PrimoGENI – Developing GENI Aggregates for Real-Time Large-Scale Network Simulation

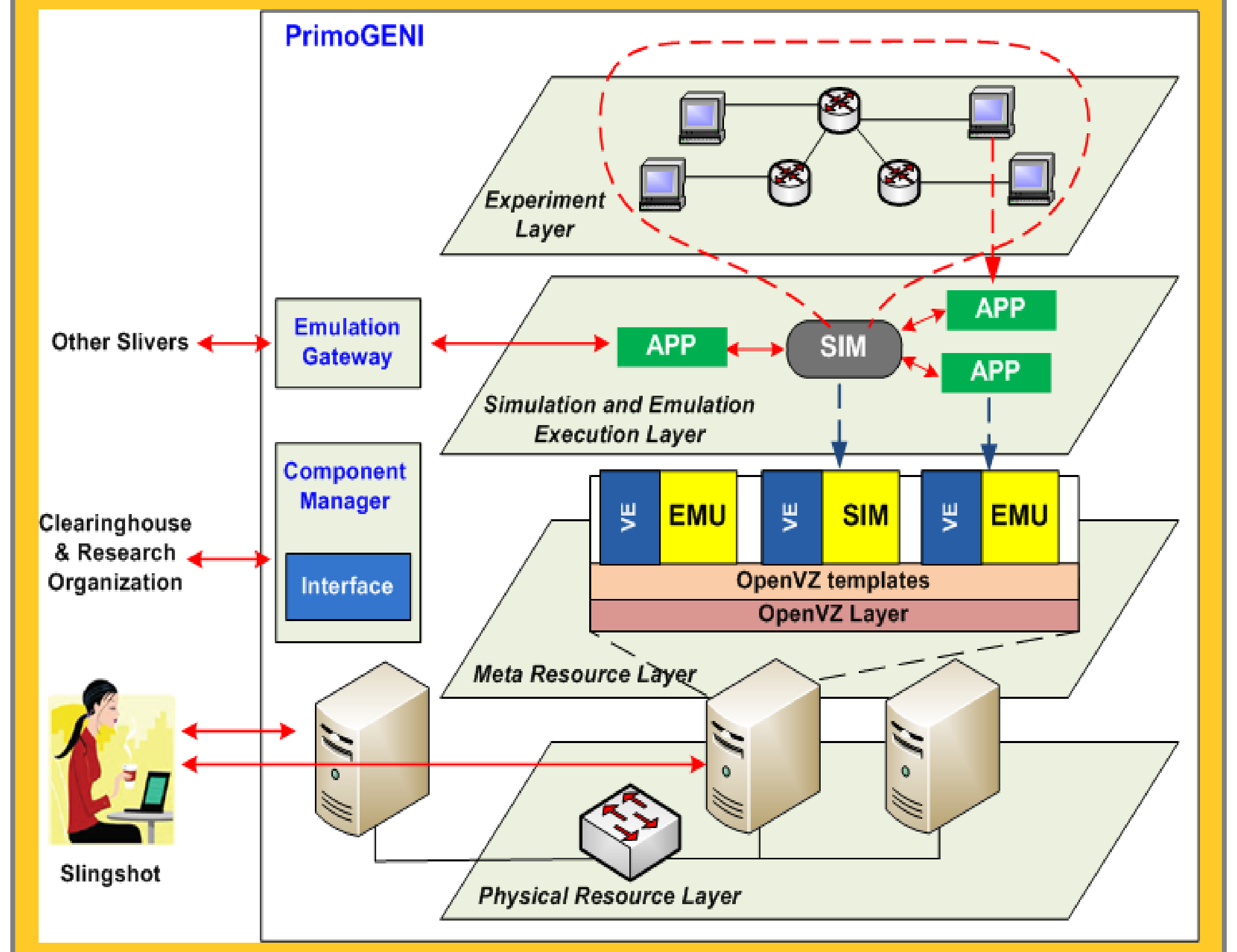
Miguel A. Erazo, Nathanael Van Vorst, Eduardo Tibau, Eduardo Peña, Hao Jiang, Ting Li, Jason Liu
School of Computing and Information Sciences, Florida International University, Miami, Florida

1 Overview

The **goal** of PrimoGENI is to incorporate **real-time network simulation** into the GENI "ecosystem". We are extending our existing real-time large-scale network simulator **PRIMEX** to become part of the GENI federation.

