G E N I

Global Environment for Network Innovations

GENI System Overview

Document ID: GENI-SE-SY-SO-02.0

September 29, 2008

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 provides an overview of the GENI system design. It is a very early draft, and should be taken as a basis for discussion only.

Much of the material in this document is drawn from other GENI documents. In particular, Larry Peterson and John Wroclawski are recognized for their significant contributions. Ted Faber and Jeff Chase provided thoughtful reviews.

1.2 Context for this Document

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

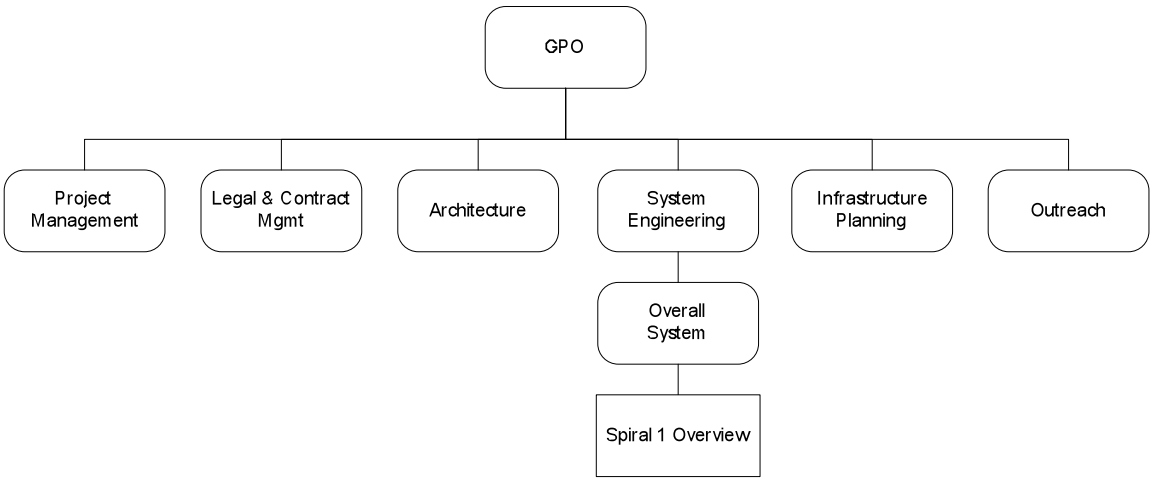

Figure 1. This Document within the GENI Document Tree.

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.

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-ARCH-CP-01	GENI Control Plane Framework (to be published)

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
N / A	

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
-01.0		A. Falk	Initial draft.
1.0	12/14/07	A. Falk	Release for Solicitation #1
1.1	12/19/07	A. Falk	Cover page reformat, new text in §2.2, reworked §6
1.2	9/13/08	H. Mussman	Revised to use current, detail system decomposition
1.3	9/15/08	A. Falk	§4 cleanup and reorganization
1.4	9/18/08	A. Falk	Revised based on GPO review
1.5	9/18/08	A. Falk	Version for Spiral 1 PI review
2.0	9/29/08	A. Falk	Version for public review

2 Introduction

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

This new research challenges us to understand networks broadly and at multiple layers of abstraction from the physical substrates through the architecture and protocols to networks of people, organizations, and societies. The intellectual space surrounding this challenge is highly interdisciplinary, ranging from new research in network and distributed system design to the theoretical underpinnings of network science, network policy and economics, societal values, and the dynamic interactions of the physical and social spheres with communications networks. Such research holds great promise for new knowledge about the structure, behavior, and dynamics of our most complex systems – networks of networks – with potentially huge social and economic impact.

As a concurrent activity, community planning for the suite of infrastructure that will support NetSE experiments has been underway since 2005. This suite is termed the Global Environment for Network Innovations (GENI). Although its specific requirements will evolve in response to the evolving NetSE research agenda, the facility's conceptual design is now clear enough to support a first spiral of planning and prototyping. The core concepts for the suite of GENI infrastructure are as follows.

- **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.

As envisioned in these community plans, the GENI suite will support a wide range of experimental protocols, and data dissemination techniques running over facilities such as fiber optics with next-generation optical switches, novel high-speed routers, city-wide experimental urban radio networks, high-end computational clusters, and sensor grids. The GENI suite is 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.

2.1 GENI's Design Goals

As the system design for GENI evolves it is guided by the following design goals. These goals have been derived to ensure the resulting infrastructure suite will be useful to the research community by encouraging wide-spread deployment, diverse and extensible technologies, and support for real-user traffic. See GDD-06-08 for a detailed discussion.

Goal	Explanation
Generality	GENI should give each experimenter the flexibility needed to perform the desired experiment. This means that each component should be programmable, so that researchers are not limited to experimenting with small changes to pre-existing functionality.
Diversity & Extensibility	GENI must include a wide class of networking technologies, spanning the spectrum of wired and wireless technologies available today. GENI must also be extensible—with explicitly defined procedures and system interfaces—making it easy to incorporate additional technologies, including those that do not exist today.
Fidelity	GENI should permit experiments that correlate to what one might expect in a real network.
Observability	GENI must offer strong support for measurement-based quantitative research.
Ease of Use	GENI must remove as many practical barriers as possible to researchers being able to make full use of its federated infrastructure.
Sliceability	To be cost-effective, GENI must be a shared infrastructure suite that can be used to support multiple experiments running on behalf of many independent research groups.
Controlled Isolation	GENI must support strong isolation between slices so that experiments do not interfere with each other.
Opt-in	To support meaningful deployment studies, GENI must make it easy for a broad mix of users to "opt in'' to experimental services.
Security	GENI must be secure, so that its resources cannot accidently or maliciously be used to attack today's Internet.
Federation & Sustainability	GENI must be designed for a 15-20 year lifetime.

Many of these goals are in tension with each other. Where possible GENI's design should permit researchers to affect the balance, for example where sliceability is in tension with fidelity, the infrastructure suite should permit some experiments on dedicated hardware to enable high-fidelity measurements in addition to supporting platforms which may be shared.

To achieve these goals, the GENI project uses an engineering approach informed by the success of the Internet and the open source software movement:

- Start with a well crafted system architecture
- Build only what you know how to build
- Build incrementally
- Design open protocols and software

- Leverage existing technology

This approach is reflected in the system design presented in the following sections.

2.2 About This Document

The GENI Project Office, in collaboration with the GENI working groups, will be publishing a series of design documents that will provide more detailed frameworks for key functions, define the major subsystems within GENI, and specify the interfaces between them. These documents will be informed by a community-led design and prototyping efforts. This collaborative effort will add detail to the GENI design, backed up by a broad range of implementations.

The GENI design process is proceeding in parallel with a separate process, led by the Network Science and Engineering (NetSE) Council, to clearly articulate the research motivations for the infrastructure suite in a Research Agenda. GENI's aim is to satisfy these research needs.

As these are concurrent processes, it may turn out that the research needs impose changes on the GENI design. The GENI Project Office anticipates this and is instituting a "spiral development" process to permit smooth evolution to changes in requirements and design.

This document is intended to help readers unfamiliar to the GENI project understand the design through examples and discussions of key elements and concepts.

Section 3 gives a narrative introduction to how a researcher would use GENI. This is followed definitions of the major GENI subsystems and a high-level description of the concept of operations in Section 4. Section 5 gives some examples of aggregates and how they might be integrated into GENI. Section 6 discusses how GENI slices connect to non-GENI networks. Section 7 gives some preliminary details on how slices work. Section 8 discusses the scope of experimenter support tools and services. Section 9 presents a justification for clearinghouses, a recent addition to the conceptual design. And Section 10 summarizes an approach to measurement and instrumentation.

The GENI design is neither fixed nor complete. In particular, the areas of security, federation, end-user opt-in, and measurement are not well defined and are important functions needed for a complete system. This document illustrates how some of them might be implemented. However, further work is needed and the solutions presented in this document should be viewed as illustrative, not final.

3 Using GENI – An Example

This section introduces GENI's basic concepts by a very simple example. It has two goals: to provide a basic understanding of how GENI might be used in practice, and to introduce most of the key system concepts. These concepts will then be treated more formally in subsequent chapters.

3.1 Resource Discovery

We start our example with a researcher who wishes to use GENI to perform an experiment. An experiment is simply some researcher-defined use of GENI resources. Figure 1 depicts our researcher finding out what GENI resources are available for her experiments. Researchers are the users of GENI; but we explicitly call them "researchers" to distinguish experiment creators from experiment participants, such as end-users or other researchers building on a long-running experiment.[1]

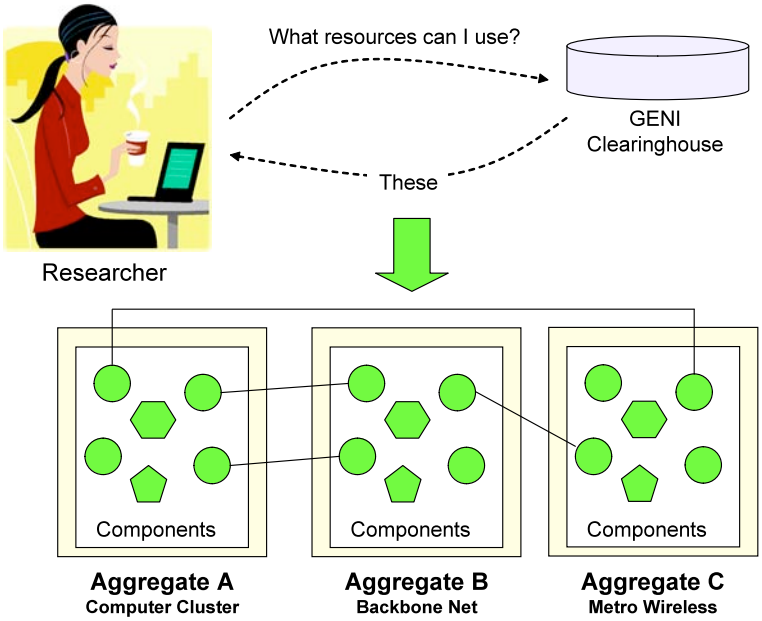

Figure 1. Resource Discovery.

To perform resource discovery, the researcher consults a resource discovery portal such as the GENI Clearinghouse, which contains information about resources available to GENI researchers. It not only knows what resources exist, but it also has some idea about which resources are currently available, which are already booked for other researchers, and which may be unavailable due to planned and scheduled events such as outages for preventative maintenance.

