G E N I

Global Environment for Network Innovations

GENI Control Framework Requirements

Document ID: GENI-SE-CF-RQ-01.3
January 9, 2009

Prepared by:
The GENI Project Office
BBN Technologies
10 Moulton Street
Cambridge, MA 02138 USA

Issued under NSF Cooperative Agreement CNS-0737890

TABLE OF CONTENTS

1 Document Scope

This section describes this document's purpose, its context within the overall GENI document tree, the set of related documents, and this document's revision history.

1.1 Purpose of this Document

This document defines the GENI control framework subsystem, and then specifies its requirements. It is a DRAFT, to be used for discussion in the GENI Facility Control Framework working group. Once approved, it can be used as a guide to judge the completeness of prototype control framework designs, and as a guide to their continued evolution.

1.2 Context for this Document

Figure 1-1. below shows the context for this document within GENI's overall document tree.

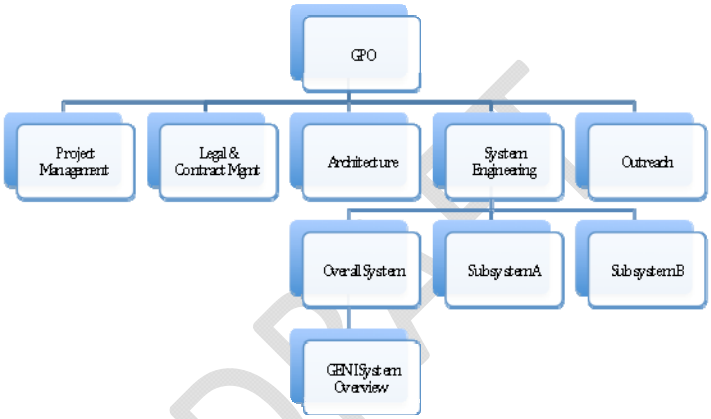

Figure 1-1. This Document within the GENI Document Tree.

1.3 Related Documents

The following documents of exact date listed are related to this document, and provide background information, requirements, etc., that are important for this document.

Some of the material in this document is drawn from the GENI System Requirements document.

Some of the material in this document is drawn from the GENI System Overview document.

Some of the material in this document is drawn from the Slice-based Facility Architecture document.

1.3.1 National Science Foundation (NSF) Documents

Document ID	Document Title and Issue Date
N / A	

1.3.2 GENI Documents

Document ID	Document Title and Issue Date
GENI-SE-SY-RQ-01.4	GENI System Requirements, September 18, 2008 http://www.geni.net/docs/GENI-SE-SY-RQ-01.7.pdf
GENI-SE-SY-SO-01.5	GENI System Overview, September 19, 2008, http://www.geni.net/docs/GENISysOvrvw092908.pdf
TBD	GENI Experiment Lifecycle TBD

1.3.3 Standards Documents

Document ID	Document Title and Issue Date
N / A	

1.3.4 Other Documents

Document ID	Document Title and Issue Date
GDD 06-10	"Towards Operational Security for GENI," by Jim Basney, Roy Campbell, Himanshu Khurana, Von Welch, GENI Design Document 06-10, July 2006. http://www.geni.net/GDD/GDD-06-10.pdf
GDD 06-23	"GENI Facility Security," by Thomas Anderson and Michael Reiter, GENI Design Document 06-23, Distributed Services Working Group, September 2006. http://www.geni.net/GDD/GDD-06-23.pdf
N/A	"GMC Specifications," edited by Ted Faber, Facility Architecture Working Group, September 2006. http://www.geni.net/wsdl.php
GDD 06-24	"GENI Distributed Services," by Thomas Anderson and Amin Vahdat, GENI Design Document 06-24, Distributed Services Working Group, November 2006. http://www.geni.net/GDD/GDD-06-24.pdf
GDD 06-38	"GENI Engineering Guidelines," edited by Ted Faber, GENI Design Document 06-38, Facility Architecture Working Group, December 2006. http://www.geni.net/GDD/GDD-06-38.pdf
GDD 06-42	"Using the Component and Aggragate Abstractions in the GENI Architecture," by John Wroclawski, GENI Design Document 06-42, Facility Architecture Working

	Group, December 2006. http://www.geni.net/GDD/GDD-06-42.pdf
N/A	Slice Based Facility Architecture, v1.10, August 8, 2008, by Larry Peterson, et.al. http://groups.geni.net/geni/attachment/wiki/GeniControlBr/v1.10%20%20080808%20%20sfa.pdf

1.4 Document Revision History

The following table provides the revision history for this document, summarizing the date at which it was revised, who revised it, and a brief summary of the changes. This list is maintained in reverse chronological order so the newest revision comes first in the list.

Revision	Date	Revised By	Summary of Changes
01.1	11/21/08	H. Mussman	Completed draft, for limited review, based on material adapted from earlier architecture document.
01.2	12/22/08	H. Mussman	Updated after review by GPO systems engineers.
01.3	1/9/09	H. Mussman	Updated after 2nd review by GPO systems engineers.
01.4			

2 GENI Core Concepts

The Global Environment for Network Innovations (GENI) is a suite of experimental network research infrastructure now being planned and prototyped. GENI prototyping is sponsored by the National Science Foundation to support experimental research in network science and engineering.

As envisioned in these community plans, this suite will support a wide range of network science and engineering experiments such as new protocols and data dissemination techniques running over a substantial fiber-optic infrastructure with next-generation optical switches, novel high-speed routers, city-wide experimental urban radio networks, high-end computational clusters, and sensor grids. All infrastructure are envisioned to be shared among a large number of individual, simultaneous experiments with extensive instrumentation that makes it easy to collect, analyze, and share real measurements.

Core concepts for a GENI infrastructure suite have been established:

- **Programmability** – researchers may download software into GENI-compatible nodes to control how those nodes behave;
- **Virtualization and Other Forms of Resource Sharing** – whenever feasible, nodes implement virtual machines, which allow multiple researchers to simultaneously share the infrastructure; and each experiment runs within its own, isolated slice created end-to-end across the experiment's GENI resources;
- **Federation** – different parts of the GENI suite are owned and/or operated by different organizations, and the NSF portion of the GENI suite forms only a part of the overall "ecosystem"; and
- **Slice-based Experimentation** – GENI experiments will be an interconnected set of reserved resources on platforms in diverse locations. Researchers will remotely discover, reserve, configure, program, debug, operate, manage, and teardown distributed systems established across parts of the GENI suite.

3 GENI System Overview

3.1 Major Entities and their Relationships

Figure 2-1 presents a block diagram of the GENI system covering the major entities within the overall system. Optional (but desirable) parts are shown "grayed-out." See the GENI System Overview document at http://www.geni.net/docs/GENISysOvrvw092908.pdf for more details.