PrimoGENI will support large-scale GENI experiments with *millions* of **simulated** network entities (hosts, routers, and links) and *thousands* of **emulated** elements running unmodified network protocols and applications.

2 PrimoGENI Architecture



Important features of **PrimoGENI** include:

Multi-layer architecture:

- ➢ ***Physical resources layer (substrate)****:* cluster nodes, switches, and other physical resources, which can be queried during resource discovery.
- ➢ ***Meta resources layer:*** virtual machines upon resource assignment during sliver creation.
- ➢ ***Simulation and emulation execution layer:*** simulator instances and emulated hosts, created upon virtual network specification, and mapped to the meta resources at the layer below.
- ➢ ***Experiment layer:*** researchers can conduct live simulation and emulation experiments on the virtual network

Decentralization:

- ➢ PrimoGENI's design allows users to instantiate experiments on **any** ProtoGENI site with minimum effort.
- ➢ The logic of PrimoGENI resides entirely in the IDE provided to the user and specific OS images installed at the ProtoGENI/Emulab experimentation nodes.

Resources:

Shararable features managed by a component manager and described in **Rspecs** and **our own language**, which define two types:

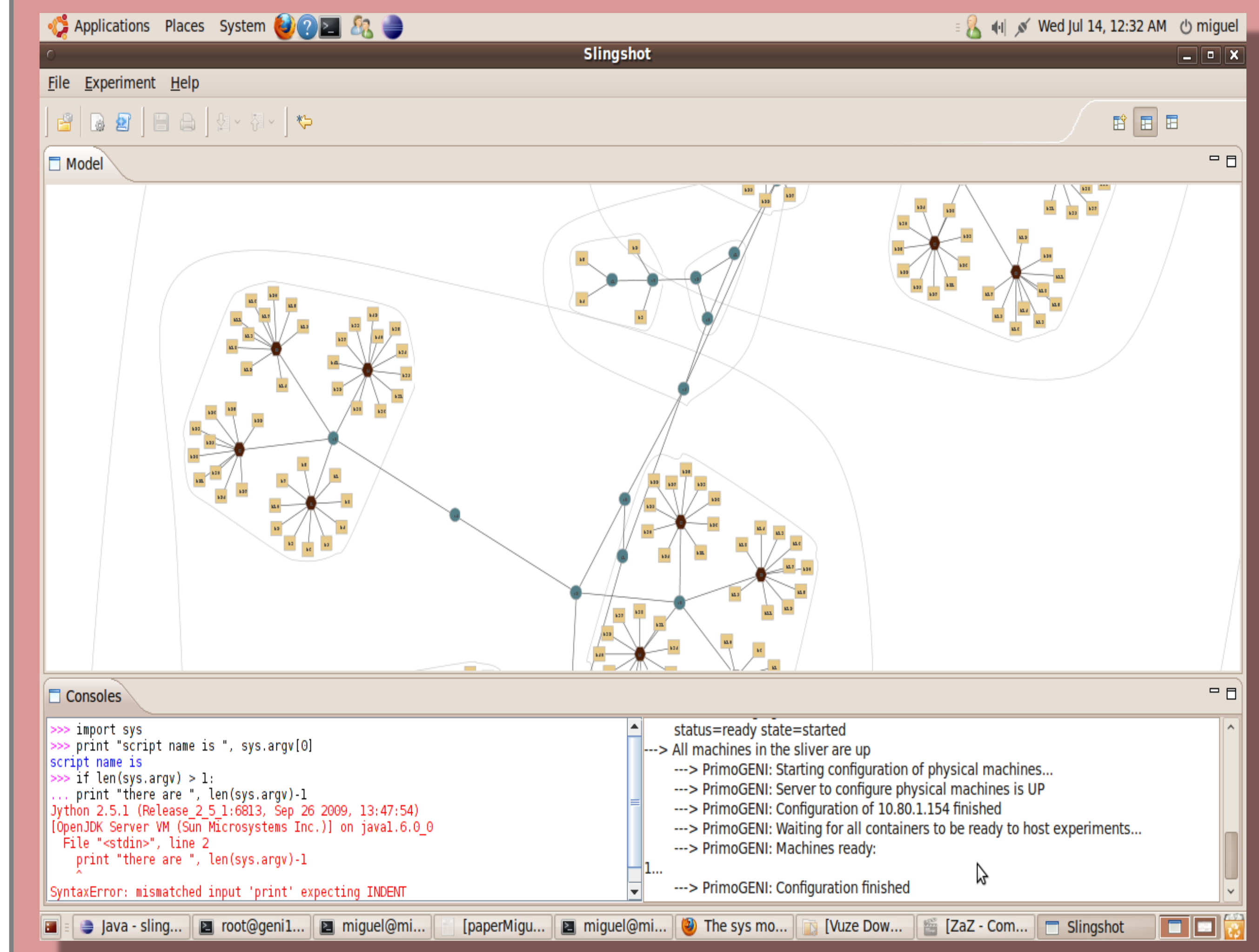
- ➢ ***Meta resources***: physical or virtual hosts, and other resources managed by EmuLab.
- ➢ ***Virtual resources***: virtual network resources simulated and emulated by **PrimoGENI**.

