2012 GENI Research and Educational Experiment Summer Camp (GREE-SC2012)

PrimoGENI Tutorial

Jason Liu and Miguel Erazo

FIU | FLORIDA INTERNATIONAL UNIVERSITY

geni Exploring Networks of the Future

NSF

PrimoGENI

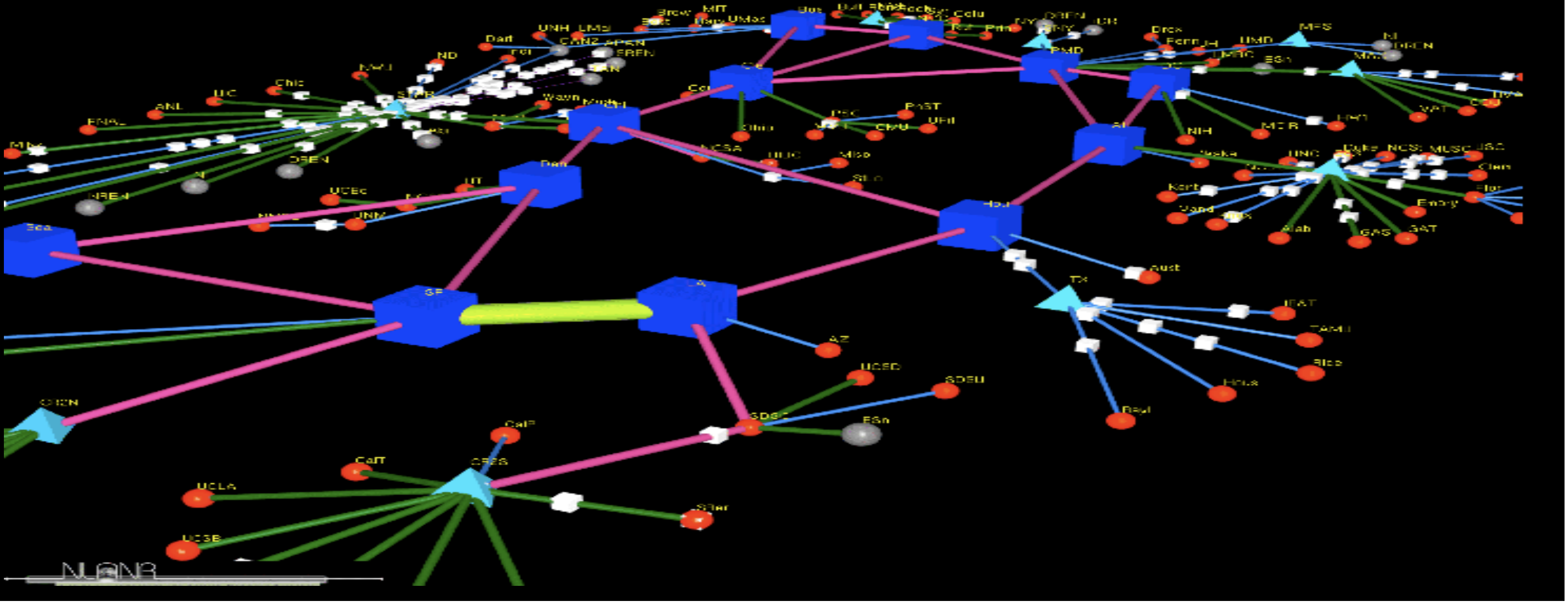
PRIME on GENI

Parallel Real-time Immersive Modeling Environment

PrimoGENI

Enable hybrid network experiments on GENI

- Including simulated, emulated, and physical components



Enable hybrid network experiments on GENI

- Including simulated, emulated, and physical components

network experiment at scale
modeling abstraction
fast prototyping
flexible

Enable hybrid network experiments on GENI

- Including simulated, emulated, and physical components

real protocols and applications
real execution environment
resource multiplexing
easily deployable

PrimoGENI

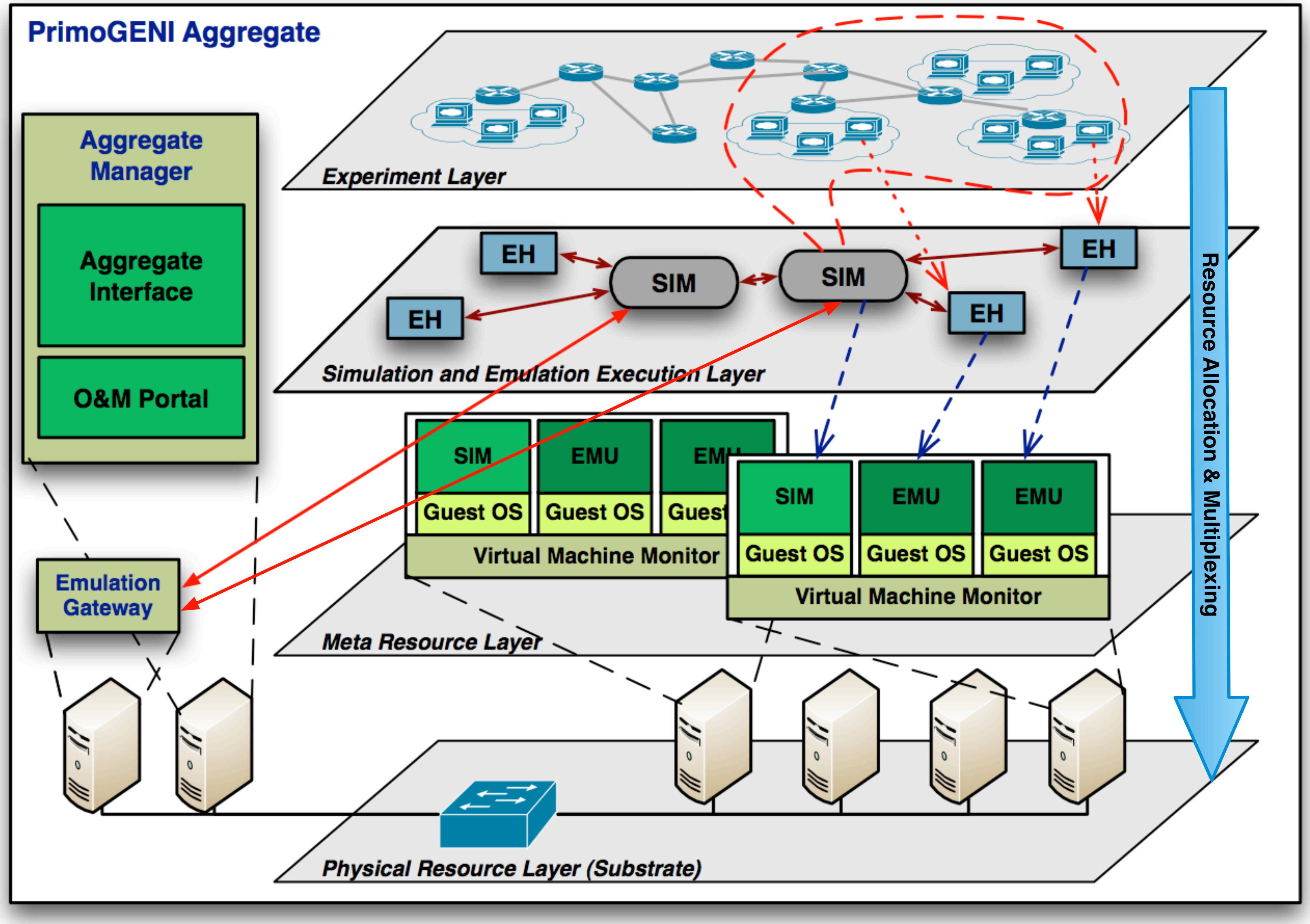
Enable hybrid network experiments on GENI