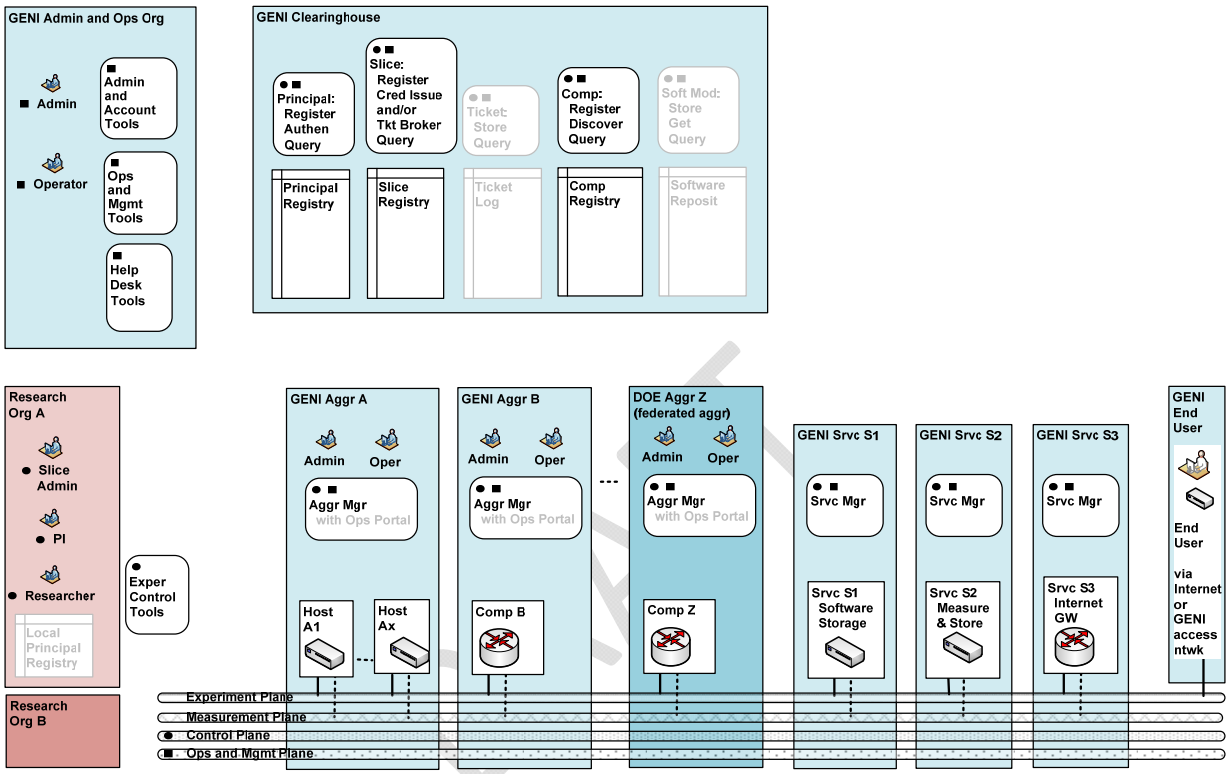

Figure 2-1. GENI System Diagram.

3.2 Federated Suites

Figure 2-2 provides a system diagram illustrating federation between one GENI suite and another. As a hypothetical example, it depicts federation between a US-based GENI suite and a compatible suite in the European Union (EU).

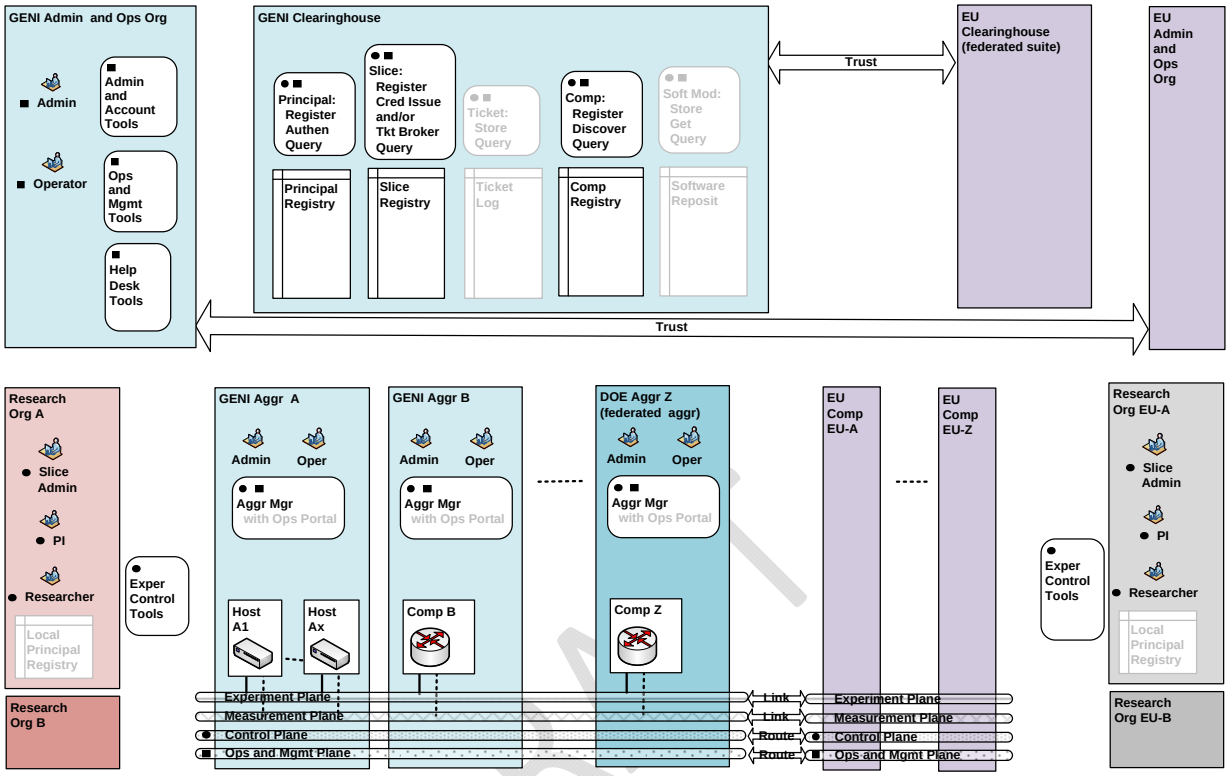

Figure 2-2. System Diagram with Federated Infrastructure Suites.

3.3 Slices

Figure 2-3 shows two researchers from different organizations managing their two experiments in two corresponding slices. Each slice spans an interconnected set of slivers on multiple aggregates and/or components in diverse locations. Each researcher remotely discovers, reserves, configures, programs, debugs, operates, manages, and teardowns the "slivers" that are required for their experiment. Note that the clearinghouse keeps track of these slices for troubleshooting or emergency shutdown.

