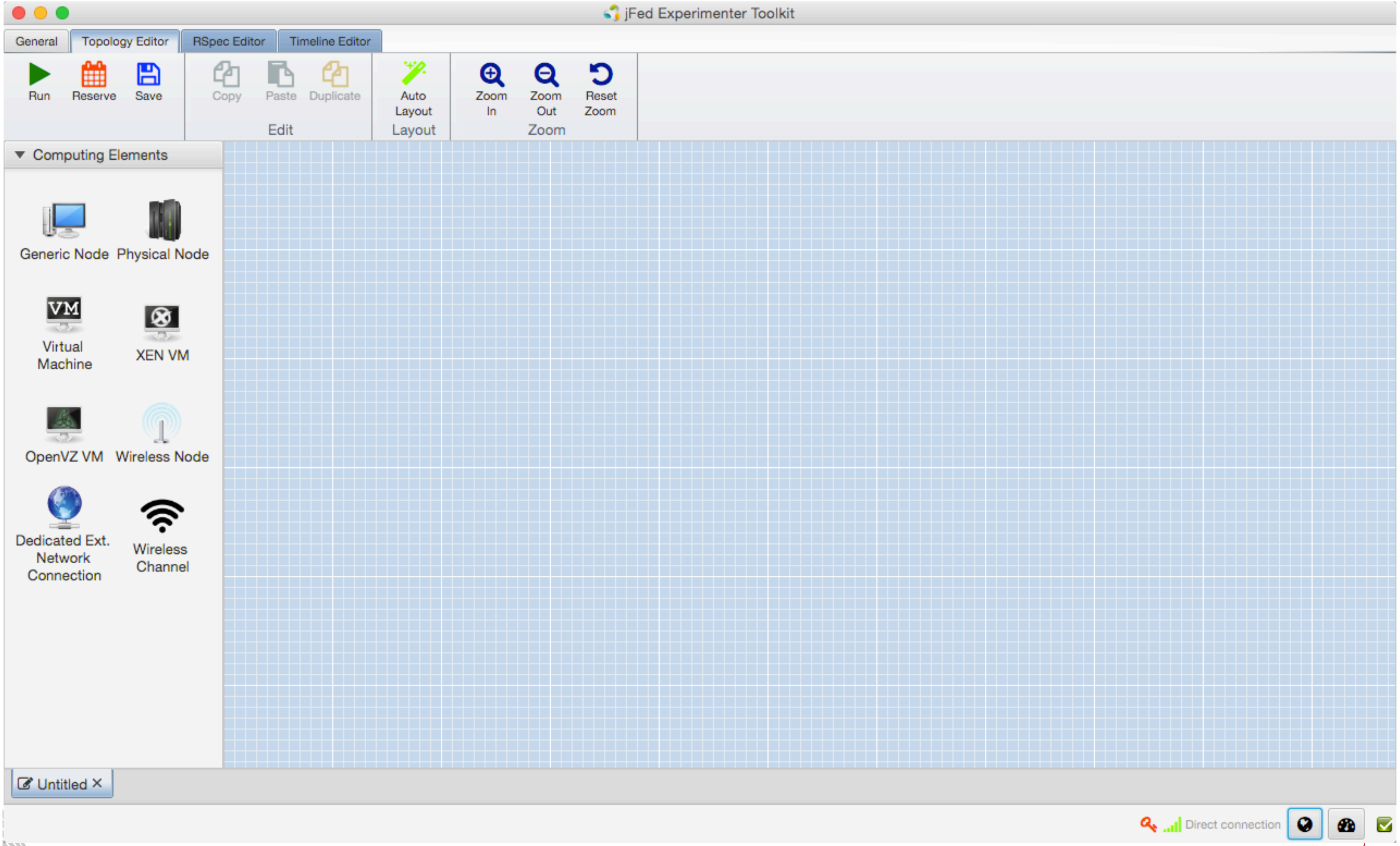
Authority: iMinds Virtual Wall 2

Cert expires: 2017-07-07 ✓

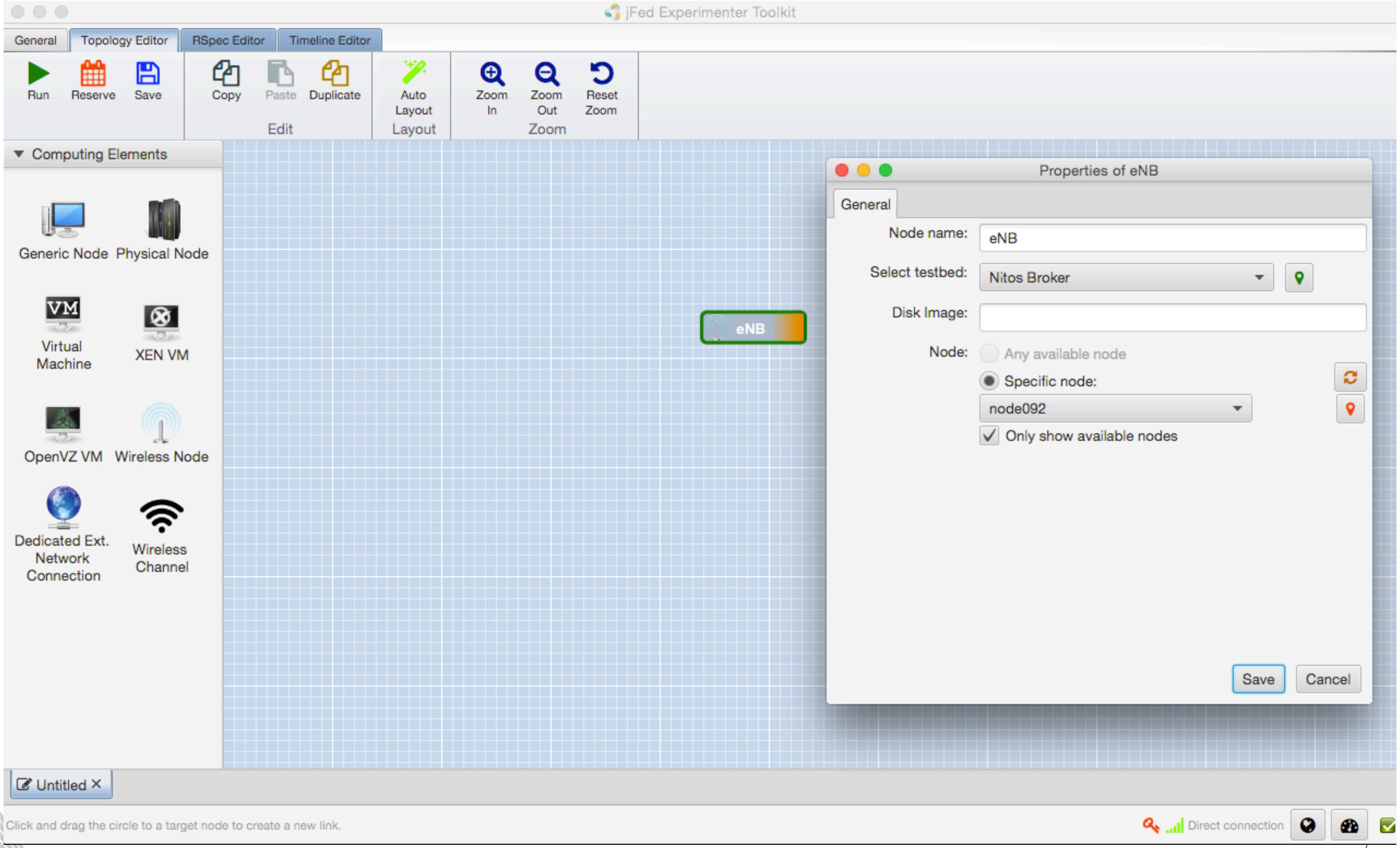
Password:

Enter the password associated with the certificate

JFED


jFed Experimenter Toolkit
General
Topology Editor
RSpec Editor
Timeline Editor
Run
Reserve
Save
Copy
Paste
Duplicate
Edit
Auto Layout
Layout
Zoom In
Zoom Out
Reset Zoom
Zoom
Computing Elements
Generic Node
Physical Node
Virtual Machine
XEN VM
OpenVZ VM
Wireless Node
Dedicated Ext. Network Connection
Wireless Channel
Untitled
Direct connection

JFED



Properties of eNB

General

Node name: eNB

Select testbed: Nitos Broker

Disk Image:

