University *of* Missouri

VIMAN Lab

VIRTUALIZATION, MULTIMEDIA AND NETWORKING LAB

Deploying Network-aware Applications in GENI

Presenter: **Prasad Calyam**, Ph.D.
Department of Computer Science

Graduate Student: Ronny Bazan Antequera

GENI Webinar Series, Fall 2016

Tutorial Topics

- Future Internet Performance Architecture (FIPA)
  - Requirements and Design
- OnTimeMeasure-GENI: FIPA Reference Implementation
  - Hands-on Demo
    - Working with instrumentation and measurement services
    - Adding custom metrics of your Application
- FIPA Case Study: Dynamic Resource Allocation in Virtual Desktop Clouds

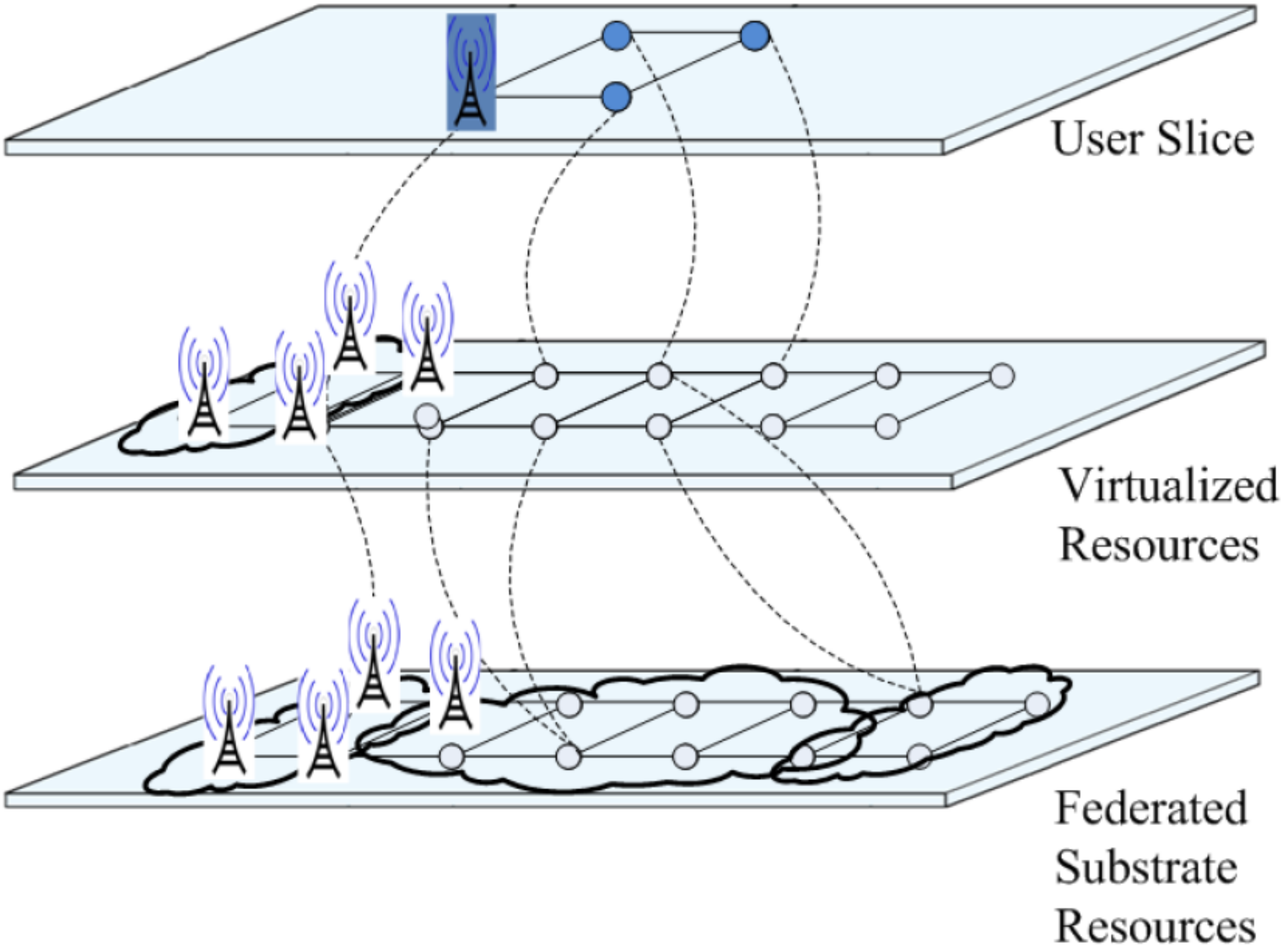
Tutorial Topics

- Future Internet Performance Architecture (FIPA)
  - Requirements and Design

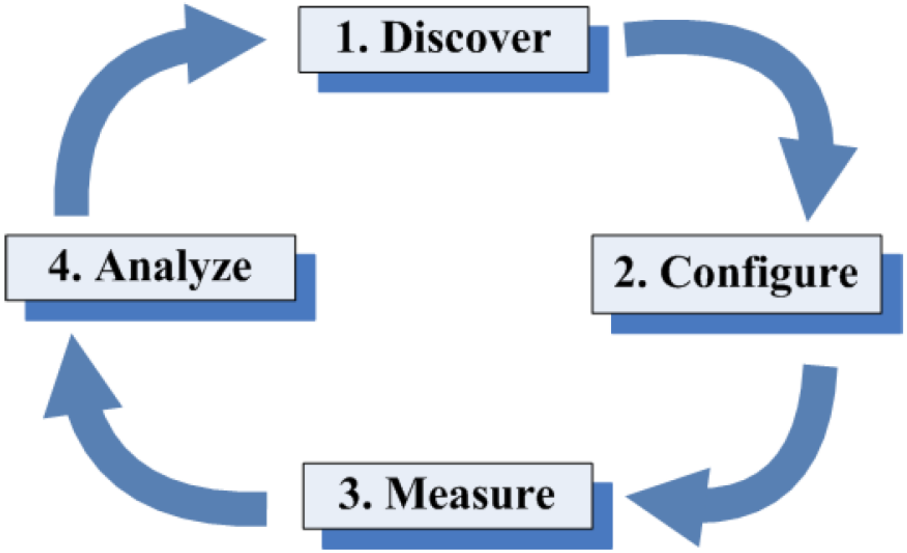
- OnTimeMeasure-GENI: FIPA Reference Implementation
  - Hands-on Demo
    - Working with instrumentation and measurement services
    - Adding custom metrics of your Application

- FIPA Case Study: Dynamic Resource Allocation in Virtual Desktop Clouds

User Slice in Future Internet

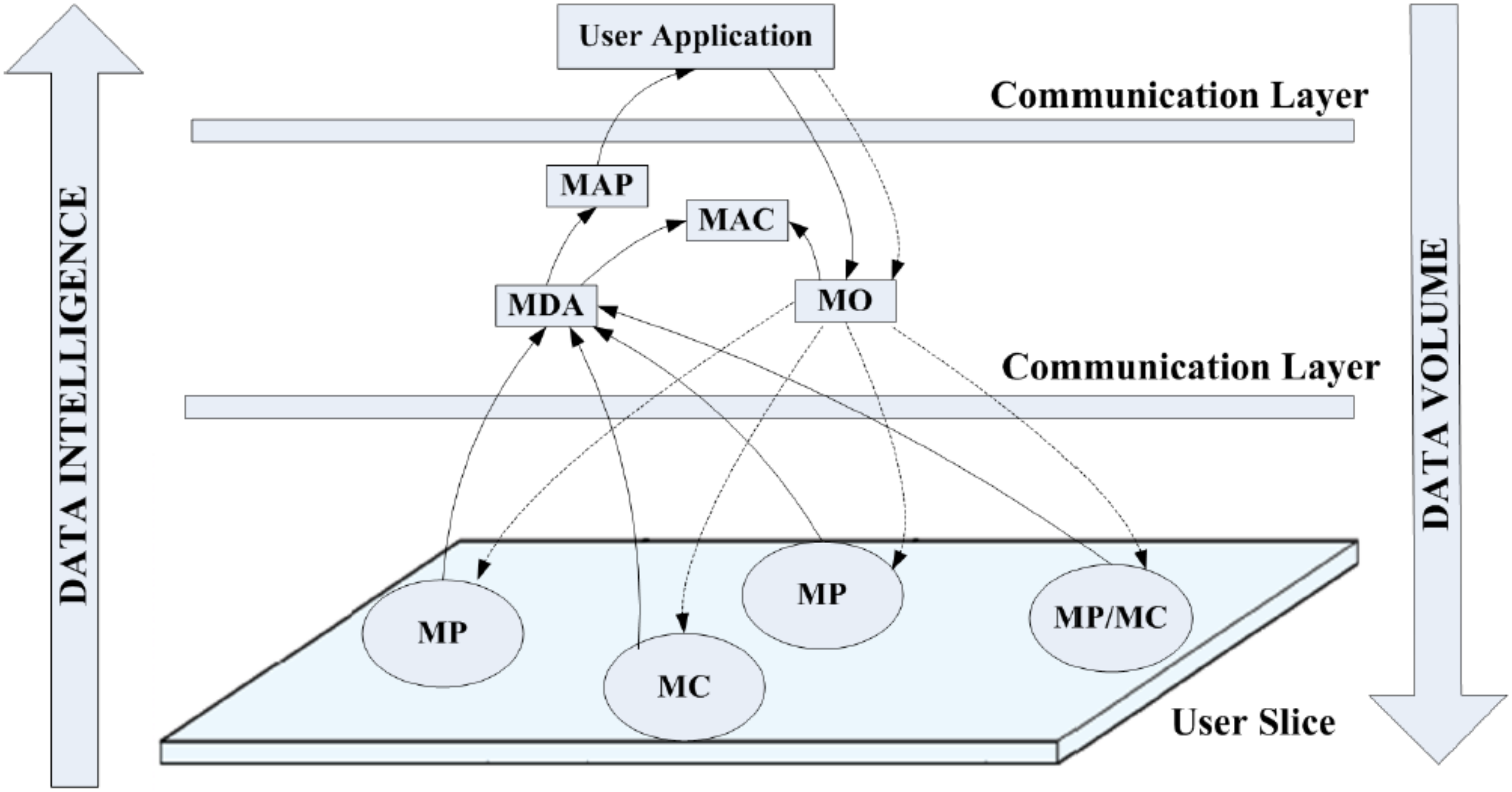


Future Internet Application Life-cycle



- Awareness in host and network components
- Future Internet application’s *autonomicity* attributes
  - Self-configuring, Self-managing, Self-monitoring, Self-optimizing
- Enabling *performance intelligence* in Future Internet applications is vital

FI Performance Architecture (FIPA)



Legend:

→ Data Flow

⇢ Control Flow

MP - Measurement Point

MC - Measurement Collector

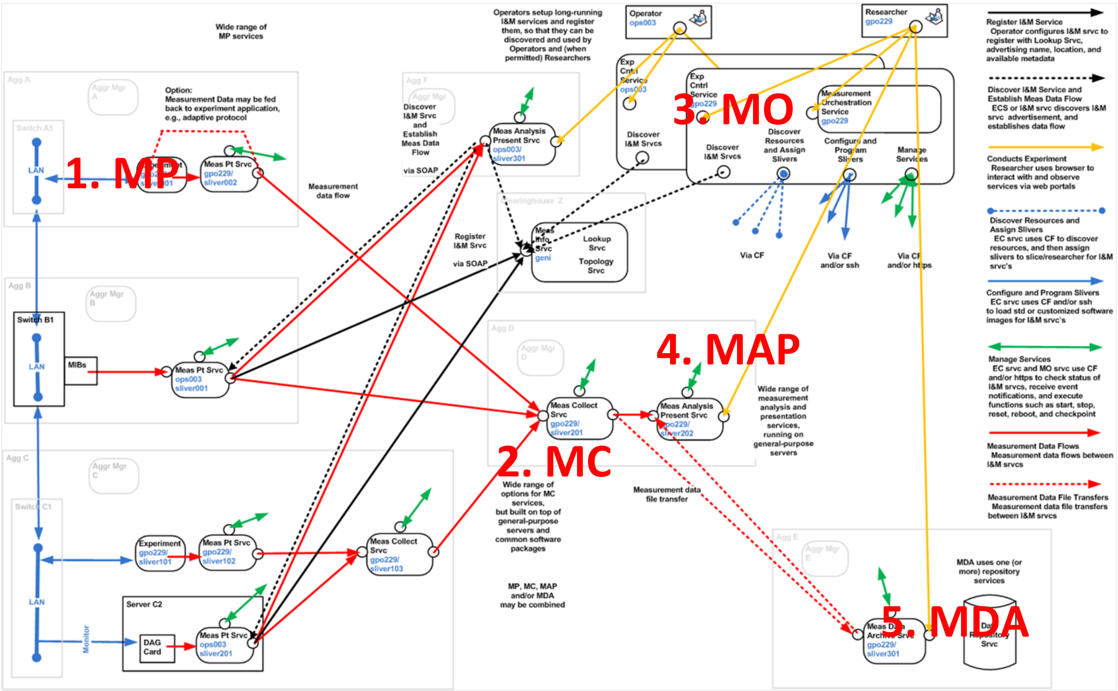
MDA – Measurement Data Archive

MO – Measurement Orchestration

MAC – Measurement Access Control

MAP – Measurement Analysis and Presentation

DIY: I&M service for GENI Experimenters!


Wide range of MP services
Operators setup long-running I&M services and register them, so that they can be discovered and used by Operators and (when permitted) Researchers
Operator
ops003
Researcher
gpo229
Agg A
Aggr Mgr A
Option: Measurement Data may be fed back to experiment application, e.g., adaptive protocol
Switch A1
LAN
1. MP
Experiment
gpo229/ sliver001
Meas Pt Srvc
gpo229/ sliver002
Measurement data flow
Agg F
Aggr Mgr
Discover I&M Srvc and Establish Meas Data Flow
via SOAP
Meas Analysis Present Srvc
ops003/ sliver301
Exp Cntrl Service
ops003
Discover I&M Srvcs
Exp Cntrl Service
gpo229
3. MO
Measurement Orchestration Service
gpo229
Discover I&M Srvcs
Discover Resources and Assign Slivers
Configure and Program Slivers
Manage Services
Via CF
Via CF and/or ssh
Via CF and/or https
Clearinghouse Z
Meas Info Srvc
geni
Lookup Srvc
Topology Srvc
Register I&M Srvc
via SOAP
Agg B
Aggr Mgr B
Switch B1
LAN
MIBs
Meas Pt Srvc
ops003
sliver001
Agg D
Aggr Mgr D
Meas Collect Srvc
gpo229/ sliver201
4. MAP
Meas Analysis Present Srvc
gpo229/ sliver202
Wide range of measurement analysis and presentation services, running on general-purpose servers
2. MC
Wide range of options for MC services, but built on top of general-purpose servers and common software packages
Measurement data file transfer
Agg C
Aggr Mgr C
Switch C1
LAN
Experiment
gpo229/ sliver101
Meas Pt Srvc
gpo229/ sliver102
Meas Collect Srvc
gpo229/ sliver103
Server C2
Monitor
DAG Card
Meas Pt Srvc
ops003
sliver201
MP, MC, MAP and/or MDA may be combined
Agg E
Aggr Mgr E
MDA uses one (or more) repository services
Meas Data Archive Srvc
gpo229/ sliver301
5. MDA
Data Repository Srvc
Register I&M Service
Operator configures I&M srvc to register with Lookup Srvc, advertising name, location, and available metadata
Discover I&M Service and Establish Meas Data Flow
ECS or I&M srvc discovers I&M srvc advertisement, and establishes data flow
Conducts Experiment
Researcher uses browser to interact with and observe services via web portals
Discover Resources and Assign Slivers
EC srvc uses CF to discover resources, and then assign slivers to slice/researcher for I&M srvc's
Configure and Program Slivers
EC srvc uses CF and/or ssh to load std or customized software images for I&M srvc's
Manage Services
EC srvc and MO srvc use CF and/or https to check status of I&M srvcs, receive event notifications, and execute functions such as start, stop, reset, reboot, and checkpoint
Measurement Data Flows
Measurement data flows between I&M srvcs
Measurement Data File Transfers
Measurement data file transfers between I&M srvcs

GENI I&M Architecture (Credit: Harry Mussman, GPO)

Every GENI experiment requires a custom I&M service; a GENI I&M service should remove most of that burden for an experimenter….

GENI Experimenter Use Cases

- Experiment Environment Monitoring
  - Have I got the system and network resources I asked in my slice?
  - Is the experiment environment functioning as expected over my slice lifetime?
  - Can I trace my non-intuitive results in my experiment to a problem in the slice environment?

- Experiment Instrumentation and Measurements (I&M) Management
  - I want to collect active and passive measurements at hop, link, path and slice levels for my experiments…
  - I need measurements feedback with anomaly detection and forecasts to control my experiment progress…
  - I would like to have a measurement framework in my experiment to be configured once and be re-usable for subsequent experiment runs…
  - I want an archive of the experiment measurements collected for offline analysis or to share with colleagues…

Tutorial Topics

- Future Internet Performance Architecture (FIPA)
  - Requirements and Design

- OnTimeMeasure-GENI: FIPA Reference Implementation
  - Hands-on Demo
    - Working with instrumentation and measurement services
    - Adding custom metrics of your Application

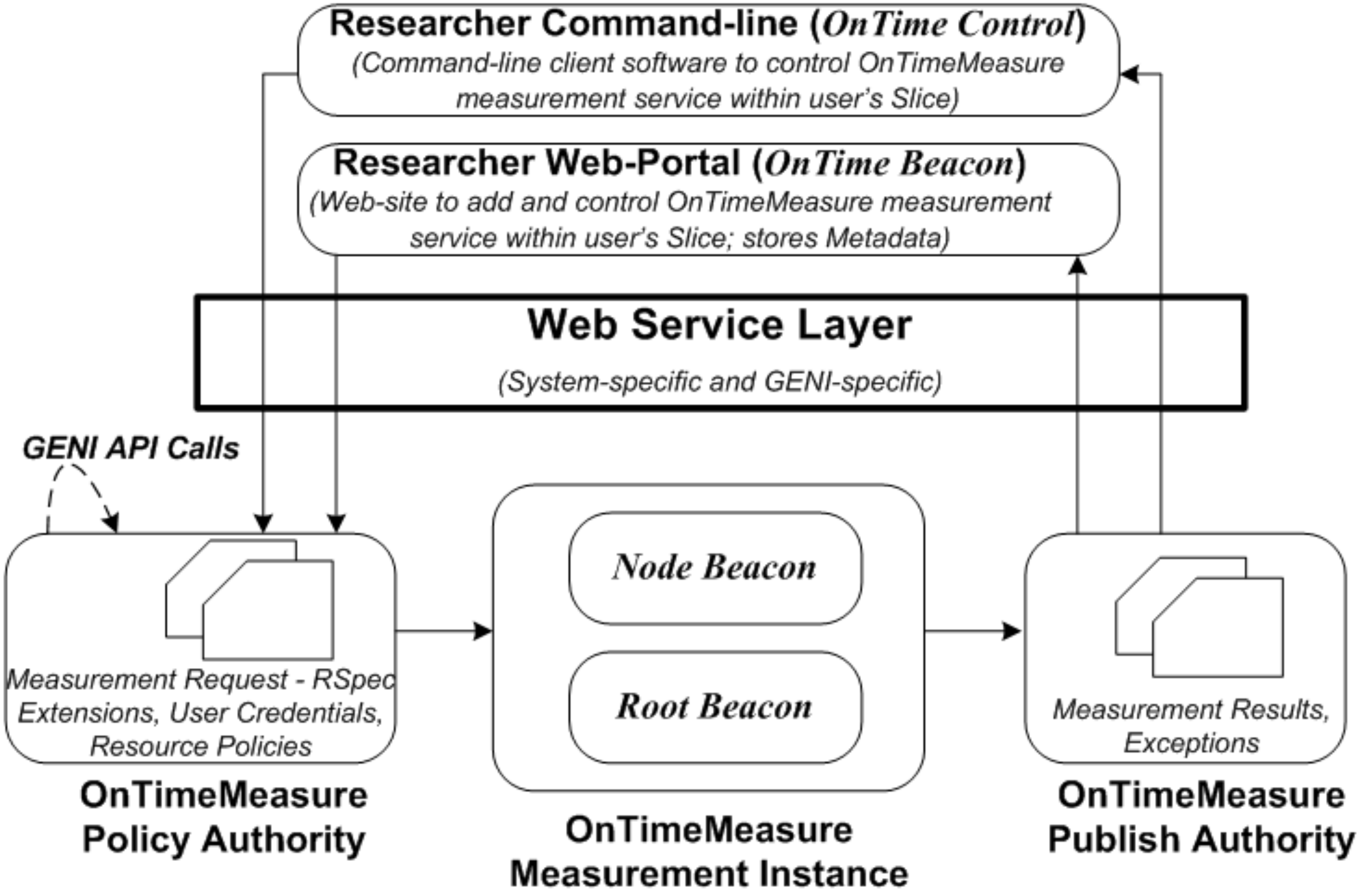
- FIPA Case Study: Dynamic Resource Allocation in Virtual Desktop Clouds

OnTimeMeasure-GENI Overview

GENI Project Wiki – http://groups.geni.net/geni/wiki/OnTimeMeasure

- Goal: Provide GENI community with capabilities for provisioning on-going and on-demand measurement requests
  - Used in forecasting, anomaly detection, and fault-location diagnosis in GENI experiments and GENI operations
- Outcomes:
  - *Software* to perform measurement orchestration and provisioning of measurements
    - For continuous monitoring, persistent measurements storage and processed network measurement feeds
  - *Measurement service* that enables users to utilize OnTimeMeasure software in GENI experiments
    - Registers users, slices, maintains meta-data, and allows user control of measurement service functions
      - Researcher Web-portal – http://ontime.rnet.missouri.edu for *interactive user control* of measurement service
      - Command-line tools for measurement service *control automation*

OnTimeMeasure Architecture



Researcher Command-line (*OnTime Control*)

(Command-line client software to control OnTimeMeasure measurement service within user's Slice)

Researcher Web-Portal (*OnTime Beacon*)

(Web-site to add and control OnTimeMeasure measurement service within user's Slice; stores Metadata)

Web Service Layer

(System-specific and GENI-specific)

GENI API Calls

Measurement Request - RSpec Extensions, User Credentials, Resource Policies

OnTimeMeasure Policy Authority

Node Beacon

Root Beacon

OnTimeMeasure Measurement Instance

Measurement Results, Exceptions

OnTimeMeasure Publish Authority

OnTimeMeasure Software Modules

- Customizable software developed by P. Calyam & Team at The Ohio State University
- Two main modules installed within a GENI experiment slice
  - **Node Beacon**
    - Installs tools that measure network health metrics such as: route changes, delay, jitter, loss, bandwidth
      - TCP/UDP Iperf, Traceroute, Ping, Pathload, OWAMP, etc.
    - Runs measurements based on a schedule and outputs results
  - **Root Beacon**
    - Installs Apache, MySQL and other packages
    - Creates database tables and configuration files
    - Generates measurement schedules for node beacons
    - Collects data and provides dashboard visualization, statistical analysis (i.e., anomaly detection and weather forecasting) with alarm generation

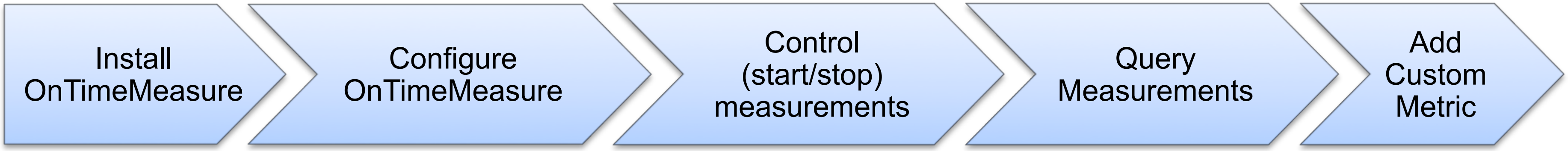
    NOTE: Typically, two or more Node Beacons and one Root Beacon need to be installed on separate dedicated servers in a slice; it is however possible to install Root Beacon on one of the Node Beacon servers

What OnTimeMeasure can provide Experimenters?