The GENI Clearinghouse also knows who is *authorized* to use resources. Later chapters will explain exactly how the clearinghouse knows about these researchers and their authorizations, but here we note that the clearinghouse not only knows about existing GENI researchers and resources, but also

[1] Note that other GENI documentation is inconsistent on this point.

includes interfaces to a policy engine that automates community decisions about who can use which resources, and when[2].

Finally, notice that most GENI components are not treated as isolated units. Instead they are parts of *aggregates*, which are collections of resources managed as a coherent whole. (Aggregates and components are defined in Section 4.1.1. and 4.1.7.) GENI will contain many different kinds of aggregates – for example, we expect that even early versions of GENI will contain one or more aggregates of each of the following types: computing / storage cluster; regional network; backbone network; metropolitan wireless network and sensor network

An aggregate consists of far more than just CPUs. As we shall see, most aggregates can be shaped in various ways – for example, a topology can be instantiated on a backbone network, or specific sets of radios can be enabled or disabled for ensemble effects in wireless networks. This should become clear as this example unfolds.

One last word on aggregates – each aggregate is managed by some organization. We do not expect that there will be a "GENI Central" that has direct, hands-on control of all machinery within GENI. Instead, a clearinghouse will coordinate the activities of a number of aggregates, each controlling its own local operations and management. This approach has many advantages, explained in greater detail in subsequent chapters.

3.2 Slice Creation

Figure 2 depicts "slice creation" in a highly stylized form. A slice is an empty container into which experiments can be instantiated and to which researchers and resources may be bound. Here specifically we see a slice that extends across three aggregates: a computer cluster, a backbone network, and a metro wireless network. The resources within this slice are linked together to form a coherent virtual network in which an experiment can run.

[2] A centralized approach is simple to operate and straightforward to develop but centralized policy engines will have scaling limitations. Therefore, distributed policy mechanisms such as the use of virtual (or real) currency or reputation systems are also of interest. Such systems may not be located within the clearinghouse. In fact, some policies will be purely of local concern to the component owner and would naturally be applied at the aggregate.

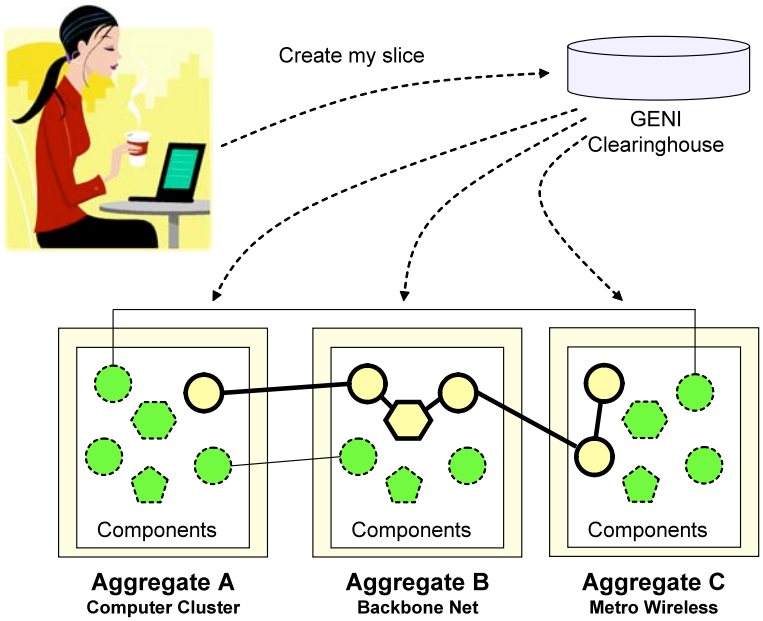

Figure 2. Slice Creation.

Ideally, this slice (virtual CPUs, virtual network, etc.) will be well isolated from other researchers' slices so that experiments running within the slice will behave consistently no matter what other researchers are doing within their own GENI slices.

Slice creation involves several activities.

- First, individual resources must be allocated within an aggregate; examples include real or virtual CPUs, storage, wireless nodes, etc.
- Second, these resources must be woven into a coherent topology within the aggregate. For example, in a backbone network this could involve setting up a virtual private network (VPN) to link the resources, which might be quite far apart geographically; alternatively this could be accomplished by provisioning lightpaths, setting up an Ethernet virtual LAN, etc. Note that this kind of activity is not confined to backbone networks; it will probably be required in most aggregates within GENI. For example, topology creation will probably be required within storage area networks, wireless networks, and within the backplanes of larger CPU clusters.
- Third, the aggregates must be stitched together to form a coherent slice. For example, a compute cluster in Aggregate A must be stitched to its corresponding backbone node in Aggregate B. Later chapters explain how this dataplane stitching is performed.

When these steps are finished, the researcher has a complete "blank" slice. She is now ready to download her software into the programmable elements within the slice, and then to start her experiment.

3.3 Experimentation

Figure 3 shows the researcher actually performing her experiment. She downloads code into her slice, debugs, collects measurements, and iterates. Note that these interactions occur directly between the researcher and the aggregates; the clearinghouse does not participate.

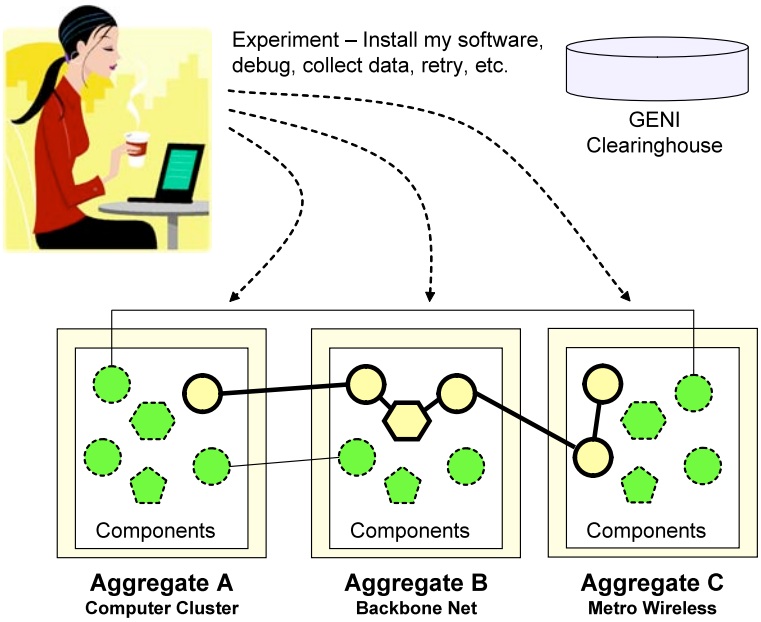

Figure 3. Experimentation.

Experimentation may take place over different timeframes. It is possible that a researcher may be able to plan a small experiment, and then execute it and collect results, all within a few hours. This would be made easier if GENI provides "canned" experiments that can be modified to produce new student experiments, for example. However we expect that many GENI experiments will be quite open-ended and long running; some experiments may run for years.

Over this period of time, many planned and unplanned events can occur within the infrastructure suite. Disks and processor cards may fail; backhoes may cut optical fibers; bad weather or power outages may take down portions of metro wireless networks, individual component software and hardware may be upgraded, policies for accessing overloaded aggregates may change. Because GENI cannot protect against these events, researchers must be able to obtain enough information about the federated infrastructure to understand why certain portions of their experiments are experiencing interruptions and respond appropriately.

Because GENI will contain many different kinds of computational and networking resources, and these will surely change as technology progresses, GENI cannot provide a single, uniform interface for programming or debugging every kind of GENI element. Consider software download as a specific case. GENI will almost certainly contain various types of CPUs in its cluster computers (different manufacturers, different product families, different generations). In addition, it will probably contain a variety of small, embedded processors, e.g., in handheld devices, sensors, etc. Finally, it seems highly likely that GENI will contain resources with Field Programmable Gate Arrays (FPGAs), network processors, and other specialized types of processing engines. Clearly a range of development and debugging tools will be needed for this wide variety of processors.

The same holds true for instrumentation and measurements. For operational reasons as well as to support research, GENI will be instrumented to the teeth, and the instrumentation streams will be open unless there is a compelling reason to block access (e.g., privacy issues for opt-in users). This is one of GENI's distinguishing characteristics.

For example, the types of measurements needed for good experiments with radio networks (e.g. RF spectrum measurements) are extremely different from those needed for higher-level congestion control algorithms. Therefore we expect GENI to define a general measurement framework into which one can plug specific kinds of instrumentation and measurement devices. (Right now most aspects of measurement are unclear.)

Because GENI is meant to enable experimentation with, and understanding of, large-scale distributed systems, experiments will in general always contain multiple communicating software entities within a slice. Large experiments may well contain tens of thousands of such communicating entities, spread across continental or transcontinental distances. At present, it seems that inter-entity communications within most such experiments will be packet-based (although often not IP); a key open challenge in GENI is how to best incorporate circuit switching.

3.4 Growing an Experiment (Modifying a Slice)

Figure 4 shows that a researcher may "grow" an experiment by modifying the slice in which it runs. This will be a natural action as a successful experiment evolves over time. She may also shrink or otherwise modify her existing slice.

All these actions are accomplished by altering the resources devoted to that slice, e.g., the computational elements within the slice, and the connections between these elements. In Figure 4, we see that the researcher has added new computational elements and new links within aggregates, and also added new links between aggregates. Although it is not visible in the figure, she may also have adjusted link bandwidths, and perhaps some of their operational characteristics (discard rate, etc.).

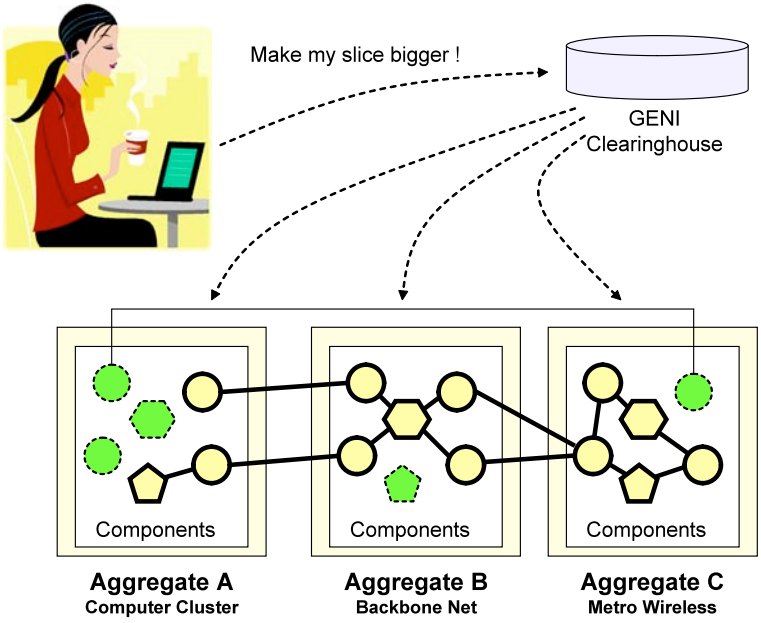

Figure 4. Growing an Experiment (Modifying a Slice).