Node: ○ Any available node

● Specific node:

node092

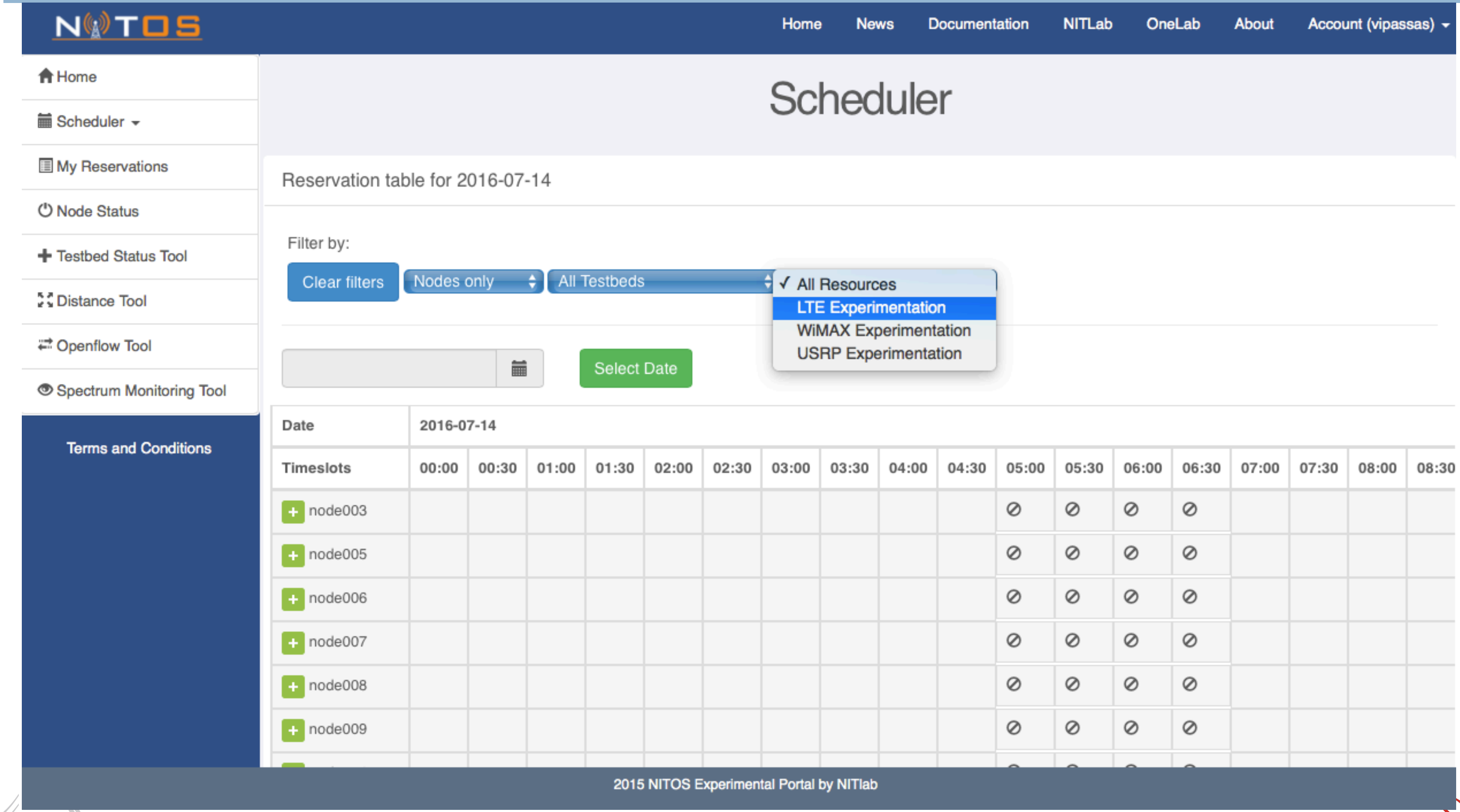
☑ Only show available nodes

Click and drag the circle to a target node to create a new link.

NITOS Scheduler

- □ Access to the NITOS server via the NITOS portal.
- □ Available at http://nitos.inf.uth.gr
- □ REST interface for reserving LTE resources.
- □ Advanced filtering of resources (e.g. LTE, USRPs, indoor/outdoor, etc.).
- □ Communicates directly with the NITOS Broker in order to handle the reservations and access to the testbed.

Making a reservation (1/2)



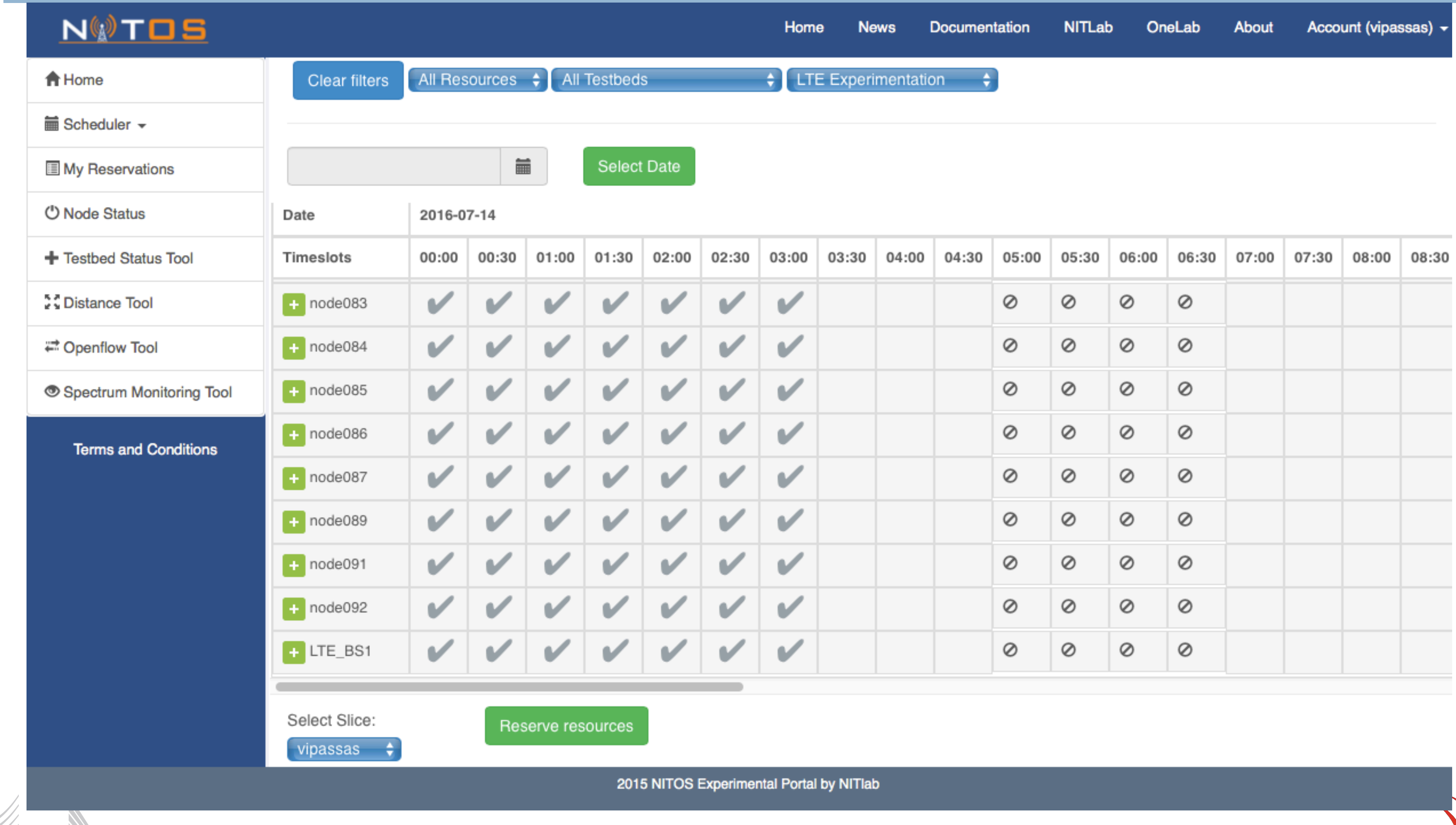
Scheduler

Reservation table for 2016-07-14

Filter by:

Date	2016-07-14																	
Timeslots	00:00	00:30	01:00	01:30	02:00	02:30	03:00	03:30	04:00	04:30	05:00	05:30	06:00	06:30	07:00	07:30	08:00	08:30
+ node003											⊘	⊘	⊘	⊘				
+ node005											⊘	⊘	⊘	⊘				
+ node006											⊘	⊘	⊘	⊘				
+ node007											⊘	⊘	⊘	⊘				
+ node008											⊘	⊘	⊘	⊘				
+ node009											⊘	⊘	⊘	⊘				

Making a reservation (2/2)