- Data Aggregation
  - Your distributed data sources can be controlled and accessed (start/stop/query) in a centralized manner via web-portal or command line
- Data Visualization
  - Measurement data graphs, dashboards
- Data Analysis
  - Time Series files with Anomalies/Time Series with Forecasts/others
  - Ability to use analysis of the measurements to reconfigure the measurement specifications without human intervention or use in research experiment schemes
- Data Archive
  - Slice owners can access and download data and metadata; transfer to GENI archive

OnTimeMeasure-GENI Hands-on Tutorial

Follow OnTimeMeasure Tutorial Steps at –
http://groups.geni.net/geni/wiki/GENIExperimenter/GENIWebinars/DeployingNetworkAwareApplicationsInGENI



Install OnTimeMeasure

Configure OnTimeMeasure

Control (start/stop) measurements

Query Measurements

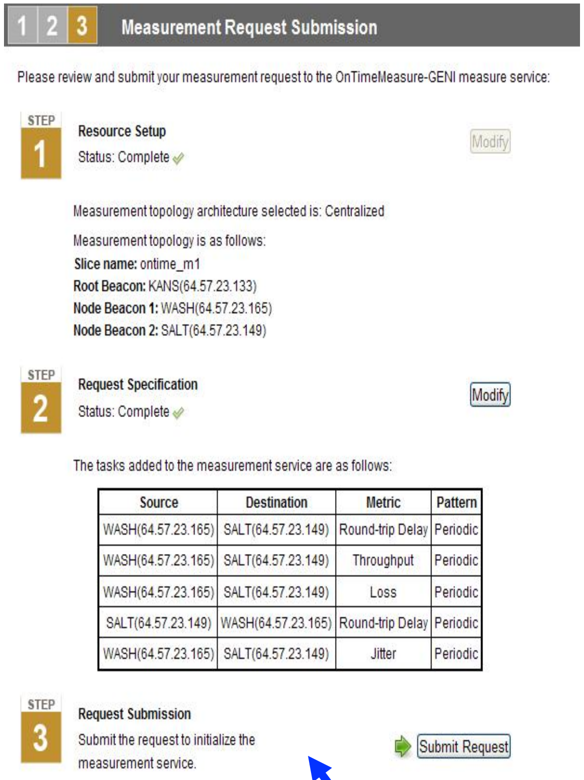
Add Custom Metric

- **Purpose of the Tutorial**
  - **Perform Instrumentation and Measurement of a GENI Slice**
    - Install and configure a distributed measurement Web App viz., “OnTimeMeasure” within a slice. Configuration involves ‘Node Beacons’ at measurement points, and a ‘Root Beacon’ that acts as a central intelligence module.
    - Schedule measurements from the Root Beacon and query/ visualize performance between two compute nodes that host Node Beacons. Performance data collection comprises of network health metrics such as round trip delay, loss, jitter and throughput.
    - Add custom metrics on system health metrics such as CPU utilization

Screenshots

View OnTimeMeasure Demo Videos at – http://ontime.rnet.missouri.edu/demo

Centralized Service Control



1 2 3 **Measurement Request Submission**

Please review and submit your measurement request to the OnTimeMeasure-GENI measure service:

STEP 1 **Resource Setup** Modify

Status: Complete ✓

Measurement topology architecture selected is: Centralized

Measurement topology is as follows:

Slice name: ontime_m1
Root Beacon: KANS(64.57.23.133)
Node Beacon 1: WASH(64.57.23.165)
Node Beacon 2: SALT(64.57.23.149)

STEP 2 **Request Specification** Modify

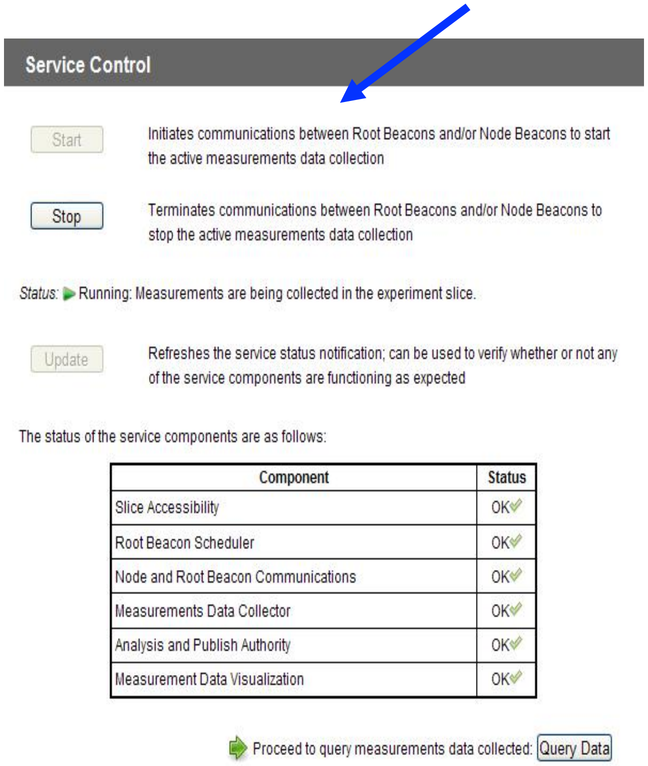
Status: Complete ✓

The tasks added to the measurement service are as follows:

Source	Destination	Metric	Pattern
WASH(64.57.23.165)	SALT(64.57.23.149)	Round-trip Delay	Periodic
WASH(64.57.23.165)	SALT(64.57.23.149)	Throughput	Periodic
WASH(64.57.23.165)	SALT(64.57.23.149)	Loss	Periodic
SALT(64.57.23.149)	WASH(64.57.23.165)	Round-trip Delay	Periodic
WASH(64.57.23.165)	SALT(64.57.23.149)	Jitter	Periodic

STEP 3 **Request Submission**

Submit the request to initialize the measurement service. Submit Request



Service Control

Start — Initiates communications between Root Beacons and/or Node Beacons to start the active measurements data collection

Stop — Terminates communications between Root Beacons and/or Node Beacons to stop the active measurements data collection

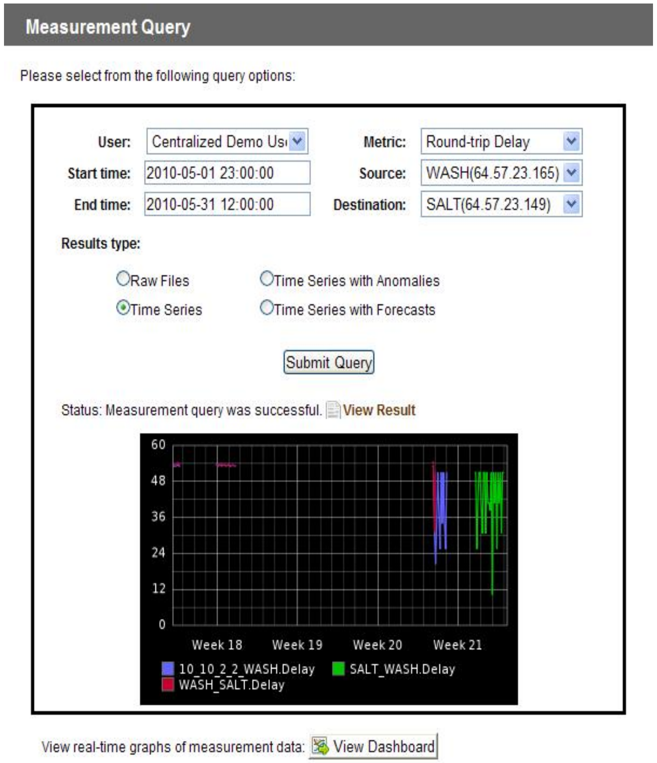
Status: ▶ Running: Measurements are being collected in the experiment slice.

Update — Refreshes the service status notification; can be used to verify whether or not any of the service components are functioning as expected

The status of the service components are as follows:

Component	Status
Slice Accessibility	OK✓
Root Beacon Scheduler	OK✓
Node and Root Beacon Communications	OK✓
Measurements Data Collector	OK✓
Analysis and Publish Authority	OK✓
Measurement Data Visualization	OK✓

Proceed to query measurements data collected: Query Data



Measurement Query

Please select from the following query options:

User:	Centralized Demo Us	**Metric:**	Round-trip Delay
Start time:	2010-05-01 23:00:00	**Source:**	WASH(64.57.23.165)
End time:	2010-05-31 12:00:00	**Destination:**	SALT(64.57.23.149)

Results type:

○ Raw Files ○ Time Series with Anomalies
● Time Series ○ Time Series with Forecasts

Submit Query

Status: Measurement query was successful. View Result

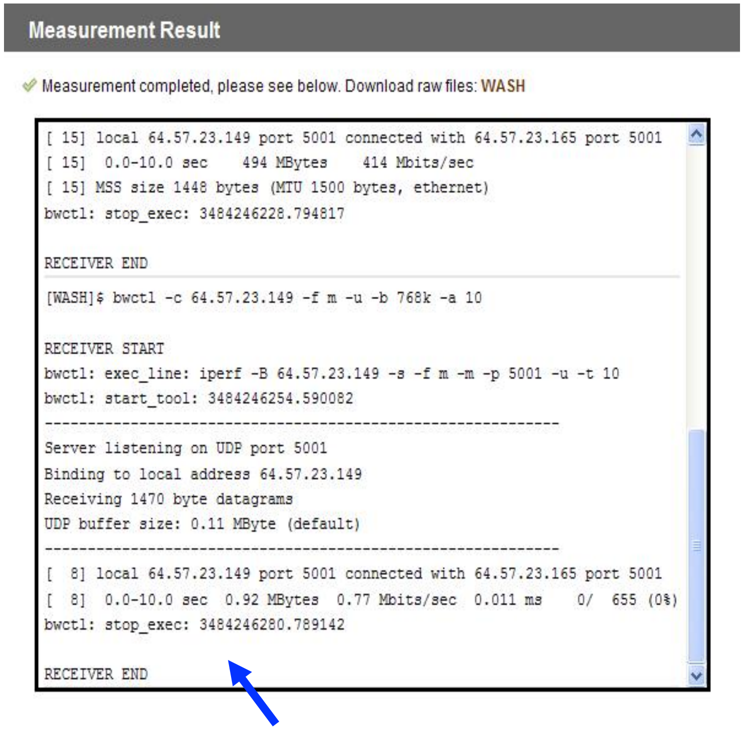
View real-time graphs of measurement data: View Dashboard

