

# Improving Component Integration in GEMINI Instrumentize

## Current workflow

1. Instrumentize is invoked by user
2. GENI DesktopParser (GDP) collects manifests from AMs in experiment
3. GDP creates a JSON object with experiment information
  - a. Object includes original XML GENI documents
  - b. GDP representation of the experiment (nodes, links, etc)
4. The GENI Desktop (GD) uses its JSON representation to display slice topology and populate GUI elements.
5. Instrumentize also continues to use GDP-created JSON to completion.
6. Instrumentize invokes unisencoder to convert manifests to UNIS JSON format.
  - a. Uses original XML manifests in GDP-provided JSON
  - b. Sends UNIS JSON to global UNIS
7. The MS, BLiPP, and former Periscope visualization components use UNIS for topology info, service registration, and measurement metadata.

## Notes

While the above approach works, the concern is having UKY and IU using two different representations/systems for keeping track of the experiment in GEMINI. When we begin to support active modifications to the slice, this would entail a requirement to re-parse manifests and update two different information stores. Work is also duplicated in keeping up with changes and supporting parsing from different RSpec formats across control frameworks.

## Proposal #1

Keep current workflow as-is with the addition of having the GDP be tasked with keeping the GEMINI representations up-to-date when the experiment is modified. This means polling or subscribing to each relevant AM, updating the GDP JSON, and invoking unisencoder and pushing updates to UNIS. This doesn't solve the duplication of work since we have to maintain two different parsers, but it allows the system to keep functioning. In effect, UKY and IU maintain two different systems within GEMINI.

## Proposal #2

GDP collects manifests from AMs as before but does not parse. Unisencoder is used to load UNIS with the experiment slice information. The burden of managing differences in RSpecs and converting into a general representation falls to UNIS. Instrumentize and GD query UNIS directly for topology info and for populating GUI elements. UNIS agent subscribes to AMs associated with experiment and maintains a consistent view with what is happening in the control framework after instrumentize completes.

Note #1: the GD could pull the topology information from UNIS and convert to its current expected representation, but that would mean developing a new parser that converts from one JSON representation to another.

Note #2: The proposed change does not eliminate the GDP collecting the initial experiment manifests. The instrumentize process and the GDP interface is needed to kick-off the GEMINI system and figure out what is happening across AMs when the user is ready to do so. The question is what to do with the experiment information internally once that process starts.

## **Recommendations**

UKY team should look at existing UNIS representation to see if it makes sense to use that JSON directly for the GD. A conversion from UNIS->GDP representation could work, but might not be necessary if UNIS can represent everything the GD needs. The schema is extensible.

IU team should investigate what needs to be changed to allow instrumentize to complete with UNIS JSON representation. A version of the scripts could be developed now that uses the UNIS JSON topology object instead of the GDP JSON.

IU team will investigate a UNIS agent that subscribes to AMs (e.g., event channels in Emulab/ProtoGENI) and keeps experiment information up-to-date in GEMINI.