NITOS

Date	2016-07-14																	
Timeslots	00:00	00:30	01:00	01:30	02:00	02:30	03:00	03:30	04:00	04:30	05:00	05:30	06:00	06:30	07:00	07:30	08:00	08:30
+ node083	✓	✓	✓	✓	✓	✓	✓				⊘	⊘	⊘	⊘				
+ node084	✓	✓	✓	✓	✓	✓	✓				⊘	⊘	⊘	⊘				
+ node085	✓	✓	✓	✓	✓	✓	✓				⊘	⊘	⊘	⊘				
+ node086	✓	✓	✓	✓	✓	✓	✓				⊘	⊘	⊘	⊘				
+ node087	✓	✓	✓	✓	✓	✓	✓				⊘	⊘	⊘	⊘				
+ node089	✓	✓	✓	✓	✓	✓	✓				⊘	⊘	⊘	⊘				
+ node091	✓	✓	✓	✓	✓	✓	✓				⊘	⊘	⊘	⊘				
+ node092	✓	✓	✓	✓	✓	✓	✓				⊘	⊘	⊘	⊘				
+ LTE_BS1	✓	✓	✓	✓	✓	✓	✓				⊘	⊘	⊘	⊘				

Select Slice: vipassas

2015 NITOS Experimental Portal by NITlab

LTE equipment

- 4 different types of nodes with 2 different PLMNIDs
  - 46099 PLMN for the commercial setup
  - 20893 PLMN for the OAI setup
- Different types of UEs with different behaviour
- Expanding the resources in the scheduler will reveal what type of dongle they have mounted on
- Cross checking it with the NITOS documentation will be enough for learning each node's IMSI

Understanding the topology

- Each node has two Ethernet interfaces
  - 1st intf: Control Network (10.0.1.0/24)
  - 2nd intf: Experimental Network (Free addressing)
- The LTE network is communicating over the experimental network
  - Femtocell is using 192.168.200.1/24
  - EPC is using 192.168.200.200/24

Preparing the testbed

□ Once your reservation starts, you can load an image on each node

□ All the available images are located in the /var/lib/omf-images-5.4/ folder of each server

□ Images that start with the *baseline_* prefix are prepared by us and should be the most stable, using any new tools

Loading an image on the nodes

- Image loading is handled by omf
- e.g.

  ```
  nimakris@nitlab3:~$ omf load –t node0XX,node0YY –i baseline_icarus_lte.ndz
  ```
- At the end of your experiment, you can save your image if you want to continue your work in the next timeslot
- e.g.

  ```
  nimakris@nitlab3:~$ omf save –n node055
  ```
- Saving the image will save it with a timestamp at the /var/lib/omf-images-5.4/ folder
- You can rename it so as to use a more user-friendly name

Preparing the LTE testbed

- The femtocell is by default turned off at the beginning of your reservation
- Turn it on by using the "omf tell" command
  - E.g. omf tell –a on –t e_node_b_001
- You can query for its status by using the "omf stat" command
  - E.g. omf stat –t e_node_b_001
- Unless the femtocell is on, all the commands sent to it will fail

Preparing the LTE testbed

- Always reset the femtocell and EPC to their default settings
- Using the LTErf service, send from the NITOS server the following commands:
  - wget -qO- ‘http://lterf:5054/lterf/bs/default?node=1’
  - wget -qO- ‘http://lterf:5054/lterf/bs/restart?node=1’
  - wget -qO- ‘http://lterf:5054/lterf/epc/default’
  - wget -qO- ‘http://lterf:5054/lterf/epc/restart’
  - wget -qO- "http://lterf:5054/lterf/epc/get?function=getSystemStatus" | xml_pp

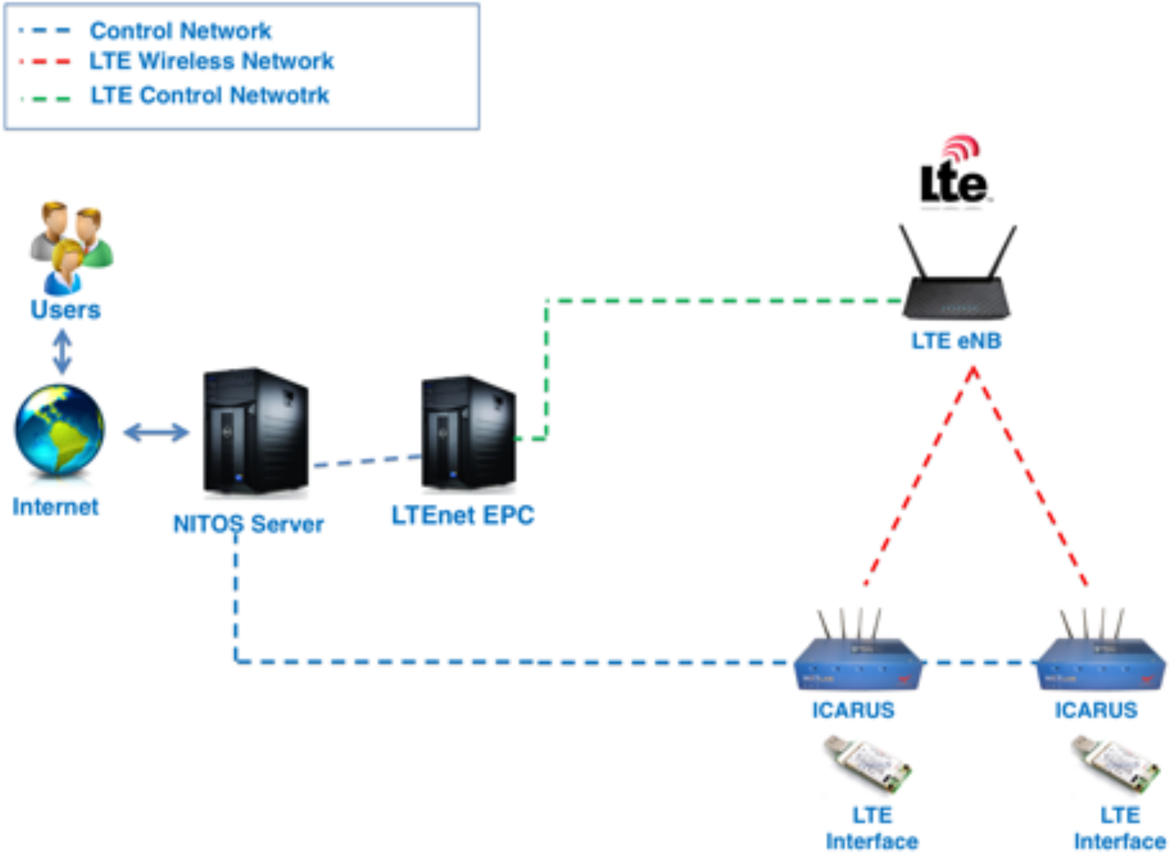
Overview

- □ FLEX project
- □ Tools for this tutorial
- □ Making a Reservation
- □ Experiment using the Commercial Setup
- □ Tutorial with the Open Source Setup (OAI)
- □ Experiment using LTE-U and Wi-Fi

1st Experiment

- For this experiment we will use
  - The commercial EPC
  - One commercial femtocell as eNB
  - Two commercial LTE-Dongles as UEs
- The scenario of this experiment is to connect the 2 UEs to the LTE network
- Then start some measurement applications sending traffic between the two UEs

Commercial Setup



Commercial Setup

- Setting up a sample experiment:
  - Load an LTE UE compatible image on the nodes
  - E.g. omf load –t node052 –i baseline_icarus_lte.ndz
- Login on the node:
  - ssh root@node052
- Turn on the dongle (E3272)
  - lte_dongle –o
- Bring it to a configurable state
  - lte_dongle –s

Connecting the LTE dongle

- Send AT commands over the serial to the dongle
- Minicom –D /dev/ttyUSB0
  at+cgdcont=1,1,"default"
  at^ndisdup=1,1,"default"
  at^dhcp?
- If the dongle reports a HEX string, it is successfully connected to the network
- Run dhclient on the wwan0 interface to get an IP address
  - dhclient wwan0
  - Ifconfig wwan0 netmask 255.255.255.0 –arp up
- Default APN of the NITOS testbed is using the 10.0.3.0/24 network, with 10.0.3.1 being the PGW
- Add a route to this network
  - route add –net 10.0.3.0/24 dev wwan0

Running the OMF Experiment (1/4)

