

Spiral 4 Target AM APIs: must use GENI AM API, for universality

GENI AM API v1
GENI rspec v3 (PG rspec v2)
Now

GetVersion
ListResources
CreateSliver
DeleteSliver
SliverStatus
RenewSliver
Shutdown

request all resources at start
no UpdateSliver (therefore, can't
add node or add link)

GENI AM API v2
GENI rspec v3
Avail GEC13

flexible arguments and returns
(property lists)

request all resources at start
no UpdateSliver (therefore, can't
add node or add link)

Spiral 5 Target AM APIs

GENI AM API v3
Avail GEC15? GEC16?

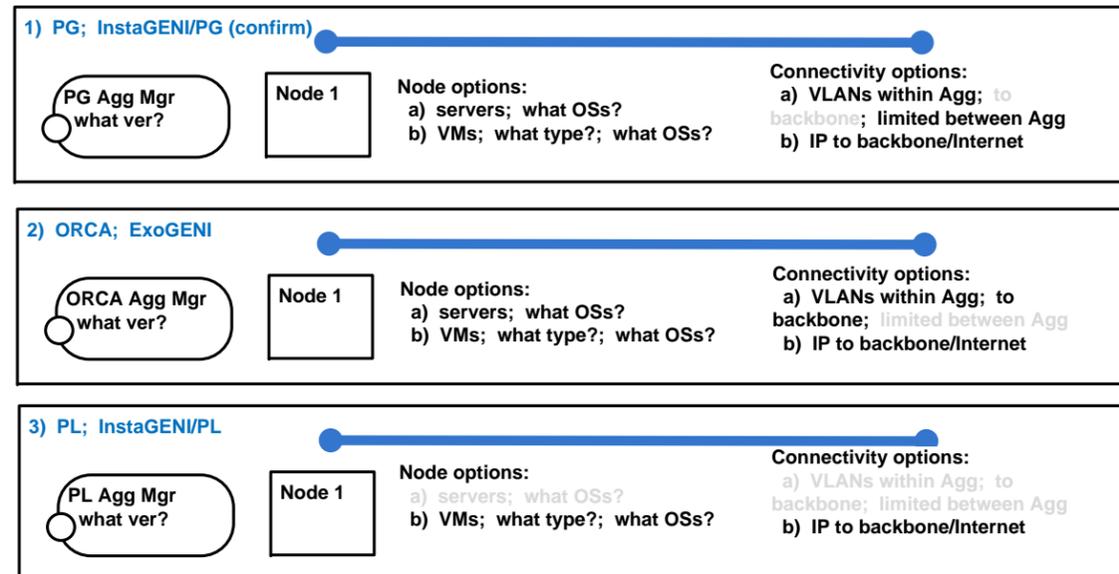
add UpdateSliver?
add Tickets?