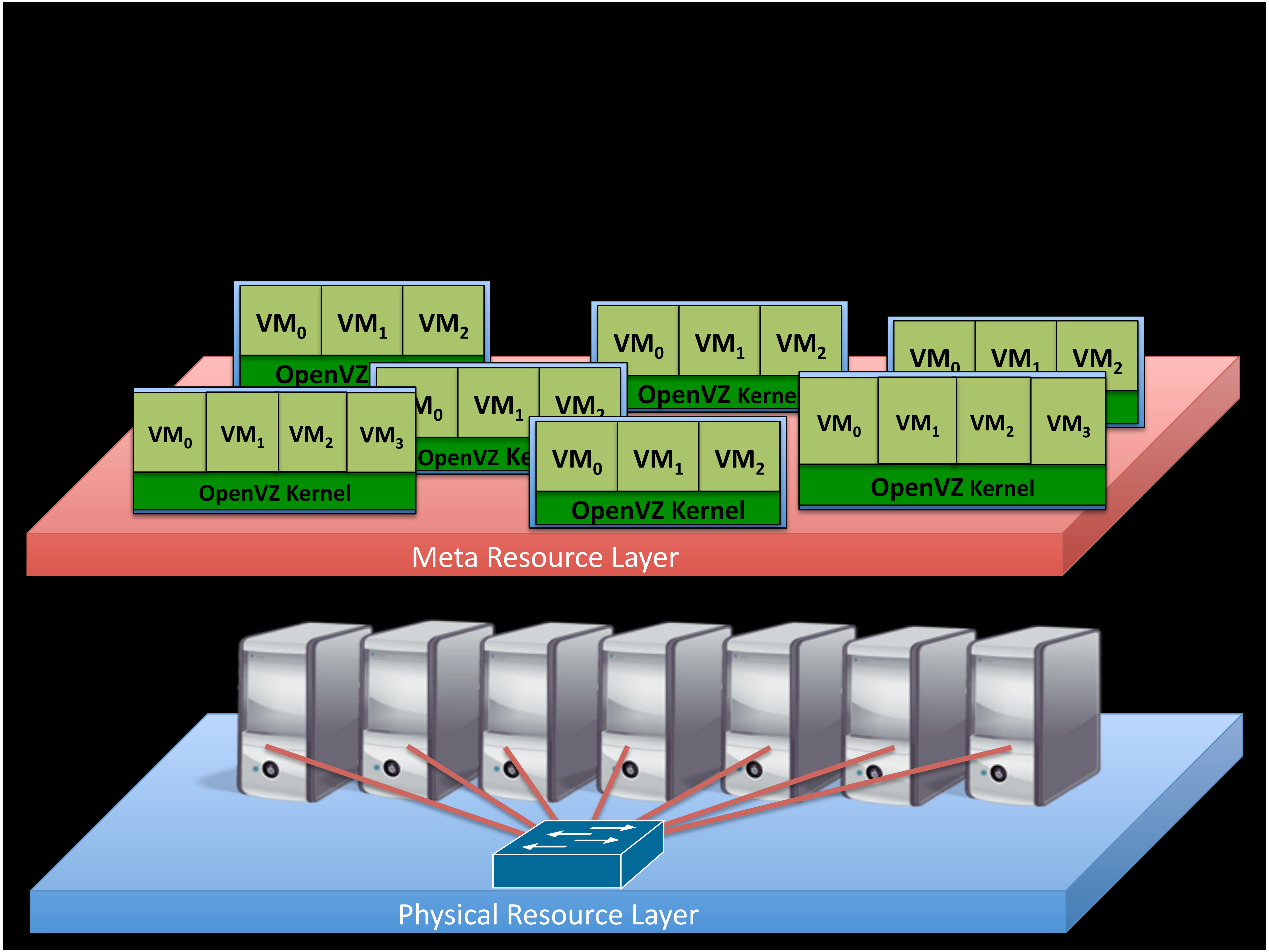
- Including simulated, emulated, and physical components

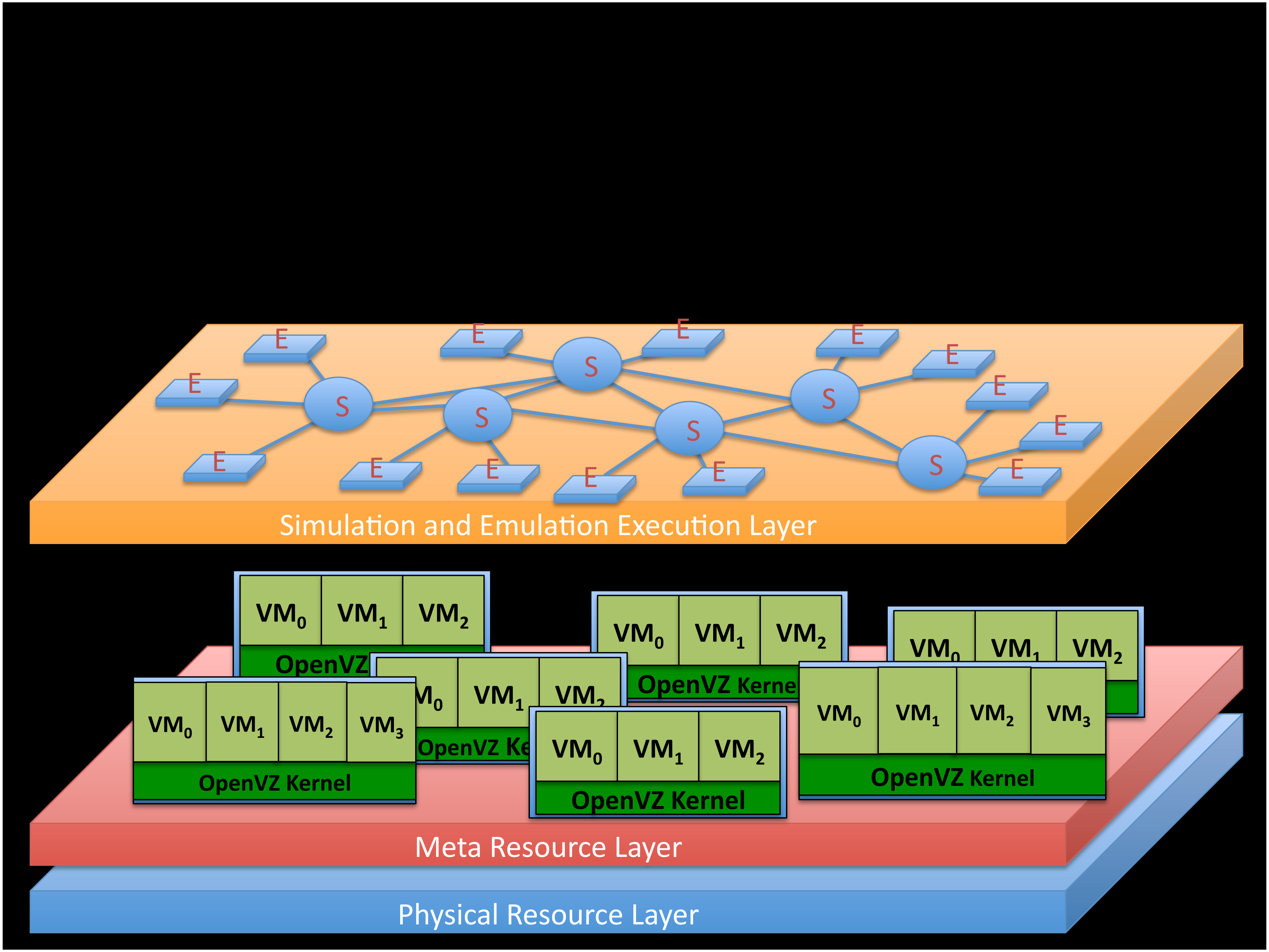
interact with real network
real traffic conditions
real devices