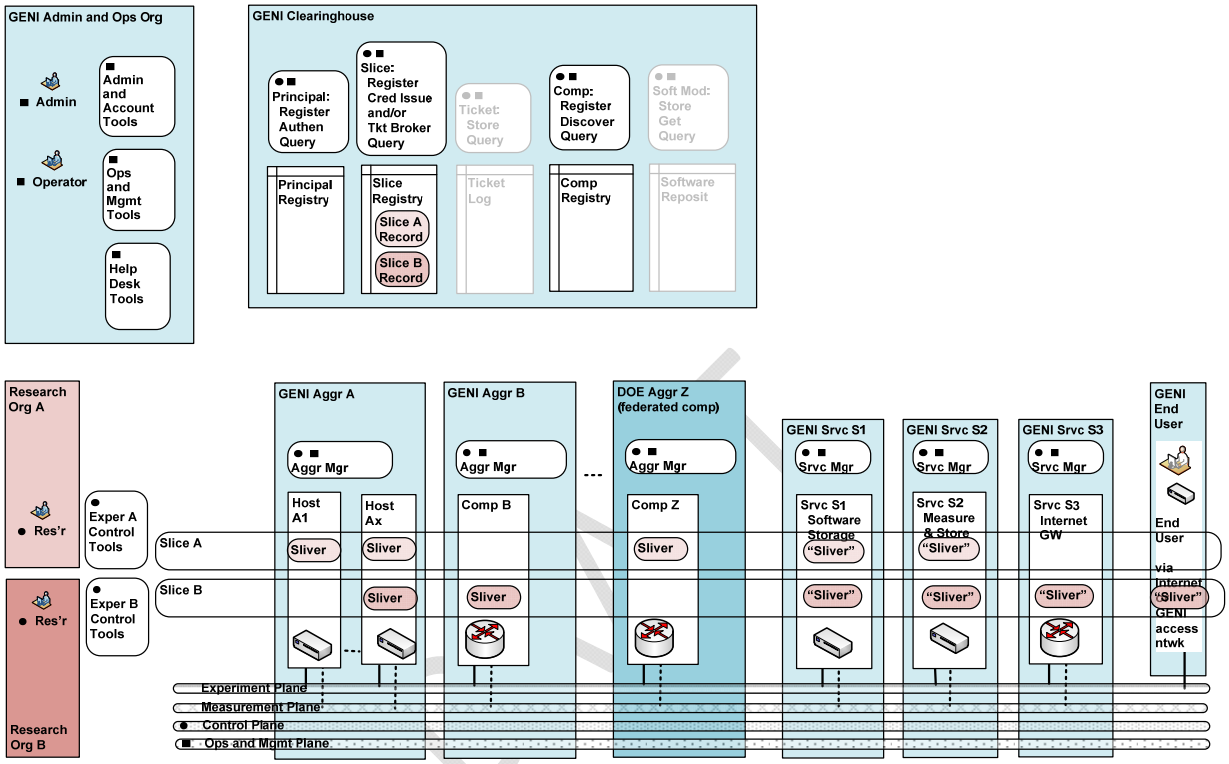

Figure 2-3. Two GENI Slices.

An aggregate manager a) interacting with the researcher (or her proxies) via the control plane and b) configuring the devices over internal interfaces establishes Slivers. Components may be virtualized, and can thus provide resources for multiple experiments at the same time, but keep the experiments isolated from one another. In addition, each slice requires its own set of experiment support services. Furthermore, as shown in Slice B, "opt-in" users may join the experiment running in a slice, and thus be associated with that slice.

4 GENI Control Framework Definition

The GENI control framework includes the entities shown in Figure 4-1, and the Control Plane for transporting messages between these entities.

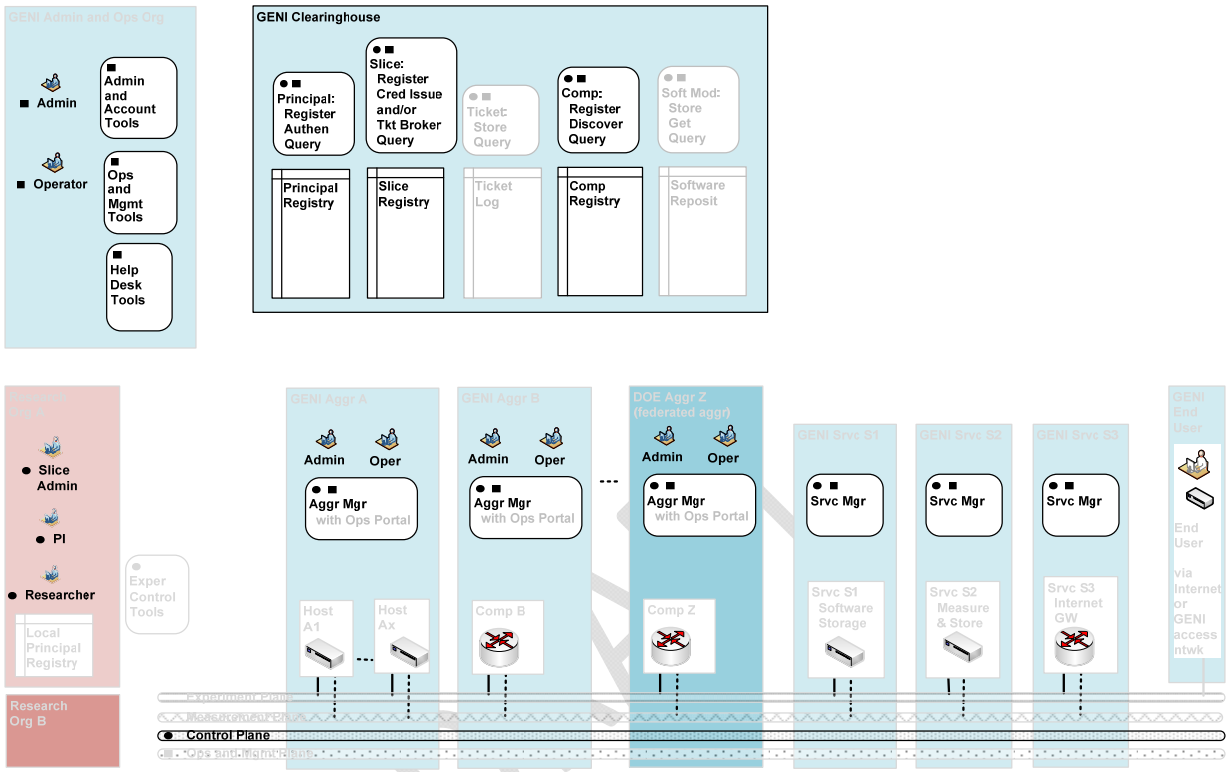

Figure 4-1. GENI Control Framework Entities.