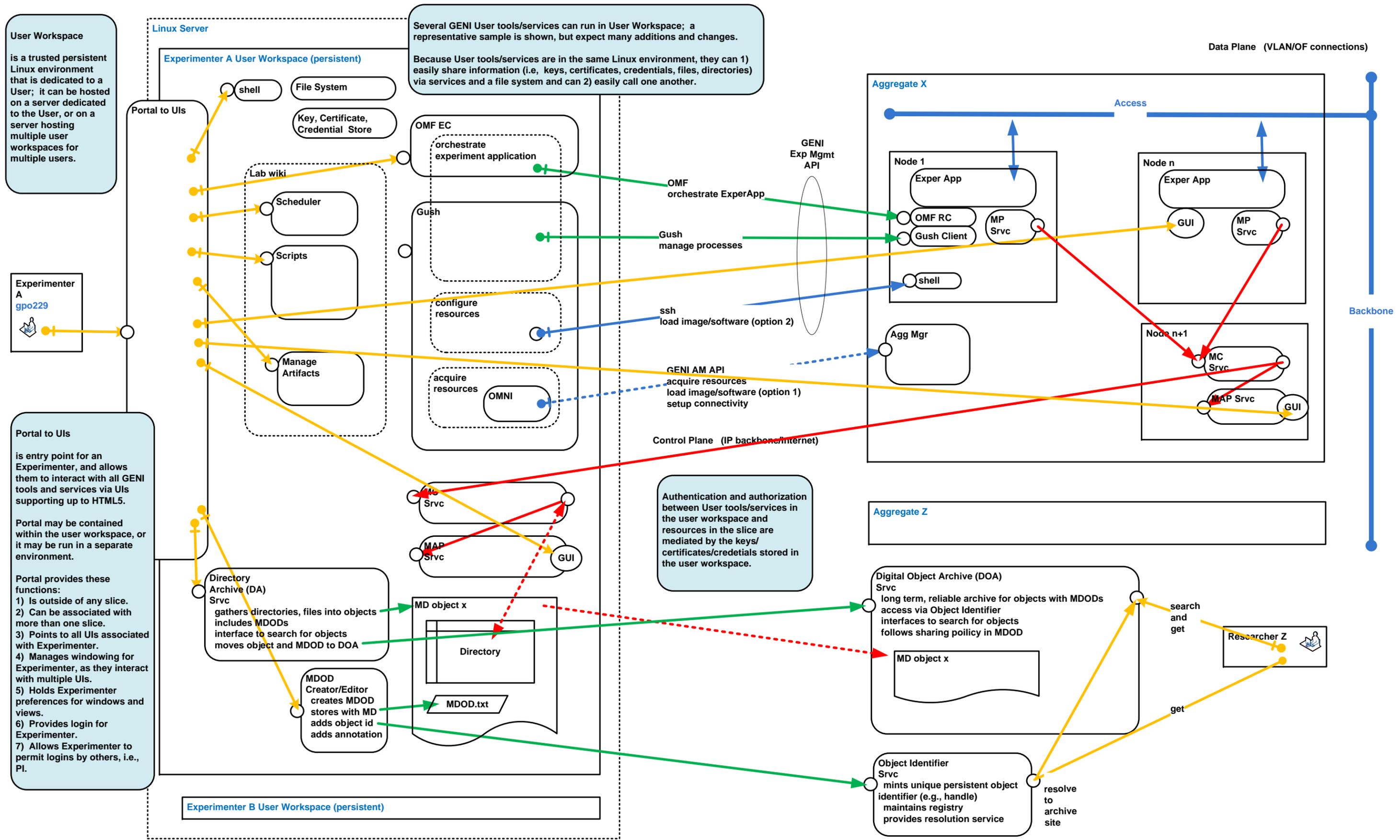
can add node or add link after start

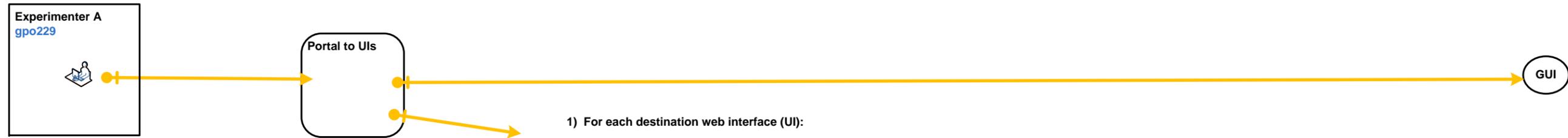
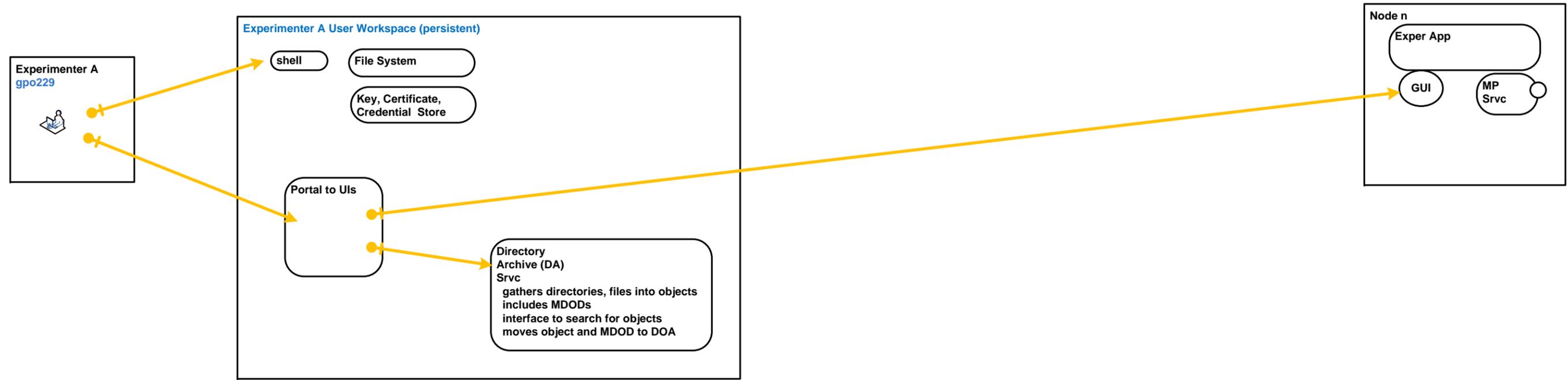
GENI rspec v4?

What extensions are being
considered to GENI rspec v3?
When will GENI rspec v4 be defined?

