

# **Developer Topic: AM API Review**

## **GEC 19**

### **Atlanta, GA**

**Aaron Helsing, GPO**  
**March 19, 2014**  
**[www.geni.net](http://www.geni.net)**

- A common API for reserving resources at Aggregates
- Version 2 is widely supported
  - [http://groups.geni.net/geni/wiki/GAPI\\_AM\\_API\\_V2](http://groups.geni.net/geni/wiki/GAPI_AM_API_V2)
- Version 3 has been adopted
  - [http://groups.geni.net/geni/wiki/GAPI\\_AM\\_API\\_V3](http://groups.geni.net/geni/wiki/GAPI_AM_API_V3)
- Some changes for Version 4 have been adopted
- Multiple change proposals exist
  - [http://groups.geni.net/geni/wiki/GAPI\\_AM\\_API\\_DRAFT](http://groups.geni.net/geni/wiki/GAPI_AM_API_DRAFT)
- The Federation Aggregate API will build on this
  - As Brecht described earlier

- Implement Version 3!
  - Particularly Allocate/Provision for 2 phase commit
- Ratify some widely accepted (and often implemented) change proposals
  - Now
- Discuss any other required items
  - Now and at the Coding Sprint

- Some change proposals seem widely accepted, often implemented – but not formally adopted.
  - Adopt these for v4 and optional in v3
  - Process: “Hearing no reasoned objection, considered adopted.”
- **Change Set N: Add `geni_am_code` and `geni_am_type` to `GetVersion`, and make AM-specific prefixes consistent**
  - Approve?
- **Change Set O1: Allow periods and underscores in sliver names**
  - Approve?

- **Change Set O2:** Formally specify legal characters in `RSpec type`, `version`, `schema`, and `namespace`; **credential type**; `geni_operational_status`
  - Approve?
- **Change Set P:** Support Speaks-For
  - Approve?
- **Change Set Q:** Allow changing users and keys on existing slivers
  - Through `Provision()` and `Update()`
  - Through `PerformOperationalAction()`
  - Approve?

- **Change Set T: Long Lived Slices**
- **Change Set AB: Case Sensitivity**
- **Change Set S: AM URN in GetVersion**
- **Change Set O3: Support Unicode**
- **Change Set V: Disk Images**

- It is a pain to renew your slivers at all your aggregates, particularly if you are running a service
- AMs ultimately control how long they will give out resources
- Experimenters would prefer to set a long expiration once
- Options
  - Add a privilege in the existing slice credential from SA
  - New slice credential (same schema) from existing SA
  - New credential from *each* AM
  - Run a cron job to incrementally renew

- Should slice names be case INsensitive?
- URNs, URLs, etc are case sensitive
- Authorities in URNs are really domain names, which are case INsensitive
- Usernames are explicitly case INsensitive but may be case preserving
- The AM API does not specify if slice names or sub-authorities (projects) are case sensitive



- Some aggregates use project names and slice names to generate hostnames – which are case INsensitive
- The Federation API does not specify if these fields are case sensitive
- **Proposal:**
  - Slice names and sub-authorities should be case INsensitive for project names and slice names, but case preserving\* to support existing aggregates
    - \*: Match case insensitively, but report the slice name using the case specified.
  - GENI URNs remain technically case sensitive

- What is the URN (name) of this aggregate?
- Having this information would allow tools to verify which RSpec elements map to the service they are talking to
  - As in the `component_manager` in the RSpec
- Or is this a list of URNs, since aggregates may front for multiple component managers?

# Change Set O3: Support Unicode

- The AM API should allow for unicode data to be passed
- We should allow tools & servers that do not support unicode to *co-exist* with those that do
  - Do not *require* anything of existing tools & aggregates
- Aggregates must advertise that they accept unicode
- Clients should expect unicode from such an aggregate
  - Specifically in response to a query that used unicode input
  - Hopefully not in response to a query that did not use unicode input, to allow for clients that cannot handle unicode
- Or do calls need an option  
“`client_supports_unicode`”?

- Support creating custom images, saving VM state
- ProtoGENI provides an API for creating, listing and deleting disk images based on snapshots of VMs, but you have to start with their images
- ExoGENI provides ImageProxy for managing images plus a sandbox for creating images, but you cannot save state
- Proposal to provoke discussion
  - Optional API
  - Based on ProtoGENI API
  - Save the state of running nodes and use those saved images on other nodes