Although "growing an experiment," i.e. modifying a slice, is a straightforward concept, it does pose some challenges for system design. We do not yet fully understand this level of design, but we outline three important issues here.

One important issue will be the resource allocation problem; GENI cannot perform a once-and-for-all global optimization of resource usage across slices, because slices are created and deleted at unpredictable times, and existing slices can be grown, shrunk, or otherwise modified. We do not currently expect that GENI will reshuffle a researcher's resources on the fly, e.g., by employing process migration to rearrange resources for better overall usage. Thus the "free pool" of resources will need to be managed by some algorithm that performs reasonably well under a variety of request scenarios.

Another issue is the means by which an existing computational element "learns" that it has new links. As a concrete example, suppose that an experiment is already running software in a Linux operating system on some computer in Aggregate A, and that the researcher's slice has now added a new link from this computer to a new component in Aggregate B. How exactly is this new link accessed from the existing software? Has a new network interface been added to the computer? Or has the existing network interface been extended from a point-to-point link to a mesh of links or a multicast path to multiple destinations?

A third issue arises with the technical means employed to manipulate aggregate topologies. It is highly desirable that existing topologies continue to function throughout the modification process; however, the underlying tools for implementing these topologies (e.g. VPNs, MPLS, …) may not support such modifications on the fly, but instead may need to tear down the existing topology before building the new one.

3.5 Federated Facilities

Figure 5 shows how a really successful experiment may begin to expand from the parts of GENI that are managed via the GENI Clearinghouse to other "federated" parts of the greater GENI ecosystem. We expect there to be many other parts of GENI, each of which will have its own clearinghouse. Some may be owned by private corporations, others by different US government agencies, and perhaps some by governments and organizations outside the United States.

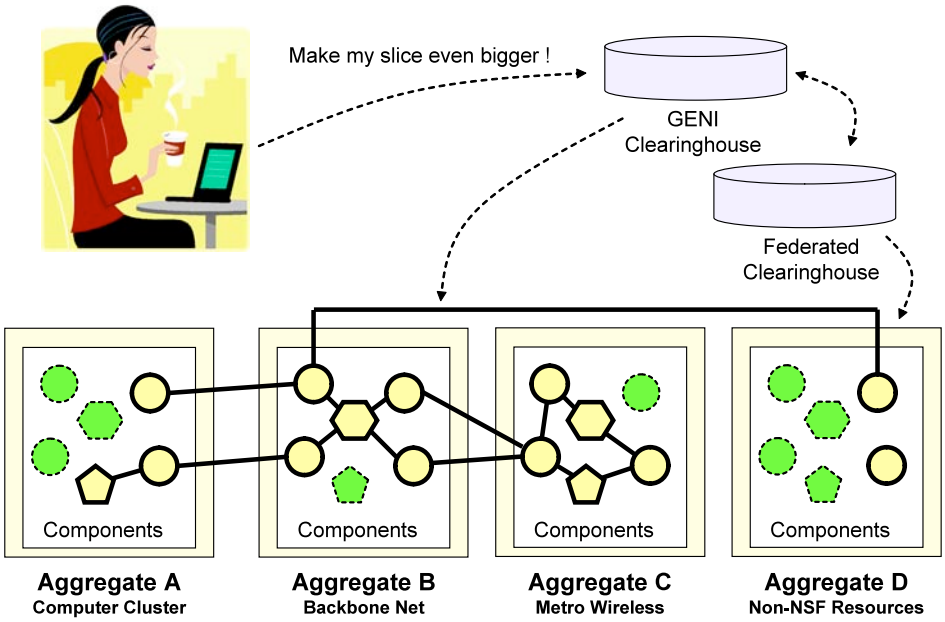

Figure 5. Using Federated Facilities.

Federation is a very important concept for GENI because it allows the federated suite of infrastructure to expand beyond its original, NSF-funded portions. Just as the NSFNET was only part of the early Internet, so the NSF GENI may be only part of a larger GENI ecosystem. There are many architectural issues about federation and policy administration that are still open. But these are problems that will need to be solved to manage contention for resources in some reasonable way.

One way in which federation can be accomplished is via clearinghouses. Here we show that Aggregates A, B, and C are affiliated with the (NSF) GENI Clearinghouse, while Aggregate D is one of the aggregates affiliated with another Federated Clearinghouse. That clearinghouse may be operated by a private company, other US agency, or indeed a separate nation.

In order for the researcher to use federated parts of GENI, several services must be implemented. First, she must be able to see these other parts; thus clearinghouses must be able to share information about their own aggregates with each other. Note that clearinghouses may choose to reveal only a portion of their resource information. For example, a private company may run a large aggregate but reveal only a portion of it to NSF researchers.

Second, she must be able to request that federated resources be incorporated into her slice. This may involve running policy engines on both her own clearinghouse and the other federated clearinghouse before she is given any resources; for example, the NetSE Council may wish to impose limits on trans-Pacific bandwidth for experiments running between the US and Asia, while an Asian government might wish to restrict the number of resources that can be obtained for an experiment that does not have local partners.

Third, in order to perform these joint actions, the federated clearinghouses must establish some degree of trust and authorization, and perhaps accounting, between themselves; relationships between federated clearinghouses are likely to be carefully negotiated rather than “free for all.”

3.6 GENI Operations and Management

Finally, we bring this scenario to its conclusion with Figure 6, which shows that GENI's federated management must be always present, although often invisible to researchers, and may need to take prompt action to stop experiments that are running out of control.

Even at this early stage in GENI's design, it is clear that the infrastructure suite must have a highly reliable "emergency shutdown" feature that stops a run-away experiment. Imagine, for instance, a researcher that obtains tens of thousands of computers in her slice, then either accidentally or on purpose uses these computers to launch an attack on the Internet. This situation must be detected, diagnosed, and addressed very quickly, for example by shutting down all or part of an experiment.

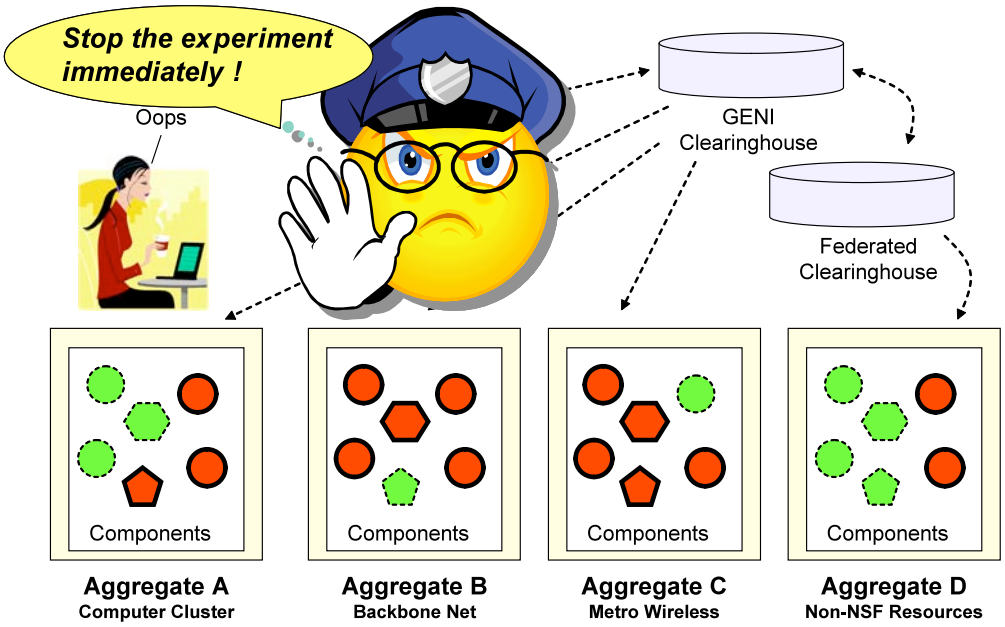

Figure 6. Immediate Shutdown when Needed.

Although the exact means to implement emergency shutdown are not yet clear, they probably involve some combination of resetting the slice's computational elements to a known benign state (e.g. off or running trusted software), or ensuring that these elements cannot communicate. Once an incident has been resolved, Operations and Management (O&M) will also need a means to put the affected resources back in service. Communications between O&M organizations affected by an incident (for example other federated clearinghouse or aggregate O&M groups will also be important in any shutdown scenario.

Other anticipated GENI O&M functions are described in section 4.4.4. Although not generally visible to researchers, they are critical for GENI's successful operation and thus play a major role in driving GENI's overall design.

4 GENI System Overview

This chapter provides a technical system overview of the current state of GENI design, showing all the major parts of the GENI suite and how they fit together. Subsequent chapters provide details on these parts.

This chapter is a work in progress. GENI is still early in its design phase, and the diagrams and descriptions in this chapter are somewhat inconsistent. We expect them to evolve considerably as prototyping and design work progress. Even so it may be helpful in obtaining a basic understanding of the technical aspects of the GENI system.

4.1 Major Entities and their Relationships

The major subsystems in GENI are defined in a very general way to permit the widest possible range of technologies, usage, and operational models. Each is discussed briefly in the sections below with an emphasis on identifying the key functions and interfaces.

Figure 7 presents a block diagram of the GENI system covering the major entities within the overall system. Optional but desirable parts of each entity are shown "grayed-out." This section provides a top-level overview of these entities and their relationships.