Spiral 4 Target Aggregates

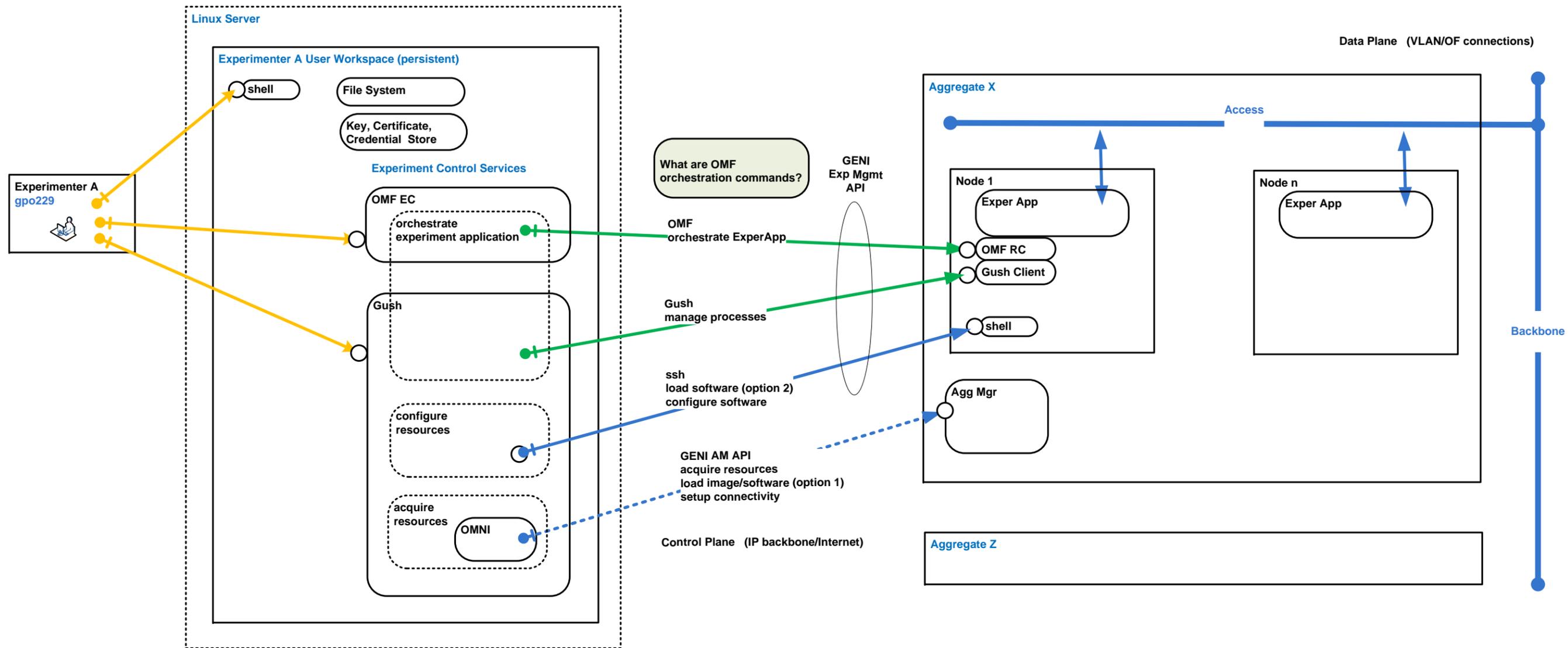






Portal to UIs
 is entry point for an Experimenter, and allows them to interact with all GENI tools and services via UIs supporting up to HTML5.
 Portal may be contained within the user workspace, or it may be run in a separate environment.
 Portal provides these functions:
 1) Is outside of any slice.
 2) Can be associated with more than one slice.
 3) Points to all UIs associated with Experimenter.
 4) Manages windowing for Experimenter, as they interact with multiple UIs.
 5) Holds Experimenter preferences for windows and views.
 6) Provides login for Experimenter.
 7) Allows Experimenter to permit logins by others, i.e., PI.

- 1) For each destination web interface (UI):
- a) Load reachable url
 - b) Use http(s)
 - c) Include client authentication that is recognized by web server interface
- note: Experimenter can install matching keys/certificates in portal and web interface when slice is



Spiral 4 Target AM APIs: must use GENI AM API, for universality

GENI AM API v1
GENI rspec v3 (PG rspec v2)
Now

- GetVersion
- ListResources
- CreateSliver
- DeleteSliver
- SliverStatus
- RenewSliver
- Shutdown

request all resources at start
no UpdateSliver (therefore, can't add node or add link)

GENI AM API v2
GENI rspec v3
Avail GEC13

flexible arguments and returns (property lists)

request all resources at start
no UpdateSliver (therefore, can't add node or add link)

Spiral 5 Target AM APIs

GENI AM API v3
Avail GEC15? GEC16?

add UpdateSliver?
add Tickets?

can add node or add link after start

GENI rspec v4?

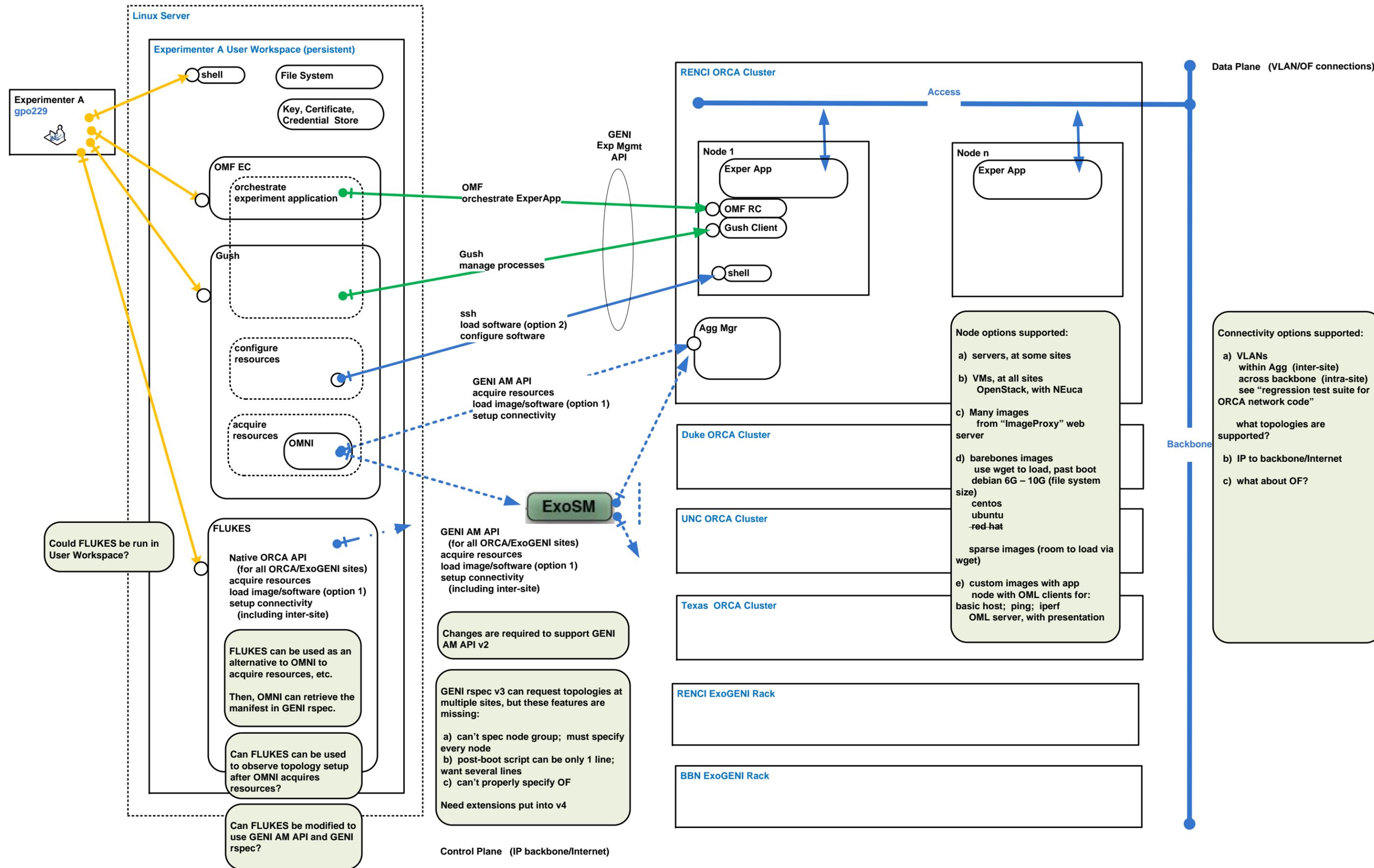
What extensions are being considered to GENI rspec v3?
When will GENI rspec v4 be defined?