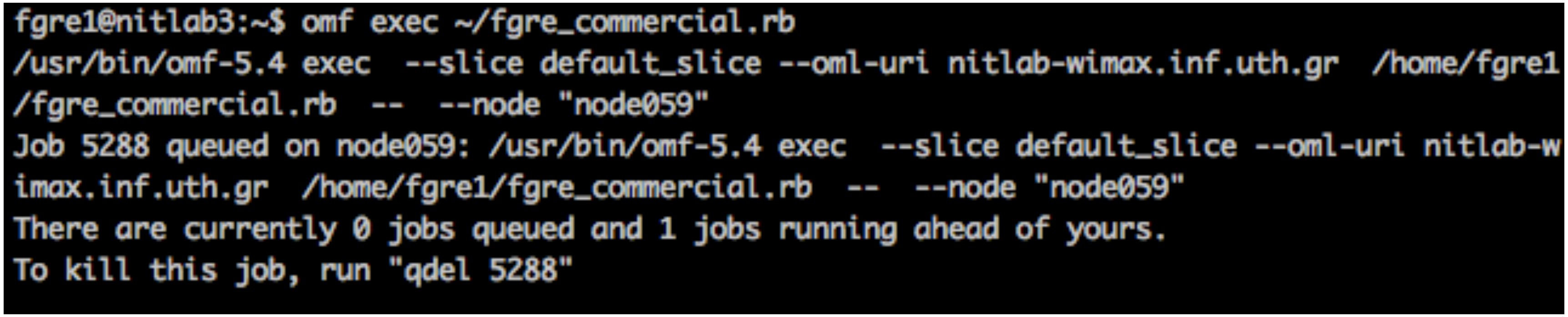
- To run the experiment you will have to remotely connect to the NITOS server
- In order to do that you will have to use an application
  - For windows download the **Mobaxterm** application (there is a portable version) or **Putty**
  - For Linux/Mac open the application **Terminal**

Running the OMF Experiment (2/4)

- ssh fgre**Y**@nitlab3.inf.uth.gr
- Password: fgre**Y**_nitos
  - If **Y**=1
    - ssh fgre1@nitlab3.inf.uth.gr
    - Password: fgre1_nitos
- ssh fgre**Y**@irodsweb.inf.uth.gr
- Password: fgre**Y**_nitos

Running the OMF Experiment (3/4)

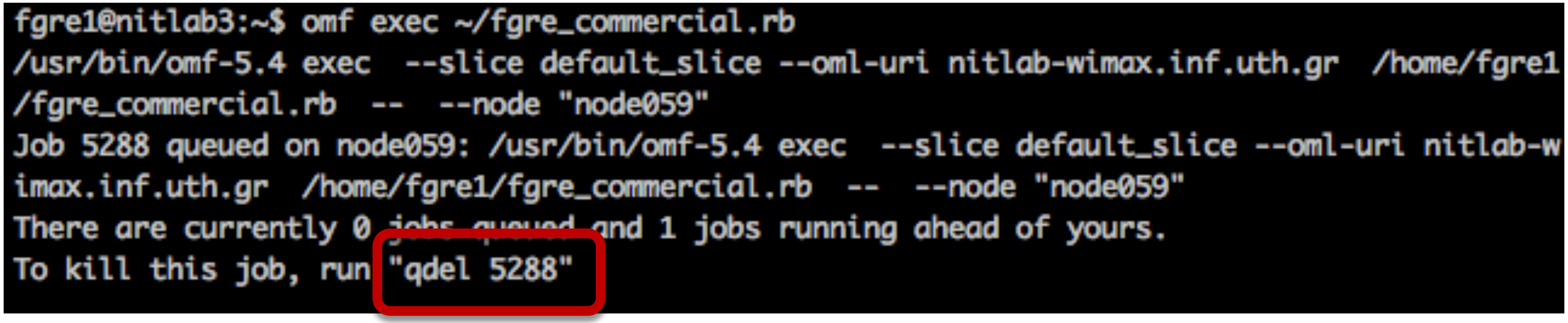
- We have employed a queue system because you will use the same resources for this experiment
- Run the following command:
  - omf exec ~/fgre_commercial.rb
- The output will be like the following



```
fgre1@nitlab3:~$ omf exec ~/fgre_commercial.rb
/usr/bin/omf-5.4 exec  --slice default_slice --oml-uri nitlab-wimax.inf.uth.gr  /home/fgre1
/fgre_commercial.rb  --  --node "node059"
Job 5288 queued on node059: /usr/bin/omf-5.4 exec  --slice default_slice --oml-uri nitlab-w
imax.inf.uth.gr  /home/fgre1/fgre_commercial.rb  --  --node "node059"
There are currently 0 jobs queued and 1 jobs running ahead of yours.
To kill this job, run "qdel 5288"
```

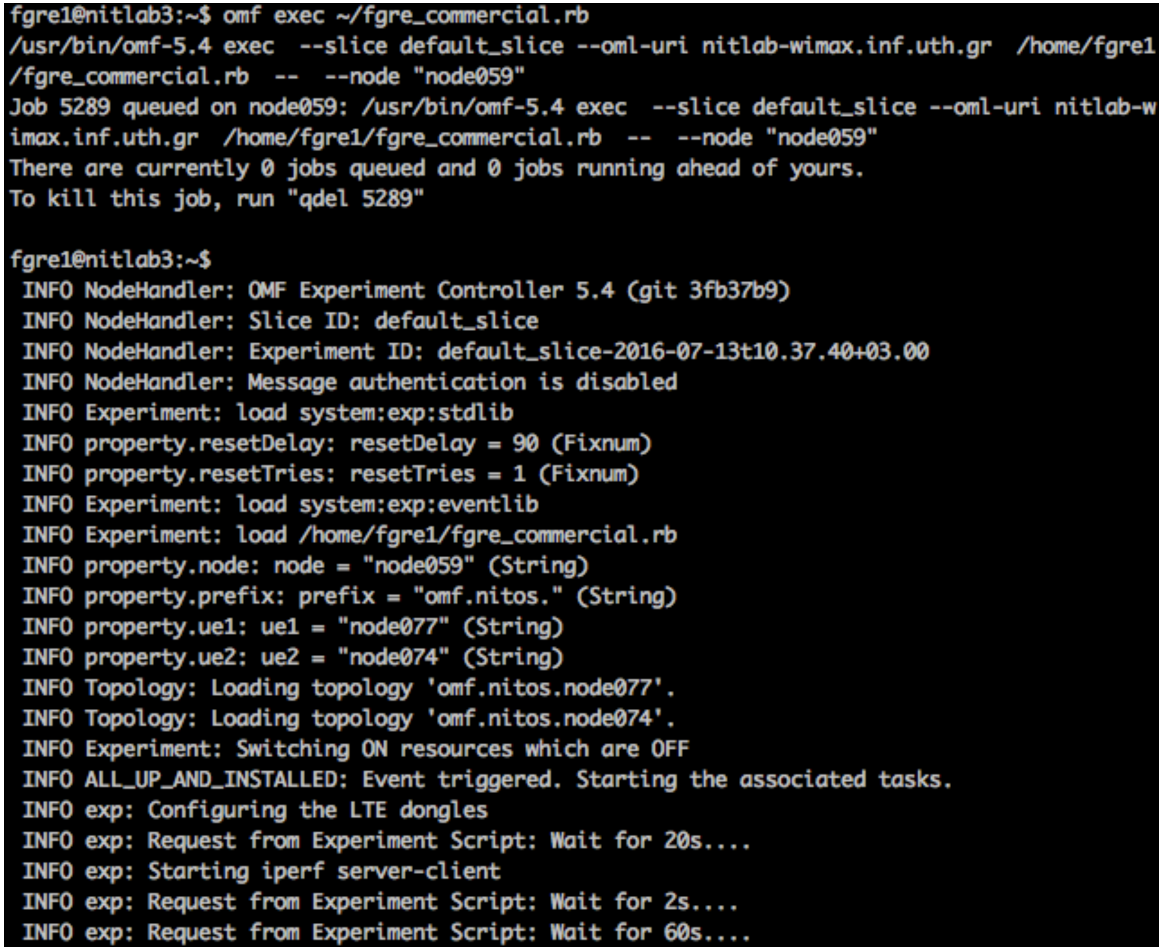
Running the OMF Experiment (4/4)

- Run it once and wait for the experiment to start
  - It may start immediately, or you may have to wait for a few minutes
- If you press more than one the command then cancel the command with qdel **XXXX**