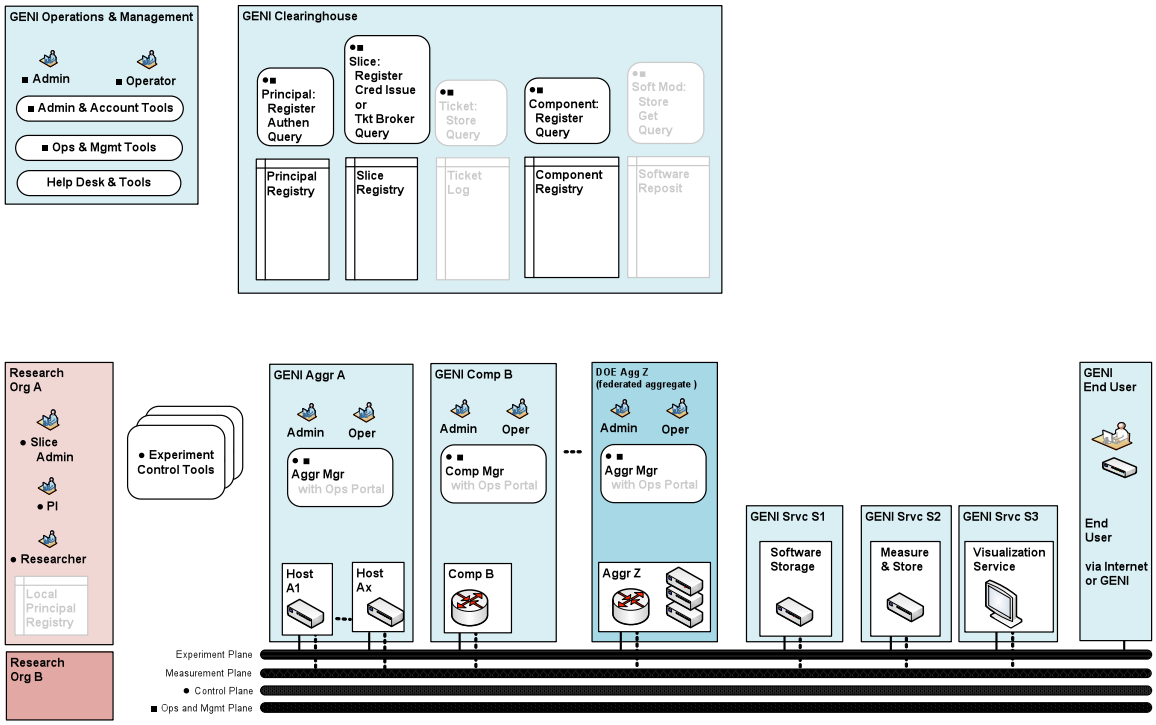

Figure 7. GENI Block Diagram.

4.1.1 Aggregates and Components

The GENI suite is expected to include many different **aggregates** and **components**, all independently owned and operated, though available for experiments via a control framework run by the clearinghouse. Components are, speaking intuitively though not precisely, the individual "things"

that can be obtained and programmed for running experiments. They can be organized into aggregates, which are groups of things owned and administered as an ensemble by some organization. Aggregates do not have to be homogeneous; for example, a single aggregate may contain a collection of cluster computers, Ethernet switches, disk storage networks, etc. Individual components (aggregated or not) are typically "virtualized" (or shared in another way) and are often programmable, so that they can be uniquely configured for a particular experiment. Many aggregates or components will be "federated", i.e., owned and/or operated by different organizations, but nonetheless affiliated with the GENI clearinghouse.

Researchers use GENI by acquiring **resources** from components through the GENI control framework. Resources may be virtualized or real and may be on a single component or require coordination from multiple components within an aggregate. The **component manager** provides the interface to the control framework, manages resource allocation, and – using internal communications – configures components to provide **slivers**. When components are organized into aggregates, an **aggregate manager** provides the above functions plus any needed organization of components to provide resources that span multiple components. For example, a network might be treated as an aggregate that provides researchers Ethernet VLANs. The network would implement an aggregate manager that configures the Ethernet switches based on researchers' connectivity requests.

The aggregate/component manager may apply access control or resource allocation policies.

Most aggregates/components provide an **operations portal** to export operational data to GENI O&M. This data, provided in a standardized format, can be used to provide help-desk services, maintain high-level views of system status, and identify network events (e.g., failures or attacks). The portal also permits privileged access by GENI O&M for diagnostic and management purposes (such as requesting shutdown of slivers associated with an out-of-control slice). A privileged access path is typically provided so that an operator can bypass a congestion or failure in the control plane.

4.1.2 Clearinghouse

The **clearinghouse** is a software entity that is logically centralized, probably implemented in a distributed fashion for robustness. The clearinghouse includes **principal, slice, and component registries**, with related services. In addition, the clearinghouse may contain the following optional entities: a **ticket log** and/or a **software repository**, with related services. It is likely that the clearinghouse will also maintain other registry transaction logs to allow for later troubleshooting and system utilization studies.

The **principal registry** holds a record for each GENI-associated researcher, PI, administrator, operator, etc. (or a pointer to a record in another trusted component registry, such as that of a trusted research organization). Each principal record includes: a global name, contact information, authentication key (or a pointer to it in another trusted registry), roles, and status (active, suspended, etc.). This registry includes services for: principal registration and management; principal authentication; and related queries. In particular, the principal registry includes records for of all researchers who have permission to establish slices, each vouched for by an associated research organization.

The **slice registry** holds a record for each slice, equivalent to a "bank account" for that slice in that it records transactions and authorized accesses. Each slice record includes: the responsible organization (e.g., the slice administrator) and its permissions; associated principals (e.g., researchers)

and their individual permissions; and the slice status (active, suspended, etc.). This registry includes services for: slice registration (creation) and management; issuing slice credentials; brokering tickets; and making slice queries.

The **component registry** holds a record for each affiliated substrate component or aggregate, possibly via a pointer to a record in another trusted component registry. Each record includes: the responsible organization (i.e., management authority); associated principals (e.g., operators) and their individual permissions; the interface(s) to query for available resources (e.g., on the component manager); other contact information; and (optionally) policy to be applied to use of this component or aggregate. This registry includes services for: registration and management; and related queries. Thus, this registry provides records for all components or aggregates that have agreed to participate in experiments that utilize this clearinghouse, entered by the owners/operators of the components. Aggregates may choose to have records for each constituent component.

A **ticket** is a "sliver record" that specifies the resources that a component allocates (or promises to allocate) to a given slice. Depending upon the approach used to obtain a ticket, the clearinghouse contains a credential issuing service or a ticket broker service associated with the slice registry. A **credential issuing service** in the clearinghouse issues a credential (a cryptographically-signed certificate) to a researcher, who then uses the credential to obtain a ticket directly from a component or aggregate. A **ticket broker service** in the clearinghouse (or elsewhere) brokers the initial ticket for a researcher from a component and can apply a clearinghouse policy to determine who gets a ticket. An example of such a policy might be an upper bound on the duration of tickets for certain classes of users (e.g. undergraduates) on some components.

A **ticket log** holds copies of the initial ticket (sliver record) and subsequent updates, or their equivalents. This could allow administrators and operators to find and manage all slivers related to a particular part of GENI, e.g., to find all slivers associated with a set of slices. A ticket log could provide useful diagnostic information for the control plane (e.g., for troubleshooting slice establishment problems or associating network events with slice activities). Additionally, it could allow administrators to track usage patterns and forecast growth. This log would include a service for queries. Ticket log information might also be useful to researchers and end-users, assuming appropriate security and privacy protection were available.

A clearinghouse can "federate" with another clearinghouse, recognizing the other's slices, principals, and components based on some negotiated policy. Therefore, clearinghouses will have **federation interfaces**. This is discussed further in Section 4.2.

4.1.3 Research Organizations, including Researchers and their Experiment Control Tools

GENI also includes **research organizations** (primarily universities) including **researchers** with associated **experiment control tools**. Optional **local principal registries** in a research organization keep track of that organization's own researchers and their roles (lab director vs. first year grad student), linkages between the organizations researchers and those at other organizations, and so forth. It is these organizations that know whether Researcher X is a graduate student, whether he/she is currently working on a given experiment for Professor Y, or whether he/she has left the organization (for example). Research organizations will typically include the following roles: **slice administrators**, who are responsible for creating a slice record, and authorizing PIs and researchers to utilize the slice to

setup and run experiments; **PIs**, who manage the use of a slice by researchers to setup and run an experiment; and **researchers**, who actually setup and run experiments.

Due to the complexity of dealing with multiple components (and other services) to setup and run an experiment, researchers may use **experiment control tools** which act as proxies for researchers. These tools can help with software package acquisition; resource discovery; resource reservation; slice setup and debugging; slice operations; experiment measurements; archiving results; and forensic recordkeeping. When slices are large, tools will be helpful for coordinated scheduling and sliver interaction. Experiment control tools are expected to use slice-specific state (such as slice credentials) and will usually be dedicated to one slice. The research organization, the GENI clearinghouse or a third party may host such tools.

4.1.4 Experiment Support Services

The GPO expects the GENI community to develop a wide range of **experiment support services**. These services might include Software Storage services, for researchers to archive code, configurations and experiment results; Measurement Storage services, to make, gather and archive experiment measurements; and Visualization Services, to provide ways to view relationships between experiment plane data flows and experiment users.

A research organization, the GENI clearinghouse or a third party may host experiment support services. Services may be implemented with interfaces to substrate components aggregates, or slices. A service might use a dedicated slice of its own to collect, organize, and deliver information for the service. Particular components and aggregates might even incorporate service controllers. The GPO expects that the research community will build, deploy and share an ever-growing number of such services.

4.1.5 Opt-in End Users

GENI **Opt-in End Users** are those who choose to participate or "opt-in" to a GENI experiment, and become part of the slice. These users may access GENI through the general-purpose Internet, or through a network that supports the GENI natively (for example at a university LAN that is part of a GENI project). The users may have no understanding of GENI, but be running an application or service that takes advantage of GENI resources. Including real-world users and traffic in GENI is key to providing the fidelity experimenters need in the GENI suite of infrastructures to make their experimental results potentially relevant to real-world networks. The number of opt-in users may easily exceed the number of research users. Unlike research users, the opt-in users may not be individually registered and authenticated in the GENI clearinghouse or aggregates (probably not in most cases), so experiments will need to provide recording, tracking, and security/privacy functions for their opt-in users when they operate on GENI.

4.1.6 GENI Operations and Management

GENI **Operations and Management** are provided by people, tools, and services who administer and operate some or all parts of the GENI system, and in particular its clearinghouse. Operations and Management (O&M) for GENI will be distributed among many organizations and individuals, because of the collaborative nature of the infrastructure suites. Management and operations tools may be

similarly distributed, or local tools may be used for specific functions, and then resultant actions coordinated (for example with agreed procedures in a concept of operations) in a distributed fashion. Researchers may also use O&M tools and vice versa (for example an end-to-end monitoring tool would be useful to both communities). GENI O&M functions should be coordinated with Aggregate O&M functions and with federated Clearinghouse O&M functions, and also provide interfaces to O&M outside the native GENI infrastructure, for example in the general Internet.