The GENI control framework includes the following clearinghouse entities in a GENI suite:

- Principal registry and related services.
- Component registry and related services.
- Slice registry and related services.
- An optional ticket log and related services, for holding “sliver records”, used in administering and managing the GENI suite.
- An optional software repository, for holding software objects that are required to administer, operate or manage the GENI suite.

It includes the following entities associated with each aggregate or component that is providing experiment resources in a GENI suite:

- An aggregate manager and related services.

- An optional component manager and related services, for components that are part of an aggregate.
- An optional broker service and related services, that typically functions as an aggregate-of-aggregates manager in the GENI suite.

It includes the following entities associated with a principal who is utilizing, administering or managing experiment resources in a GENI suite.

- A principal acting from a server utilizing a browser client and/or a set of helper tools
- A principal service acting on behalf of a principal , utilizing a browser client and/or a set of helper tools, that appears as a principal in the GENI suite.

The GENI control framework defines:

- Interfaces between all entities.
- Message types including basic protocols and required functions.
- Message flows necessary to realize key experiment scenarios.

5 GENI Control Framework Requirements

5.1 Origin

These GENI control framework requirements originate from the following:

- GENI core concepts; see http://www.geni.net/docs/GENI-SE-SY-RQ-01.7.pdf and Section 2.
- GENI system overview; see http://www.geni.net/docs/GENISysOvrvw092908.pdf and Section 3.
- GENI system requirements; see [http://www.geni.net/docs/GENI-SE-SY-RQ-01.7.pdf].

5.2 Principals

5.2.1 Definitions

a) A GENI principal is a person acting from a server utilizing a browser client and/or a set of helper tools, who is utilizing, administering or managing experiment resources in a GENI suite.

b) A GENI service acting on behalf of a principal, utilizing a browser client and/or a set of helper tools, may function as a principal in a GENI suite.

5.2.2 Identification

a) Each principal shall have a globally-unique name and/or a globally-unique numerical identifier.

b) It shall be possible to identify a principal who is acting within the GENI suite.

Issue: c) Should there be a principal who is anonymous, perhaps with strictly limited privileges?

5.2.3 Registration

a) Each principal shall be registered within the GENI suite, which then holds a "principal registration record", or a "principal record".

b) A principal may be indirectly registered, i.e., the GENI suite may recognize their registration within their home organization, and check with its registration service as needed.

c) A principal shall be registered jointly by a principal administrator who acts for the GENI suite and by one who has been authorized to act for a research organization, or their delegates, who are then jointly responsible for the registration record of that principal.

d) The principal record shall include their identity and their contact information.

e) The principal record shall include the status and quality of verifying their identity.

f) The principal record shall include their status to operate within the GENI suite.

g) The principal record shall include information (e.g., a PublicKey) so that they can be authenticated when operating within the GENI suite.

Issue: h) Should there be a principal who is "casually registered" in the GENI suite, perhaps with strictly limited privileges?

5.2.4 Authentication

a) It shall be possible to authenticate a principal who is acting within the GENI suite by utilizing information (e.g., a public key) stored within the registry.

Note: This involves a check with the registry that is either positive (status is active and here is the current public key) or negative (here is certificate revocation list for you to check).

5.2.5 Privileges and Roles

a) It shall be possible to assign privileges and/or roles to a principal who is acting within the GENI suite.

b) A principal shall be able to serve more than one role within the GENI suite, but they shall not require multiple registrations for multiple roles.

Note: Privileges should precisely define what principals can and cannot do within the GENI suite, and in a particular situation.

Note: Roles are the traditional, broad-brush way to categorize how a principal can act within the GENI suite. The following are typical roles that are expected in the GENI suite:

- Administrators, who act for the GENI suite, and are responsible for administrative functions within the GENI suite, including authorizing other administrators.
- Operators, who act for the GENI suite, and are responsible for operations and management functions within the GENI suite.
- Principal administrators, who act for the GENI suite or a research organization, and are responsible for principal records and the authentication of principals.
- Aggregate (or component) administrators, who act for the GENI suite or an owning organization, and are responsible for aggregate (or component) records.
- Operators, who act for the GENI suite or an owning organization, and are responsible for operations and management functions within an aggregate (or component).
- Slice administrators, who act for the GENI suite or a research organization, and are responsible for slice records.
- PIs, who act for a research organization, and are responsible for slice records, the researchers assigned to a slice, and for managing slices, including all of their slivers.
- Researchers, who utilize the GENI suite for running experiments, deploying experimental services, measuring aspects of the platform, and so on.

c) Where possible, a precise set of privileges shall be assigned to a principal , instead of a broad-brush role.

Issue: d) Opt-in users are not considered principals. Should they be defined? How?

5.3 Aggregates and Components

5.3.1 Definitions

a) Aggregates, and the components which comprise them, are the primary building blocks of the GENI suite.

b) An aggregate may include zero, one or many components.

c) An aggregate may optionally reveal an "internal structure" of one or more components.

Note: This definition is consistent with the traditional GENI definition at http://www.geni.net/GDD/GDD-06-42.pdf , except that it proceeds "from the outside to the inside" in terminology, and avoids the awkward "aggregate/component" term. See Figure 5.1.

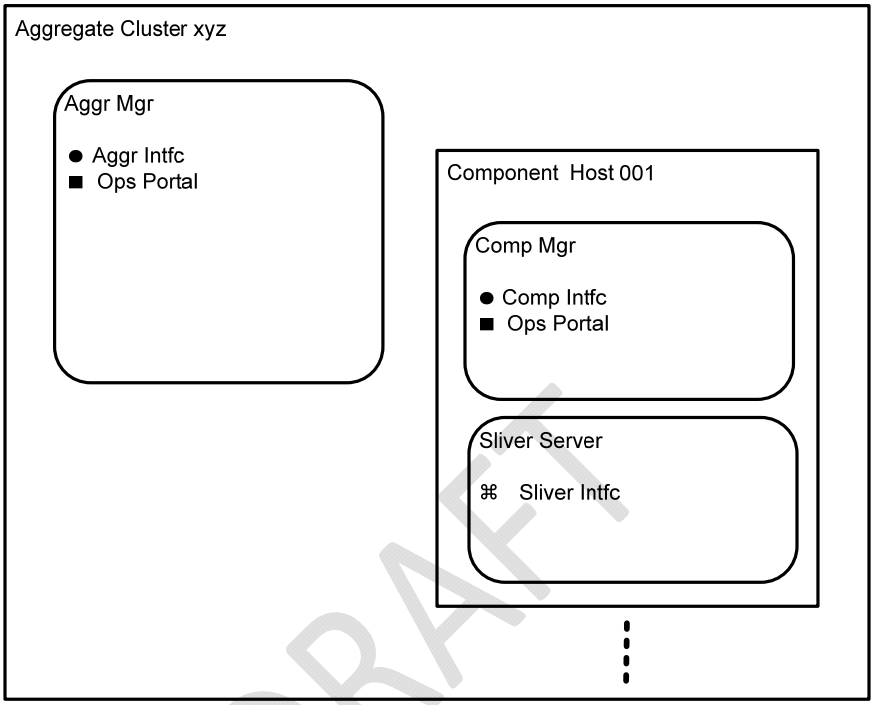

Figure 5.1 – An Aggregate and Its Internal Structure

Each aggregate is controlled via an aggregate manager, which exports a well-defined, remotely accessible interface to the GENI suite.