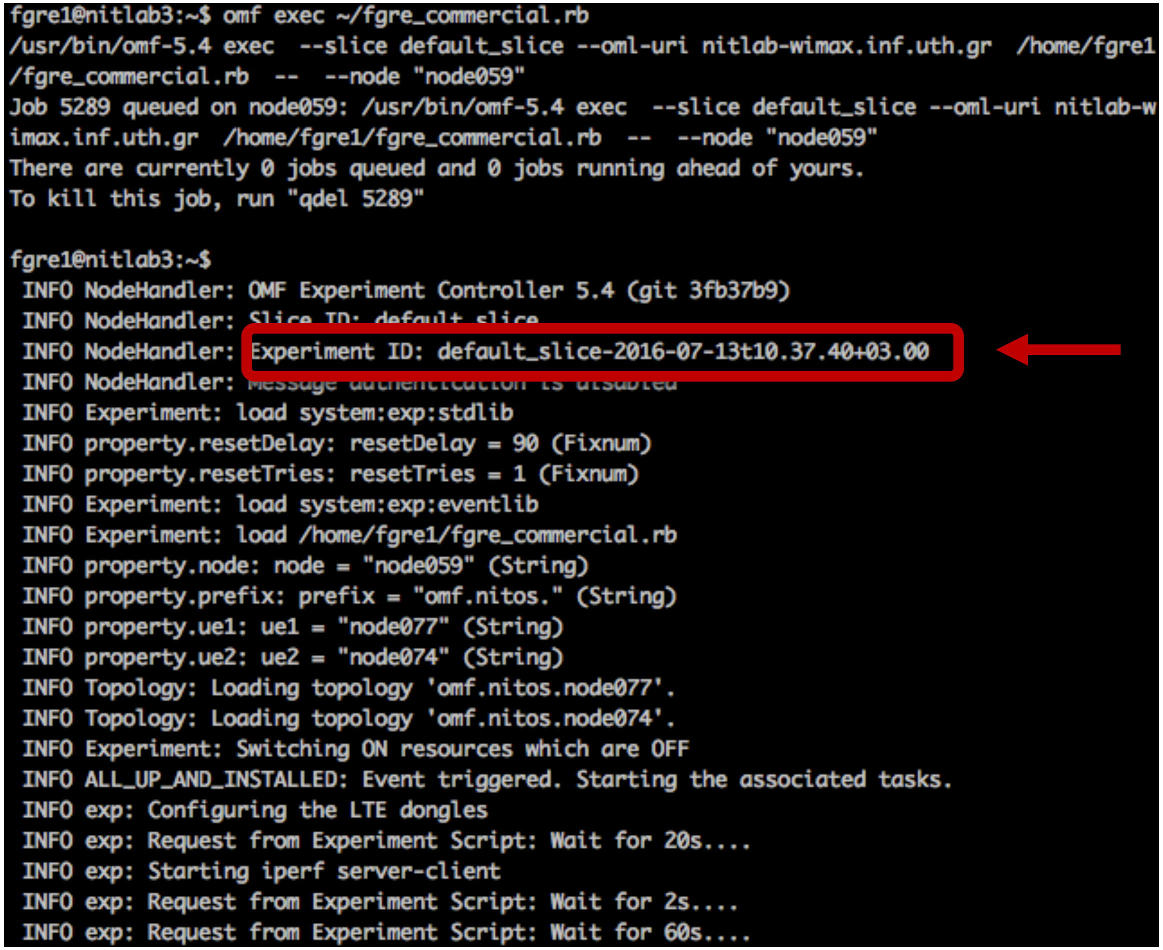
```
fgre1@nitlab3:~$ omf exec ~/fgre_commercial.rb
/usr/bin/omf-5.4 exec  --slice default_slice --oml-uri nitlab-wimax.inf.uth.gr  /home/fgre1
/fgre_commercial.rb  --  --node "node059"
Job 5288 queued on node059: /usr/bin/omf-5.4 exec  --slice default_slice --oml-uri nitlab-w
imax.inf.uth.gr  /home/fgre1/fgre_commercial.rb  --  --node "node059"
There are currently 0 jobs queued and 1 jobs running ahead of yours.
To kill this job, run "qdel 5288"
```

Example Experiment Run



```
fgre1@nitlab3:~$ omf exec ~/fgre_commercial.rb
/usr/bin/omf-5.4 exec  --slice default_slice --oml-uri nitlab-wimax.inf.uth.gr  /home/fgre1
/fgre_commercial.rb  --  --node "node059"
Job 5289 queued on node059: /usr/bin/omf-5.4 exec  --slice default_slice --oml-uri nitlab-w
imax.inf.uth.gr  /home/fgre1/fgre_commercial.rb  --  --node "node059"
There are currently 0 jobs queued and 0 jobs running ahead of yours.
To kill this job, run "qdel 5289"

fgre1@nitlab3:~$
 INFO NodeHandler: OMF Experiment Controller 5.4 (git 3fb37b9)
 INFO NodeHandler: Slice ID: default_slice
 INFO NodeHandler: Experiment ID: default_slice-2016-07-13t10.37.40+03.00
 INFO NodeHandler: Message authentication is disabled
 INFO Experiment: load system:exp:stdlib
 INFO property.resetDelay: resetDelay = 90 (Fixnum)
 INFO property.resetTries: resetTries = 1 (Fixnum)
 INFO Experiment: load system:exp:eventlib
 INFO Experiment: load /home/fgre1/fgre_commercial.rb
 INFO property.node: node = "node059" (String)
 INFO property.prefix: prefix = "omf.nitos." (String)
 INFO property.ue1: ue1 = "node077" (String)
 INFO property.ue2: ue2 = "node074" (String)
 INFO Topology: Loading topology 'omf.nitos.node077'.
 INFO Topology: Loading topology 'omf.nitos.node074'.
 INFO Experiment: Switching ON resources which are OFF
 INFO ALL_UP_AND_INSTALLED: Event triggered. Starting the associated tasks.
 INFO exp: Configuring the LTE dongles
 INFO exp: Request from Experiment Script: Wait for 20s....
 INFO exp: Starting iperf server-client
 INFO exp: Request from Experiment Script: Wait for 2s....
 INFO exp: Request from Experiment Script: Wait for 60s....
```

Example Experiment Run



```
fgre1@nitlab3:~$ omf exec ~/fgre_commercial.rb
/usr/bin/omf-5.4 exec  --slice default_slice --oml-uri nitlab-wimax.inf.uth.gr  /home/fgre1
/fgre_commercial.rb  --  --node "node059"
Job 5289 queued on node059: /usr/bin/omf-5.4 exec  --slice default_slice --oml-uri nitlab-w
imax.inf.uth.gr  /home/fgre1/fgre_commercial.rb  --  --node "node059"
There are currently 0 jobs queued and 0 jobs running ahead of yours.
To kill this job, run "qdel 5289"

fgre1@nitlab3:~$
 INFO NodeHandler: OMF Experiment Controller 5.4 (git 3fb37b9)
 INFO NodeHandler: Slice ID: default_slice
 INFO NodeHandler: Experiment ID: default_slice-2016-07-13t10.37.40+03.00
 INFO NodeHandler: Message authentication is disabled
 INFO Experiment: load system:exp:stdlib
 INFO property.resetDelay: resetDelay = 90 (Fixnum)
 INFO property.resetTries: resetTries = 1 (Fixnum)
 INFO Experiment: load system:exp:eventlib
 INFO Experiment: load /home/fgre1/fgre_commercial.rb
 INFO property.node: node = "node059" (String)
 INFO property.prefix: prefix = "omf.nitos." (String)
 INFO property.ue1: ue1 = "node077" (String)
 INFO property.ue2: ue2 = "node074" (String)
 INFO Topology: Loading topology 'omf.nitos.node077'.
 INFO Topology: Loading topology 'omf.nitos.node074'.
 INFO Experiment: Switching ON resources which are OFF
 INFO ALL_UP_AND_INSTALLED: Event triggered. Starting the associated tasks.
 INFO exp: Configuring the LTE dongles
 INFO exp: Request from Experiment Script: Wait for 20s....
 INFO exp: Starting iperf server-client
 INFO exp: Request from Experiment Script: Wait for 2s....
 INFO exp: Request from Experiment Script: Wait for 60s....
```