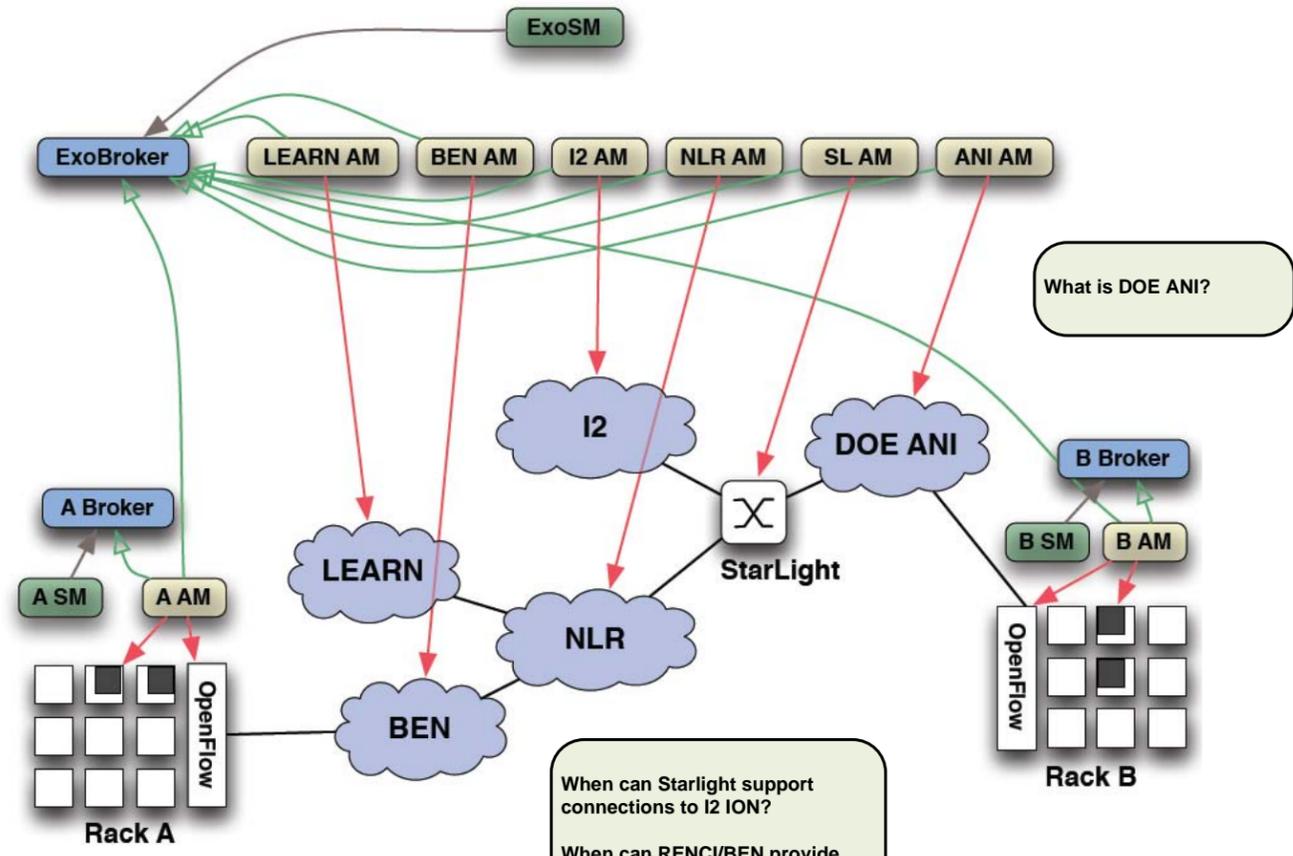
What can be specified using GENI rspec v3?
What extensions are being considered?
When will GENI rspec v4 be defined?

Node options:
a) servers; what OSs?
b) VMs; what type?; what OSs?

Connectivity options:
a) VLANs within Agg; to backbone; across backbone
b) IP to backbone/Internet

What is supported by various target Agg's?
What is common set ?





What is DOE ANI?

Need to verify that VLAN connections will be supported between the following sites:

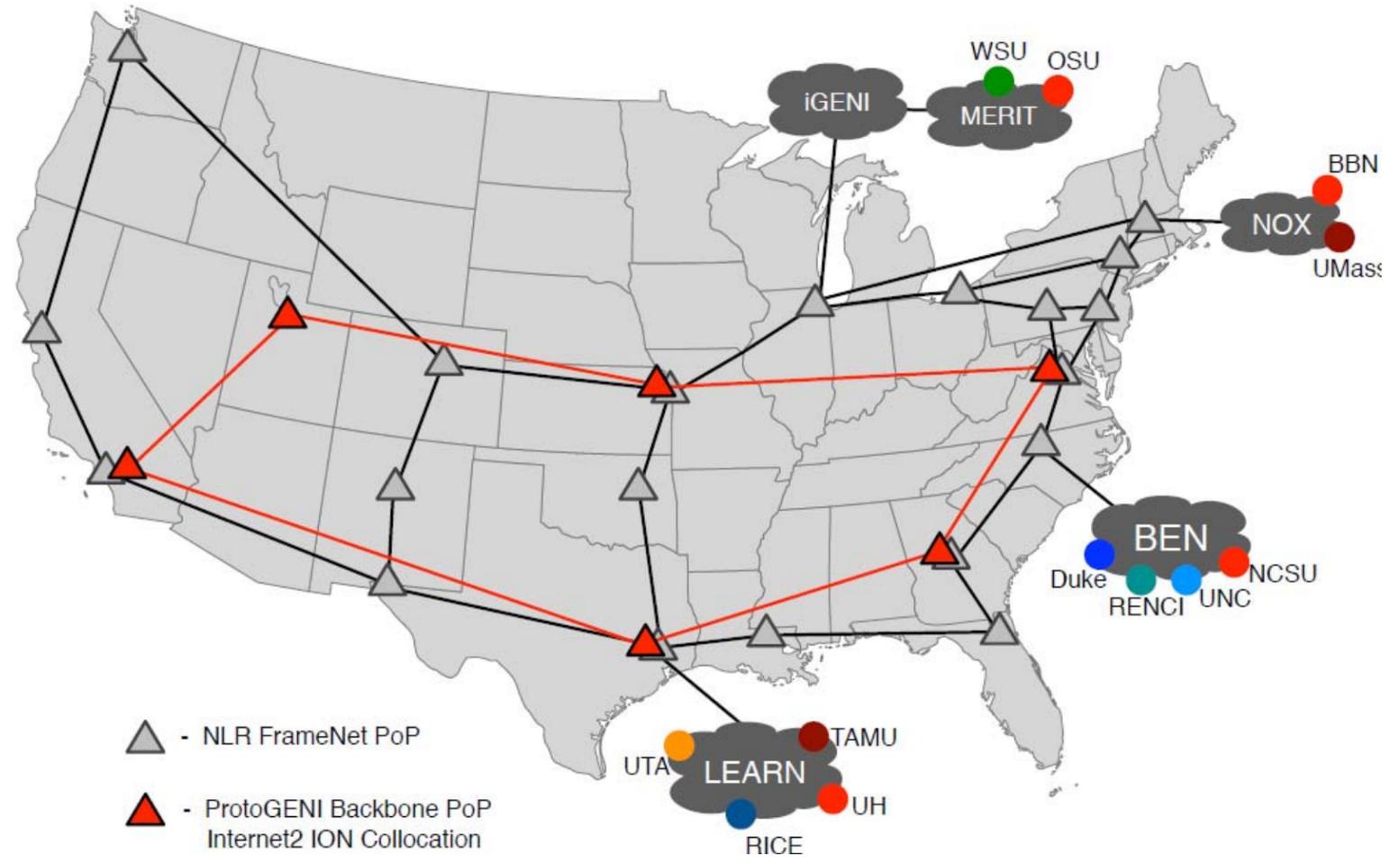
- via BEN
 - RENCI ORCA cluster
 - UNC ORCA cluster
 - Duke ORCA cluster
 - RENCI ExoGENI rack
- via LEARN
 - UH ORCA cluster
- via NOX
 - BBN ExoGENI rack

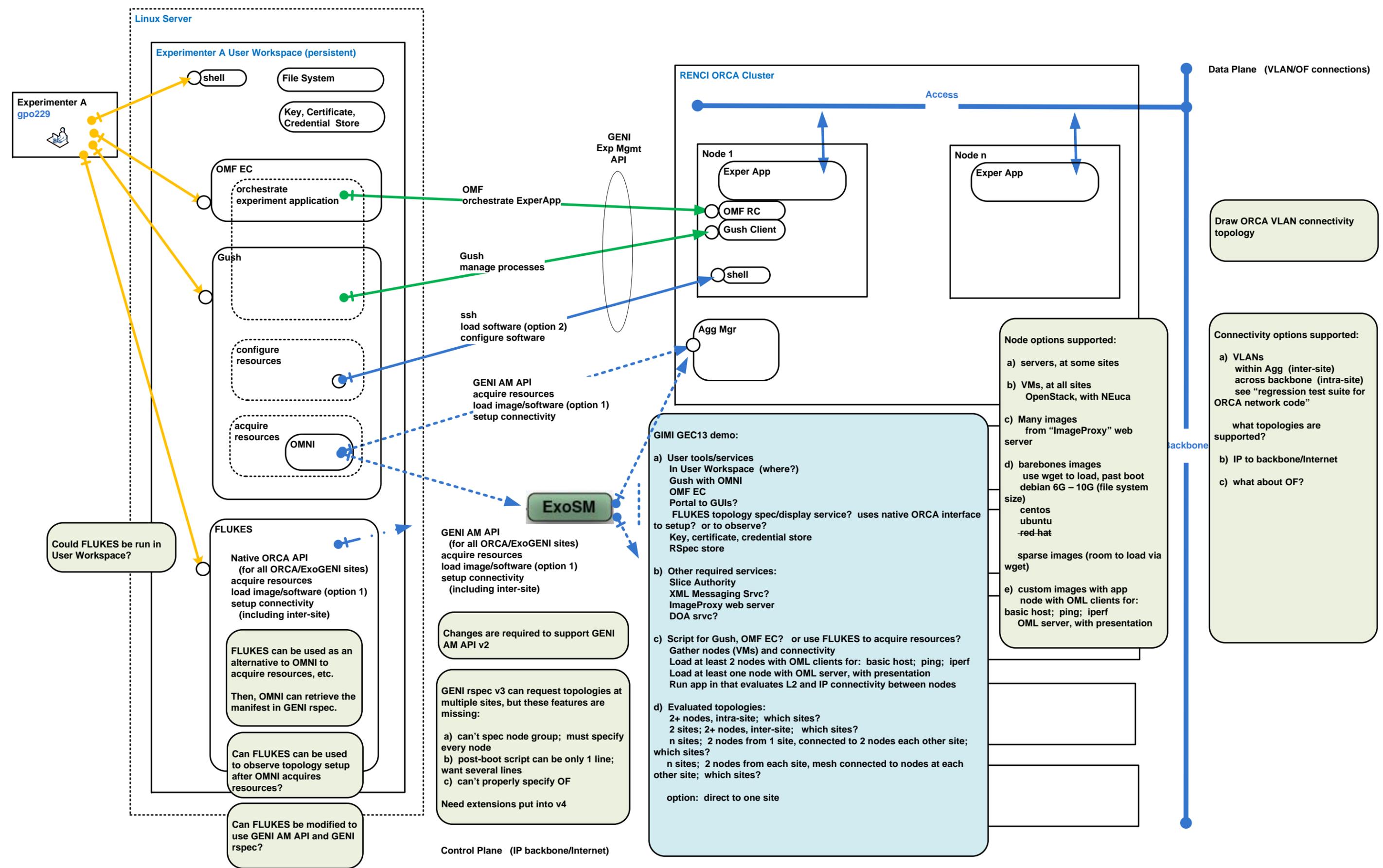
what are limits?
how would these be specified in rspec?