An aggregate encapsulates a collection of *resources*, including physical resources (e.g., CPU, memory, disk, bandwidth), logical resources (e.g., file descriptors, port numbers), and synthetic resources (e.g., packet forwarding fast paths). These resources can be contained in a single physical device or distributed across a set of devices, depending on the nature of the aggregate.

An aggregate might correspond to a backbone network, a customizable router, an edge computer, or a cluster of hosts.

Components within an aggregate may include their own component managers, which can also export well-defined, remotely accessible interfaces. For example, a cluster of hosts (an aggregate) may reveal that it has 100 hosts (components), and that it has assigned resources on Host 29 (a component) for an experiment. Then, for example, it is possible to program Host 29 to meet the needs of this experiment.

Issue: d) What if an aggregate contains aggregates, i.e., it is an aggregate of aggregates? How is this presented and controlled?

5.3.2 Identification

a) Each aggregate shall have a globally-unique name and/or a globally-unique numerical identifier.

b) Each component that is revealed by an aggregate shall have a globally-unique name and/or a globally-unique numerical identifier.

c) It shall be possible to identify an aggregate, or component revealed by an aggregate, within the GENI suite.

5.3.3 Registration

a) Each aggregate shall be registered within the GENI suite, which then holds an "aggregate registration record", or an "aggregate record".

b) An aggregate may be indirectly registered, i.e., the GENI suite may recognize their registration within their home organization, and check with its registration service as needed.

c) An aggregate may be registered within the GENI suite even if it is associated with a completely different "home suite".

d) An aggregate shall be registered jointly by an administrator who acts for the GENI suite and by one who acts for the owning organization, or their delegates, who are then jointly responsible for the aggregate record.

e) The aggregate record shall include its identity and its owner.

f) The aggregate record shall include the associated administrators, who are authorized to act for the GENI suite and for the owning organization, and who are responsible for the aggregate record.

g) The aggregate record shall indicate the associated operators, who authorized to act for the GENI suite and for the owning organization, and who are responsible for operations and management functions within the aggregate.

h) The aggregate record shall include pointers to the aggregate manager for use in discovering and requesting resources, etc.

i) The aggregate record shall include pointers to the aggregate manager for use in operating and managing the aggregate.

5.3.4 Resource Allocation

Note: By registering an aggregate in the GENI suite, the administrator/owner of the aggregate indicates that they are willing to allocate resources to experiments in the GENI suite.

a) The registration record of an aggregate shall indicate the nature and extent of the resources that are being offered.

b) When queried, the aggregate manager shall indicate the nature and extent of the resources that are available to the principal making the query.

5.4 Slices

5.4.1 Definitions

a) A slice is an interconnected set of reserved resources, or slivers, on heterogeneous substrate aggregates (components). Researchers can remotely discover, reserve, configure, program, debug, operate, manage, and teardown resources within a slice to complete an experiment. See Figure 2-3.

b) Slices are expected to have a long lifetime, and be utilized for multiple experiments that come and go, all within the same slice.

c) A slice is also the primary abstraction for accounting and accountability—resources are acquired and consumed by slices, and external program behavior is traceable to a slice, respectively.

Issue: d) Shall there be a sub-slice entity, to allow for delineation of experiments within a slice?

5.4.2 Identification

a) Each slice shall have a globally-unique name and/or a globally-unique numerical identifier.

b) It shall be possible to identify a slice within the GENI suite.

5.4.3 Registration

a) Each slice shall be registered within the GENI suite, which then holds a "slice registration record", or a "slice record".

b) A slice may be indirectly registered, i.e., the GENI suite may recognize its registration within its home organization, and check with its registration service as needed.

c) A slice shall be registered jointly by an administrator who acts for the GENI suite and by one who acts for the research organization, who are then jointly responsible for the slice record.

Note: The registration of a slice (and its active status) indicates that the owner of this slice has a trust and/or contractual relationship with the GENI suite, and through it, with all (or some) of its aggregates, so that researchers can be granted resources by aggregates within the GENI suite.

d) A slice record shall indicate the identity of the slice, and the owner of the slice (e.g., a research organization) who takes responsibility for this slice, and for all experiments done by this slice.

e) A slice record shall indicate the status of this slice, i.e., that it is active and can be utilized to gather resources and complete experiments.

f) A slice record shall indicate the associated slice administrators, who are authorized to act for the GENI suite and for the research organization, and who are responsible for the slice record.

g) A slice record shall indicate the associated PI(s), who are authorized to act for the owner of the slice, and who are responsible for the researchers assigned to the slice, and for operating and managing all of the slivers associated with this slice.

h) A slice record shall indicate the associated researchers, who are authorized to utilize this slice to request resources from the GENI suite to run experiments.

i) The slice record may point to an associated "slice account", when necessary to provide extended accounting features. For example, a "slice account" may contain "GENI bucks" that are used to "purchase" resources.

5.5 Experiment Setup

The GENI control framework provides the functions required for a GENI researcher to setup an experiment, as detailed in the following sections. See also the GENI Experiment Lifecycle document at TBD.

5.5.1 Resource and Topology Discovery

a) The control framework shall allow a researcher, using the component registry, to discover all of the resources available to them from the aggregates associated with the GENI suite.

b) The control framework shall allow a researcher, using the component registry, to discover the interconnection topology of the resources available to them from the aggregates associated with the GENI suite.

5.5.2 Resource Sharing

Note: A core concept of a GENI suite is to provide: **Virtualization and other forms of resource sharing** – Whenever feasible, substrate components are virtualized to allow multiple researchers to simultaneously share them, and operate without disturbing another experiment, or being disturbed. Thus, each experiment runs within its own, isolated slice, created end-to-end across GENI resources. For example, this may be accomplished by dividing a host component into multiple virtual machines or by assigning separate connections across a network component.

a) The control framework shall allow multiple researchers, referencing multiple slices, to request and be assigned multiple sets of resources (slivers) on a given aggregate.

b) When this involves the assignment of a component, the control framework shall allow multiple researchers, referencing multiple slices, to request and be assigned multiple sets of resources (or slivers) on a given component.