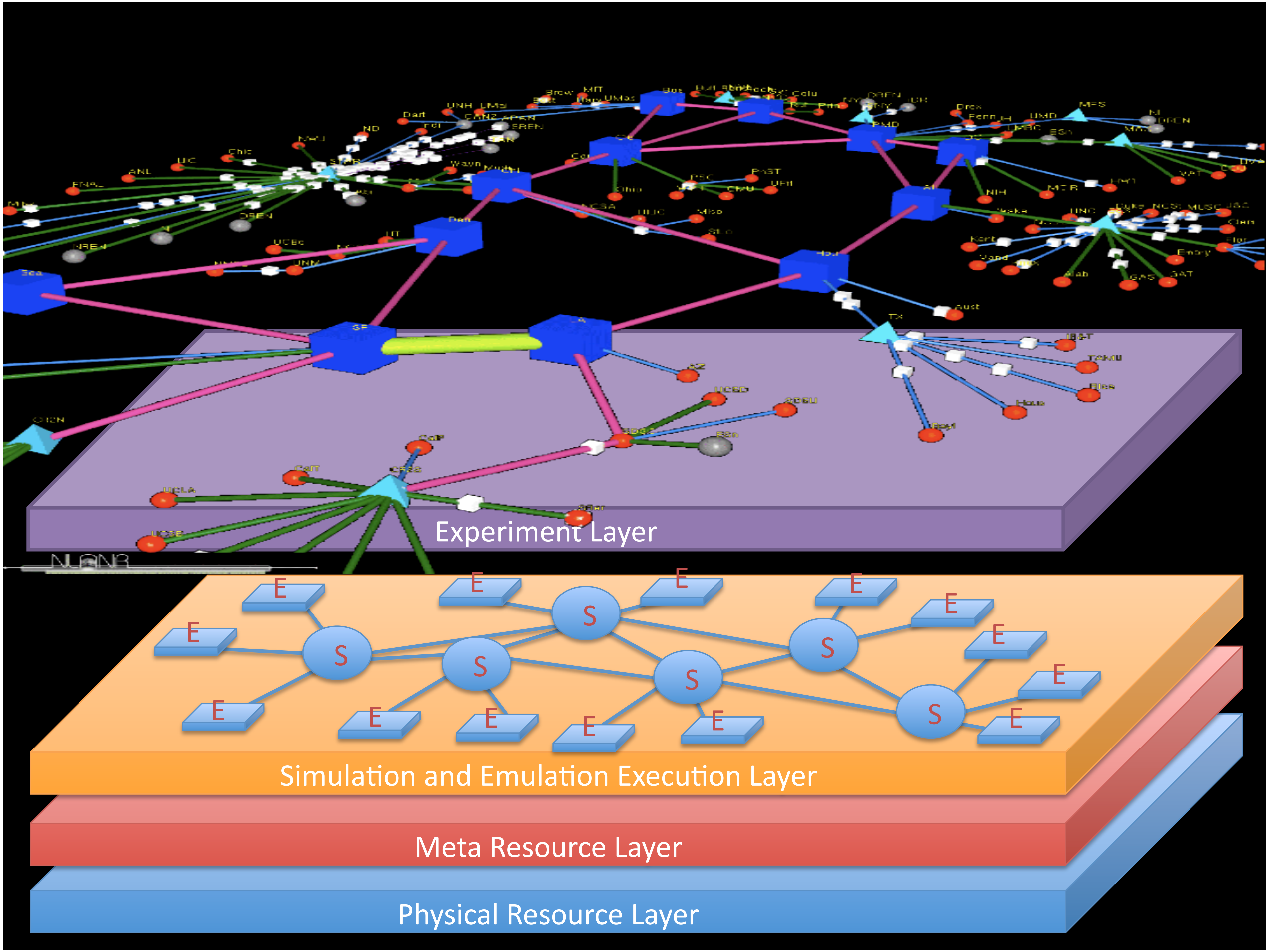
PrimoGENI is an Aggregate

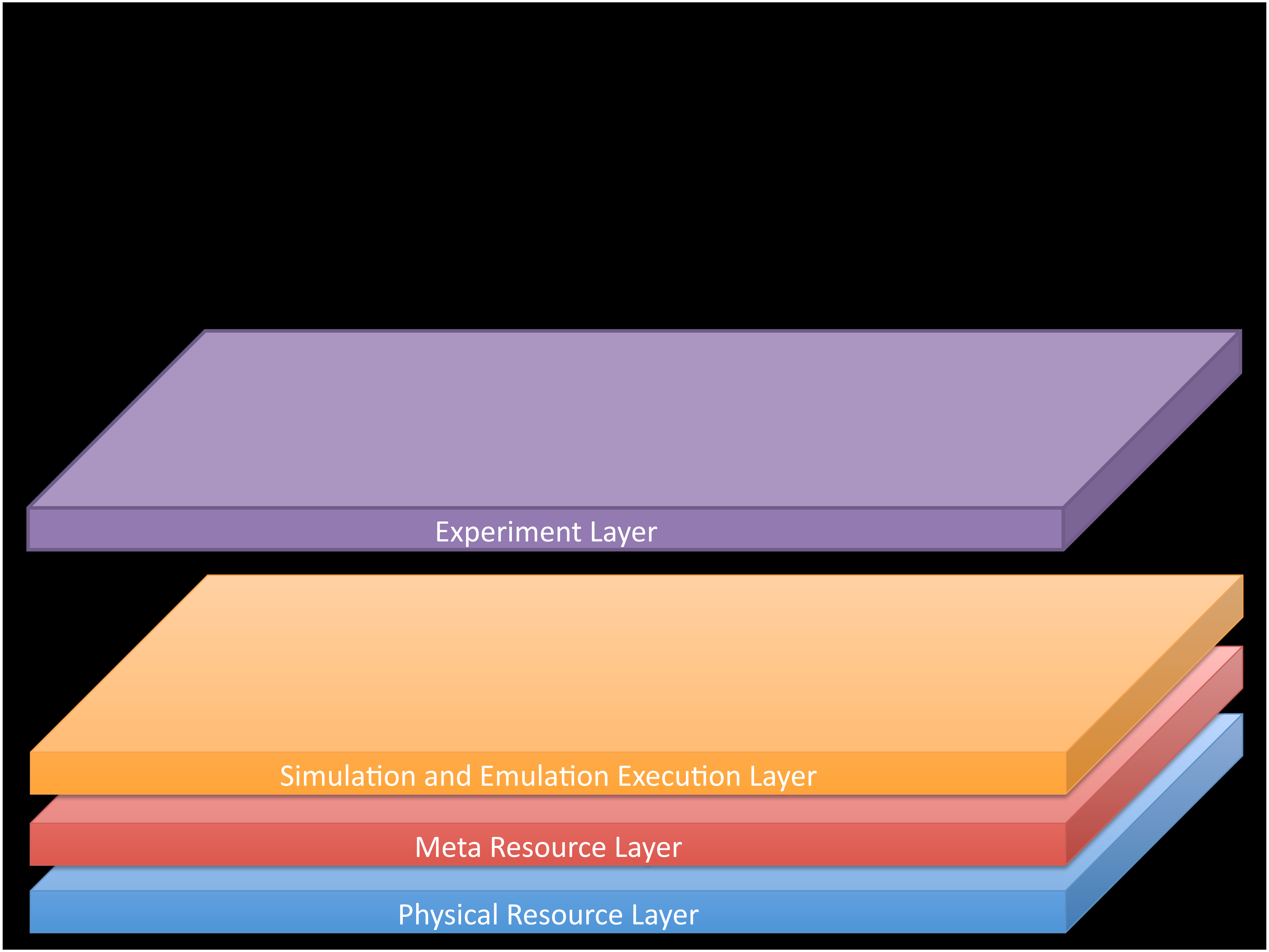
PrimoGENI is a GENI Aggregate

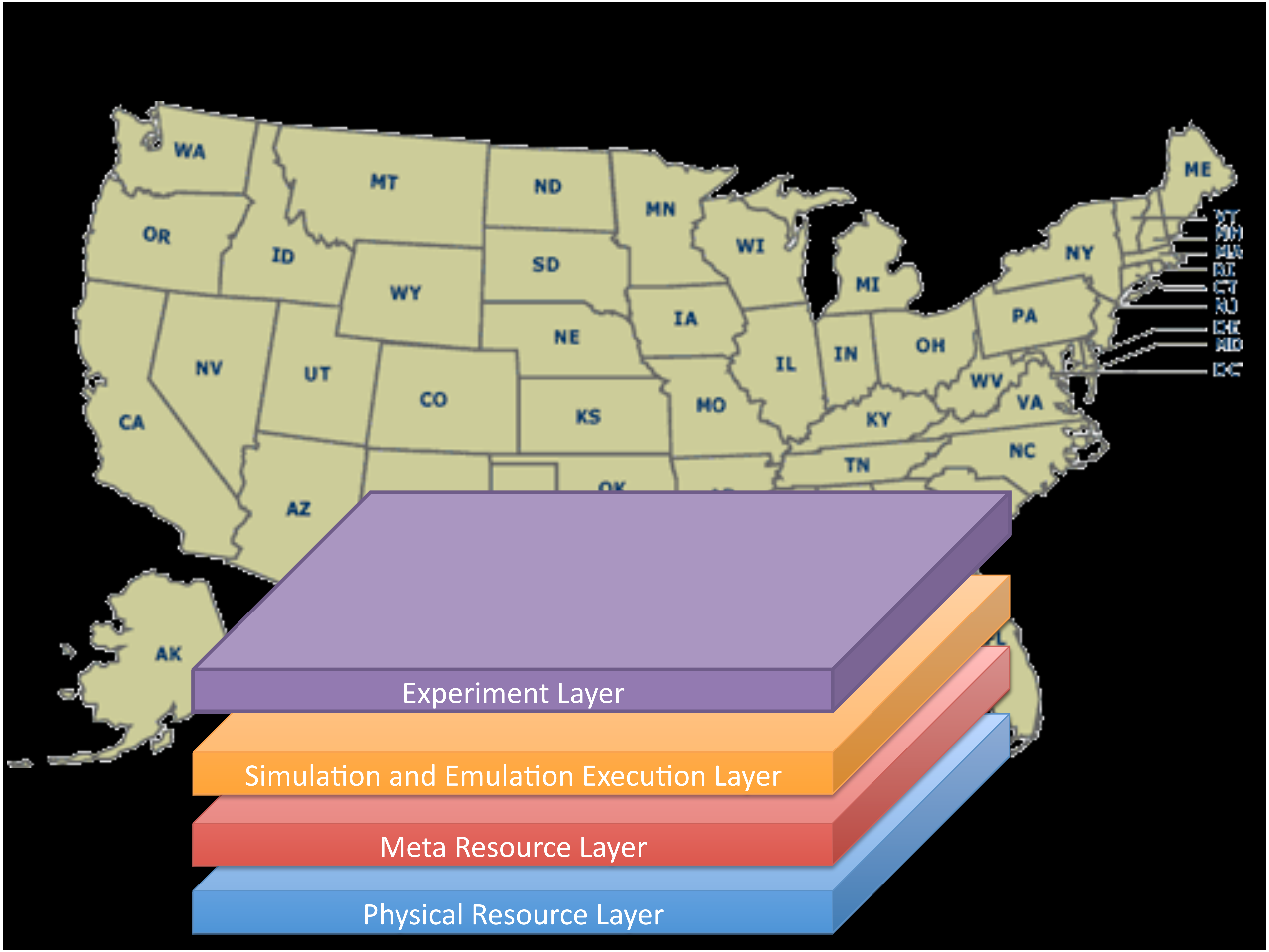



VM$_0$ VM$_1$ VM$_2$
OpenVZ
VM$_0$ VM$_1$ VM$_2$
OpenVZ Kerne
VM$_0$ VM$_1$ VM$_2$
M$_0$ VM$_1$ VM$_2$
OpenVZ Ke
VM$_0$ VM$_1$ VM$_2$ VM$_3$
OpenVZ Kernel
VM$_0$ VM$_1$ VM$_2$
OpenVZ Kernel
VM$_0$ VM$_1$ VM$_2$ VM$_3$
OpenVZ Kernel
Meta Resource Layer
Physical Resource Layer


E
E
E
E
E
E
E
E
E
E
E
E
E
E
E
E
S
S
S
S
S
S
Simulation and Emulation Execution Layer
VM$_0$
VM$_1$
VM$_2$
OpenVZ
VM$_0$
VM$_1$
VM$_2$
OpenVZ Kerne
VM$_0$
VM$_1$
VM$_2$
M$_0$
VM$_1$
VM$_2$
OpenVZ Ke
VM$_0$
VM$_1$
VM$_2$
VM$_3$
OpenVZ Kernel
VM$_0$
VM$_1$
VM$_2$
OpenVZ Kernel
VM$_0$
VM$_1$
VM$_2$
VM$_3$
OpenVZ Kernel
Meta Resource Layer
Physical Resource Layer


NLANR
Experiment Layer
E
S
Simulation and Emulation Execution Layer
Meta Resource Layer
Physical Resource Layer


Experiment Layer
Simulation and Emulation Execution Layer
Meta Resource Layer
Physical Resource Layer

