

Project Overview

- Goal:** Provide GENI community with a shared measurement service for provisioning on-going and on-demand measurement requests

Expected Outcomes:

- OnTimeMeasure Software to perform centralized and distributed measurement orchestration and provisioning of active measurements
 - Centralized orchestration for continuous monitoring, persistent measurements storage and processed network measurement feeds
 - Distributed orchestration for on-demand measurement requests without need for persistent measurements storage
- Measurement service that uses OnTimeMeasure software in GENI experiments to enable:
 - Network paths monitoring
 - Network weather forecasting
 - Network performance anomaly detection
 - Network-bottleneck fault-location diagnosis

Software Modules

- Customizable software [*] developed at OSC and OARnet
- Two main modules installed within a GENI experiment slice as part of an active measurement service

- 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 in "raw" and "processed" formats

- Root Beacon

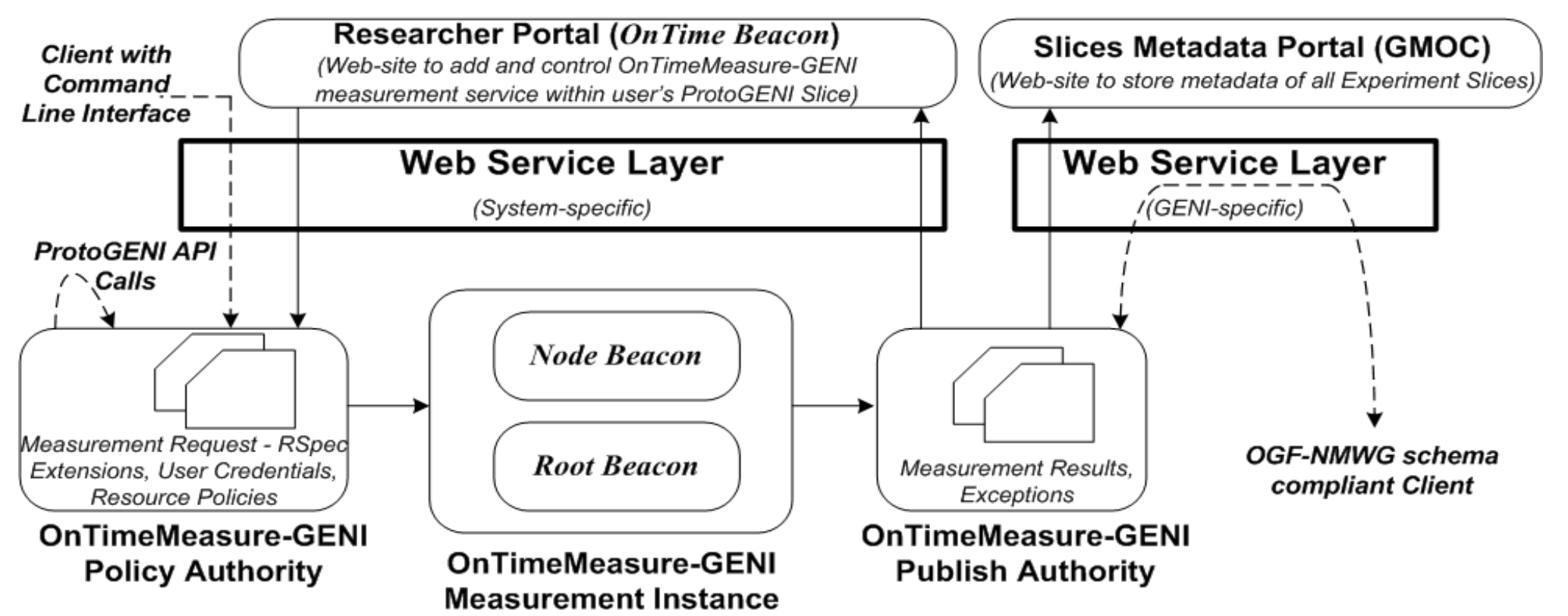
- Installs Apache, MySQL and other packages
- Creates database tables and configuration files
- Generates measurement schedules for node beacons
- Collects data and provides dashboard visualization, statistical analysis (i.e., anomaly detection and weather forecasting) with alarm generation

[*] P. Calyam, C.-G. Lee, E. Ekici, M. Haffner, N. Howes, "Orchestrating Network-wide Active Measurements for Supporting Distributed Computing Applications", *IEEE Transactions on Computers Journal (TC)*, 2006