5.5.3 Resource Authorization and Policy Implementation

a) The GENI control framework shall allow the authorization and assignment of resources from aggregates or federated aggregates to GENI researchers following established policies.

b) The control framework shall allow this to be done through the interaction of some or all of these entities, records and accounts:

- The GENI researcher.
- The GENI and/or federated clearinghouse. (one or more)
- The designated slice record, or optional slice account.
- A broker service. (zero, one or more)
- The GENI or federated aggregate.

c) The control framework shall support a rich variety of resource authorization and policy mechanisms.

d) The control framework shall support resource authorization by an aggregate based on resource availability and its local policies.

e) The control framework shall support resource authorization that includes policies associated with a clearinghouse.

f) The control framework shall support resource authorization that includes policies associated with an intermediate broker.

g) The control framework shall allow policies that can be based on a variety of parameters, including:

- Trust and contractual relationships established between actors and entities.
- Researcher lineage and status
- Slice lineage and status
- Presence of electronic currency, i.e., "GENI tokens"
- Resource availability

For example, in the simplest case, the control framework shall allow an aggregate to check the slice lineage of a request against a local list of trusted (supported) slices to decide whether to grant a resource (or not).

5.5.4 Resource Assignment

a) The GENI control framework shall allow the authorization and assignment of resources from GENI or federated aggregates to GENI researchers on a best-effort basis, without specific starting and stopping dates/times.

b) The GENI control framework shall allow the authorization and assignment of resources from GENI or federated aggregates to GENI researchers on a best-effort basis, with specific starting and stopping dates/times.

c) The GENI control framework shall allow the authorization and assignment of resources from GENI or federated aggregates to GENI researchers on an assured basis, with specific starting and stopping dates/times, where the starting date/ time can be now.

d) The GENI control framework shall allow GENI researchers and GENI (or federated) aggregates to revise their agreed upon authorization or assignment of resources at any time, changing its basis (say, from best-effort to assured) and/or its date/time.

e) The GENI control framework shall allow a GENI (or federated) aggregate to change the authorization and assignment of a resource from less specific (one host, with these characteristics) to more specific (Host 69, reachable at this IP address).

For example, a researcher may request, and an aggregate may authorize, a resource (one host, with these characteristics) on a best effort basis, and then later a specific assignment can be made (Host 69, reachable at this IP address, starting at this date and time, for a one hour period).

f) The GENI control framework shall allow GENI researchers and GENI (or federated) aggregates to revise their agreed upon authorization or assignment of resources at any time, changing from one specific to another specific (e.g., from Host 69 to Host 92)

5.5.5 Component Programming

Note: A core concept of the GENI suite is to provide: **Programmability** – Whenever feasible, a researcher can download software into a (virtual) machine or network node component to define the behavior of the resultant sliver. For example, programming a network node component could define a custom routing function.

a) The control framework shall allow a GENI or federated aggregate to assign a specific component to a researcher, and and then it shall provide a means for the researcher program that component, e.g., a means to login to that component, load code, and then boot it.

5.5.6 Disconnected Operation of Components

Note: In a GENI suite, some of the components (such as wireless servers) will require "disconnected operation", where they are controlled and polled in the short periods of time that they are connected to the suite.

b) The control framework shall allow disconnected operation for designated components.

Issue: c) What shall be done? Can this be hidden behind an aggregate manager that is never disconnected? How can the status of a communication with a disconnected component (waiting; in progress; completed) be made available to the remainder of the suite via the aggregate manager?

5.5.7 Disconnected Operation of Researchers

Note: In a GENI suite, some researchers, will connect to the GENI suite to setup an experiment, e.g., by reserving resources for use at a later time, and then will disconnect until they are ready to execute the experiment.

a) The control framework shall allow disconnected operation for researchers after an experiment has been scheduled.

Issue: b) Can a researcher be disconnected when an experiment is being executed? If so, must some principal or service be designated to be in charge? What about long-term experiments?

5.5.8 Resource to Resource Connections

a) When a researcher has been assigned resources from two (or more) GENI or federated aggregates that are to be connected together, the control framework shall provide a way for the researcher to complete the necessary connections, including the ability to: learn about the connection points; request the connections in the necessary sequence; and receive a verification that the connection has been completed.

For example, after assignments in two components have been completed, they both may "revise" their agreements with the researcher by adding the connect points. Then, the researcher may "revise" both agreements to tell each component where to connect. Finally, each component may "revise" their agreement to indicate that they are connected.

5.5.9 Setup Verification

a) When a researcher has been assigned resources on GENI (or federated) aggregates for an experiment, the control framework shall provide a way for the researcher to ask the aggregates to verify the setup before it is time for the experiment to start.

Issue: b) How can this be done? Always include a background best effort resource assignment for setup verification? How can results be returned to help in debugging?

5.6 Experiment Execution

The GENI control framework provides the functions required for a GENI researcher to execute an experiment, as detailed in the following sections. See also the GENI Experiment Lifecycle document at TBD.

5.6.1 Experiment and Sliver Control

a) When a researcher, associated with a designated slice, has been assigned resources (slivers) on GENI or federated aggregates for an experiment, the control framework shall provide a way for designated principals to discover and control all of the slivers in the aggregates and included components.

b) When a researcher, associated with a designated slice, has been assigned resources (slivers) on GENI or federated aggregates for an experiment, the control framework shall provide a way for designated principals to discover and control all of the slivers associated with the slice, as a group, in the aggregates and included components.

c) Designated principals shall include: the researchers associated with the slice; slice administrators, PIs, etc.; aggregate administrators, operators, etc.

d) Control shall include a comprehensive set of commands appropriate to the nature of the sliver. For example: start, stop, reboot for a process running on a host; connect, disconnect, loopback for a path in a network.

5.6.2 Experiment Data Collection and Management

Note: The GENI suite provides for experiment data collection and measurement, both locally within aggregates (components) and globally in designated measurement services. It is expected that large data files will be gathered both locally and globally. After an experiment, these will typically have to transferred to a software repository and/or an experiment analysis service.

a) To accomplish this, the control framework shall provide the mechanism(s) to allow a researcher to transfer large software objects between components, software repositories, etc.

For example: Permit the researcher to login to a component and use ftp to transfer a file to a repository.

Issue: b) How can these transfers be made without interrupting normal functions within the control framework? Is a dedicated path required? Is a "scheduler" required?