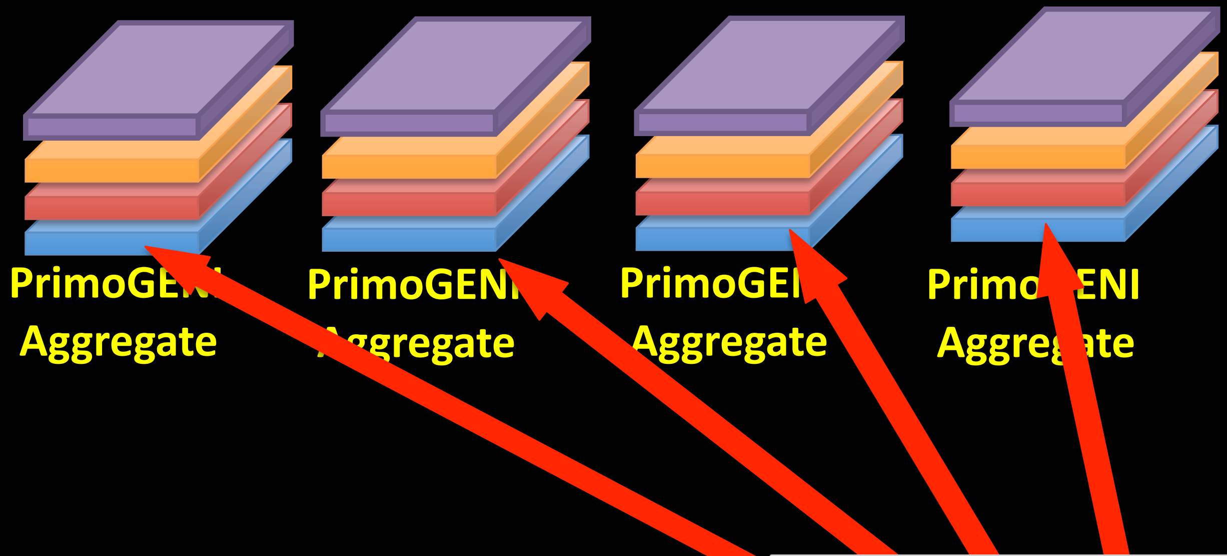


Experiment Layer

Simulation and Emulation Execution Layer

Meta Resource Layer

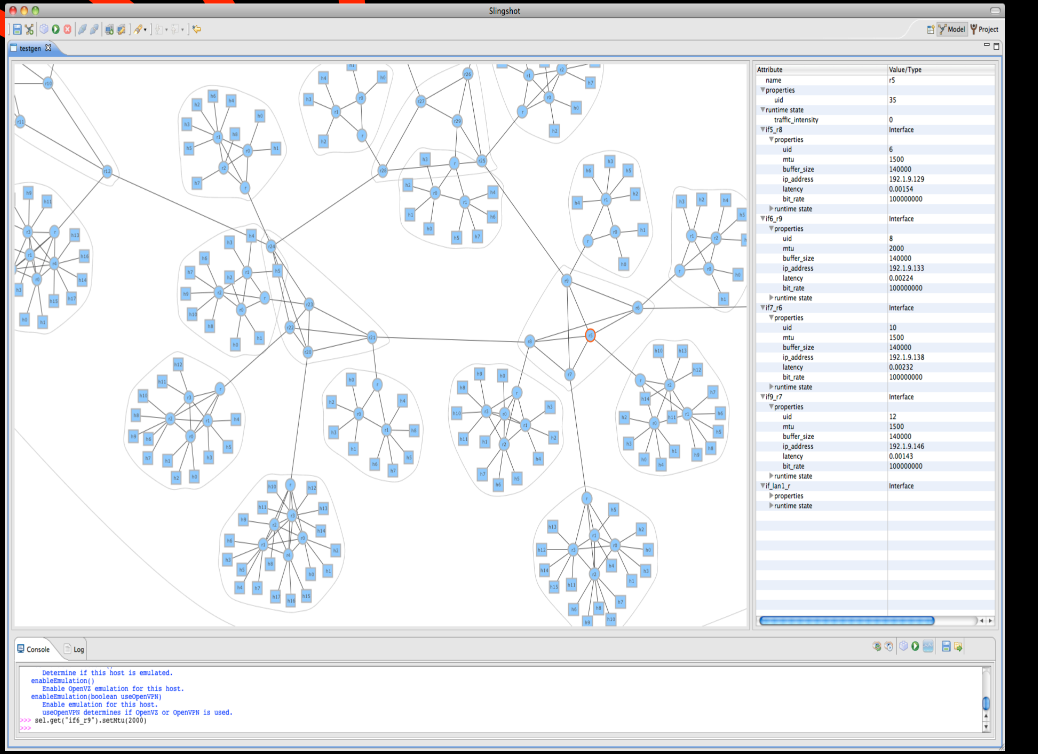
Physical Resource Layer



Slingshot

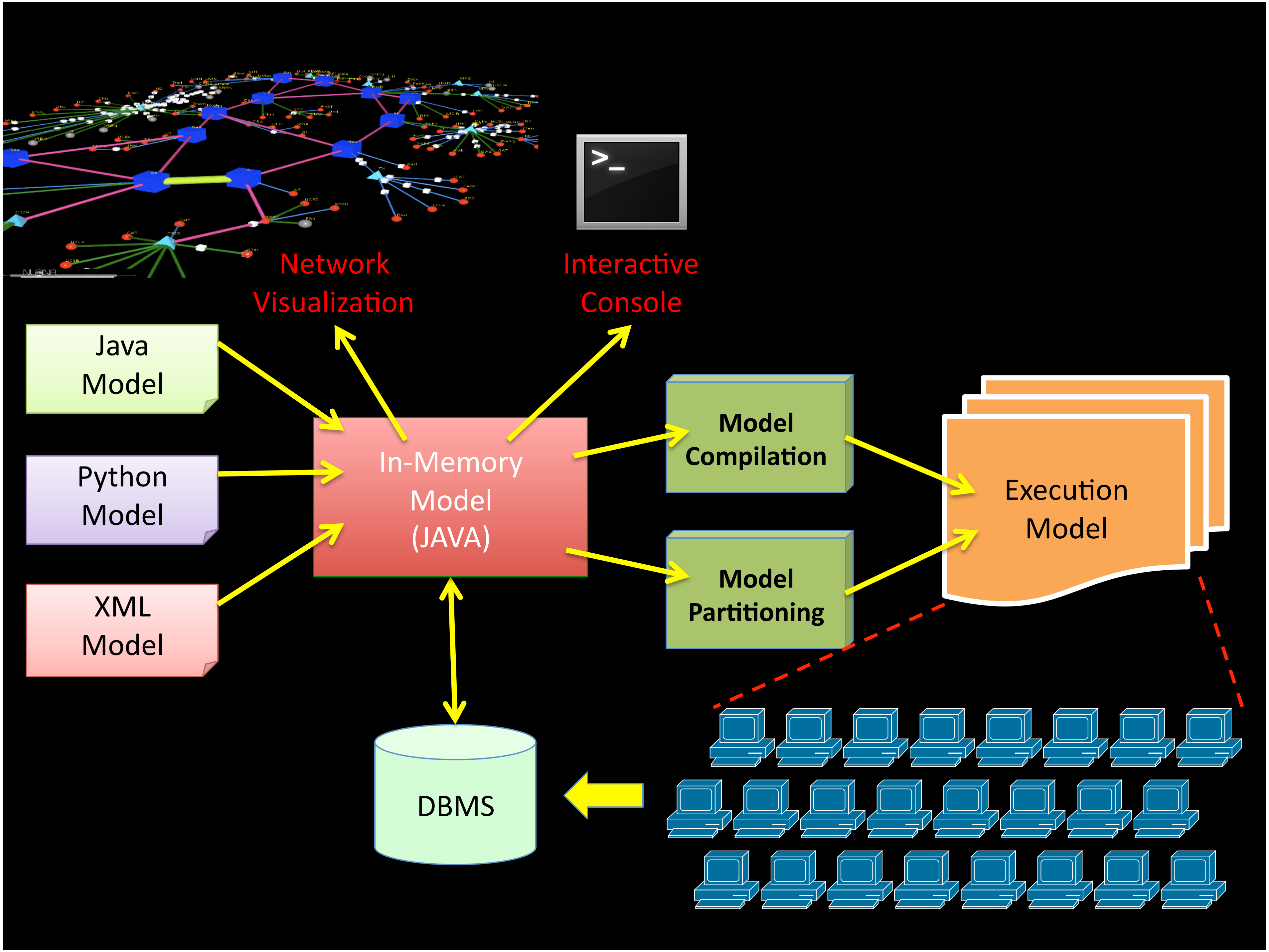
Manage the *life cycle* of network experiments

- model configuration
- resource specification
- deployment & execution
- online control & monitoring



Attribute	Value/Type
name	r5
▼properties	
uid	35
▼runtime state	
traffic_intensity	0
▼if5_r8	Interface
▼properties	
uid	6
mtu	1500
buffer_size	140000
ip_address	192.1.9.129
latency	0.00154
bit_rate	100000000
▶runtime state	
▼if6_r9	Interface
▼properties	
uid	8
mtu	2000
buffer_size	140000
ip_address	192.1.9.133
latency	0.00224
bit_rate	100000000
▶runtime state	
▼if7_r6	Interface
▼properties	
uid	10
mtu	1500
buffer_size	140000
ip_address	192.1.9.138
latency	0.00232
bit_rate	100000000
▶runtime state	
▼if9_r7	Interface
▼properties	
uid	12
mtu	1500
buffer_size	140000
ip_address	192.1.9.146
latency	0.00143
bit_rate	100000000
▶runtime state	
▼if_lan1_r	Interface
▶properties	
▶runtime state	

```
      Determine if this host is emulated.
   enableEmulation()
      Enable OpenVZ emulation for this host.
   enableEmulation(boolean useOpenVPN)
      Enable emulation for this host.
      useOpenVPN determines if OpenVZ or OpenVPN is used.
>>> sel.get("if6_r9").setMtu(2000)
>>>
```