3 PRIMEX

• **PRIMEX** is a real-time network simulator, capable of simulating large-scale networks and emulating unmodified network protocols, services, and applications.

- ➢ ***Scalable network models***: Capable of conducting large-scale experiments on parallel and distributed platforms.
- ➢***High-speed emulation***: Provide high-performance interoperability between network applications running on virtual machines and the network simulator.
- ➢***Real-time interaction***: User can directly monitor and modify the state of the network model during experiments.
- ➢**Network scripting**: User can construct large complex network experiments using simple scripting languages (in Java and Python).
- ➢***Model databases***: Existing network models and experiment results are stored in databases for reusability and validation.

4 Slingshot



```
>>> import sys
>>> print "script name is ", sys.argv[0]
script name is
>>> if len(sys.argv) > 1:
... print "there are ", len(sys.argv)-1
Jython 2.5.1 (Release_2_5_1:6813, Sep 26 2009, 13:47:54)
[OpenJDK Server VM (Sun Microsystems Inc.)] on java1.6.0_0
  File "<stdin>", line 2
    print "there are ", len(sys.argv)-1
    ^
SyntaxError: mismatched input 'print' expecting INDENT
```

```
    status=ready state=started
---> All machines in the sliver are up
    ---> PrimoGENI: Starting configuration of physical machines...
    ---> PrimoGENI: Server to configure physical machines is UP
    ---> PrimoGENI: Configuration of 10.80.1.154 finished
    ---> PrimoGENI: Waiting for all containers to be ready to host experiments...
    ---> PrimoGENI: Machines ready:
1...
    ---> PrimoGENI: Configuration finished
```

Slingshot is an Integrated Development Environment (IDE), which aims to provide an end-to-end solution for **PrimoGENI** experiments.

- ➢ ***Model development***: Use a python console for constructing large complex network models.
- ➢ ***Model visualization***: Use prefuse for visualizing the network configurations.
- ➢ ***Experiment setup***: Deploy and launch experiments on ProtoGENI/Emulab environments.
- ➢ ***Experiment monitoring and control***: Monitor the experiment and possibly change the state of the network entities during experiment run time.