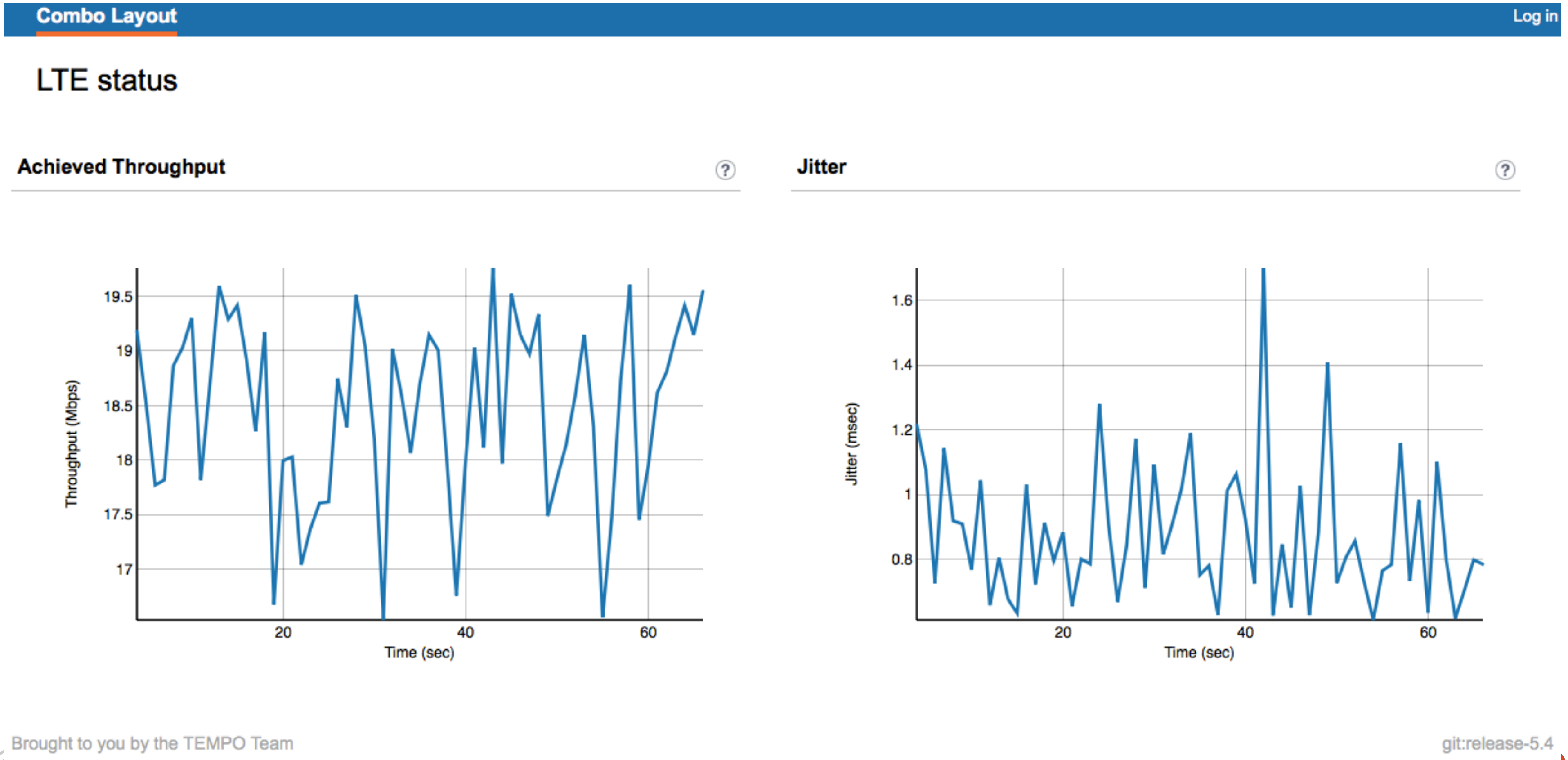
ExpID

OMF Web

- The **ExpID** is different for every experiment, save it because you will need it for the next step
- In order to visualize the results of the experiment open a new terminal and connect to the following server
  - ssh fgre**Y**@nitlab-wimax.inf.uth.gr
  - Password: fgre**Y**_nitos
  - cd omf_web_nitos_fgre2016/
  - ruby1.9.1 -I lib/ example/fgre/simple_viz_server.rb start -p 300**Z** --db **ExpID**

OMF Web GUI

- □ Visit irodsweb.inf.uth.gr:300Z



LTE status

Achieved Throughput

Jitter

Brought to you by the TEMPO Team

git:release-5.4

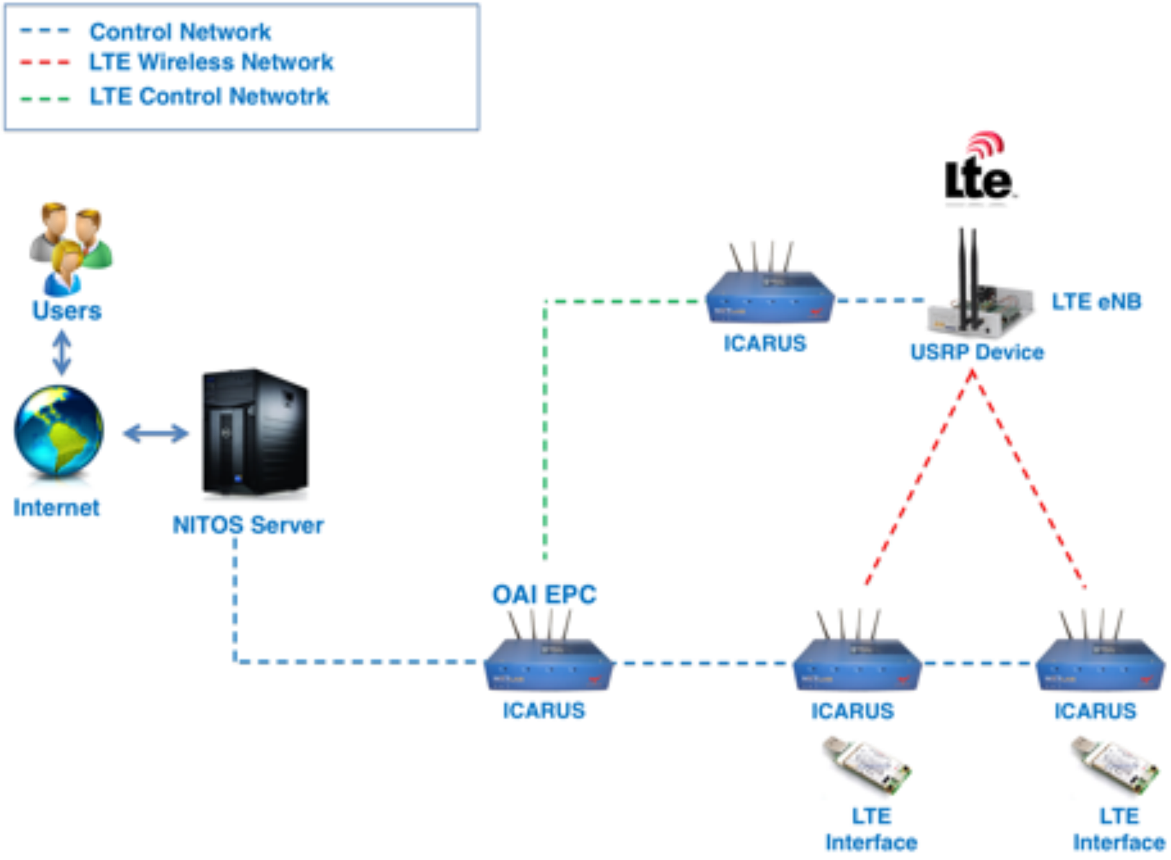
Overview

- □ FLEX project
- □ Tools for this tutorial
- □ Making a Reservation
- □ Experiment using the Commercial Setup
- □ Tutorial with the Open Source Setup (OAI)
- □ Experiment using LTE-U and Wi-Fi

OAI tutorial

- For this tutorial we will use
  - The openair-cn as EPC
  - One node employed with USRP B210 and OAI as eNB
  - Two commercial LTE-Dongles as UEs
- The scenario is the same with the previous experiment

Open-Source Setup



Running OAI on NITOS

- □ You can load one of the compatible OAI images on any B210 enabled node
- □ Any other node can be used for running the EPC software
- □ PLMNs and clients that can be used with OAI are freely used and altered via its .conf file
- □ As the baseline images can be setup with different nodes, you will need to change the conf files for EPC, HSS and eNB

Client Connection

- Once all entities are connected, you will be able to list the network at the UE side
- If the UE is present at the database, then you can connect it to the network
- Otherwise you will have to insert it manually on the HSS entity and try again to connect the UE

Experimenting with OAI (1/2)

- Various RF frontend can be used by the OAI eNodeB such as EXMIMO2, USRP B210/X3100, bladerf, limesdr
- OAI eNodeB is operating giving as an input a configuration file where you can define:
  - PLMN
  - Band
  - PRB – Bandwidth
  - TX/RX Gain
  - MME IP address, etc

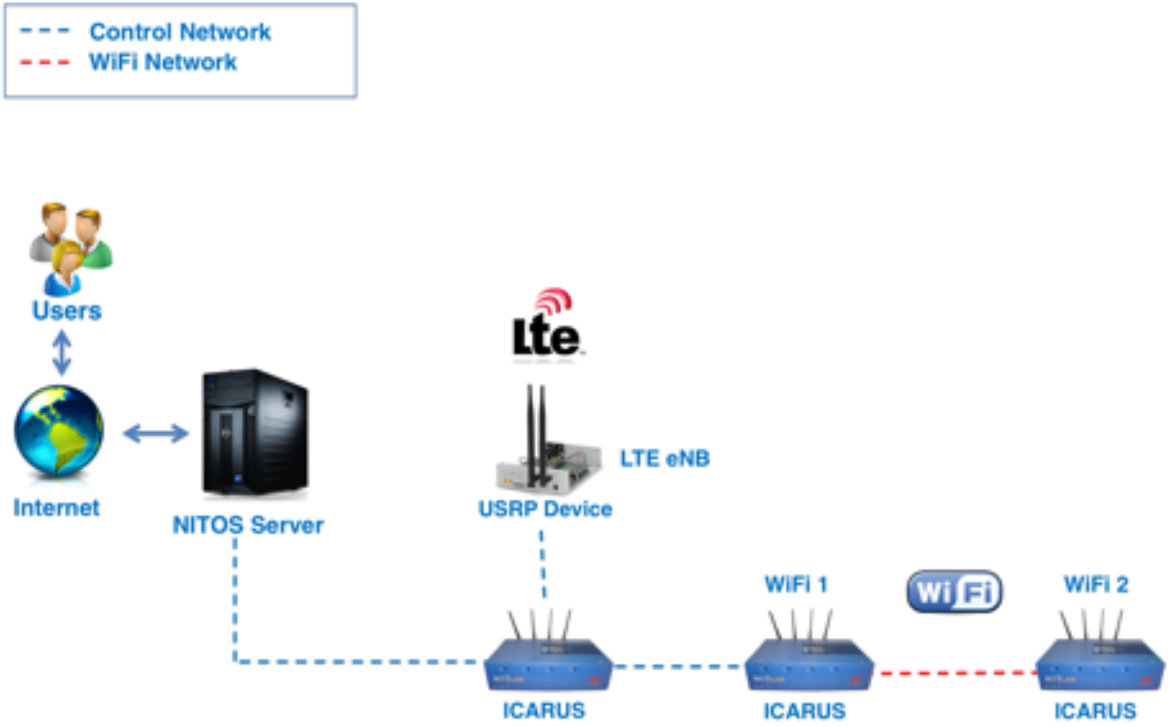
Experimenting with OAI (2/2)