5.6.3 Forensic and Usage Data Collection and Management

Note: Forensic and usage data has many uses, including:

- Keeping track of suite and aggregate resource usage, including immediate usage, recent usage and trends.
- Permitting proper administration and management of suite resources.
- Permitting financial accounting where necessary.
- Finding anomalies that indicate errors, faults, malicious activity, etc.
- Allowing help desk functions to be provided to researchers.

a) The control framework shall provide a structure for collecting and managing forensic and usage data records.

b) The control framework shall specify the basic information and the formats for the forensic and usage data records that need to be saved.

c) The forensic and usage data records shall always include the identity of the slice (or slices) associated with each record.

d) The control framework shall provide a structure for the GENI suite administrators and operators to gather, archive, and analyze forensic and usage data records associated with the entire GENI suite.

e) The control framework shall provide a structure for an aggregate's administrators and operators to gather, archive, and analyze forensic and usage data records associated with their aggregate.

f) The control framework shall provide the local and global log structures for these records, and functions to access these structures.

g) In particular, the control framework shall provide login (or request) logs in clearinghouse entities, aggregate services and component services to indicate what principals have been logged in, and what they have requested, etc.

h) In particular, the control framework shall provide ticket logs in each aggregate, and gathered in a ticket log in the clearinghouse, to indicate what resources have been authorized, assigned, revised, etc. These logs shall be in a searchable repository.

5.6.4 Experiment Status Events and Notifications

a) The control framework shall provide a structure for defining experiment status events, triggered by the use of resources in an aggregate or component, and ways to delivery notifications of these events to principals or entities.

b) It shall be possible for these events to be defined by a researcher and/or by the aggregate or component administrator or operator.

For example, a network gateway may indicate that the following event has occurred: "traffic outbound to the Internet from slice 62 has exceeded it pre-determined threshold".

Issue: c) What can be defined to trigger an experiment status event?

d) It shall be possible for notifications to be sent out to a researcher, an administrator, an operator or any other principal (or entity) who wants to see them.

Issue: e) What is the format of a notification? What data shall be included in the notification?

Issue: f) Should a publish - subscribe protocol be used for notifications? If so, which one?

g) A local log of experiment status event records shall be maintained. formatting, generating, delivering and logging

Note: By sending event records to a repository, a global log of event records can be maintained.

h) It shall be possible to poll an aggregate or component to see if an experiment status event has occurred.

5.6.5 Experiment Status Commands and Responses

a) The control framework shall provide a structure for defining experiment status commands, and ways to deliver these commands to an aggregate or component, that responds with a change in the use of resources within the aggregate or component.

b) It shall be possible for the responses to be defined by the aggregate or component administrator or operator, or by the researcher.

For example, a command may be sent to an aggregate "to shutdown all slivers in this aggregate associated with slice 62".

For example, a network gateway may be sent commands to "begin to advertise route 189 to attract traffic" and alter "stop advertising route 189 to attract traffic".

Issue: c) What kind of response can be defined?

d) It shall be possible to make an experiment status command using a browser interface.

e) It shall be possible to subscribe to a published event, the receipt of which would make an experiment status command .

Note: When experiment status events are combined with experiment status responses, a wide range of actions can be triggered by events, without or with an involved principal.

For example, a rogue traffic flow could trigger an experiment status event, with a notification that is then published, subscribed to by an operator, who issues an experiment status command to do emergency shutdown.

5.7 Federation

Note: A core concept of the GENI suite is to provide: **Federation** – Different parts of the GENI suite of infrastructure are owned and/or operated by different organizations, and the NSF portion of the GENI suite forms only a part of the overall 'ecosystem'.

The control framework provides for federated aggregates (and components) and for federated suites, as detailed in the following sections.

5.7.1 Federated Aggregates and Components

a) The GENI control framework shall provide for the inclusion of a wide variety of federated aggregates (and their included components) into a GENI suite to provide a wide range of resources to the researchers, and thus help meet the core GENI concept of federation.

b) To recognize requests from GENI researchers and designated GENI slices, a trust or contractual agreement shall be completed between the GENI suite and the owner of the aggregate, so that the aggregate can recognize requests for resources, authorize and then assign them.

c) The GENI control framework shall provide for the inclusion of a wide variety of federated aggregates (and their included components) into a GENI suite, whose native control framework is the GENI control framework.

d) The GENI control framework shall provide for the inclusion of a wide variety of federated aggregates (and their included components) into a GENI suite, whose native control framework is different than the GENI control framework.

5.7.2 Federated Suites

a) The control framework shall provide for the federation of a GENI suite with one or more suites that utilize the same control framework structure as the GENI suite.

For example, the federation of an NSF-sponsored GENI suite with an EU-sponsored GENI suite.

For example, the federation of an NSF-sponsored GENI suite with twenty university-sponsored GENI suites.

b) The control framework shall provide for the federation of a GENI suite with one or more suites that do not utilize the same control framework structure as the GENI suite.

Note: This type of federation may be quite complicated and difficult. The following approaches can be considered:

- Is it possible to put wrapper on aggregates in some or all of the suites?
- Is it possible to use the experimenter helper tools from multiple suites?
- Is it possible to include a “protocol converter box”, or would that become too complex, or limit scaling of the solution?

5.8 Reliable Operation with High Availability

a) The control framework shall be designed to assure reliable operation of the GENI suite, in both expected and unexpected conditions.

b) Since there are typically many steps and operations required to setup an experiment, individual operations shall be completed with a high degree of reliability.

c) When an operation fails or is delayed, the control framework shall provide an error indication that gives some indication of the cause and possible solution(s), so that the operation can be retried with a better chance of success.

d) When there is a problem, the control framework shall provide enough forensic information to allow an administrator or operator to understand and rectify the problem.

e) The control framework shall be designed to assure high availability of the GENI suite, in both expected and unexpected conditions.

f) The clearinghouse entities in the GENI suite shall provide very high availability, plus the ability to fully restore their stat if there is a failure.

g) The aggregate entities in the GENI suite shall provide high availability, plus the ability to fully restore their stat if there is a failure.

Issue: h) What about services acting for a principal? Do these need to have high availability? Is it important that their state can be restored?

5.9 Responsive Operation

a) The control framework shall be designed to assure responsive operation of the GENI suite, in both expected and unexpected conditions.

Consider these parameters: [TBD]

Consider these scenarios: [TBD]

5.10 Scaling Benchmarks

a) The control framework shall be designed to operate at the following scaling benchmarks:

Consider these parameters:

- Number of federated suites, similar and dissimilar.
- Number of aggregates, and included components.
- Number of research organizations, and associated principals.
- Number of slices, registered and active.
- Number of slivers, reserved and active; setups per second.

Consider these scenarios:

- GENI prototype at end of Spiral 1
- GENI prototype at end of Spiral 2
- GENI prototype at end of Spiral 3
- GENI at 5 years later
- GENI at 10 years later
- Include for reference: PlanetLab now.
- Include for reference: ProtoGENI now.

[Table TBD]

5.11 Secure Operation

a) The control framework shall use best practices to assure secure operation of the GENI suite.

b) The control framework shall use best practices to assure that servers cannot be attacked and compromised.

c) The control framework shall use secure protocols and best practices so that principals, objects and slices can be reliably identified and authenticated.

