



NEPI: Newtork Experiment Programming Interface

Alina Quereilhac

INRIA

Sophia Antipolis, France

Network Experiment Management

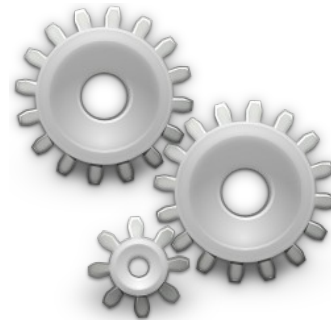
- NEPI is a framework to manage network experiments

Design



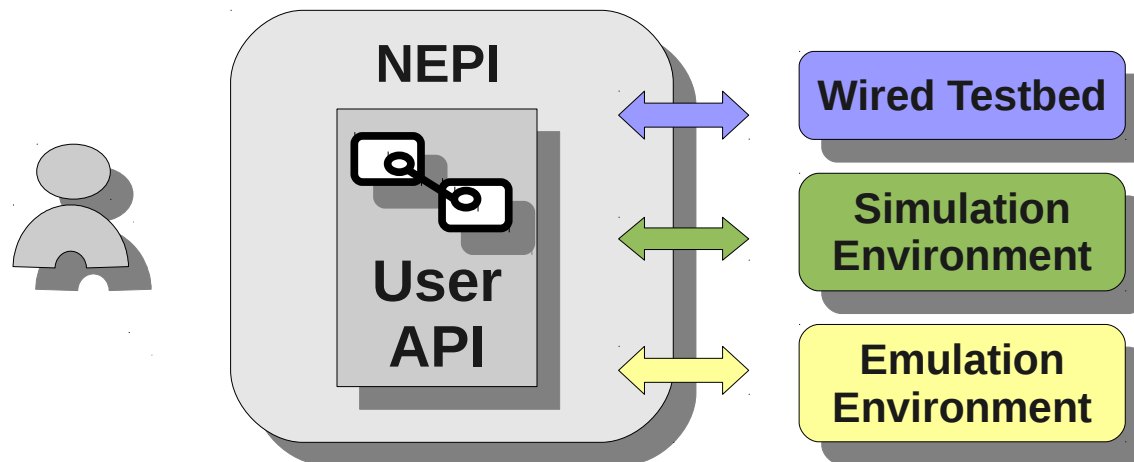
(Offline)

Execution



Uniform API for different environments

- NEPI presents a uniform user API to manage resources from arbitrary experimentation environments (Physical testbeds, emulators, simulators)
- NEPI enables to easily mix heterogeneous resources in a same experiment (real, emulated and simulated resources)





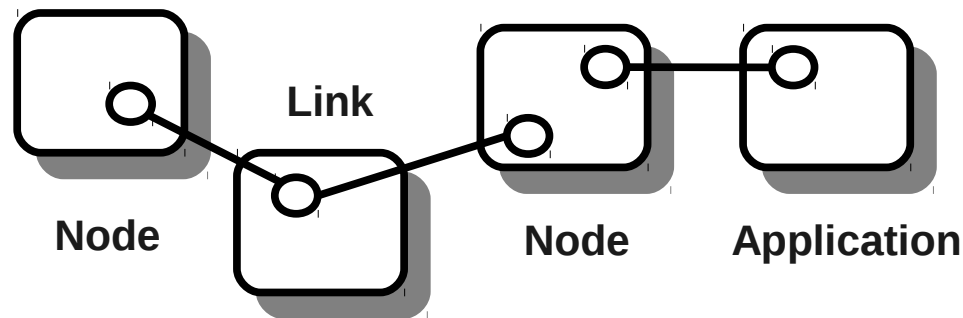
Experiment Design

- Describe the experiment with enough detail to enable reproduction
 - Describe resources to be provisioned (nodes, switches, apps... requirements they have)
 - Describe how resources interact
 - Describe how resources behave (start/stop)
 - Describe results to be collected



Experiment Design in NEPI

- An experiment is described as a graph of 'Boxes and Connectors'
- Boxes have attributes
- Boxes are associated to traces (results)
- Boxes belong to a backend (environment)