- For the core network (OAI EPC) you can changing anything, e.g. scheduling mechanism, CQI mechanism, etc
- EPC also has a configuration file where you can define the interfaces and the IPs that you will use for the communication between the MME and the eNB, the HSS, the PGW. Also you can define the IP addressing pool for the UEs and a lot of other settings
- For adding a user to the HSS database there are two options either by mysql queries or graphical interface using phpmyadmin

Overview

- □ FLEX project
- □ Tools for this tutorial
- □ Making a Reservation
- □ Experiment using the Commercial Setup
- □ Tutorial with the Open Source Setup (OAI)
- □ Experiment using LTE-U and Wi-Fi

LTE-U Setup

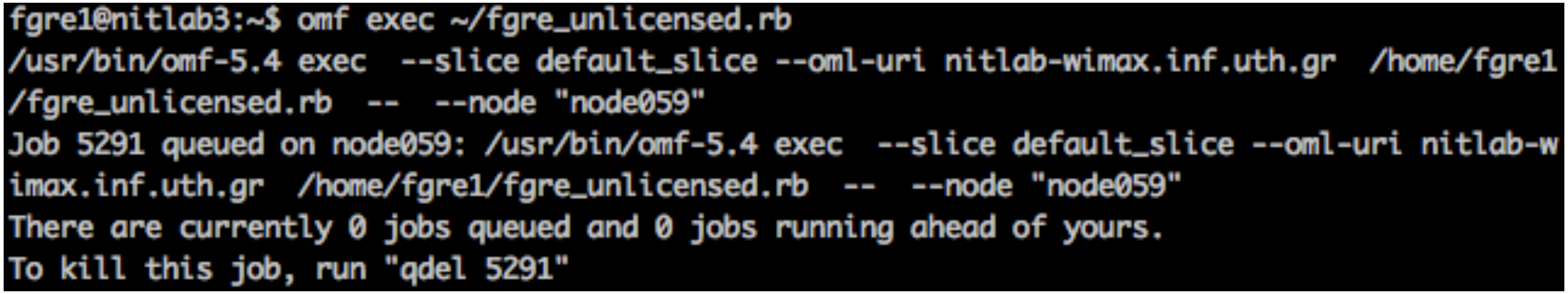

Control Network
WiFi Network
Users
Internet
NITOS Server
Lte
LTE eNB
USRP Device
WiFi 1
Wi Fi
WiFi 2
ICARUS
ICARUS
ICARUS

2nd Experiment

- For this experiment will use
  - One node employed with USRP B210 and OAI as eNB
  - Two wireless nodes for setting up a WiFi network
- The scenario of this experiment is to connect the 2 wireless nodes in ad-hoc mode
- Start some measurement applications sending traffic between the two nodes
- Then activate the LTE eNB in the same frequency as WiFi and monitor how LTE affects the WiFi communication between the two nodes

Running the OMF Experiment (1/2)

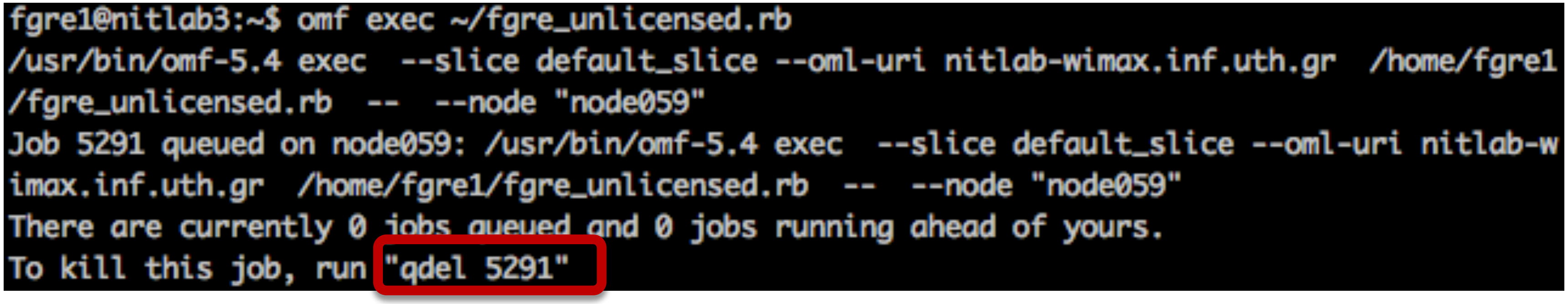
- We have employed a queue system because you will use the same resources for this experiment
- Run the following command:
  - omf exec ~/fgre_unlicensed.rb
- The output will be like the following



```
fgre1@nitlab3:~$ omf exec ~/fgre_unlicensed.rb
/usr/bin/omf-5.4 exec  --slice default_slice --oml-uri nitlab-wimax.inf.uth.gr  /home/fgre1
/fgre_unlicensed.rb  --  --node "node059"
Job 5291 queued on node059: /usr/bin/omf-5.4 exec  --slice default_slice --oml-uri nitlab-w
imax.inf.uth.gr  /home/fgre1/fgre_unlicensed.rb  --  --node "node059"
There are currently 0 jobs queued and 0 jobs running ahead of yours.
To kill this job, run "qdel 5291"
```

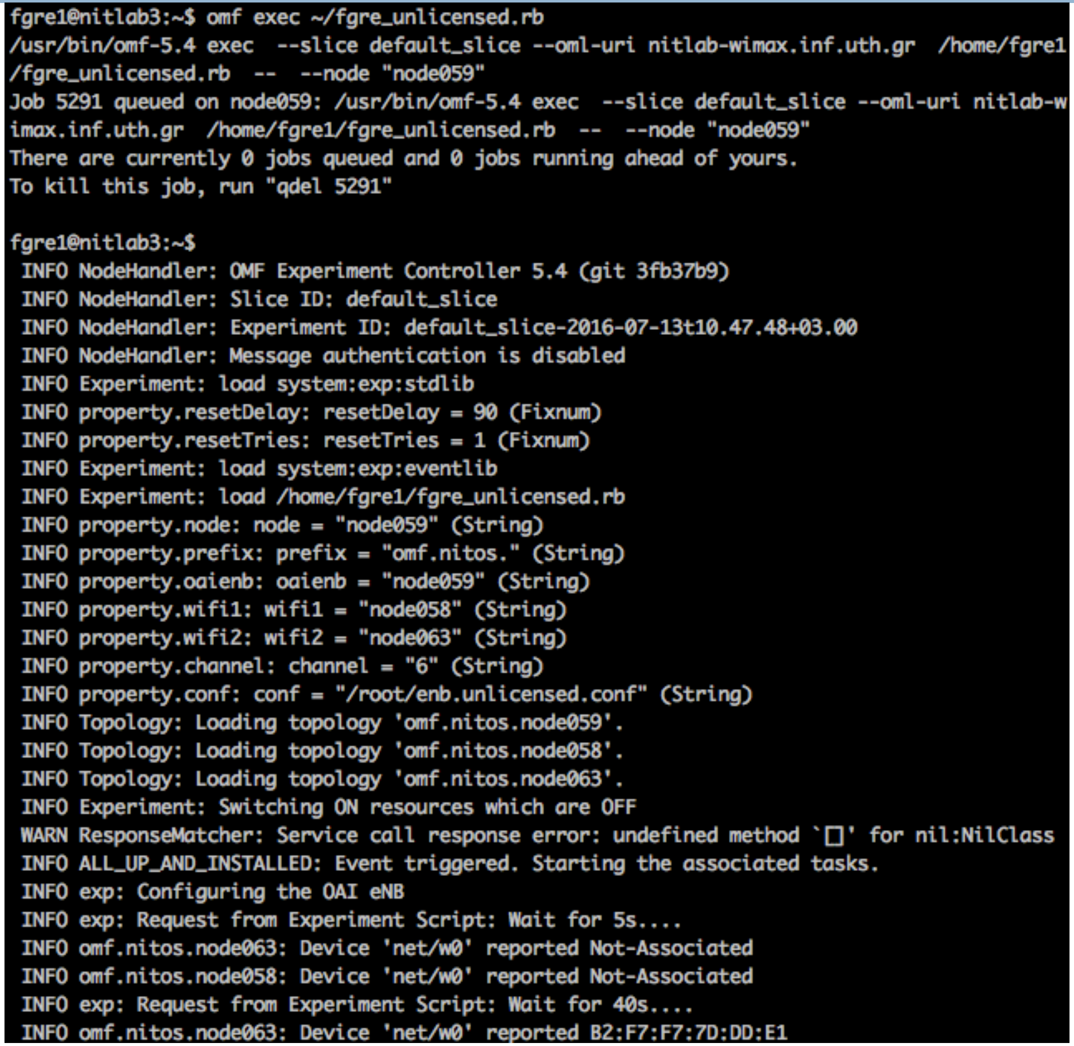
Running the OMF Experiment (2/2)