Centralized Measurement Request

Centralized Measurement Query

Screenshots (2)

View OnTimeMeasure Demo Videos at – http://ontime.rnet.missouri.edu/demo



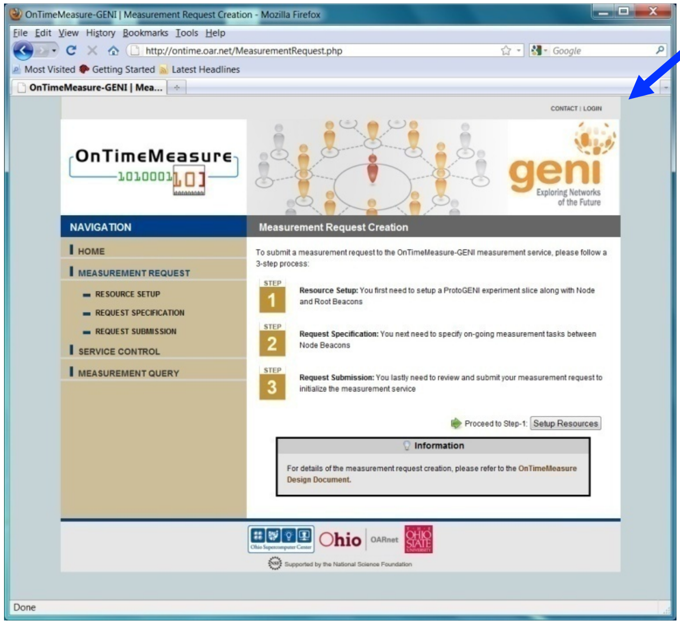
Measurement Result

✓ Measurement completed, please see below. Download raw files: WASH

```
[ 15] local 64.57.23.149 port 5001 connected with 64.57.23.165 port 5001
[ 15]  0.0-10.0 sec    494 MBytes    414 Mbits/sec
[ 15] MSS size 1448 bytes (MTU 1500 bytes, ethernet)
bwctl: stop_exec: 3484246228.794817

RECEIVER END

[WASH]$ bwctl -c 64.57.23.149 -f m -u -b 768k -a 10

RECEIVER START
bwctl: exec_line: iperf -B 64.57.23.149 -s -f m -m -p 5001 -u -t 10
bwctl: start_tool: 3484246254.590082
------------------------------------------------------------
Server listening on UDP port 5001
Binding to local address 64.57.23.149
Receiving 1470 byte datagrams
UDP buffer size: 0.11 MByte (default)
------------------------------------------------------------
[  8] local 64.57.23.149 port 5001 connected with 64.57.23.165 port 5001
[  8]  0.0-10.0 sec  0.92 MBytes  0.77 Mbits/sec  0.011 ms    0/  655 (0%)
bwctl: stop_exec: 3484246280.789142

RECEIVER END
```

Distributed Measurement Result

Researcher Web-Portal



OnTimeMeasure-GENI | Measurement Request Creation - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://ontime.oar.net/MeasurementRequest.php
Google
Most Visited Getting Started Latest Headlines
OnTimeMeasure-GENI | Mea...
CONTACT | LOGIN
OnTimeMeasure
1010001101
geni
Exploring Networks of the Future
NAVIGATION
HOME
MEASUREMENT REQUEST
RESOURCE SETUP
REQUEST SPECIFICATION
REQUEST SUBMISSION
SERVICE CONTROL
MEASUREMENT QUERY
Ohio
OARnet
Done

Measurement Request Creation

To submit a measurement request to the OnTimeMeasure-GENI measurement service, please follow a 3-step process:

STEP 1 **Resource Setup:** You first need to setup a ProtoGENI experiment slice along with Node and Root Beacons

STEP 2 **Request Specification:** You next need to specify on-going measurement tasks between Node Beacons

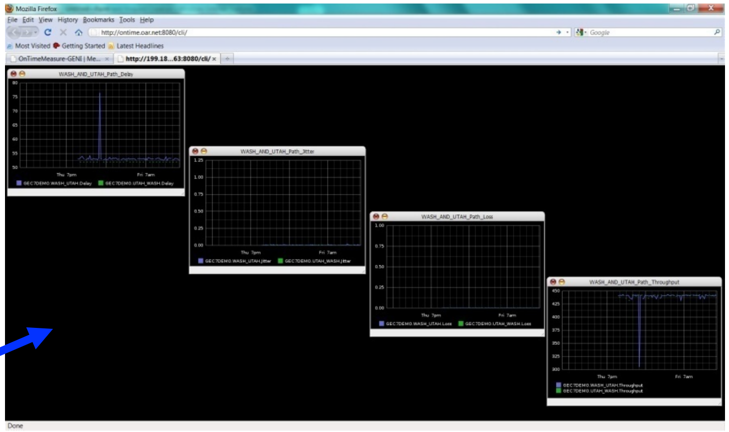
STEP 3 **Request Submission:** You lastly need to review and submit your measurement request to initialize the measurement service

Proceed to Step-1: Setup Resources

Information

For details of the measurement request creation, please refer to the OnTimeMeasure Design Document.

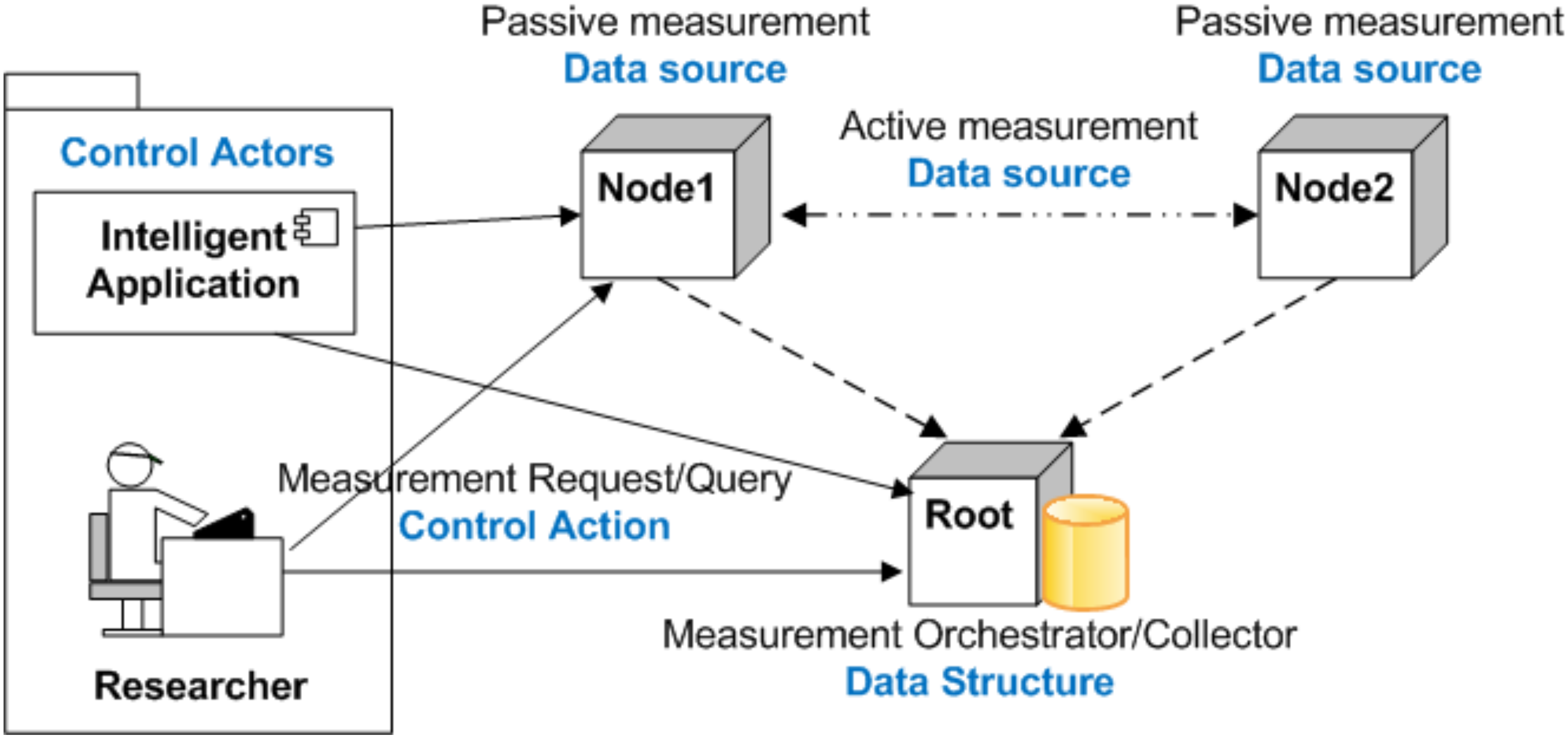
Supported by the National Science Foundation



Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://ontime.oar.net:8080/cli/
WASH_AND_UTAH_Path_Delay
WASH_AND_UTAH_Path_Jitter
WASH_AND_UTAH_Path_Loss
WASH_AND_UTAH_Path_Throughput
Done

User Customizable Dashboard

Adding a Custom Metric into OnTimeMeasure



Definitions for Custom Metric Specification

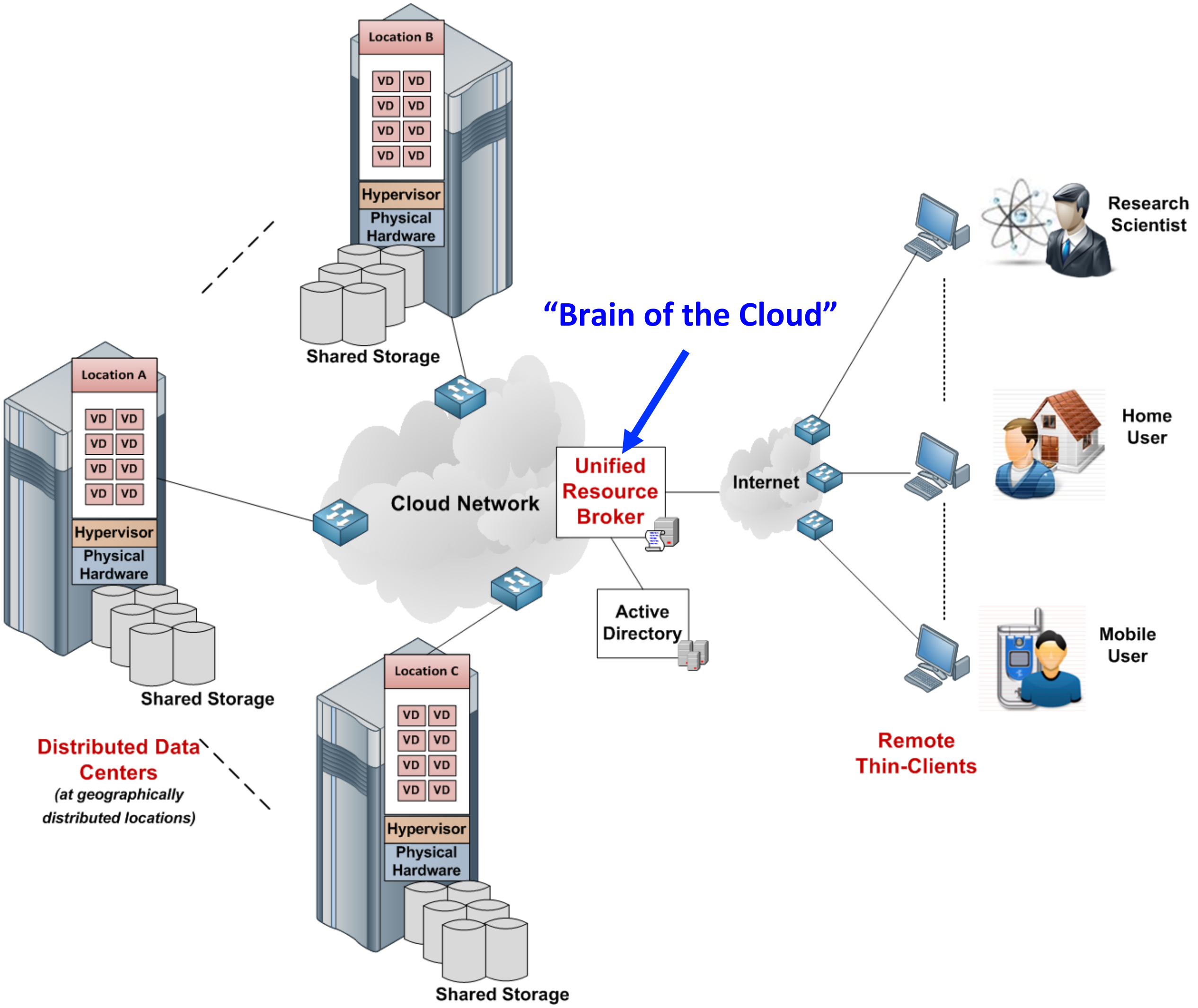
- ***Control Actors***: Project members or your research experiment who can provide the “slice password” to the OnTimeMeasure instance in an experiment slice, and would access the I&M data or share the data with other actors
- ***Data Sources***: Data generation tools deployed in slice nodes; the tool would communicate with other nodes to perform active measurements or inherently collects passive measurements in on-going and on-demand manners
- ***Data Structures /Data Types***: Measurement data that needs to be stored in a database with appropriate data structure and data types
- ***Control Actions***: Control both start and stop of the data generation tool, control how to retrieve and utilize the data e.g., dashboard, plots, time series files with anomaly annotation