5 PrimoGENI Experiments

PrimoGENI uses ProtoGENI/Emulab suite to manage physical and meta resources

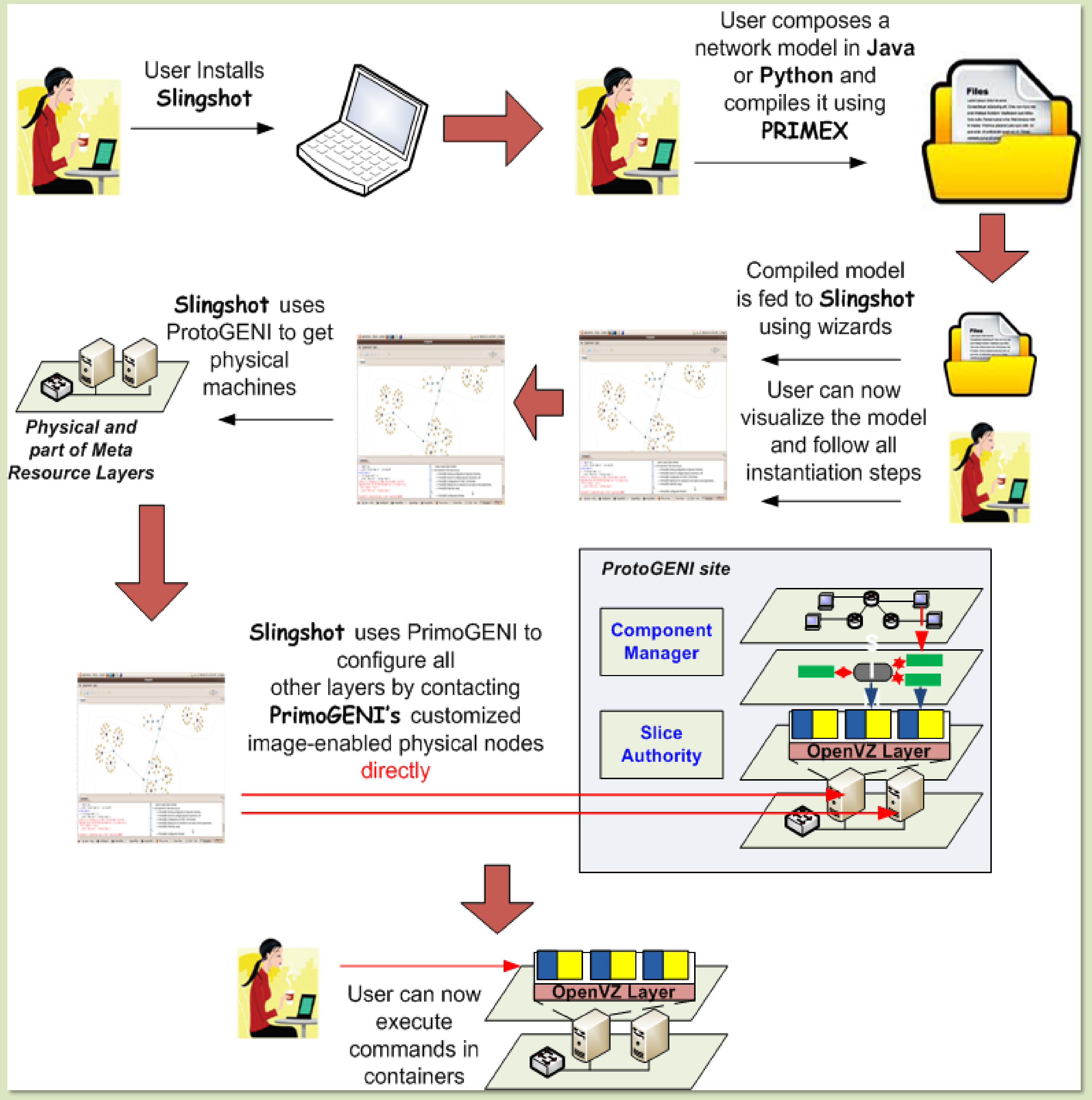
- To configure **PrimoGENI**:

I. The user needs to install **Slingshot** at the client side.

II. The user needs to acquire and install ProtoGENI credentials in **SlingShot**.

III. The user needs to install **PrimoGENI customized images** at the ProtoGENI site.

- After instantiating an experiment, the user can access individual emulated hosts (i.e., openvz containers) through ssh.


User Installs Slingshot
User composes a network model in Java or Python and compiles it using PRIMEX
Files
Compiled model is fed to Slingshot using wizards
User can now visualize the model and follow all instantiation steps
Slingshot uses ProtoGENI to get physical machines
Physical and part of Meta Resource Layers
Slingshot uses PrimoGENI to configure all other layers by contacting PrimoGENI's customized image-enabled physical nodes directly
ProtoGENI site
Component Manager
Slice Authority
OpenVZ Layer
User can now execute commands in containers
OpenVZ Layer

PrimoGENI is available at http://groups.geni.net/geni/wiki/PrimoGENI.