Network Visualization

Interactive Console

Java Model

Python Model

XML Model

In-Memory Model (JAVA)

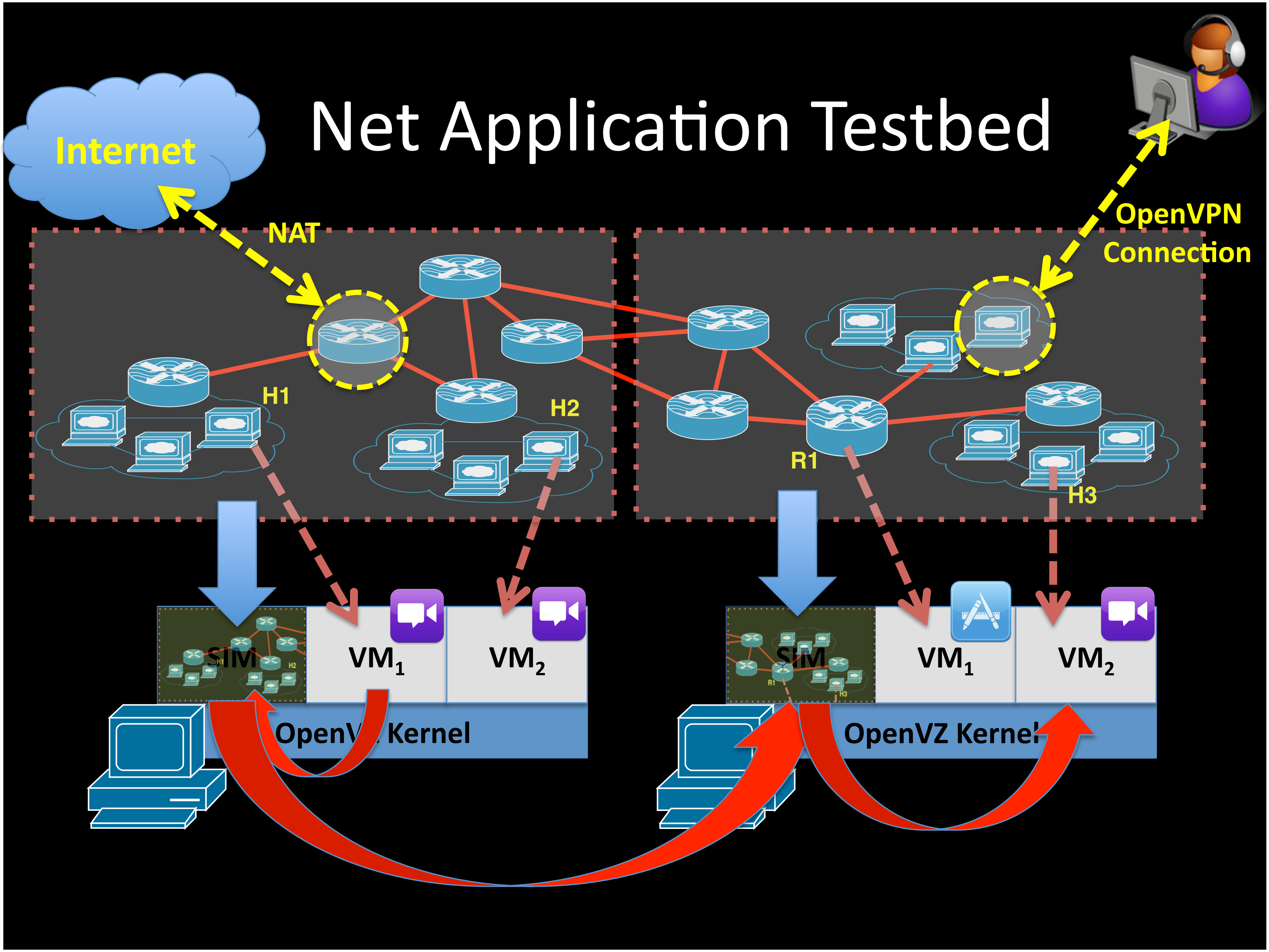
Model Compilation

Model Partitioning

Execution Model

DBMS

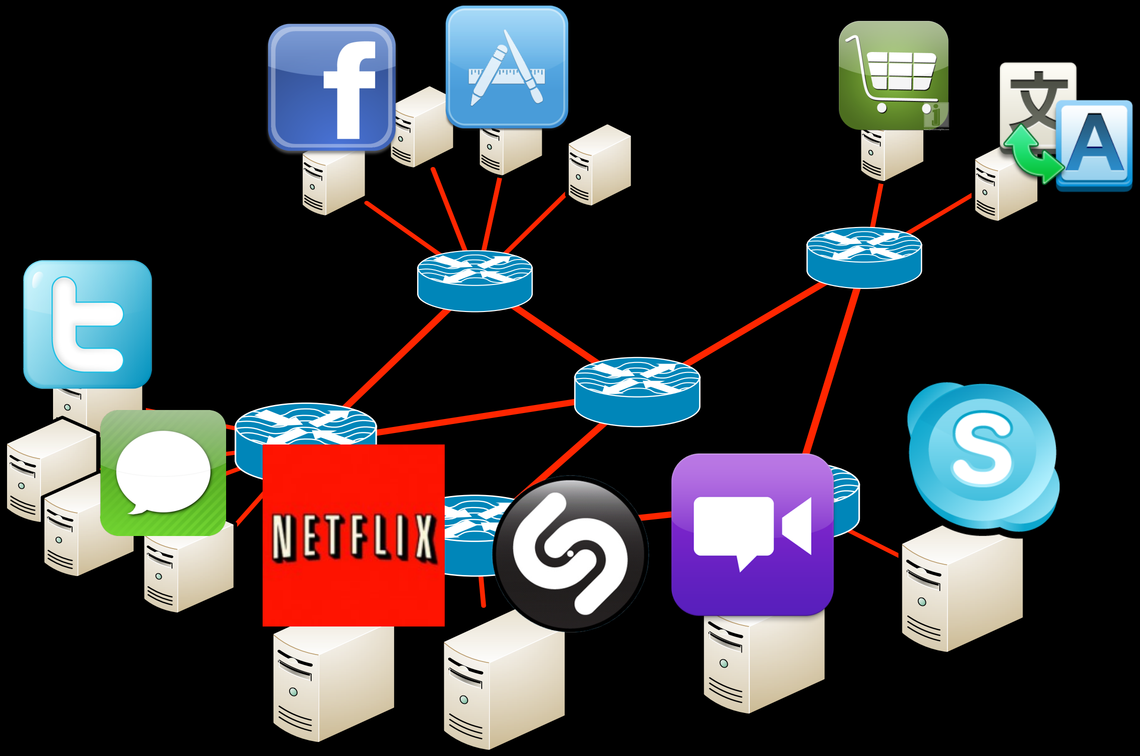
Net Application Testbed



What’s the Use?