More Information at - http://groups.geni.net/geni/wiki/OTM-CustomMetric

Tutorial Topics

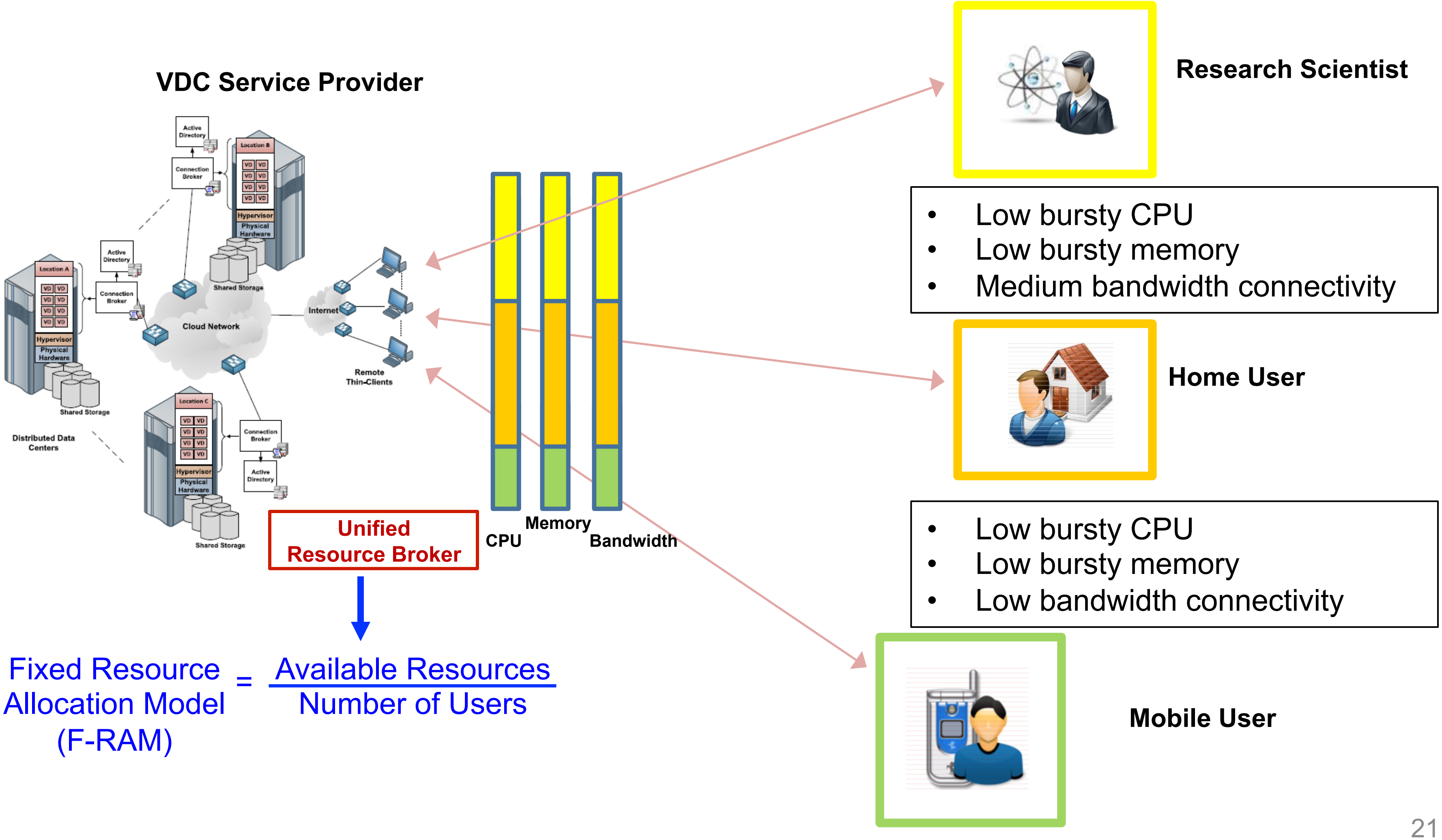
- Future Internet Performance Architecture (FIPA)
  - Requirements and Design
- OnTimeMeasure-GENI: FIPA Reference Implementation
  - Hands-on Demo
    - Working with instrumentation and measurement services
    - Adding custom metrics of your Application
- FIPA Case Study: Dynamic Resource Allocation in Virtual Desktop Clouds

Virtual Desktop Clouds (DaaS)



VDCs Today – Overprovisioning and Guesswork…

- High consistent CPU
- High consistent memory
- High bandwidth connectivity



- Low bursty CPU
- Low bursty memory
- Medium bandwidth connectivity

- Low bursty CPU
- Low bursty memory
- Low bandwidth connectivity

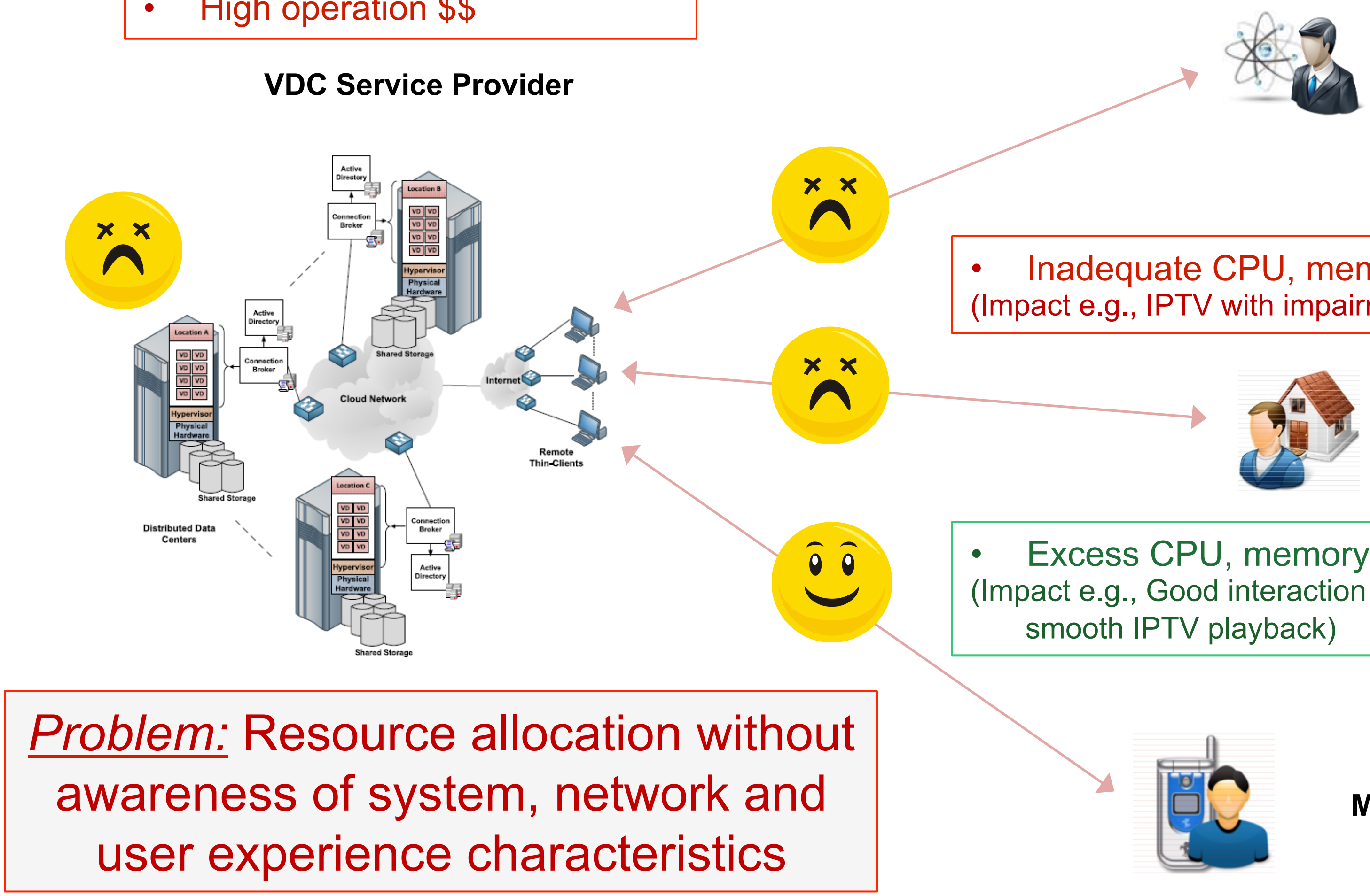
Unified Resource Broker

$$\text{Fixed Resource Allocation Model (F-RAM)} = \frac{\text{Available Resources}}{\text{Number of Users}}$$

Overprovisioning and Guesswork Fails!