The **Help Desk** function is implemented here. The help desk is a way for Researchers to communicate with the Ops team and get help in setting up and fixing experiments etc.

Administration and accounting tools help administrators to provide routine functions such as: authorizing new research organizations to the GENI infrastructure suite; registering new slices; registering new principals; registering new components; modifying or deleting existing registrations; and reporting on changes in registries. Accounting (for example recording usage) functions may also be needed, particularly if GENI implements clearinghouse policies that require it.

Operations and management tools help operators to manage the overall GENI system and its interfaces to other systems and to troubleshoot, resolve, and record issues in the suite of GENI infrastructures. Depending on the component or aggregate, the operators may be able to access a separate O&M plane or interface to debug, reset, and query parts of the infrastructure in ways that differ from the normal experimental accesses. Operators may also use specialized functions such as emergency shutdown that are not available to the community at large.

Help desk tools support researchers, PIs, slice administrators in setting up slices, running experiments and reporting and escalating problems. Operators use these tools as well, of course

4.1.7 Definitions

The following table provides somewhat more formal definitions of each of these entities with descriptions of how they inter-relate.

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.

Entity	Explanation
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.)
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.

4.2 Federation

Figure 8 provides a system diagram illustrating federation between one GENI clearinghouse and another. As a hypothetical example, it depicts federation between a US-based clearinghouse and a compatible framework in the European Union (EU).

The GENI design does not assume that it is federating with identical "GENI-like" systems. Instead it provides a relatively narrow, clearly defined set of interfaces for federation, and can federate with any entity that implements those interfaces. For clarity, however, we show a similar type of system with its own clearinghouse functionality, as well as administration and operations functions, researchers, and aggregates that can be allocated and programmed for experiments.

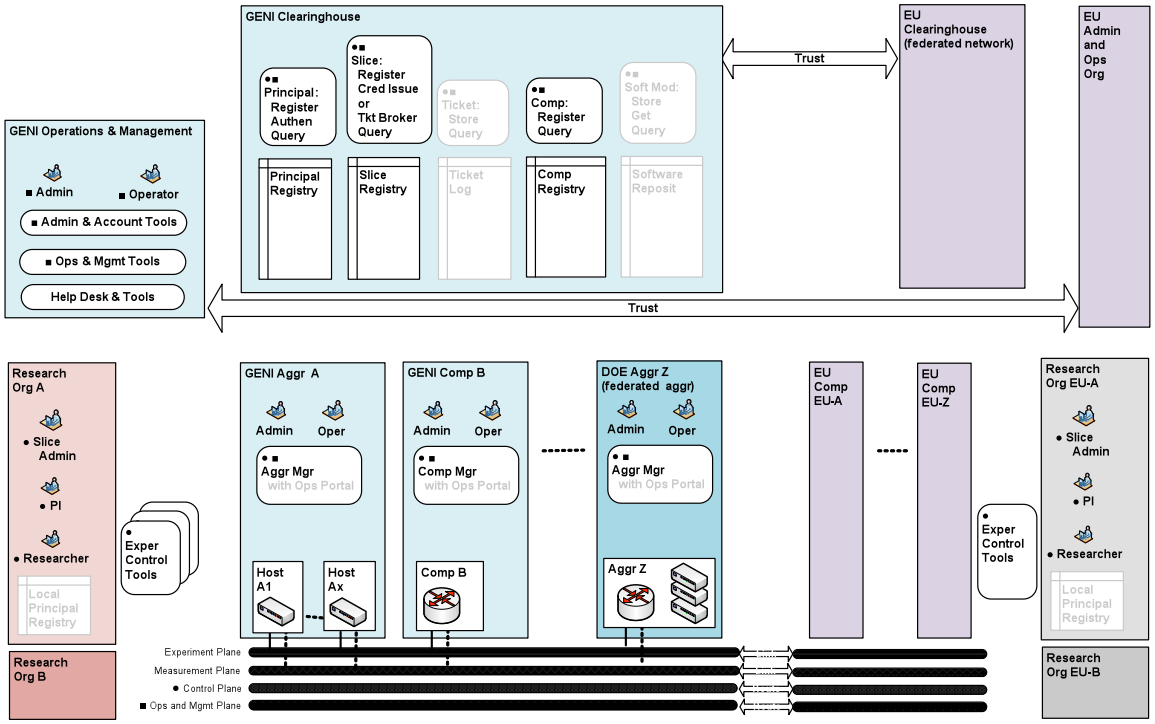

Figure 8. System Diagram with Federated Infrastructure Suites.

It is important to point out that several different kinds of interconnections are required for federation – for example, not only must the experiment control frameworks interoperate, but O&M systems must also interwork (even if only at the telephone and email level between human operators). And of course the various interconnection planes must also be compatible between at least some of the aggregates on either side.

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.

4.3 Slices with Opt-In End Users

Figure 9 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.

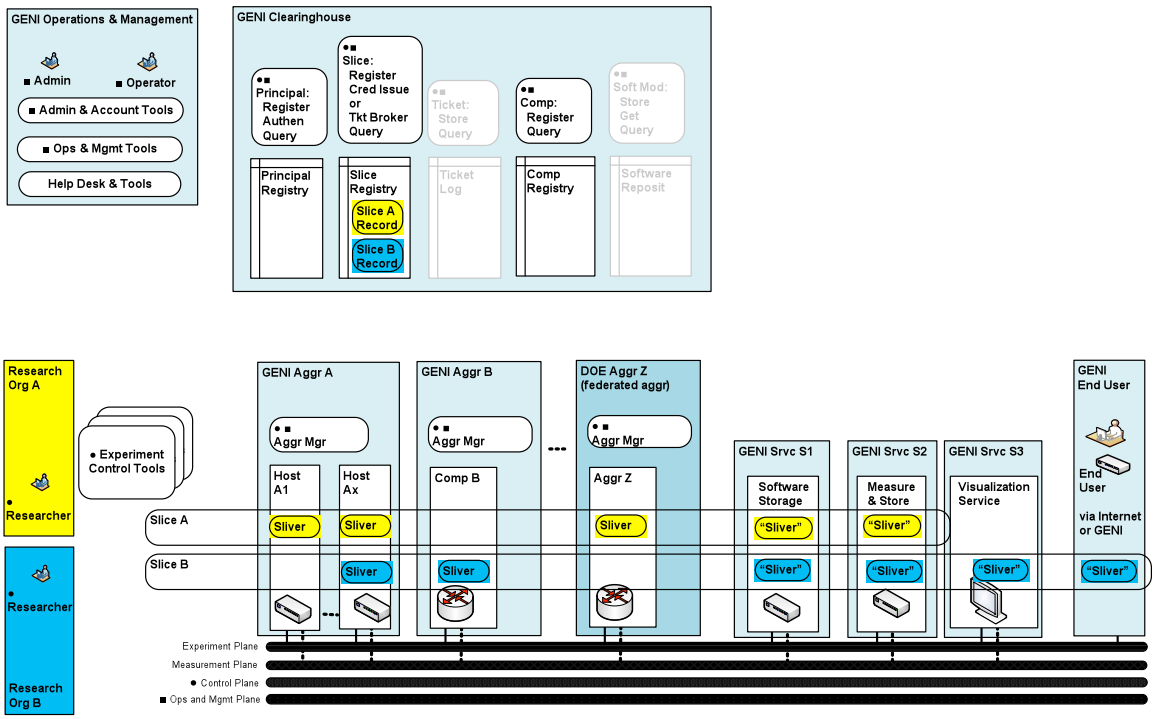

Figure 9. Two GENI Slices in a System Diagram.

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.

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.
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	When possible, a component should be able to share it's 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 must 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.

4.4 Concept of Operations

The following sections describe, in very high-level form, how the entities shown above interact in a number of key operations that will be performed on the infrastructure suite:

- Setting up a GENI suite
- Running an Experiment
- Crash and Restart Scenarios
- Operations and Management

4.4.1 Setting up a GENI Suite

First, a *clearinghouse is established*. It is a set of high-availability software services managed by an operations staff.

Second, one or more *aggregates register* with the clearinghouse. This is a trust relationship – they must be certain that the clearinghouse is who it claims to be, and will behave in a responsible fashion, and the clearinghouse must have similar faith in the aggregate's operators. Since this forms a chain of trust upon which GENI will rely, some form of mutual authentication will be used. (For example, aggregates might include managed systems of computer clusters, regional optical networks, or metro wireless networks.)

Third, the registered *aggregates publish their resource information* to the clearinghouse. The exact information is currently undefined, but probably includes lists of resources and up-to-date schedules for resource availability. This information will change continually, particularly if an aggregate belongs to multiple clearinghouses, so the aggregate is responsible for keeping its clearinghouse(s) up to date.

Fourth, *research organizations register* with the clearinghouse. Again, this is a trust relationship – they must be certain that the clearinghouse is who it claims to be, and will behave in a responsible fashion, and the clearinghouse must have similar faith in the research organization.[3] (For example, research organizations might include university computer science departments.)

[3] This step may not be required in all cases. It has been suggested, for example, providing some low level of resources to anonymous users might be a useful characteristic of the system and should not be designed out.

Fifth, *researchers register with research organizations* on the basis of existing or planned experiments. In essence, the research organization vouches that a particular experiment is indeed being planned or conducted, and that this particular researcher is authorized to manipulate the slice that contains that experiment. Note that the clearinghouse itself does not need to authorize individual researchers: that function is carried out by the research organizations. (For example, a graduate student at University X might register with University Y's research organization to join a collaborative experiment run by a principal investigator at Y.)

Sixth, the *NetSE Council establishes policy* about who may access which resources, and under which constraints. This policy is codified into a rule set, and instantiated within the clearinghouse, where it governs all subsequent resource requests.

Seventh, the *clearinghouse federates with other clearinghouses*. Again this is a trust relationship, although its details are unclear at present.

At the end of this stage:

- The clearinghouse contains linkages to its trusted aggregates and trusted research organizations, NetSE Council-specified policy for resource allocation, and linkages to federated clearinghouses.
- The aggregate contains linkages to one or more trusted clearinghouses, and is periodically publishing up-to-date views and schedules for its resources to these clearinghouses.
- The research organization contains linkages to one or more trusted clearinghouses, and lists of authorized researchers for various experiments.