For example, protocols shall be used that are not susceptible to replay attacks.

d) The control framework shall use best practices to detect and respond to any compromise in security.

e) The control framework shall use secure protocols and best practices so that aggregates can properly authorize and assign resources, and not have them used by those who are not authorized.

Note: These issues shall be covered by the GENI Security Architecture (SA), which is not yet complete, but is being addressed by a Spiral 1 project.

Early work on GENI security is summarized in [http://www.geni.net/GDD/GDD-06-10.pdf] and [http://www.geni.net/GDD/GDD-06-23.pdf].

In particular, [http://www.geni.net/GDD/GDD-06-23.pdf], outlines:

- Threat models
- Security requirements
- Access control and authorization mechanisms
- Protection of private keys
- Audit trails and intrusion detection

6 Glossary

Entity	Explanation
Aggregate	An *aggregate* is an object representing a group of components, where a given component can belong to zero, one, or more aggregates. Aggregates can be hierarchical, meaning that an aggregate can contain either components or other aggregates. Aggregates provide a way for users, developers, or administrators to view a collection of GENI nodes together with some software-defined behavior as a single identifiable unit. Generally aggregates export at least a component interface, i.e., they can be addressed as a component, although aggregates may export other interfaces, as well. Aggregates also may include (controllable) instrumentation and make measurements available. This document makes broad use of aggregates for operations and management. Internally, these aggregates may use any O&M systems they find useful.
Clearinghouse	A *clearinghouse* is a, mostly operational, grouping of a) architectural elements including trust anchors for Management Authorities and Slice Authorities and b) services including user, slice and component registries, a portal for resource discovery, a portal for managing GENI-wide policies, and services needed for operations and management. They are grouped together because it is expected that the GENI project will need to provide this set of capabilities to bootstrap the infrastructure suite and, in general, are not exclusive of other instances of similar functions. For example, there could be many resource discovery services. There will be multiple clearinghouses, which will federate. The GENI project will operate the NSF-sponsored clearinghouse. One application of 'federation' is as the interface between clearinghouses.
Components	*Components* are the primary building block of the architecture. For example, a component might correspond to an edge computer, a customizable router, or a programmable access point. A component encapsulates a collection of resources, including physical resources (e.g., CPU, memory, disk, bandwidth) logical resources (e.g., file descriptors, port numbers), and synthetic resources (e.g., packet forwarding fast paths).
Owners / Management Authorities	GENI includes *owners* of parts of the network substrate, who are therefore responsible for the externally visible behavior of their equipment, and who establish the high-level policies for how their portion of the substrate is utilized. A *management authority* (MA) is responsible for some subset of components, aggregates, or services: providing operational stability for those components, ensuring the components behave according to acceptable use policies, and executing the resource allocation wishes of the component owner. (Note that management authorities potentially conflate owners and operators. In some cases, an MA will correspond to a single organization, in which case the owner and operator are likely the same. In other cases, the owner and operator are distinct, with the owner establishing a "management agreement" with the operator.)

Entity	Explanation
Portals	A *portal* denotes the interface—graphical, programmatic, or both—that defines an "entry point" through which users access GENI. A portal is likely implemented by a combination of services. Different user communities can define portals tailored to the needs of that community, with each portal defining a different model for slice behavior, or support a different experimental modality. For example, one portal might create and schedule slices on behalf of researchers running short-term controlled experiments, while another might acquire resources needed by slices running long-term services. Yet another portal might be tailored for operators that are responsible for keeping GENI components up and running.
Resource	Resources are abstractions of the sharable features of a component that are allocated by a component manager and described by an RSpec. Resources are divided into computation, communication, measurement, and storage. Resources can be contained in a single physical device or distributed across a set of devices, depending on the nature of the component.
Substrate	GENI provides a set of physical facilities (e.g., routers, processors, links, wireless devices), which we refer to as the substrate. The design of this substrate is concerned with ensuring that physical resources, layout, and interconnection topology are sufficient to support GENI's research objectives.

Interface	Description
Measurement Plane	Configuration for measurement infrastructure; management of collected data.
Control Plane	Resource discovery, reservations, and release; slice control (e.g., experiment start and teardown); some debug.
Experiment Plane	Experiment data flow; "in-band" debugging; experiment control.
Operations and Management Plane	Operational status data; privileged slice & component/aggregate control; network event reporting.
Opt-In	Interconnecting GENI to non-GENI networks over, e.g., IP, IP tunnels, conventional (wired or wireless) link protocols. GENI experiments may run just in GENI (e.g., an experimental service accessed by Internet users) or end-users may 'opt-in' to running experimental code on their end-system.

Federation	Resource *federation* permits the interconnection of independently owned and autonomously administered facilities in a way that permits owners to declare resource allocation and usage policies for substrate facilities under their control, operators to manage the network substrate, and researchers to create and populate slices, allocate resources to them, and run experiment-specific software in them.

Experiment	An experiment is a researcher-defined use of a slice; we say an experiment runs in a slice, or in multiple slices since slices can be composed or interconnected. Experiments are not slices. Many different experiments can run in a particular slice concurrently or over time.
Sharing	Wherever possible, GENI components shall support multiple concurrent experiments. We refer to this as making components and aggregates *sharable* (or sometimes "sliceable"). Different strategies may be needed to share components based on the nature of the technologies. This can be done by a combination of virtualizing the component (where each user acquires a virtual copy of the component's resources), or by partitioning the component into distinct resource sets (where each user acquires a distinct partition of the component's resources).
Slices	From a researcher's perspective, a *slice* is a substrate-wide network of computing and communication resources capable of running one or more experiments or a wide-area network service. From an administrator's perspective, slices are the primary abstraction for accounting and accountability—resources are acquired and consumed by slices, and external program behavior is traceable to a slice. A slice is defined by a set of slivers spanning a set of network components, plus an associated set of users that are allowed to access those slivers for the purpose of running an experiment on the substrate. That is, a slice has a name, which is bound to a set of users associated with the slice and a (possibly empty) set of slivers.
Slivers	It shall be possible to share component resources among multiple users. This can be done by a combination of virtualizing the component (where each user acquires a virtual copy of the component's resources), or by partitioning the component into distinct resource sets (where each user acquires a distinct partition of the component's resources). In both cases, we say the user is granted a *sliver* of the component. Each component shall include hardware or software mechanisms that isolate slivers from each other, making it appropriate to view a sliver as a "resource container."
User Opt-In	An important feature of GENI is to permit experiments to have access to end-user traffic and behaviors. For examples, end users may access an experimental service, use experimental access technologies, or allow experimental code to run on their computer or handset. GENI will provide tools to allow users to learn about an experiment's risks and to make an explicit choice ("opt-in") to participate.