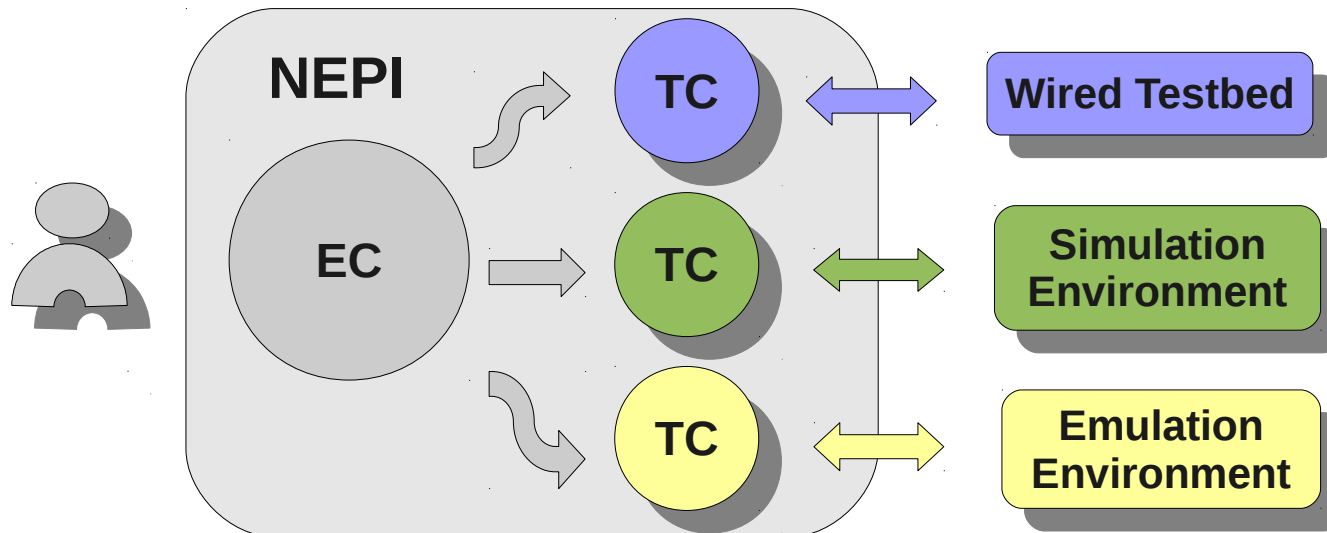
Experiment Execution

- Different aspects of execution
 - Deployment
 - Resource discovery and provision
 - Resource configuration
 - Software installation
 - Application launch
 - Control
 - Start/stop applications
 - Change resource configuration
 - Result collection
- An Experiment Controller is the entity that takes care of experiment execution



Experiment Execution in NEPI

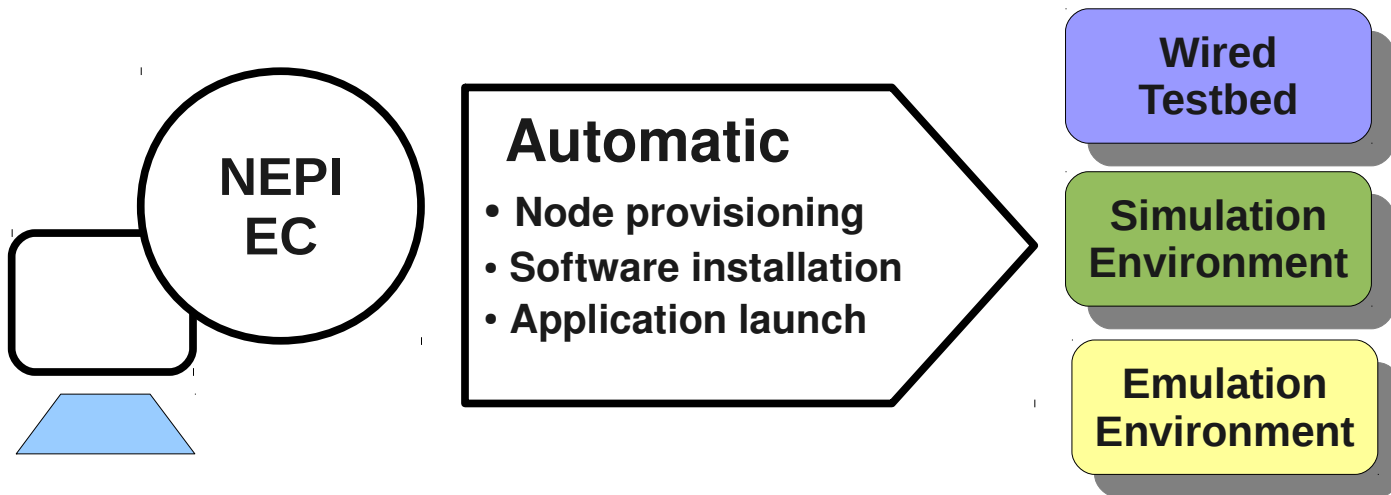
- NEPI uses two levels of controllers
 - One global generic experiment controller (EC)
 - Many specific testbed controllers (TC)
- New environments can be supported by implementing new testbed controllers