4.4.2 Running an Experiment

Now that the infrastructure suite is up and running, a researcher may perform an experiment. Section 3 provided an impressionistic, tutorial view of this process. The steps below specify the basic steps in this process in finer technical detail.

First, the researcher *acquires credentials* from her research organization. (The research organization must be registered with a GENI clearinghouse in order for these credentials to be useful.)

Second, the researcher sends her credentials to a GENI clearinghouse and requests a slice identifier from the clearinghouse. The clearinghouse validates that this research organization is indeed trusted and provides the researcher with a *globally unique slice identifier*.

Third, the researcher queries the clearinghouse (or other portal) for available resources. The clearinghouse *provides her with resource information*, including current views and projected schedules. This information comes from the lists provided by the clearinghouse's registered aggregates, possibly filtered (according to policy rules) to restrict what she is allowed to see[4]. Included in this information are high-level resource descriptions and contact points (e.g., aggregate managers) for making reservations.

Fourth, the researcher contacts each aggregate manager for a resource of interest with detailed queries about available resources. She presents credentials issued by the clearinghouse that allow her to

[4] Restricting resource visibility at the discovery stage is likely to be largely a convenience (to prevent users from asking for resources they won't be permitted to obtain) as the policy enforcement is expected to take place at the aggregate/component level.

request resources. The aggregate manager may apply locally defined policy based on her credentials (or other parameters) that will constrain the types and amount of resources the researcher can obtain. The aggregate manager responds with RSpec *describing available resources*.

Fifth, the researcher, perhaps using helper tools, makes a *resource reservation* contacting the credential issue service in the clearinghouse to acquire a "signed slice credential" (Certificate). This is then used to get a ticket from the manager of the aggregate/component. The researcher submits a description of the resources desired based on the advertised RSpec, the start time and duration of the reservation, and the Slice ID the resources are to be bound to. The manager can apply its policy and decide whether to issue the ticket or not, considering the requesting researcher and slice.

Alternately, the researcher may use a ticket broker service in the clearinghouse before contacting the aggregate manager). The ticket broker service can apply clearinghouse policy, i.e., whether the researcher gets the requested resource, considering both the requesting researcher and slice and chosen aggregate/component resource. The aggregate manager can then apply its own policy, considering the requesting researcher and slice. The ticket broker service may drop a ticket record in the ticket log, to be used later in forensic searches.

The aggregate manager then marks these resources as booked for that period, and publishes an updated schedule to their clearinghouses.

Sixth, when it is time *setup the sliver*, i.e., ticket instantiation, the researcher sends the ticket back to the aggregate manager and the resources are made available. If topologies need to be created within the aggregate, or other composite forms of action performed (such as starting measurement devices), they occur at this time.

Seventh, the researcher *downloads software images* into her resources, and starts them running. She then debugs them by mechanisms TBD, and collects measurements by mechanisms also TBD.

Eighth, the researcher may choose to *make changes to the resources used* for additional experiment runs. If subsequent *ticket management* (revisions and status) for an existing sliver is needed, the researcher goes directly to aggregate manager to make changes. Alternatively, the aggregate may need to contact the researcher to inform her of component-driven changes to the reservation. For example, if there is a failure.

Ninth, the slice is torn down and its *resources freed*. There can be many triggers for this action, including researcher request, expiration of the slice lifetime, revocation of a researcher's credentials, management decision, etc. This action is coordinated by the clearinghouse, which instructs each aggregate to free up the associated resources, and then records the slice as "ended" in its log files.

4.4.3 Crash and Restart Scenarios

Because GENI is a distributed system, it is possible that some parts of GENI will crash and restart while others will keep running. Loss of state synchronization can be an issue in such cases. GENI should avoid such problems, for example by defining a single, authoritative source for each type of shared, distributed state within the overall system.

Here we consider several scenarios; as design progresses, a thorough analysis will need to be performed. We assume that each important service is provided by replicated software / hardware, but still wish to assure ourselves that GENI will operate correctly through complete failure of such replicated services.

- If a clearinghouse crashes, it must preserve its trust relationships on stable storage; they can be restored in a straightforward manner after it restarts. Similarly if it is the authoritative source of

slice information, it must be very careful to preserve it on stable storage; an alternative scheme might be to tag this information with revision numbers and distribute it more widely through the system. It will relearn current resource availability from the ongoing (soft state) publications from its registered aggregates. [Don't know about federation information.]

- If an aggregate manager crashes, it must preserve its trust relationships on stable storage; they can be restored in a straightforward manner after it restarts. It should also store schedules for future resource booking on stable storage so they can be retried in case of restart. It seems safest to reset all its resources to a known good state, though this might not be required; if it does so, this will remove the running code for all experiments in all slices within this aggregate. When the manager is up and running, it must publish its latest views of resources and schedules to the clearinghouses with which it has registered.
- If research organization crashes, it must preserve its trust relationships on stable storage; they can be restored in a straightforward manner after it restarts. It must also store lists of authorized users on stable storage. Revocation lists should be pushed to the clearinghouse upon reboot to ensure consistency between organizations.

4.4.4 Operations and Management

GENI Operations and Management (O&M) is a system-wide function that keeps GENI resources operating and manages GENI services. Many entities have needs of the O&M functions such as: researchers, the GENI Clearinghouse (which supports external interfaces to GENI O&M), other GENI-federated clearinghouses, university and industry network management (IT) organizations that support GENI users or GENI traffic, Internet Service Providers, organizations that provide aggregates of resources to GENI, and GENI policy makers, such as the National Science Foundation. Three different arrangements can be used to provide O&M functions to these groups:

1) Implement the function entirely within a GENI Operations and Management Organization. For example, the repository for GENI O&M tickets might be implemented on a server owned and managed by the GENI O&M organization. (We'll call this organization GENI Ops from here on.)
2) Contract the function to an outside organization. For example, GENI Ops might contract with a Certificate Authority to manage certificates for GENI researchers. GENI Ops must be able to identify all registered researchers at any point in time, but may not need to manage the certificates directly. (It may be that a few thousand researchers is a reasonable estimate: 2000 PhDs in CS/year for 40 years is only 80k researchers.)
3) Divide the function between organizations to achieve particular policy or design goals, and implement standard functional interfaces and procedures between the organizations. For example, GENI Ops might have a mechanism in place to assign blocks of GENI Global Identifiers to an aggregate when the aggregate wishes to register its resources with GENI, and the aggregate might have a mechanism in place to track and report on resources when they are assigned Global IDs. The aggregate and GENI Ops manage resource identification jointly in order to provide a scalable way to allow researchers to identify components within their slice, even if a component is replaced by the aggregate due to a failure.

Note that the second and third arrangements require trust relationships between GENI Ops and any cooperating or contracted organizations in order to maintain the integrity of GENI O&M functions. They also imply standard procedures and interfaces exist for any data or control signals exchanged

between GENI Ops and the other organizations. Because very few O&M interfaces have yet been standardized in network engineering, GENI O&M will likely have to support many different types of procedures and interfaces, at least initially (as do today's Internet operators). Costs and risks can be reduced significantly if the GENI project advances standards and tools for O&M data exchange and joint management of resources.

GENI O&M will include many of the same functions performed on existing research networks: monitoring, event management, data archiving, managing planned and unplanned outages, network engineering and peering, and protecting GENI from external and internal threats. Because GENI includes slices, rather than individual resources or researchers as a fundamental managed unit, this affects all the "standard" functions to some degree, and also requires additional unique GENI functions (for example, mapping resources to slices and slices to researchers). Because a single GENI experiment can span the scope of control of several different O&M organizations, as well as several clearinghouses, aggregates, and networks that are not part of GENI, new O&M tools and procedures may need to be developed. Functions related to monitoring slices and operating jointly with multiple clearinghouses are currently unique to GENI. The emergency shutdown function that operates on slices, and may require cooperation among several clearinghouses and aggregates to work successfully, is another example of a new GENI function. Policy implementation, which is complex even in networks owned by a single entity, will need careful definition and implementation to accommodate policies from multiple clearinghouses, and GENI's own policy bodies.

The GENI environment places important constraints on O&M solutions. Although it is not possible to list them all, here are some of the most significant:

1) The GENI operations environment will change significantly over the course of GENI development and integration. Both the technology and the scale of GENI will change significantly over its notional life span. If GENI is successful, parts of it will endure even longer. The overall O&M framework must be flexible and allow various specific tools and interfaces to come and go from the design. It must also work when the numbers of managed elements (including users) in GENI grows by several orders of magnitude.
2) GENI O&M must work in an environment where GENI does not own or operate most of the resources used in GENI experiments. Although NSF may provide many of the initial aggregates of resources during early GENI design and integration, the project must operate with many more federated resources in order to function as a national or world-class service.
3) GENI Ops will be distributed throughout the U.S., not localized in a single operations center. Users and federated resources may be distributed worldwide. This is no different than many other O&M organizations, but it has a significant effect on services that must be considered in proposed solutions.
4) GENI O&M must mange experiments in slices that may use experimental protocols that differ from operations data collections and storage protocols. GENI O&M must still be able to track and monitor O&M-related data for these experiments (for example, whether an experiment's slice is active or not).
5) GENI O&M will maintain legacy Internet connectivity with GENI. This is required for some GENI experiments, and for some O&M functions. A GENI gateway, multiplexer, or similar device will be part of each legacy connection. See Section 6 for more on this topic.

5 Substrates, Aggregates, and Slices

This chapter describes the relationship between substrates, aggregates, and slices, and introduces some of GENI's control relationships and basic system interfaces.

Figure 10 depicts the basic relationship between a substrate, aggregate, and slice. They can be viewed as layers of abstraction. Reading from the bottom up:

- A *substrate* consists of physical equipment (*components)* and the management software that manipulates it; some equipment in this layer is not visible to researchers.
- An *aggregate* represents the components in a high-level, researcher-visible form as an ensemble of resources that can be manipulated as a coherent whole.
- A *slice* is a "distributed network of programmable elements" (virtual processors, links, etc.) that is instantiated within and across aggregates.