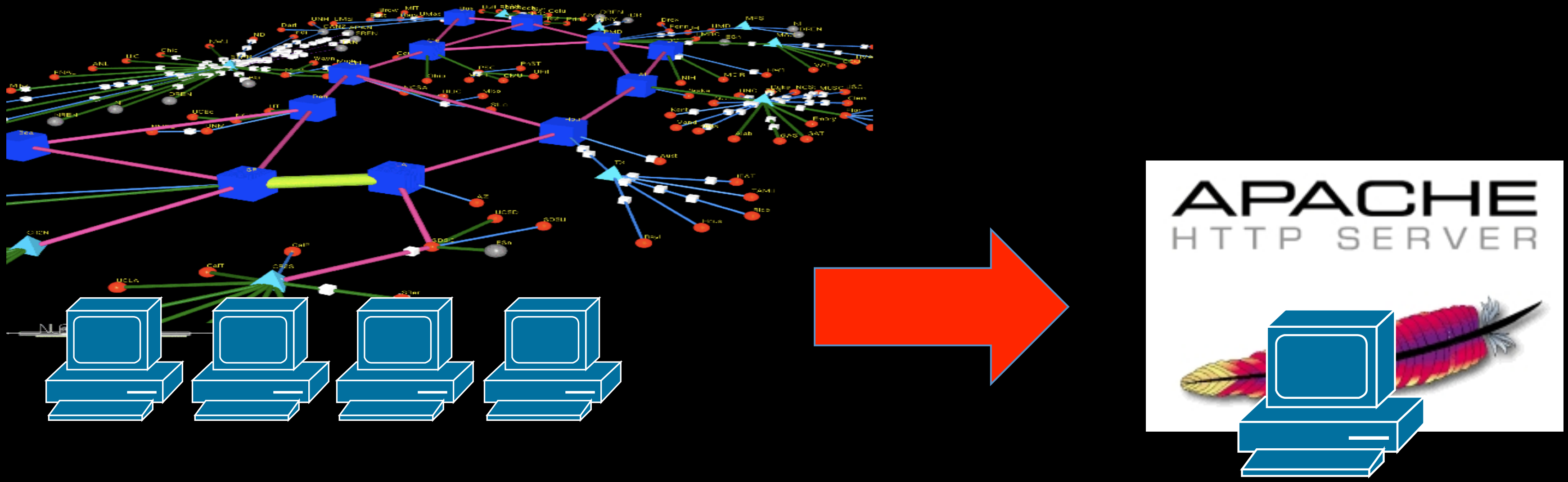
net application testbed
traffic generator
distributed virtual environment
…

Net Application Testbed



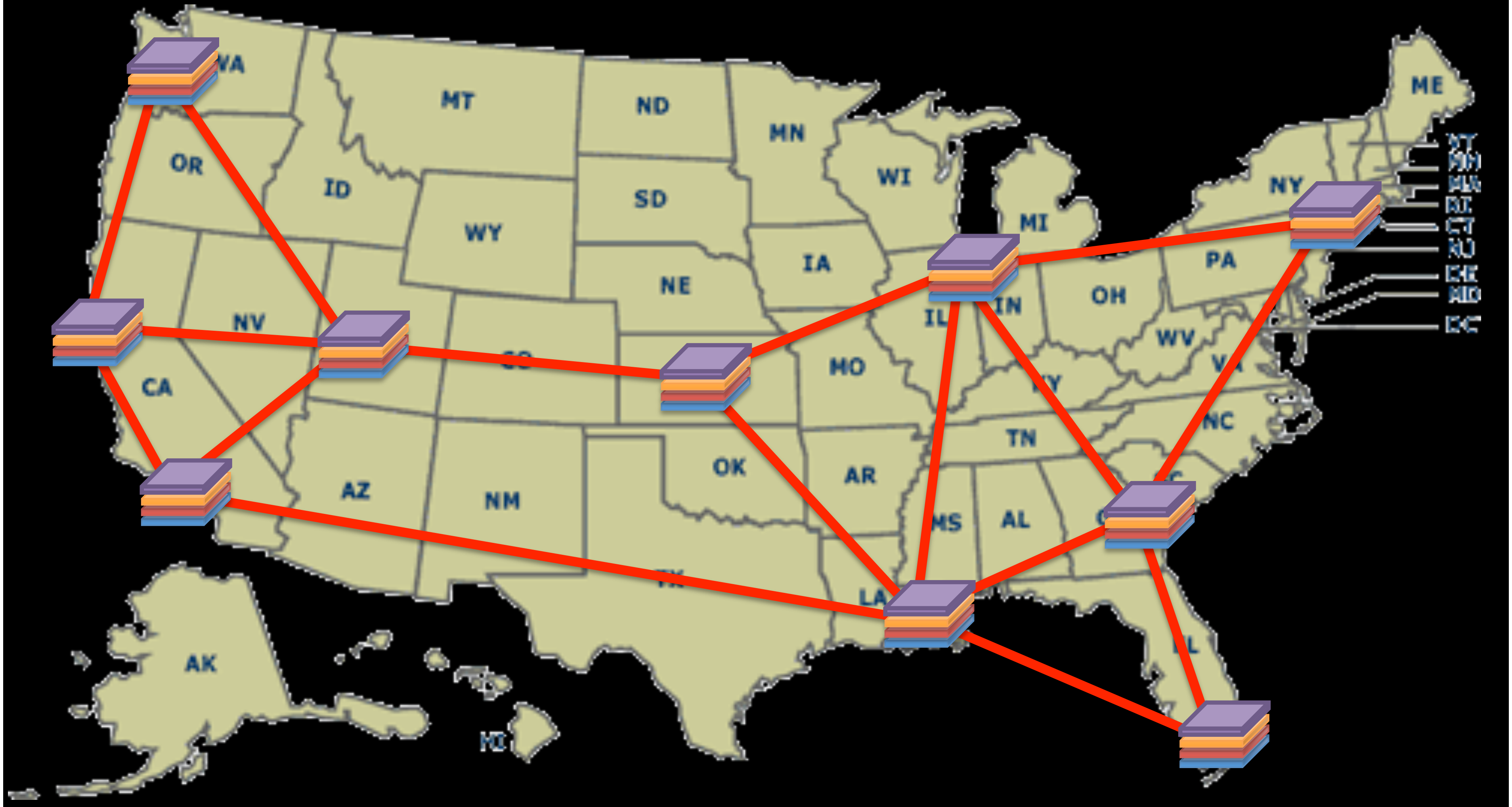
PrimoGENI can run *real* applications under simulated network conditions

Traffic Generator



PrimoGENI can generate *real* traffic according to temporal and spatial distribution

Distributed Virtual Environment



Hands-on Session

Install Software

1. Start VirtualBox VM: GENI_VM_Silver.ova (and log in using username “Geni Tutorial” and password “geniuser”)
2. Copy (by drag and drop) three files from the USB drive to the VM desktop: eclipseforsummercamp.tgz, primexforsummercamp.tgz, and summercampmodels.tgz. (You may first need to change the VirtualBox VM setting to allow the VM to access to the USB device).
3. Install eclipse and PRIMEX (the real-time network simulator):
   - mkdir ~/bin
   - cd ~/bin
   - tar xzvf ~/Desktop/eclipseforsummercamp.tgz
   - cd ~/workspace
   - tar xzvf ~/Desktop/primexforsummercamp.tgz

Install Software (cont.)