Experiment Execution in NEPI

- NEPI EC can be launched from a user machine and automates experiment deployment (without user intervention)



Using NEPI

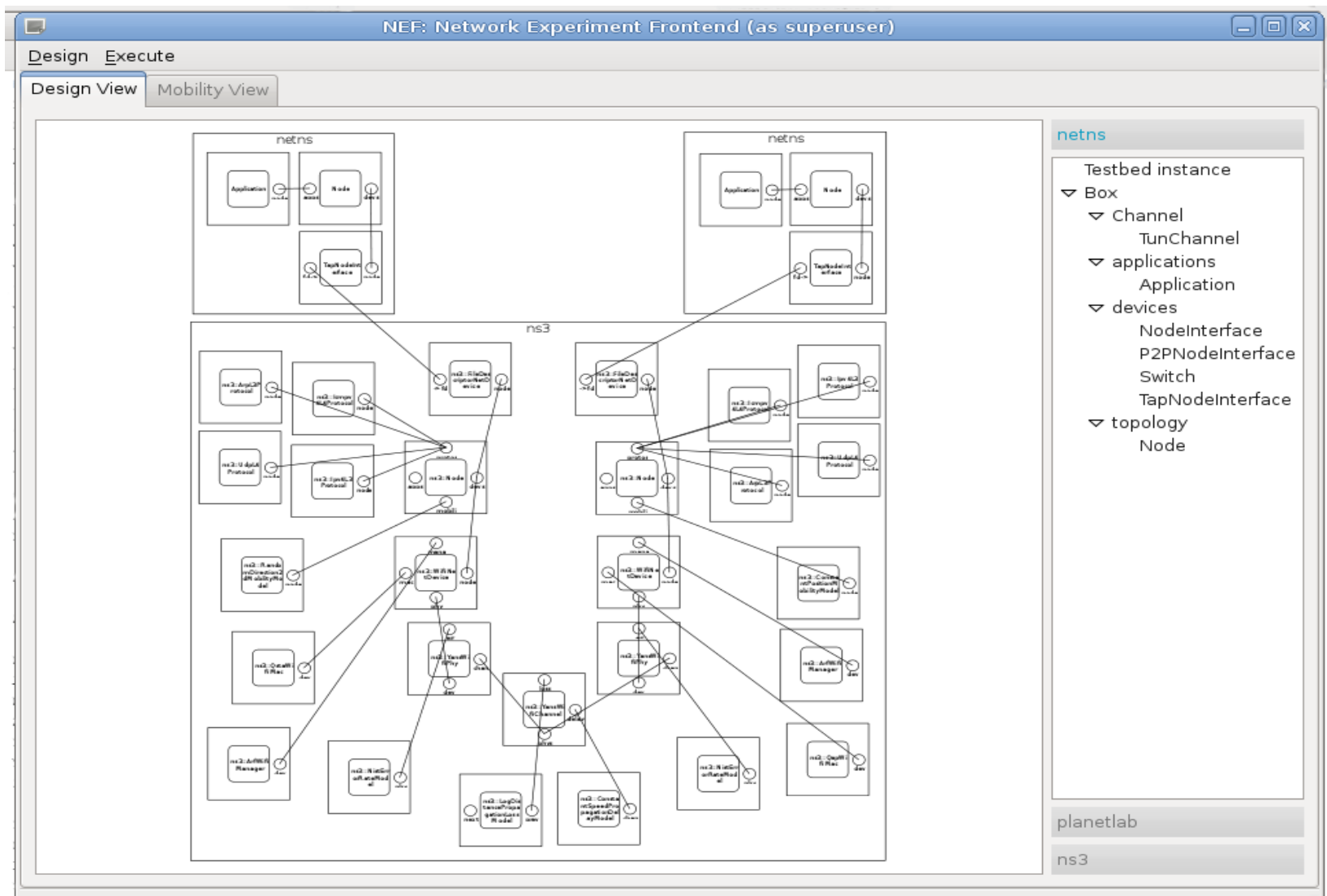
- NEPI is an open source project under GPLv2
- Is implemented in Python
- Two ways of conducting experiments
 - Writing a Python script
 - Using the GUI (NEF)
- Currently supports 4 backends