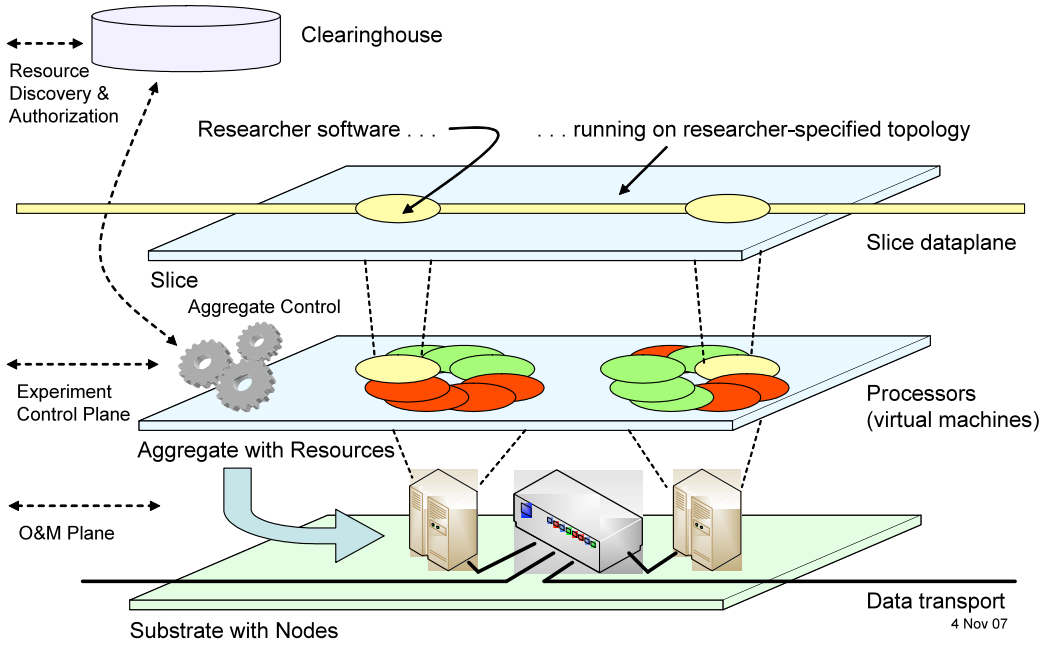

Figure 10. Substrates, Aggregates, and Slices.

5.1 An Illustrative Example

A concrete example may help to clarify this relationship. Let us consider a national backbone network that spans the continental United States and that includes 'programmable routers' at each major city within the United States. We wish to make this backbone available to researchers for experiments, where a given experiment may (a) select a subset of these routers to program, (b) create a topology that links these routers in a desired way, and (c) perform measurements at each router and store the results locally near the router for later perusal.

Substrate. There are many technologies for implementing such a national backbone, and accordingly many different components can be employed. For example, one could link 'routers' via MPLS connections, or via IP tunnels, or via switched Ethernet, or via direct lightpaths, etc. Some methods could be extremely technically advanced (R&D projects); others could be provided by existing commercial services. Each approach would imply a different set of components in the substrate.

Most substrates involve equipment that cannot be seen or directly manipulated by researchers – such as switches of various sorts, multiplexors, and perhaps even optical amplifiers. These substrates also come with their own management systems, which may be quite complex. In the case of major service providers, there is zero chance of convincing the provider to adopt a GENI-unique management system; one must live with whatever they use.

Aggregate. This layer hides the actual substrate details, and presents a simpler, coherent set of component resources to the researcher. In our example, the aggregate may consist of multiple sets of CPUs (or virtual machines), with each set located within a major city, and a set of constraints on how links can be formed between these CPUs.

To be very concrete, the aggregate might make visible 1,000 CPUs available at each city, indicating which are free and which are in-use, and indicate that connections of the such-and-such bandwidths can be formed as desired between various city-pairs. It might indicate that it has 300 free CPUs in San Francisco, 100 free CPUs in Chicago, and 450 free CPUs in Boston, with the following link availabilities: SF-Chicago up to 100 Mbps; SF Boston up to 17 Mbps; Chicago Boston up to 800 Mbps.

Note that it need not reveal much about the CPUs or how they might be interconnected. A researcher might care that the CPUs come from a specific product family, but probably not how they are packaged into racks. In this example, our researcher might care about bandwidth, delay, etc., for a link, but not care whether it is a switched or permanent circuit, or exactly what technology is used to implement data transport. (Of course some other researchers care very much!)

When a researcher requests a slice in this example, she might ask for: 50 CPUs in SF, 50 CPUs in Chicago, and 50 CPUs in Boston, with data transport between cities at 5 Mbps pairwise. The aggregate could then allocate the CPUs and create the topology on demand, e.g., by setting up MPLS circuits and VLANs.

Slice. Once the slice has been established, the researcher has her own "virtual machine" riding atop her own "virtual network." From within the slice, nothing but the slice's entities are visible; there is no hint of an external world. The slice dataplane has been stitched together as per her request so that any information (packet) leaving, say, one of her CPUs in San Francisco be delivered to another of her CPUs in Boston with no evidence of how it was managed in transport aside from underlying transports' delay, jitter, loss characteristics, etc.

5.2 Other Examples of GENI Aggregates

This section provides a series of specific examples of how a wide variety of specific GENI subsystems can be mapped into the aggregate model.

5.2.1 CPU / Storage Clusters

Figure 11 depicts a CPU / Storage cluster in schematic form. It consists of a number of processors and storage devices interconnected by a switch or switch network. GENI slivers running experimental code will run upon CPUs within such clusters, and will need access to their own local storage.

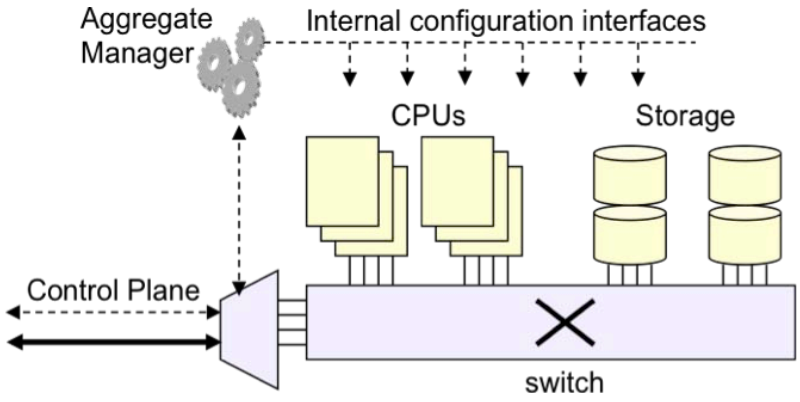

Figure 11. CPU / Storage Cluster as an Aggregate.

When a cluster is managed as an aggregate, a researcher may request resources that include CPUs, dedicated or shared storage, and the aggregate manager establishes the interconnects (e.g. storage area network VLANs) that create the requisite topology and isolation for the slice. As always, some mux function must be provided to multiplex the cluster's link to the rest of the world; in many cases, this functionality may be enabled by the cluster's own switch rather than special-built device.

5.2.2 Programmable Routers

Figure 12 depicts a high-end programmable router as an aggregate. It bears some similarities with the CPU / Storage Cluster discussed above but with two main differences: (a) it will incorporate multiple data transport links rather than act as a network stub, and (b) researcher code may be installed on portions of the mux equipment in order to manage queues, etc.

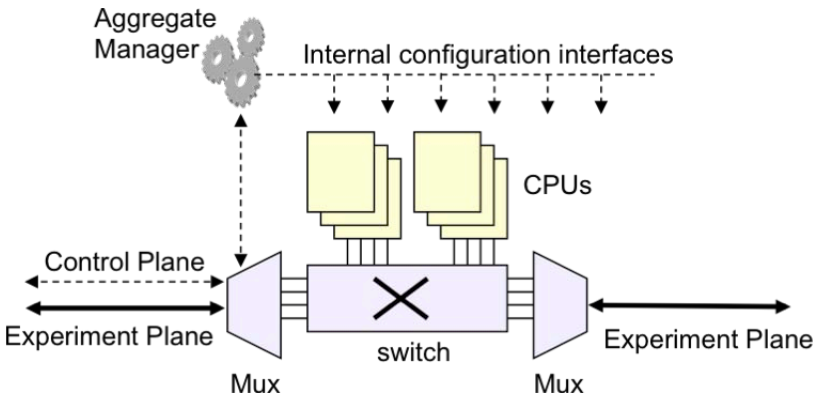

Figure 12. Programmable Router as an Aggregate.

This device clearly needs to be managed as an aggregate, so that coordinated sets of software can be installed on the main CPUs and within muxes (if desired), and so that the central switch network can establish a separate VLAN for each slice. Its muxes must also perform the generic GENI mux

functions, i.e., sharing a link fairly between multiple slices, via trusted hardware or software, even if there is also additional researcher software resident on the same hardware.

Instantiating a slice within a national backbone involves the following steps: (a) allocating computational resources at the 'programmable routers' in various cities and in general configuring each router appropriately, and (b) creating a national topology that properly links these processors into its own virtual network. Some researchers may desire that the virtual network be completely isolated from traffic effects from other slices (and other non-GENI traffic) so as to run repeatable experiments; others may wish to experience the congestion, etc., that is characteristic of a production network. Thus all slices must be isolated from each other and from real traffic, but some may wish to turn off 'fairness' in the multiplexors.

As noted above, a wide variety of technologies can be used to establish a topology on demand, and most researchers will not care exactly which technology is employed within a given substrate.

5.2.3 Sensor / Wireless Networks

Figure 13 depicts a wireless network, possibly a sensor network, which may extend across a building, campus, or metropolitan region. Here the nodes may be connected by radio communications, and each node may be small enough so that it is difficult to multiplex (e.g. not enough memory). Nonetheless such networks fit into the same overall pattern for GENI aggregates.

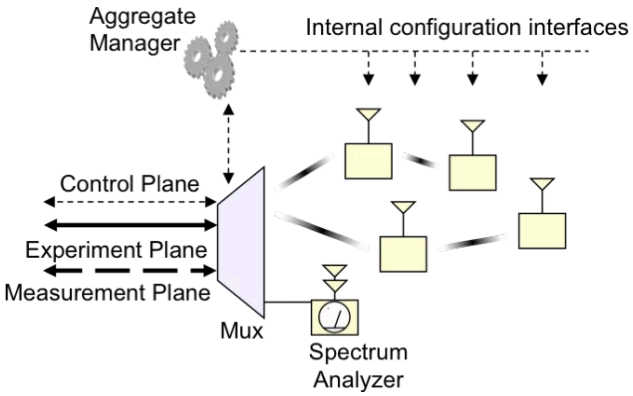

Figure 13. Wireless / Sensor Network as an Aggregate.

In these types of aggregates, researchers may wish to discover and request resources (radio nodes) on a geographic basis, and in some cases will be interested in running experiments on mobile nodes. Such experiment set-up requires its own distinct tools. Researchers may also be interested in RF spectrum measurements that are collected during their experiments, for example to discern radio interference from equipment outside of their experiment, and the aggregate will need to both measure such data and make it accessible to researchers.