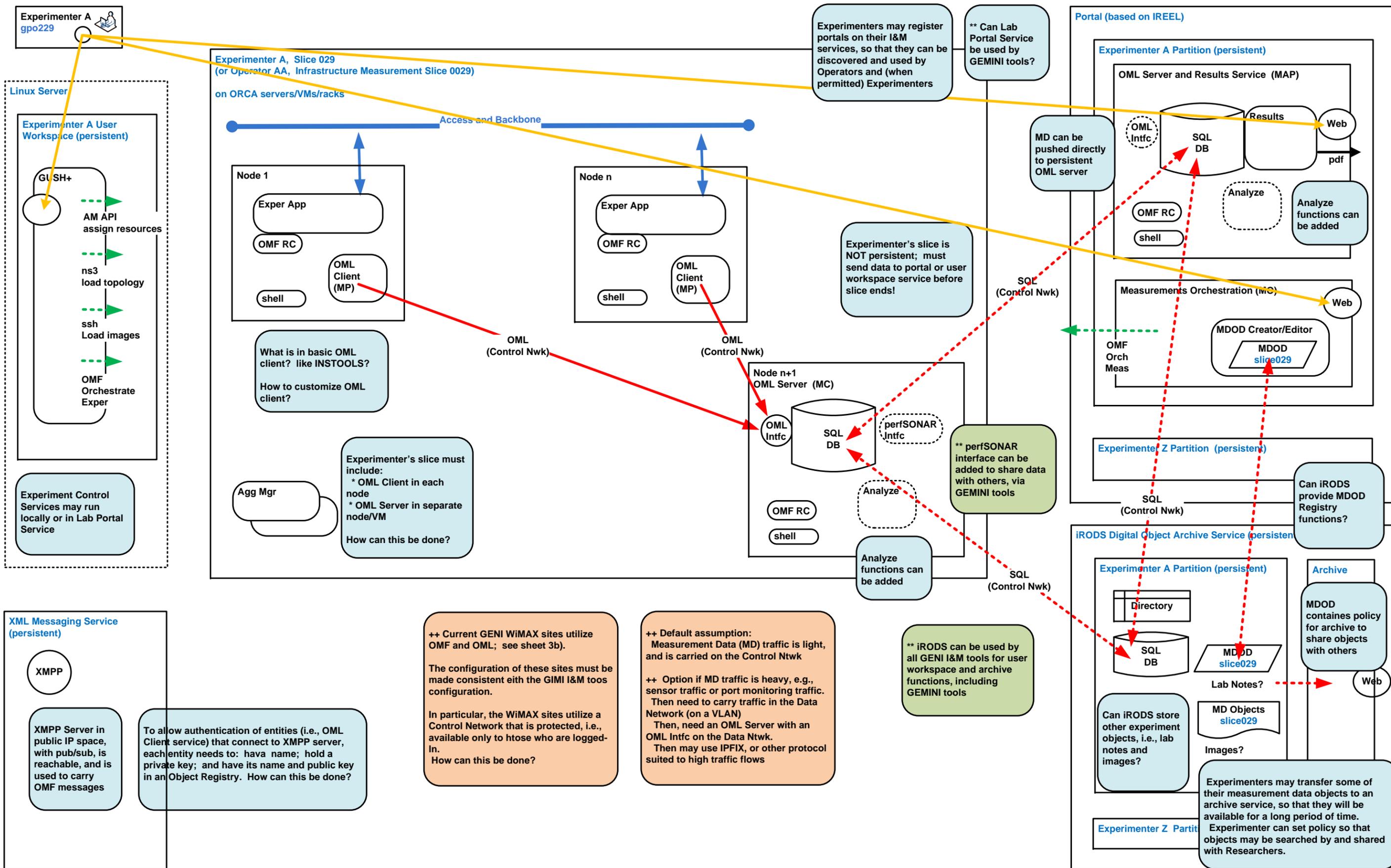
When can Starlight support connections to I2 ION?
When can RENC/BEN provide connections to I2 ION?