- Run it once and wait for the experiment to start
  - It may start immediately, or you may have to wait for a few minutes
- If you press more than one the command then cancel the command with qdel **XXXX**



```
fgre1@nitlab3:~$ omf exec ~/fgre_unlicensed.rb
/usr/bin/omf-5.4 exec  --slice default_slice --oml-uri nitlab-wimax.inf.uth.gr  /home/fgre1
/fgre_unlicensed.rb  --  --node "node059"
Job 5291 queued on node059: /usr/bin/omf-5.4 exec  --slice default_slice --oml-uri nitlab-w
imax.inf.uth.gr  /home/fgre1/fgre_unlicensed.rb  --  --node "node059"
There are currently 0 jobs queued and 0 jobs running ahead of yours.
To kill this job, run "qdel 5291"
```

Example Experiment Run



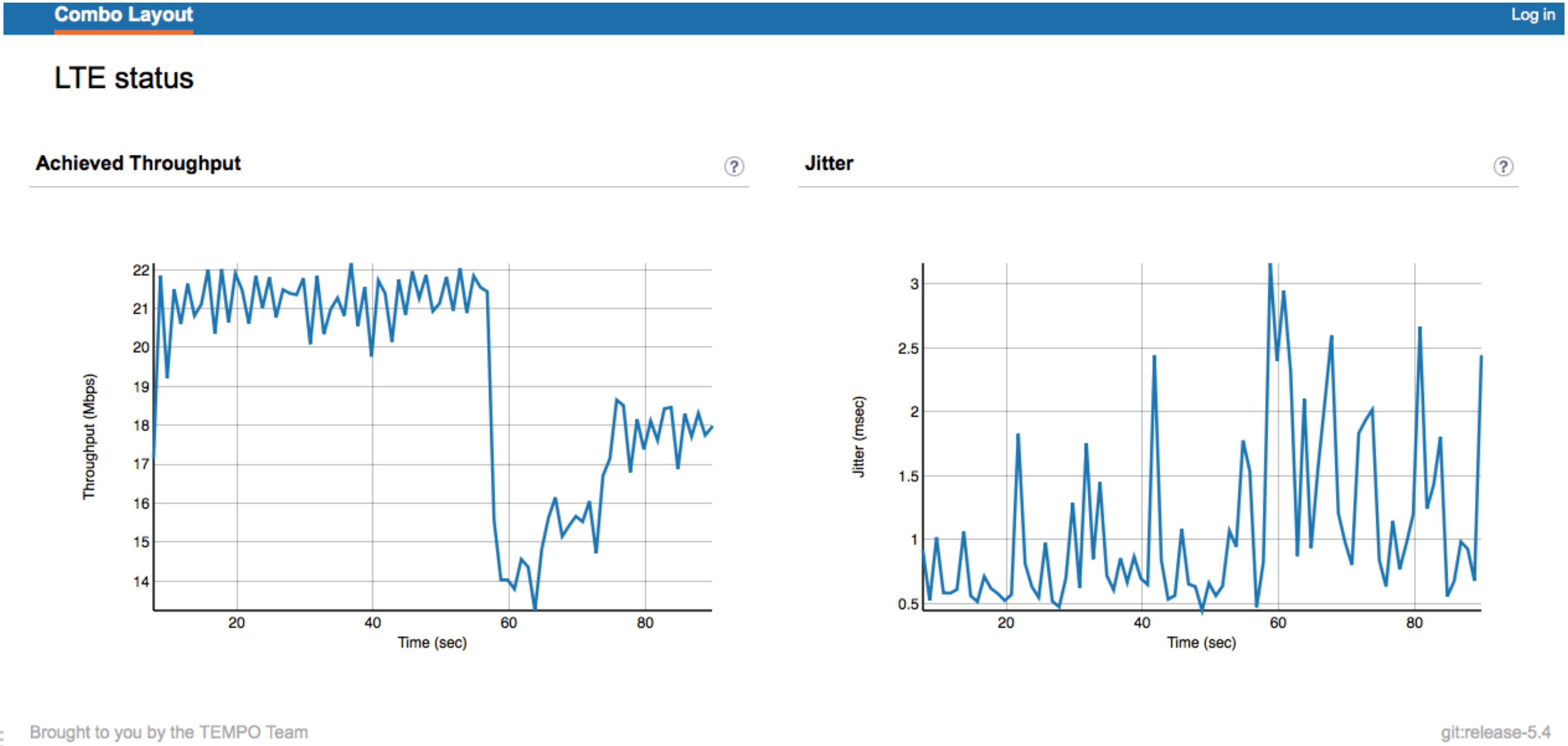
```
fgre1@nitlab3:~$ omf exec ~/fgre_unlicensed.rb
/usr/bin/omf-5.4 exec  --slice default_slice --oml-uri nitlab-wimax.inf.uth.gr  /home/fgre1
/fgre_unlicensed.rb  --  --node "node059"
Job 5291 queued on node059: /usr/bin/omf-5.4 exec  --slice default_slice --oml-uri nitlab-w
imax.inf.uth.gr  /home/fgre1/fgre_unlicensed.rb  --  --node "node059"
There are currently 0 jobs queued and 0 jobs running ahead of yours.
To kill this job, run "qdel 5291"

fgre1@nitlab3:~$
 INFO NodeHandler: OMF Experiment Controller 5.4 (git 3fb37b9)
 INFO NodeHandler: Slice ID: default_slice
 INFO NodeHandler: Experiment ID: default_slice-2016-07-13t10.47.48+03.00
 INFO NodeHandler: Message authentication is disabled
 INFO Experiment: load system:exp:stdlib
 INFO property.resetDelay: resetDelay = 90 (Fixnum)
 INFO property.resetTries: resetTries = 1 (Fixnum)
 INFO Experiment: load system:exp:eventlib
 INFO Experiment: load /home/fgre1/fgre_unlicensed.rb
 INFO property.node: node = "node059" (String)
 INFO property.prefix: prefix = "omf.nitos." (String)
 INFO property.oaienb: oaienb = "node059" (String)
 INFO property.wifi1: wifi1 = "node058" (String)
 INFO property.wifi2: wifi2 = "node063" (String)
 INFO property.channel: channel = "6" (String)
 INFO property.conf: conf = "/root/enb.unlicensed.conf" (String)
 INFO Topology: Loading topology 'omf.nitos.node059'.
 INFO Topology: Loading topology 'omf.nitos.node058'.
 INFO Topology: Loading topology 'omf.nitos.node063'.
 INFO Experiment: Switching ON resources which are OFF
 WARN ResponseMatcher: Service call response error: undefined method `[]' for nil:NilClass
 INFO ALL_UP_AND_INSTALLED: Event triggered. Starting the associated tasks.
 INFO exp: Configuring the OAI eNB
 INFO exp: Request from Experiment Script: Wait for 5s....
 INFO omf.nitos.node063: Device 'net/w0' reported Not-Associated
 INFO omf.nitos.node058: Device 'net/w0' reported Not-Associated
 INFO exp: Request from Experiment Script: Wait for 40s....
 INFO omf.nitos.node063: Device 'net/w0' reported B2:F7:F7:7D:DD:E1
```

OMF Web

- The **ExpID** is different for every experiment, save it because you will need it for the next step
- In order to visualize the results of the experiment open a new terminal and connect to the following server
  - ssh fgre**Y**@irodsweb.inf.uth.gr
  - Password: fgre**Y**_nitos
  - cd omf_web_nitos_fgre2016/
  - ruby1.9.1 -I lib/ example/fgre/simple_viz_server.rb start -p 300**Z** --db **ExpID**

OMF Web GUI

- Visit irodsweb.inf.uth.gr:300Z



Combo Layout

Log in

LTE status

Achieved Throughput

Jitter

Brought to you by the TEMPO Team

git:release-5.4

Questions?