- Calls from unhappy customers
- High operation $$

VDC Service Provider



- Inadequate CPU, memory and bandwidth

(Impact e.g., Slow interaction response times)

Research Scientist

- Inadequate CPU, memory and bandwidth

(Impact e.g., IPTV with impairments and slow playback)

Home User

- Excess CPU, memory and bandwidth

(Impact e.g., Good interaction response times and smooth IPTV playback)

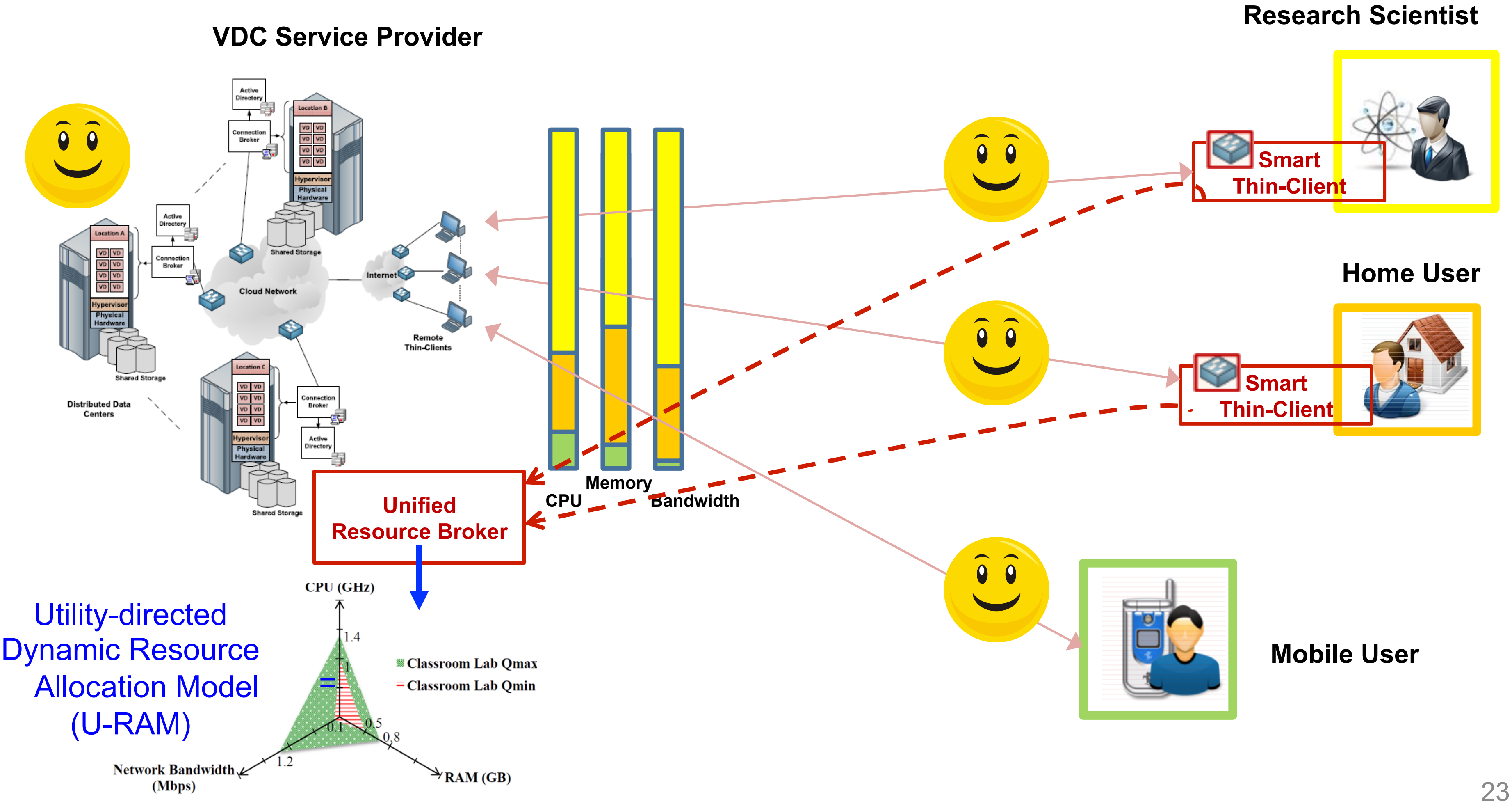
Mobile User

Problem: Resource allocation without awareness of system, network and user experience characteristics

VDCs in the Future – Smart set-top boxes at user sites

- Happy customers
- Low operation $$

- Utility-directed CPU, memory and bandwidth
(Impact e.g., Good interaction response times and smooth IPTV playback)