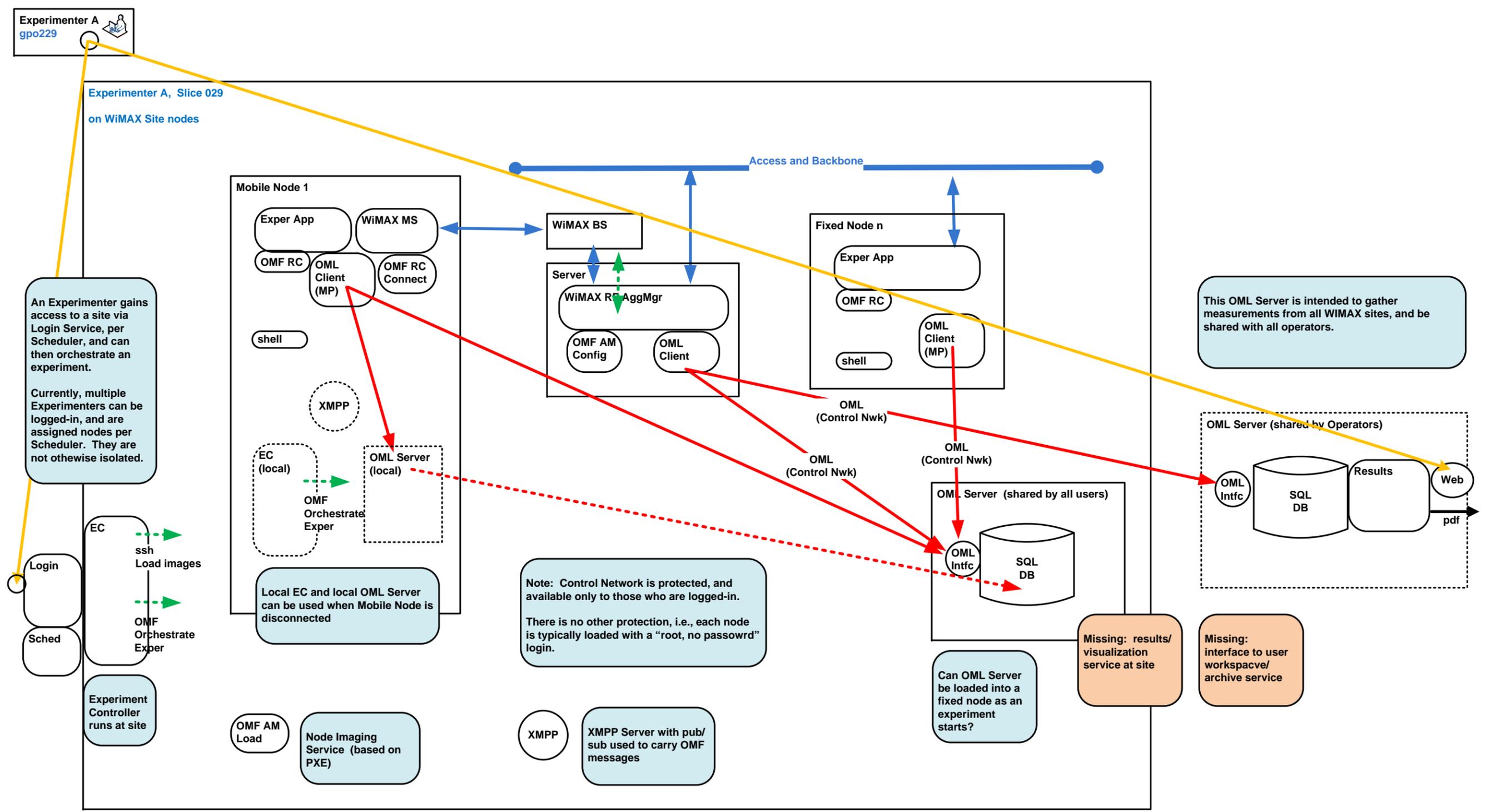
Could I2 ION extend connections to a protoGENI backbone node? how?

Could I2 ION extend connections to WiMAX sites? how?