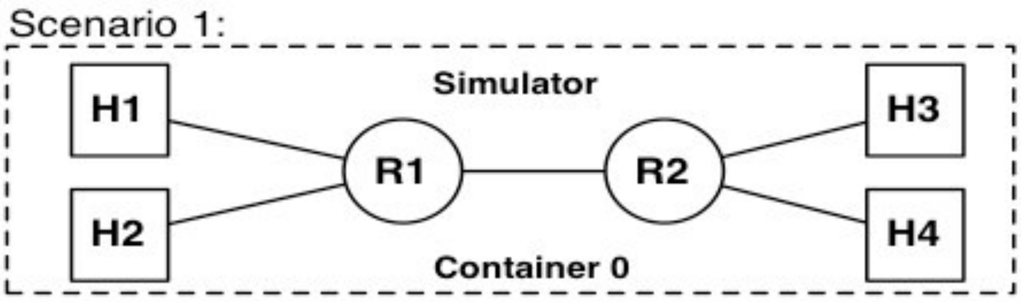
4. Create slingshot project in eclipse:
   - Start eclipse located in folder: ~/bin/eclipse
   - Create new java project, and name it “slingshot”
   - Import slingshot source code:
     - Right click on the project and click “import”
     - Go to General->File System; click ‘Next’
     - Click ‘Browse’ and choose ‘/home/geni/workspace/primex/netIDE’ folder
     - Check the box on left of netIDE and click ‘Finish’
   - Choose ‘Yes to All’ when prompted

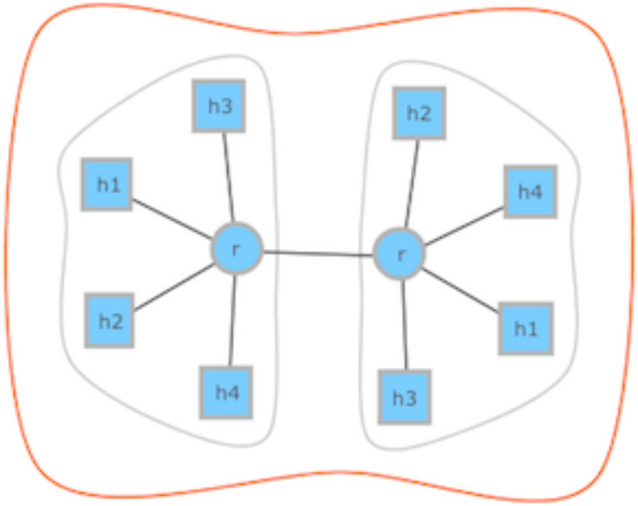
Install Software (cont.)

5. Set up slingshot (run from within eclipse):
   - Expand the slingshot project and right click on 'slingshot_linux_32.product' and choose: "Run as->Eclipse Application".
   - A small window opens up prompting for a workspace folder; we use the default: "/home/geni/slingshot"; check the box "Remember the workspace", and click "OK".
   - When prompted for creating a new folder, click "OK".
   - Quit from slingshot and install the models
     - cd ~/slingshot/Models
     - tar xzvf ~/Desktop/summercampmodels.tgz

Experiment 1: A **Simulated** Dumbbell Model

- Develop network model in Python, Java, or XML
- Create experiment
- Inspect and change network configurations in the attribute tree and the python console
- Launch experiment on local host
- Visualize traffic flowing on the network





A Simple SIMULATION Experiment

1. Start slingshot (you’ll first see the “project perspective”)
2. Select the file Models/basics/BasicDumbbellModel.py in the “project explorer”; double click to open the file in the “file editor”.
3. Inspect the python model:
   - Create top-level network “topnet”
   - Create subnet “leftnet”
   - Create four hosts within the subnet, each with a network interface
   - Create a router “r”
   - Create four links connecting a newly created interface at the router and the interface at each of the four hosts
   - Create an exact copy of the subnet to be “rightnet”
   - Create a link connecting the two routers at “leftnet” and “rightnet”

A Simple SIMULATION Experiment

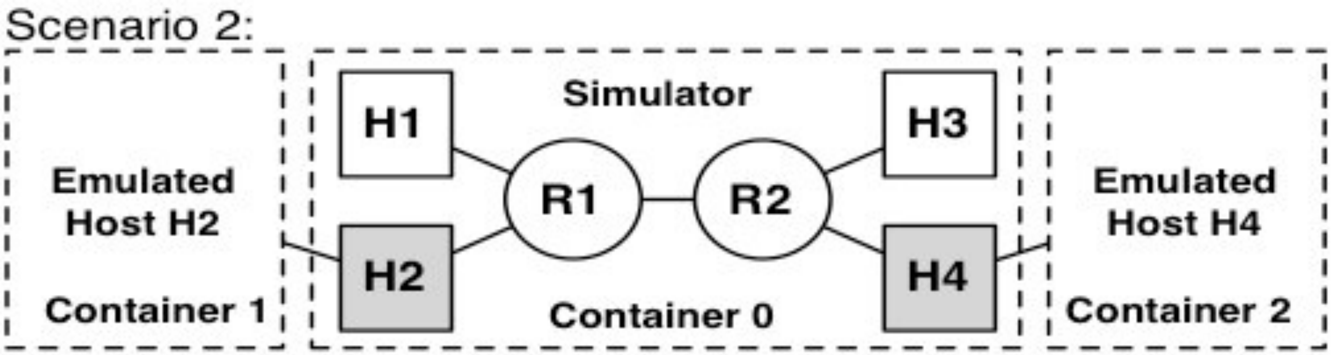
4. Create experiment:
   - At prompt, enter experiment name (say, test1), and browse for the source code for the model
   - Slingshot brings you to the “model perspective”, which consists of a “prefuse window”, a “tree view”, and a “python console”
   - The model is displayed in the “prefuse model” (using right button or control-right butter to center the network; changing view depth to view expand or collapse the network)
   - Inspect the attributes in “tree view”
     - Change the middle link’s bandwidth from 1e8 to 1e7 (you may need to use “control-return” to confirm the change)
5. Compile the experiment and run it

A Simple SIMULATION Experiment

5. Add dynamic traffic (at the “python console”):
   - topnet = exp.getRootNode() #(or select the topnet by using the mouse, then at the console: topnet=sel)
   - tf = exp.createDynamicTrafficFactory(topnet)
   - left_h1 = topnet.get("leftnet.h1")
   - right_h1 = topnet.net.get("rightnet.h1")
   - tf.createSimulatedTCP(10, 1000000000, left_h1, right_h1) #(at time 10, left_h1 will request download of 1 GB from right_h1)
   - You can also select nodes using mouse and use console to assign the selected node to a variable:
     - (select one node then at the console) x = sel
     - (select another node) y = sel
     - tf.createSimulatedTCP(40, 1000000000, x, y)

Experiment 2: An **Emulated** Network Model

- Set up execution environment
- Specify emulated hosts/routers
- Create emulated traffic (traceroute, iperf)
- Monitor real-time traffic using LiveGraph



An EMULATION Experiment

1. Untar the manifests.tgz file provided in ~/slingshot/.
2. Click on Experiment->Create/Edit Environments
3. Name the enviroment “utah” and select “ProtoGENI” as the type of environment. Click on “Add”.
4. You were assigned a group number (if you do not know please ask). Then, load the file ‘primo-scX’ where X is your group number and click on “Add”.
5. Now, you must select which machine will be your machine that will host the simulator (slave). In this case, such machine is the one which has 3 nics. Select that machine and check the ‘Slave’ checkbox. Click on Finish twice.
6. **We need to set up the password-less login to the machines**

An EMULATION Experiment

1. We create another experiment with the dumbbell model, name it “test2”
2. At the console, we select a node to be emulated: sel.enableEmulation()
3. Compile and run experiment on the protogeni environment
4. Click on the first emulated host, you should note the “compute node” at the “tree view” indicate the machine on which the emulated host is running; also make a note of the “uid” and “ip_address” of the host
5. In a command shell, ssh to the machine
6. Become root: sudo su –
7. Check the emulated host is running: vzlist
8. Enter the emulated host: vzctl enter 30 (assuming 30 is the uid)
9. Once in the emulated host: ifconfig
10. You can ping other nodes, for example: ping 192.1.0.33
11. You can download from other nodes, for example: wget 192.1.0.33/100000.html