VDC Service Provider

Research Scientist

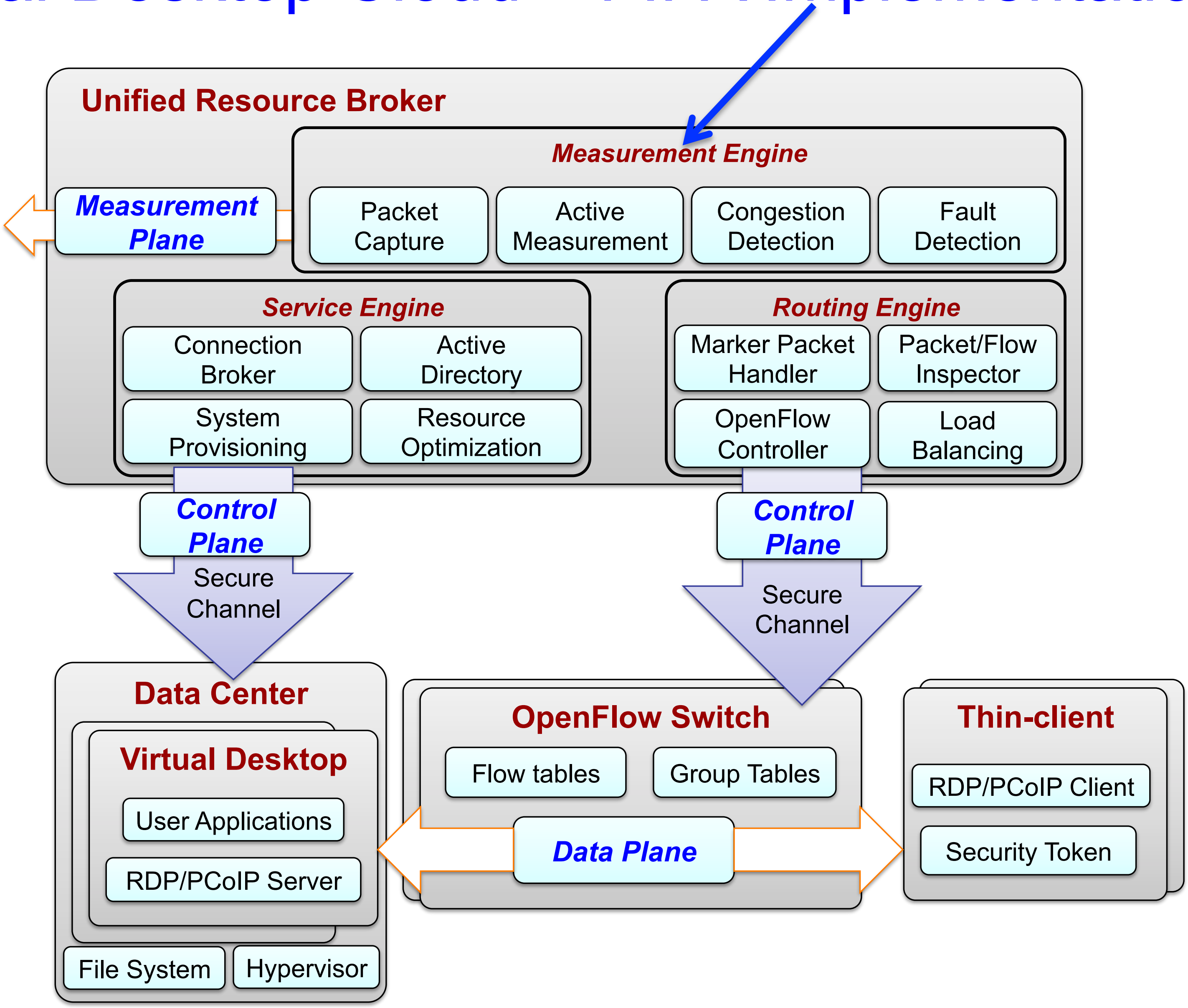
Home User

Mobile User

Unified Resource Broker

Utility-directed Dynamic Resource Allocation Model (U-RAM) =

Virtual Desktop Cloud – FIPA Implementation



Virtual Desktop Provisioning Experiment Demonstration

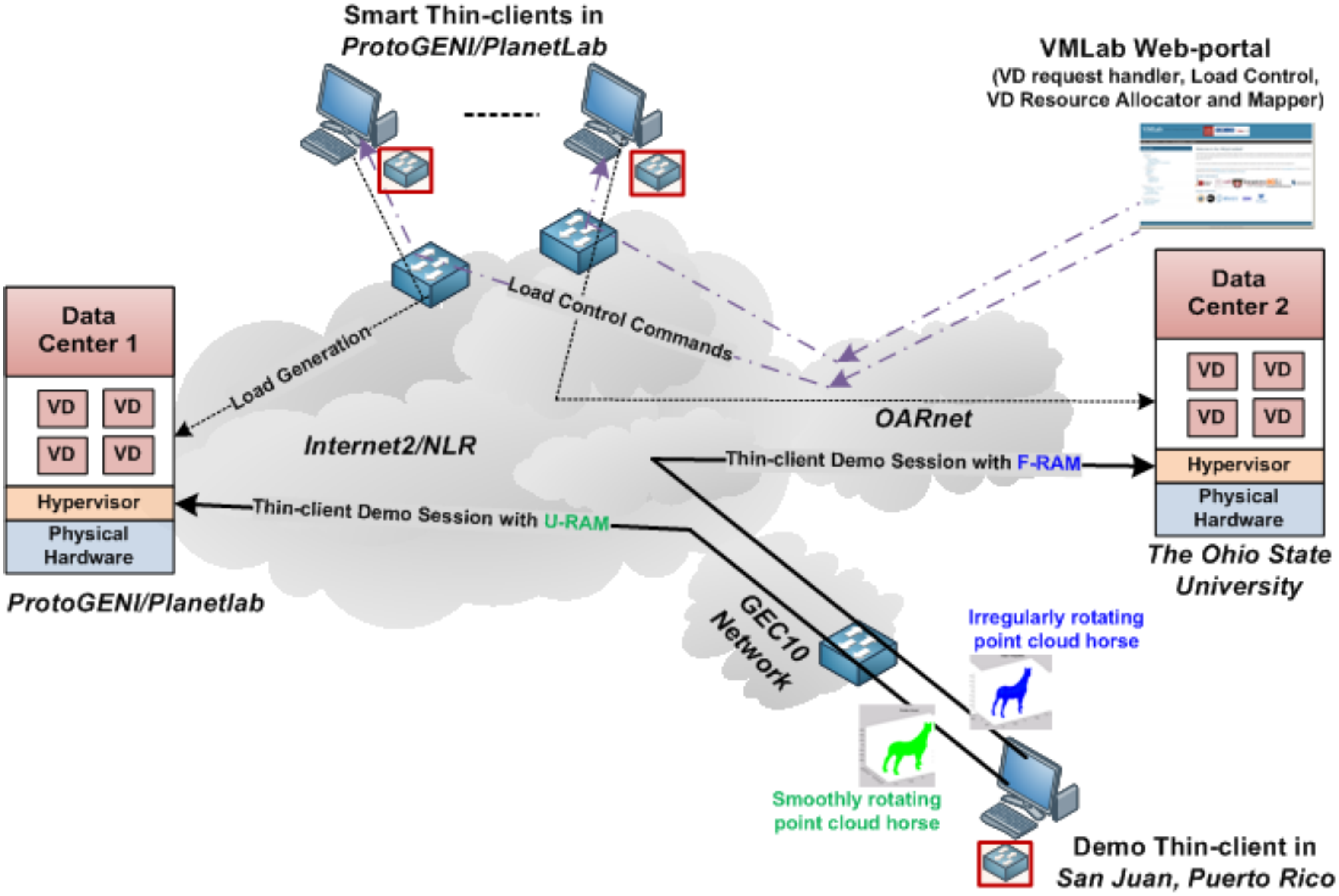
- Compared U-RAM and F-RAM performance

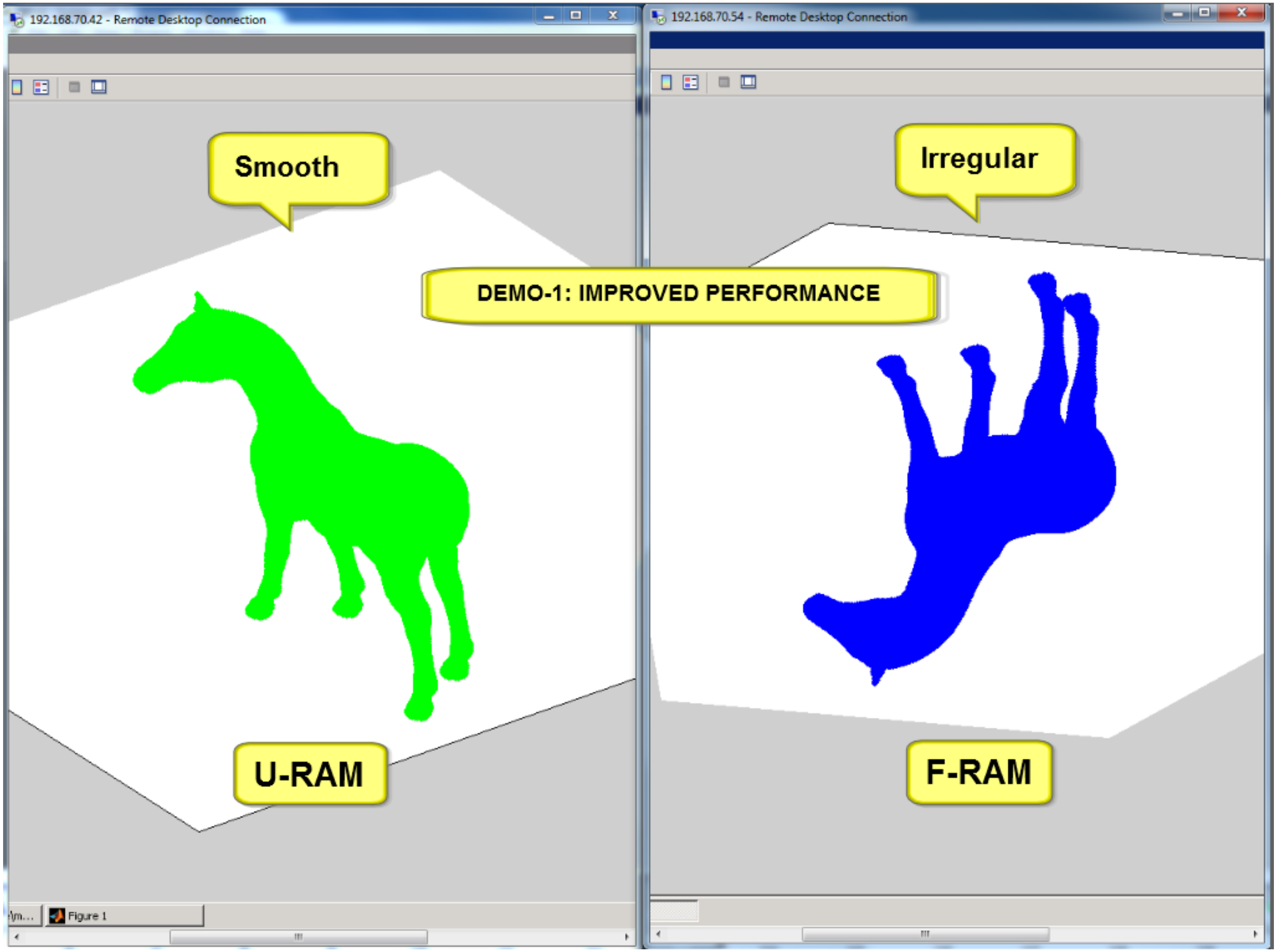
- Created a datacenter in VMLab
  - One physical server each in VMLab for U-RAM and F-RAM

- VDC clients on ProtoGENI slice in GEC10
  - Developed a web-portal to launch VDC clients and control network emulation for demonstration

- Leveraged OnTimeMeasure’s custom metric creation capability
  - Path-based measurements of network health such as delay, available bandwidth, loss
  - Host-based measurements from VMware VDI tools such as CPU, memory, number of VM connections

GEC10 Experiment Demonstration Setup





DEMO-1: IMPROVED PERFORMANCE

Smooth

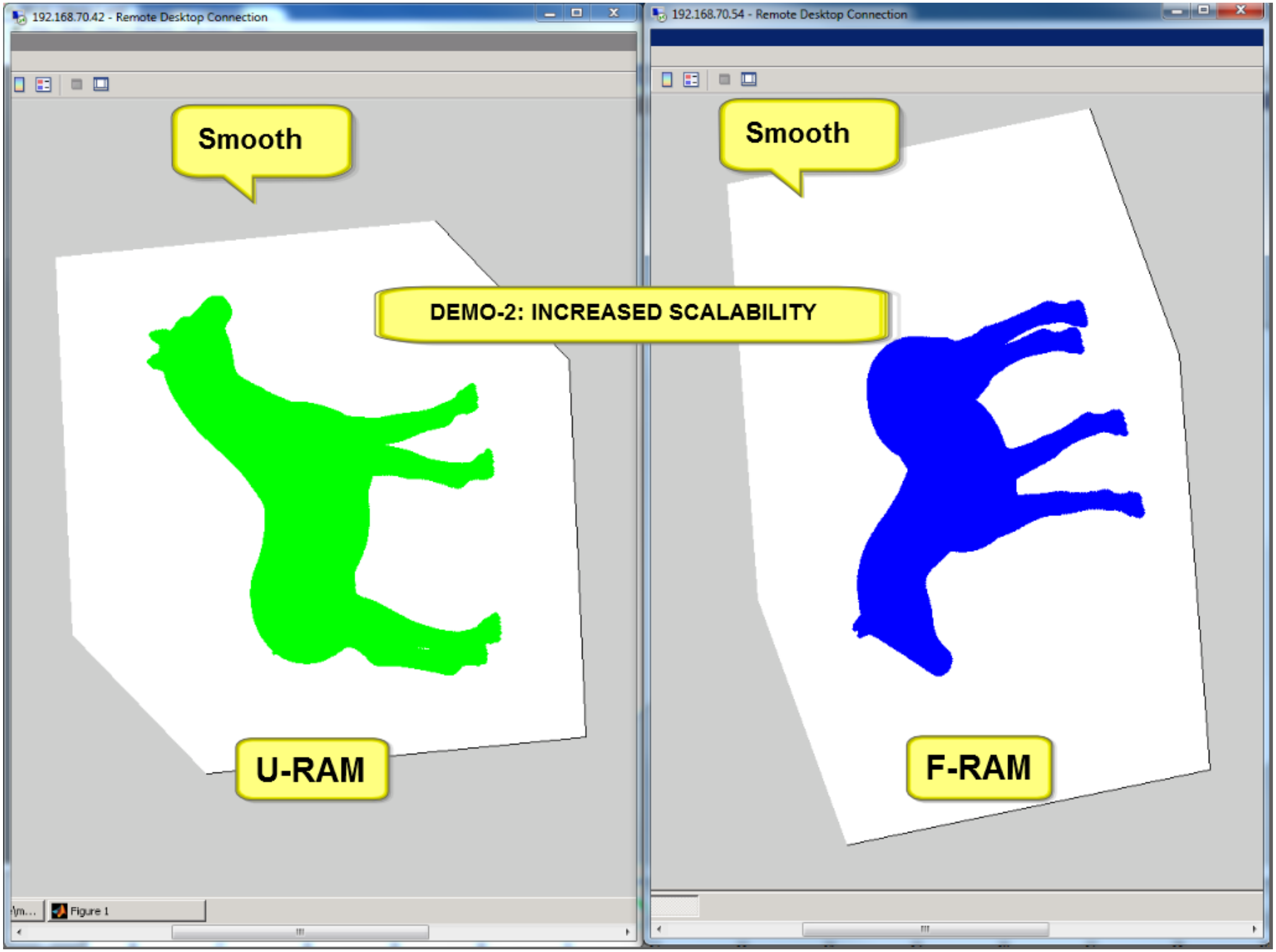
U-RAM

Irregular

F-RAM

192.168.70.42 - Remote Desktop Connection
'/m...
Figure 1
192.168.70.54 - Remote Desktop Connection

]


192.168.70.42 - Remote Desktop Connection
192.168.70.54 - Remote Desktop Connection
Figure 1

Smooth

Smooth

DEMO-2: INCREASED SCALABILITY

U-RAM

F-RAM

Virtual Desktop Placement after U-RAM Provisioning

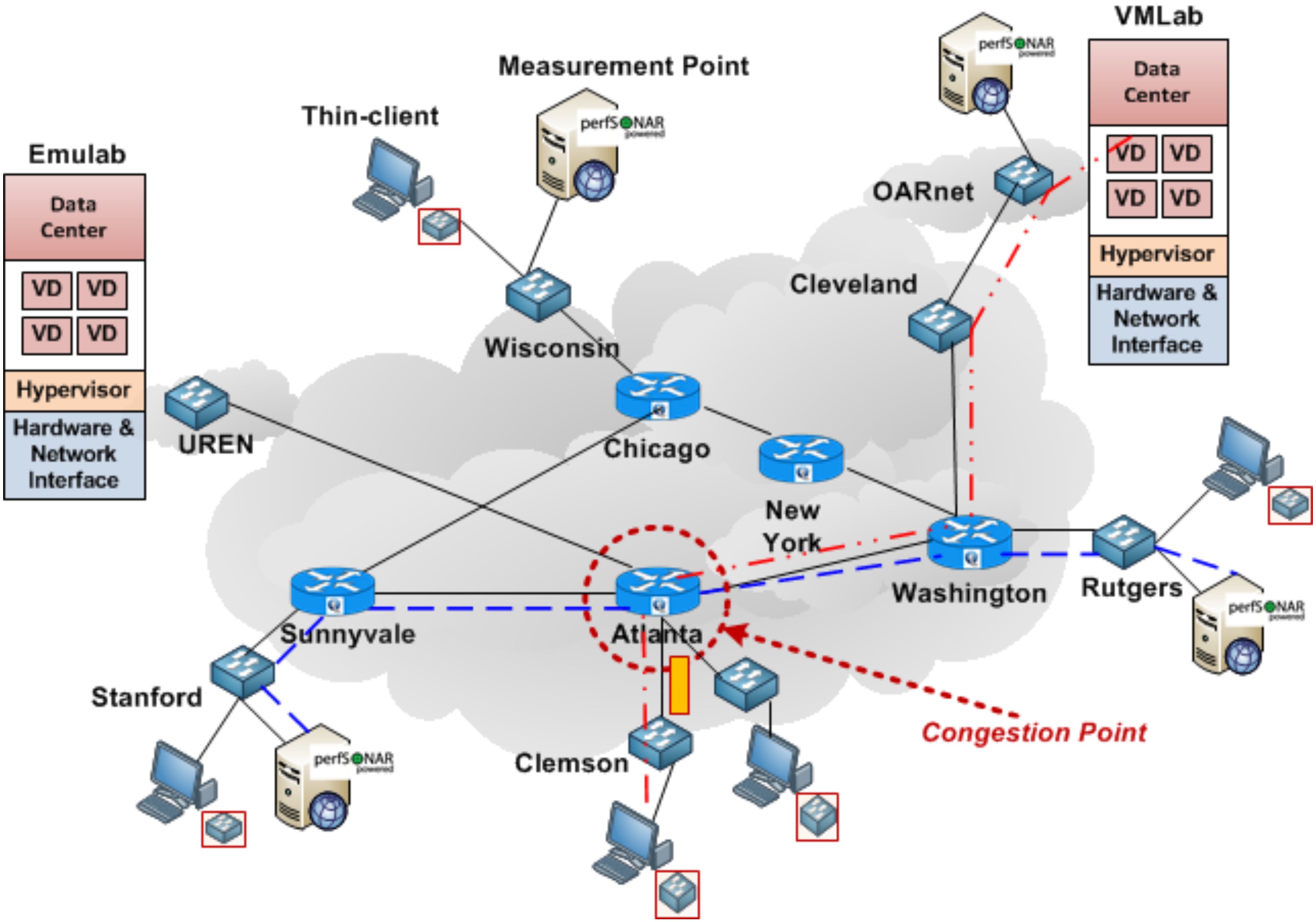
- Placement decisions involving data centers are influenced by:
  - Session latency, Load balancing, Operation cost