NETNS



The GUI (NEF)



Python script

- Import design and execution classes

```
from nepi.core.design import ExperimentDescription, FactoriesProvider  
from nepi.core.execute import ExperimentController
```

- Design

- Instantiate ExperimentDescription object

```
exp_desc = ExperimentDescription ()
```

- Create a testbed description object

```
testbed_id = "netns"  
netns_provider = FactoriesProvider(testbed_id)  
netns_desc = exp_desc.add_testbed_description(netns_provider)
```

Python script

- Create and configure boxes

```
node = netns_desc.create("Node")  
node.set_attribute_value("forward_X11", True)
```

- Interconnect boxes using connectors

```
app = netns_desc.create("Application")  
app.set_attribute_value("command", "xterm")  
app.connector("node").connect(node.connector("apps"))
```

- Enable traces

```
App.enable_trace("stdout")
```

Python script

- Execution

- Persist design to xml format

```
xml = exp_desc.to_xml()
```

- Create ExperimentController object

```
controller = ExperimentController(xml, "/tmp/root_dir")
```

- Start the experiment

```
controller.start()
```

Python script

- Wait until the experiment is finished

```
while not controller.is_finished(app.guid):  
    time.sleep(0.5)
```

- Collect results

```
result = controller.trace(app.guid, "stdout")
```

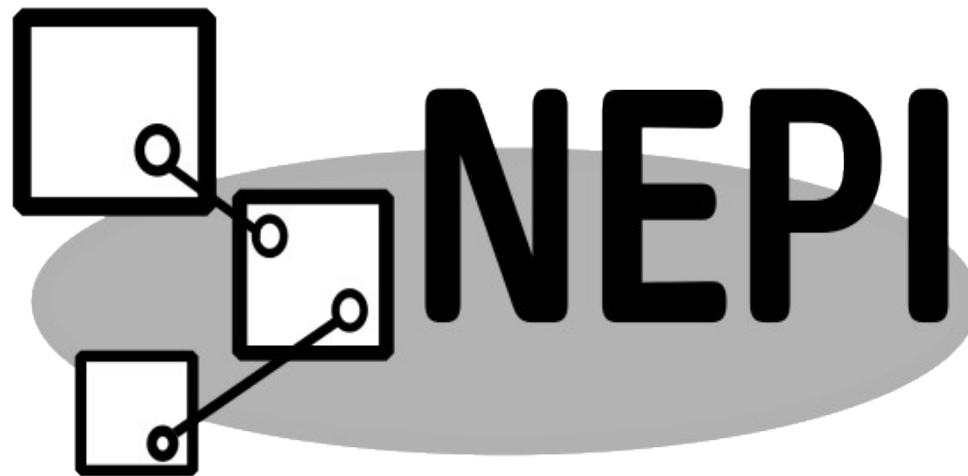
- Shutdown the experiment

```
controller.stop()  
controller.shutdown()
```

Extending NEPI

- NEPI was designed to be extended for arbitrary environments
- Steps to create a new backend
 1. Add a new directory under `src/nepi/testbeds/` (e.g. `src/nepi/testbeds/omf`)
 2. Add a `metadata.py` file and define all the boxes, connector and attributes for the boxes
 3. Implement the functions to be invoked on each type of box upon creation, connection, start, stop
 4. Add a `execute.py` file and extend the `TestbedController` class, adding environment specific behavior

- Visit NEPI wiki page for more information and examples <http://nepi.inria.fr>



Thank you



<http://nepsi.inria.fr>



Questions?



<http://nepi.inria.fr>