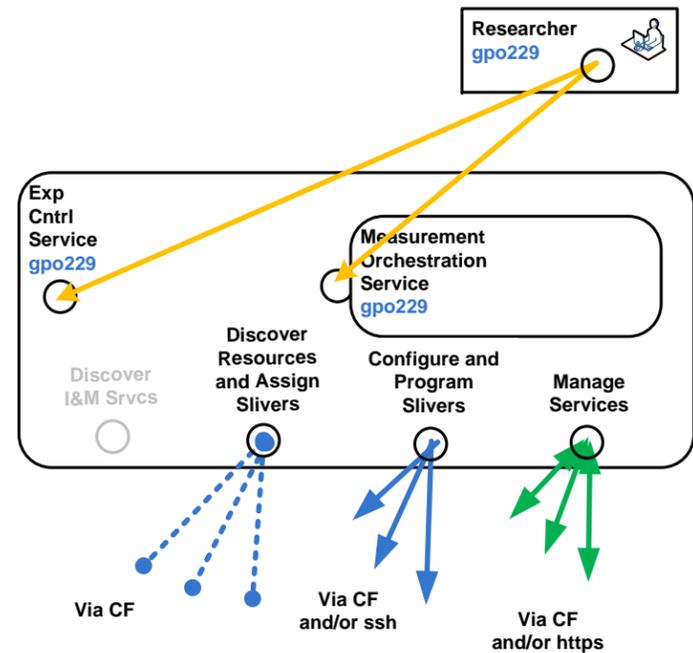






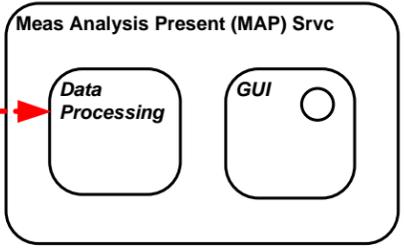
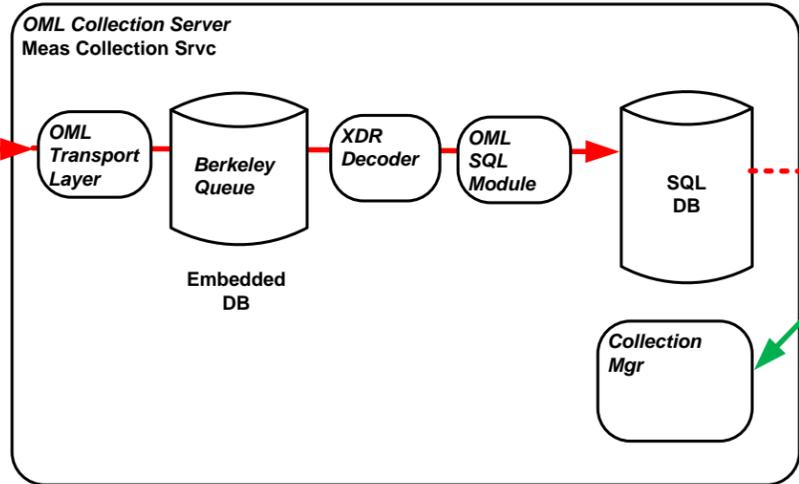
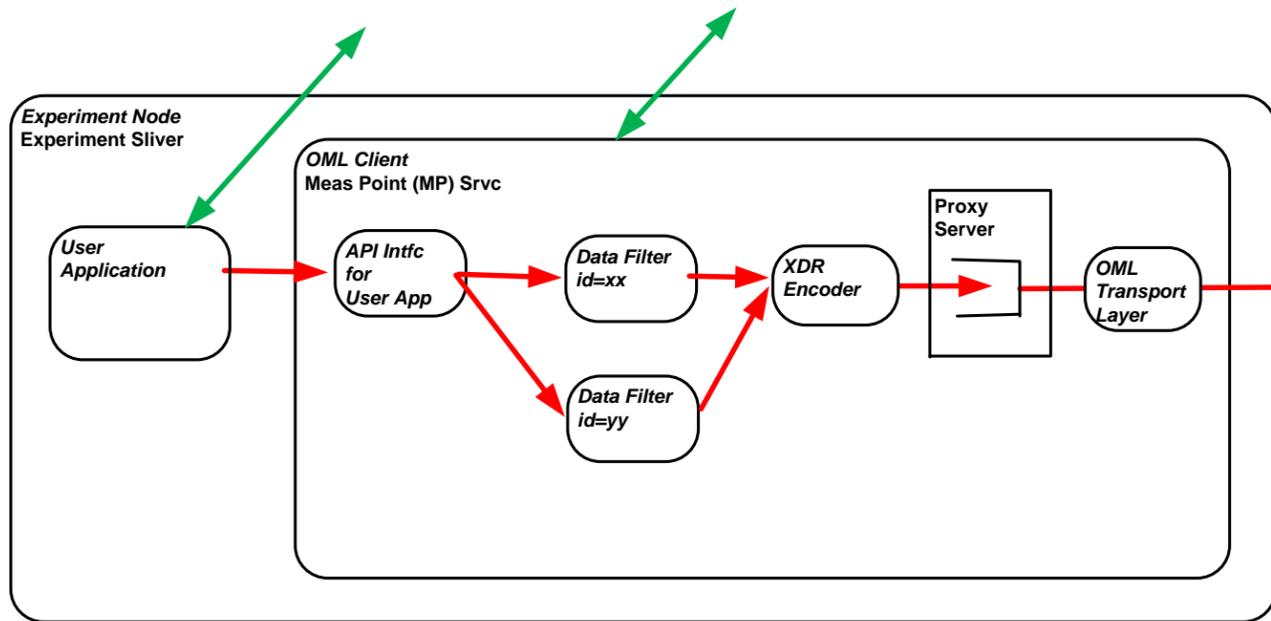
OMF/OML Structure

- * One OML Client included in each Experiment Node
- * One OML Collection Server provided for each OMF site
- * Researcher programs User Application and OML Client to gather and filter desired measurement data, and define stream to Collection Server
- * Meas Data streamed from OML Client to OML Collection Server



Manage (and Control) Svcs

- * Via HTTP to all svcs's, with APIs based on REST.
- * Via HTTP to OML Client svrc, to config files specifying filtering and streaming, which are then compiled into code



Meas Analysis Present Srvc

- * Running outside of OMF/OML.
- * Can import directly from SQL DB
- * EC can arrange to convert tables into graphs

Experiment Portal

- * Early prototype
- * Each experiment results in a separate page containing all the experiment related information (script, parameter, resources used, time) as well as a pointer to the measurement database.
- * Where?

Client API

- * Researcher uses web interface to define measurement points and parameters through a web interface, saved as XML config file; causes XSLT-based code generator to generate source code for measurement client.
- * Researcher includes measurement-point id's, and metric id's, which dynamically creates a schema.

Meas Data Filters

- * Researcher defines filters, separate from User App, so that can easily change data collection behavior
- * Can be triggered by various properties, e.g., time, by the no of data values collected, etc.
- * Can be triggered by events, e.g., passing a trip line in a mobile application

Meas Data Streams

- * Researcher defines measurement streams, gathering data samples and averaging, etc.
- * Meas data is series of typed vectors, XDR coded, and then streamed from client to collection server using proprietary OML protocol, on top of TCP, over dedicated Control VLAN
- * Considering using IPFIX instead of prop OML protocol; IPFIX typically uses SCTP for transport
- * If path becomes disconnected from time-to-time. data is cached in Proxy Server FIFO, and then forwarded when path is reestablished

Meas Data Schema

- * Meas data follows schema defined by researcher, including: measurement-point id's, metric id's, etc.
- * A sensor (or application, or service) define a set of measurement points, with each measurement point defined by a name and a typed vector (sensor schema).
- * At runtime, the experimenter (or operator) provides a streams spec which defines what measurement points are going to be activated and what initial processing is going to be performed - that defines the actual schema going over the wire and/or ending up in the collection database

DB Schema

- * Application definition is used to create DB schema for experiment, using XSLT.
- * DB table is created for each measurement point, names based on id attribute of the group element.
- * Includes mandatory fields for name/id, timestamp, sequence number
- * Protocol is self describing
- * Server automatically creates a table for every distinct stream (distinct in terms of schema not source).
- * Streams carry their own name which is translated into a database using a simple naming convention.