- Compared network-adaptation performance when backup path switching occurred upon detecting a cross-traffic congestion scenario

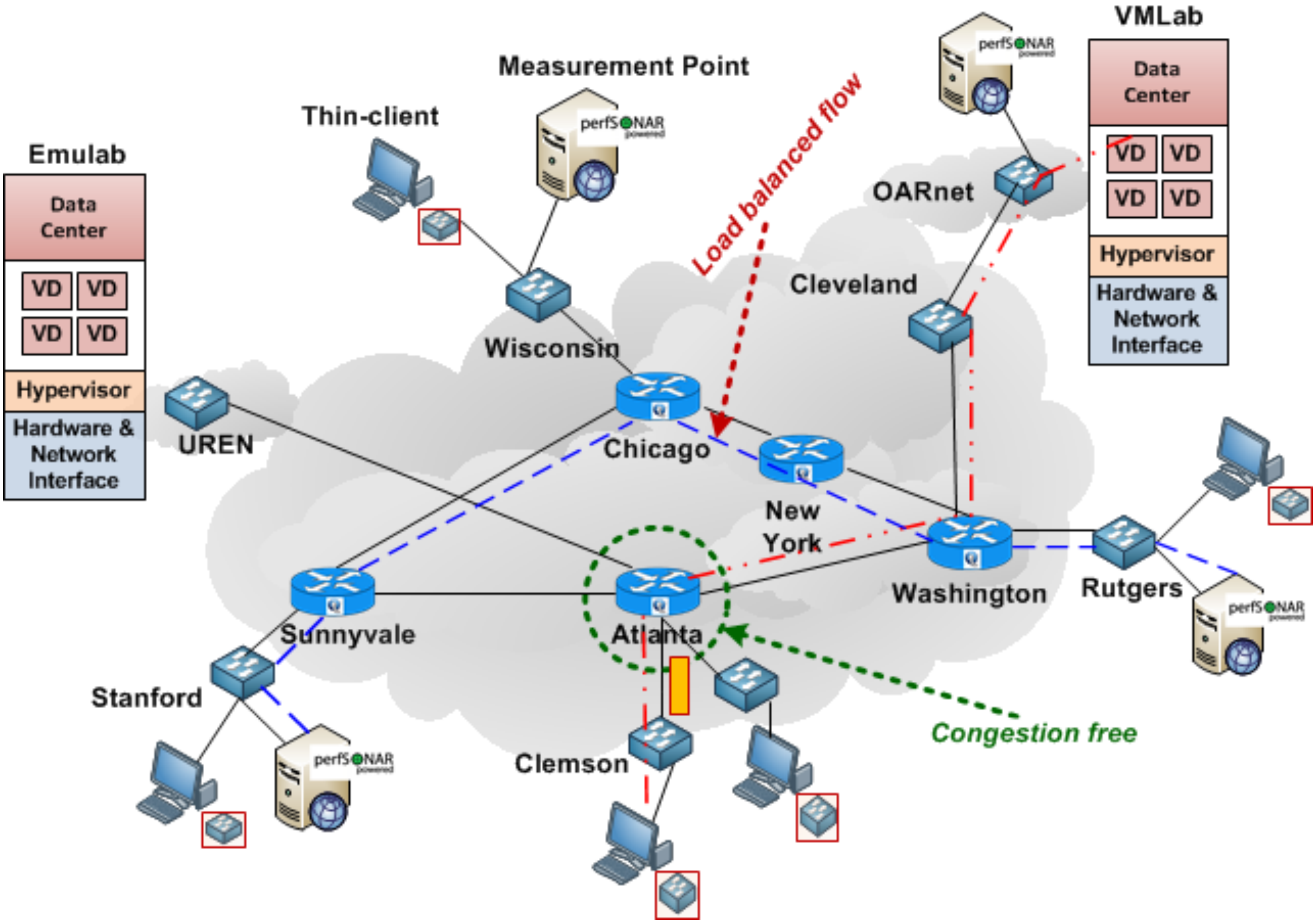
- Multi-domain GENI testbed at GEC15 involved a slice setup
  - Using "OSU VMLab – Utah Emulab" VLAN
  - End-to-end Path: OSU – OARnet – Internet2/NLR Backbone – Utah

- Leveraged OnTimeMeasure's custom metric creation capability
  - Path-based measurements of network health such as delay, available bandwidth, loss
  - Host-based measurements from VMware VDI tools such as CPU, memory, number of VM connections

Virtual Desktop Placement Experiment w/o Load-Balancing



Virtual Desktop Placement Experiment w/ Load-Balancing



OpenFlow-based Network Adaptation Demonstration

Step-1 Route setup

Step-2 Cross-traffic Impact

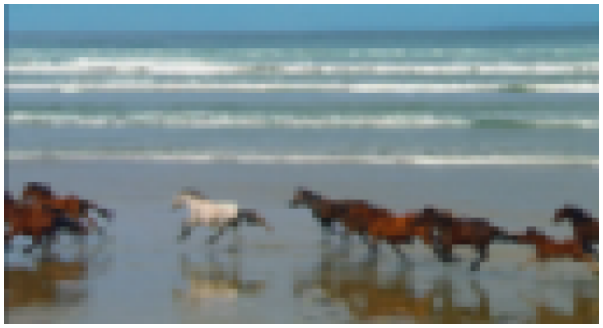
Step-3 Load-balancing Improvement

OpenFlow Switch	Client	In Port	Out Port
ATLA	PG46	20	52
ATLA	PG47	20	52

OpenFlow Switch	Client	In Port	Out Port
SUNNW	PG48	50	51
SUNNW	PG49	50	51
ATLANTA	PG46	52	52
ATLANTA	PG47	52	52
ATLANTA	PG46	20	52
ATLANTA	PG47	20	52

OpenFlow Switch	Client	In Port	Out Port
ATLANTA	PG46	20	52
ATLANTA	PG47	20	52
SUNNW	PG48	50	52
SUNNW	PG49	50	52

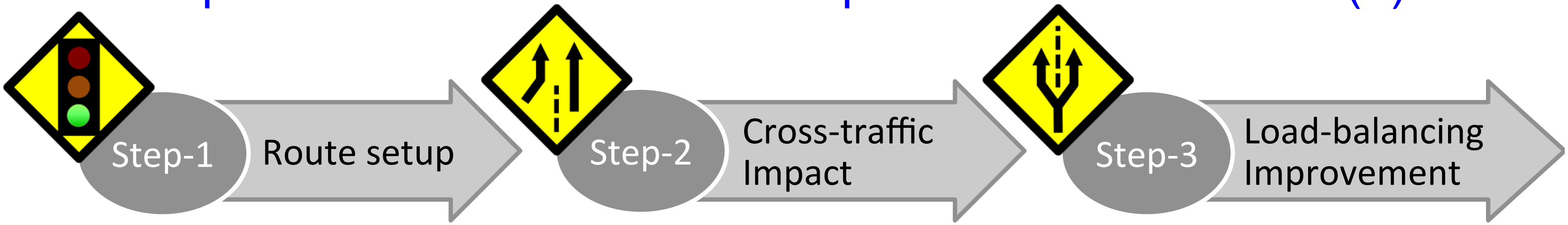
Video runs smooth, GUI applications are responsive



Video freezes, disconnects, GUI applications are not responsive

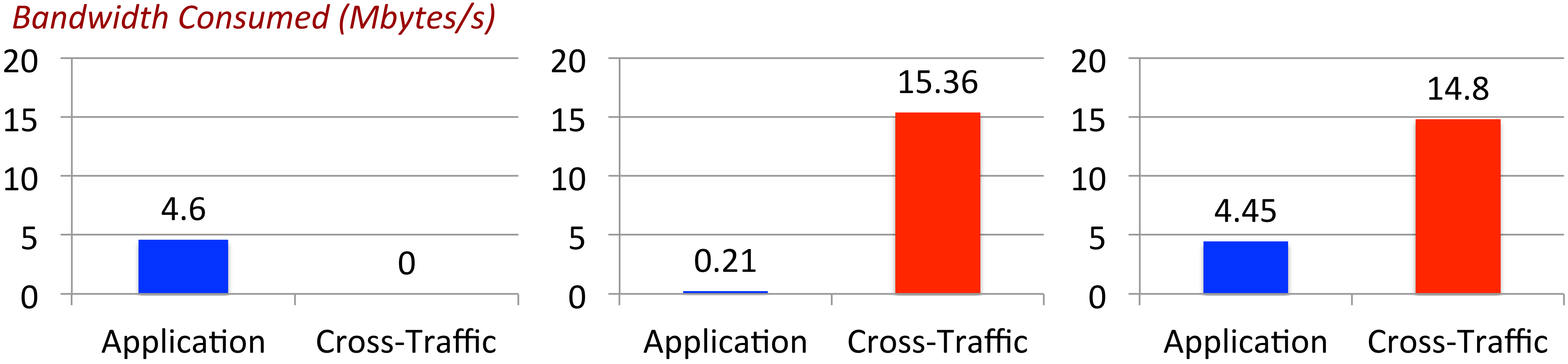
Video runs smooth, GUI applications are responsive

OpenFlow-based Network Adaptation Demonstration (2)

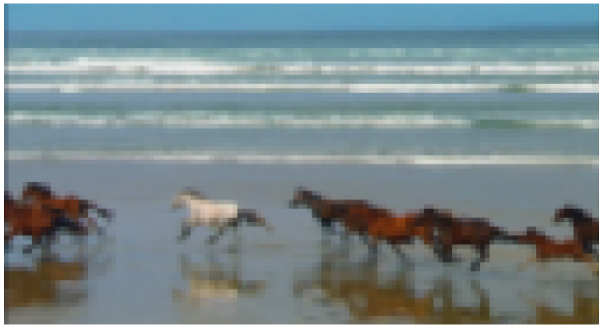

Step-1 Route setup

Step-2 Cross-traffic Impact

Step-3 Load-balancing Improvement



Video runs smooth, GUI applications are responsive


Video freezes, disconnects, GUI applications are not responsive

Video runs smooth, GUI applications are responsive

Thank you for your attention!

References:

P. Calyam, M. Sridharan, Y. Xiao, K. Zhu, A. Berryman, R. Patali, "Enabling Performance Intelligence for Application Adaptation in the Future Internet", *Journal of Communications and Networks (JCN)*, 2011.
http://faculty.missouri.edu/calyamp/publications/fi-ontimemeasure-vdcloud_jcn11.pdf

P. Calyam, S. Rajagopalan, S. Seetharam, A. Selvadhurai, K. Salah, R. Ramnath, "VDC-Analyst: Design and Verification of Virtual Desktop Cloud Resource Allocations", *Elsevier Computer Networks (COMNET)*, 2014.
http://faculty.missouri.edu/calyamp/publications/vdc-analyst-comnet14.pdf