Service Capabilities

- Active measurement request handling
 - Sampling requirements (e.g., periodic, stratified random, random, adaptive)
 - Active measurement tools to be used (e.g., Ping, Traceroute, Iperf)
 - Measurement topology (e.g., full-mesh, tree, hybrid)
- Enforce policies for measurements scheduling
 - Semantic priorities (e.g., superGENI-er vs. GENI-er)
 - Measurement level restrictions (e.g., allowable measurement bandwidth and measurement flow duration for different nodes/paths/user-roles)
- Provide raw and processed measurement
 - Measurements provisioning interfaces (e.g., raw output of a tool to human/component-service, processed output of multiple tools to a viz application)
 - Measurement use context (e.g., curiosity about network path(s) performance in a new slice, network-awareness in an experiment to develop a novel network control scheme, troubleshoot a network bottleneck affecting an advanced application)
 - Every OnTimeMeasure instance metadata gets stored for GMOC use

OnTimeMeasure Integration in ProtoGENI

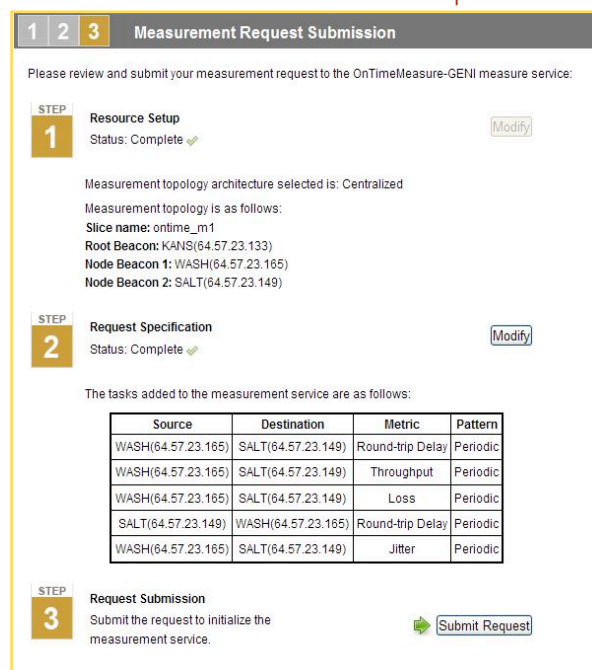


User Workflow

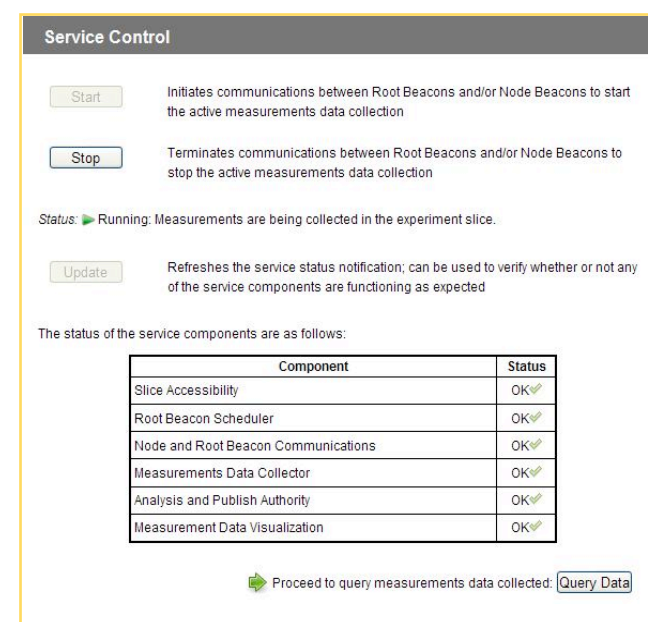
- User creates an experiment slice using ProtoGENI control framework tools
 - For ProtoGENI slice creation details, see - <https://users.emulab.net/trac/protogeni/wiki/TutorialNew>
- User registers at the "Researcher Portal" (<http://ontime.oar.net>), provides Slice Rspec information and requests installation of measurement instance
 - Slice Rspec should include reservation of any required measurement resources
- Each experiment slice needing a measurement service gets its own OnTimeMeasure software instance
 - Node/Root Beacons need to be installed as slivers based on the instructions provided at - <http://groups.geni.net/geni/wiki/OnTime-Installv1>
- Valid login to the "Researcher Portal" allows user to interact with the measurement service in his/her experiment slice. Specifically, the user can:
 - Submit measurement requests
 - Control the measurement service
 - Query measurement data

Screen Shots

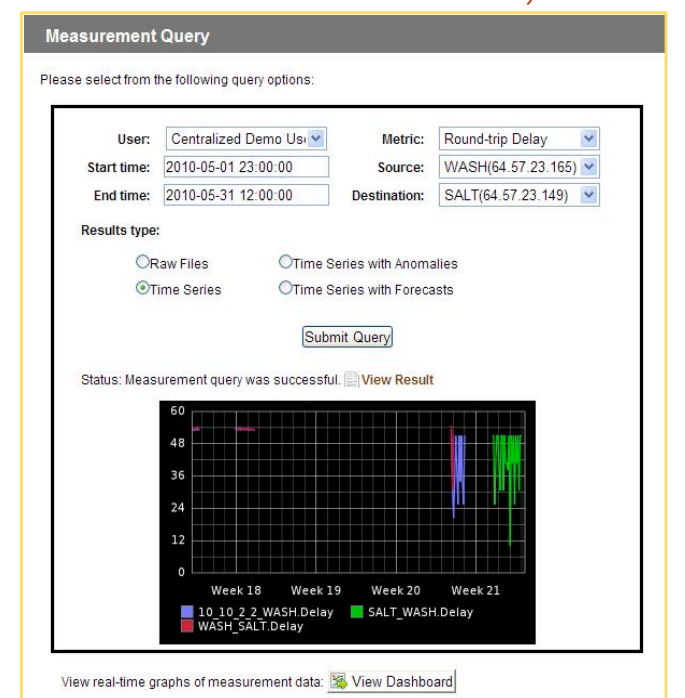
Centralized Measurement Request



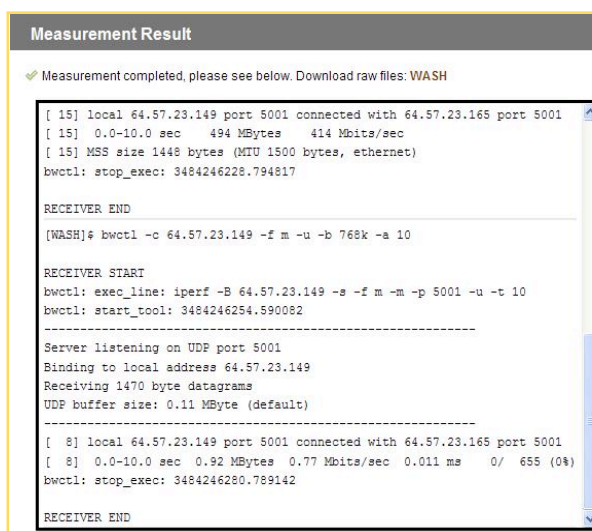
Centralized Service Control



Centralized Measurement Query



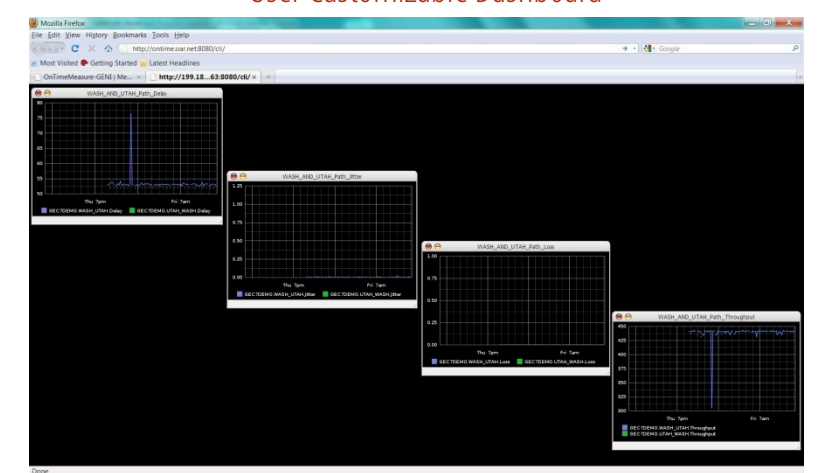
Distributed Measurement Result



Researcher Web-Portal



User Customizable Dashboard



GEC8 Demo Testbed