Each wireless network will also need a GENI mux where it plugs into the data transport system, with the usual requirements to share the link fairly between isolated slices.

6 Connecting a Slice to a Non-GENI Network

There are various reasons why a researcher might want to connect her experiment to some network outside of GENI, and particularly to the Internet, such as:

- Encouraging users to "opt in" to an experimental service
- Running Internet traffic over an experimental transport system
- Running a experimental Internet Service Provider (ISP) that peers with real ISPs

Each has its own challenges and potential dangers. A particular danger is that the experiment will run amok and attack some part of the Internet, whether accidentally or on purpose. Therefore connection of a slice to the Internet must be undertaken with some caution.

Although this aspect of GENI design is currently murky, this chapter provides an initial idea for what kinds of equipment might be required. It provides somewhat the same functionality as envisioned for the Gateway device in GENI's conceptual design. As described this approach allows for greater flexibility than simply connecting to the Internet; in principle it could connect to arbitrary networks based on a packet or frame structure.

Figure 14 shows one way to connect an experiment (running in a GENI slice) to the Internet. A controlled filter regulates which slices can access the Internet and manages assignment of public IP addresses (a limited resource) to slices. The responsibility for adapting experimental traffic to IP packets falls to the experimenter. In this approach, the Aggregate Control would request public IP addresses and passage through the filter as part of the slice set-up procedure.

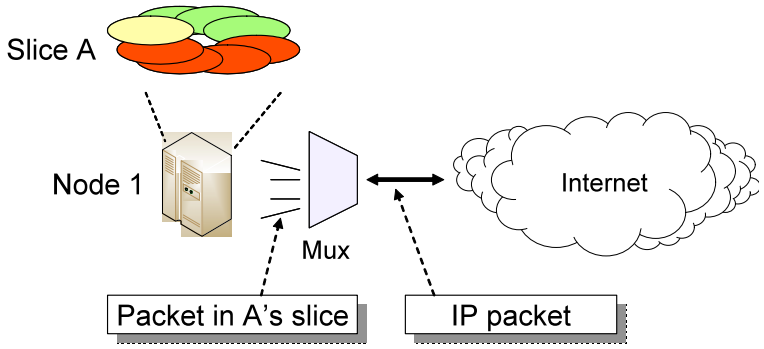

Figure 14. One way to connect an Experiment (Slice) to the Internet.

In some instances it may be desirable for end-users to "opt in" to GENI experiments without having to use any IP protocols at all, e.g., without connecting across the Internet to GENI.

One example is the use of handheld devices within a college campus, or even in a metropolitan area, which may be able to send experimental GENI packets directly within WiFi 802.11 frames. If the WiFi system can shunt such packets to a collocated GENI component, then these packets can directly enter a running experiment's slice without ever being IP packets. (Of course the same approach would also work for a wireline Ethernet.)

7 Slice Requests, Authorization, and Audit Information

The GENI control plane includes a set of core functions supported by all GENI components – supplemented by a variety of services and tools – to allow researchers to discover component capabilities, establish slices and reserve slivers, implement a basic set of controls over reserved resources (e.g., start, stop, return to known state), and perform systemic debugging operations. This section describes a few of the key architectural elements used by the control plane.

7.1 Resource Specifications

A **resource specification**, or **RSpec**, is a data structure describing a component's resources, such as their processing capabilities (such as processor architecture and speed), their network interfaces (including bandwidth and the like), and privileged operations that can be invoked on the component (such as access to instrumentation, protected kernel state, and hardware accelerators). The purpose of the RSpec standard is to give component managers, user services, and end users a common resource vocabulary.[5]

A component owner signs an RSpec—one that includes the right to allocate the corresponding resources—to produce a ticket. Such tickets are "granted" by a component owner, and later "redeemed" to acquire resources on the component.

7.2 Names & Identifiers

Unambiguous identifiers—called GENI Global Identifiers (GGID)—are defined for the set of objects that make up GENI. GGIDs form the basis for a correct and secure system, such that an entity that possesses a GGID is able to confirm that the GGID was issued properly and has not been forged, and to authenticate that the object claiming to correspond to the GGID is the one to which the GGID was actually issued.

A name registry maps strings to GGIDs, as well as to other domain-specific information about the corresponding object (e.g., the URI at which the object's manager can be reached, an IP or hardware address for the machine on which the object is implemented, the name and postal address of the organization that hosts the object, and so on). GGIDs are used to identify slices, users, and components within the system.

The GENI clearinghouse provides a default registry that defines a hierarchical name space for slices and aggregates, corresponding to the hierarchy of authorities that have been delegated the right to create and name objects. This default registry assumes a top-level naming authority trusted by all GENI entities.

7.3 Tickets

A **Ticket** is a "sliver record" that specifies the resources an aggregate/component allocates (or promises) to a given slice. The aggregate/component manager implements the following functions:

1. resource advertisement

[5] An initial description of an RSpec can be found here: http://groups.geni.net/geni/wiki/GeniRspec.

2. ticket creation, or resource reservation
3. ticket management, including revisions and status
4. ticket instantiation, or sliver setup
5. sliver activation, including start, pause, re-start, stop, etc.

Resources are described and requested using a resource specification or RSpec that includes parametric resource descriptions, level of resource commitment (e.g., best effort or assured), reservation duration, and start time.

7.4 Authentication and Authorization

GENI will require a system for identifying users, components, and slices and controlling access based on policies set by the substrate owners and the research community. Several concepts have been proposed for this and there are systems already in use that have similar needs. GENI will need to meet this need in a scalable, efficient manner that can accommodate the many substrates and operational scenarios anticipated. This section highlights a few ideas that have been suggested so far.

GENI will likely have a PKI however its use will not be mandated. For example, resources may be made available to anonymous users or reputation services may be used to make the decision as to whether a resource should be allocated.

One approach to defining GGIDs is to use an X.509 certificate that binds a Universally Unique Identifier (UUID) [X667] to a public key as illustrated in Figure 15. The object identified by the GGID holds the private key, thereby forming the basis for authentication. Each GGID (X.509 certificate) is signed by the authority that created and controls the corresponding object; this authority must be identified by its own GGID. There may be one or many authorities that each implements the GENI control framework, where an authority with the power and rights to sign GGIDs issues every GGID. Any entity may verify GGIDs via cryptographic keys that lead back, possibly in a chain, to a well-known root or roots. See GDD-06-23 for a more complete description of authentication and authorization in GENI.

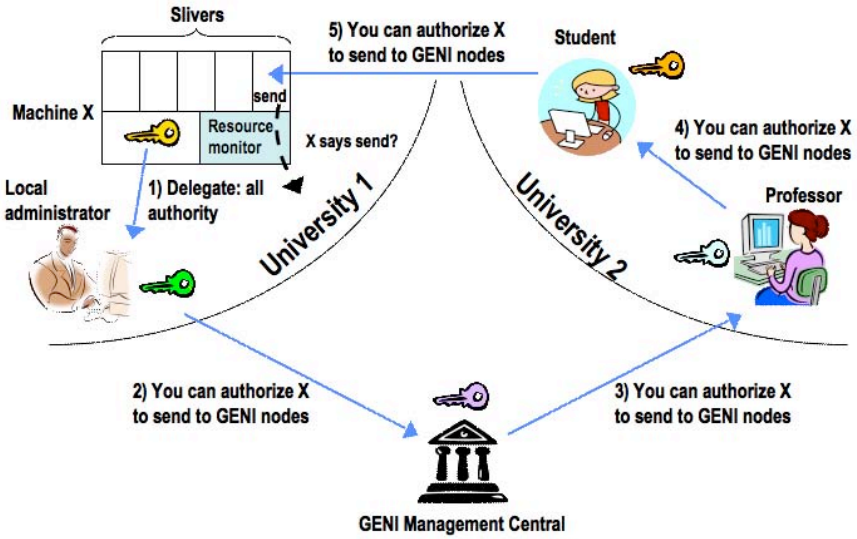

Figure 15. Delegation of Authority in GENI.

8 GENI Tools & Services

GENI will provide tools and services to make the system easier to use. An initial set of tools will be developed as part of GENI. The user community will be encouraged (with GPO facilitation) to develop and make available additional tools. Supporting the heterogeneous technical base and operational models will be a challenge for tool and service development.

Many tools and services will need to be decentralized and secure, and support federation and local site autonomy. Further, GENI will need to support different types of experiments/users:

- beginners vs. expert programmers
- short-term experiments vs. long-running services
- homogeneous deployments vs. heterogeneous deployments

The goal is to make GENI accessible to the broadest set of researchers, including those at places with little prior institutional experience.

The table below briefly introduces the range of tools that will be needed.

Researcher Tools & Services	
Resource Discovery	The GENI clearinghouse will contain a high level registry of components. In addition, specialized catalogs will help researchers find collections of resources that will meet the requirements for their experiments. GENI will support open interfaces for identifying low-level component information making it possible for a variety of resource discovery tools and services to exist.
Experiment Status & Discovery	Portals will be needed for cataloging running experiments that a researcher might build on or connect to. That includes ongoing deployments as well as other artifacts such as canned experiment software, workloads, or faultloads.
Debugging	Experimenters will need facilities, services, and tools for debugging active component code before deployment and tracing failures in experiments. When a sliver does not perform as expected, researchers will need tools to determine whether the host component is functioning properly. For example, virtual CPU resources should provide something like a terminal interface for low-level diagnostics. Experiment-specific diagnostics should be provided by the experimenter.
Slice Management	Slice management tools will push component code out to GENI components in an efficient manner (manual ftp of code to active components will not scale to large systems). Slices will need to tools to bring them active or inactive in a coordinated manner (i.e., the collection of slivers made concurrently active or inactive).
Measurement	Measurement tools will assist researchers to collect system and experiment metrics. Commonly used measurements can be made available, for example, from components hosting virtualized processes or specialized test equipment. Management services can help apply standard meta-data to collected measurements making it easier for researchers to reference and build on each other's results.
Documentation	Tools assisting creation of regularized documentation, for example describing experiments, will encourage reuse of experimental code.

Code Repository	GENI will provide a code repository under the GENI IPR rules to encourage code reuse and rapid sharing of useful innovations.
Operator Tools & Services	
Status Monitoring	GENI will provide high-level status information for the infrastructure suite. Operators of GENI components will make component and system status available for researchers to monitor and build on.
Identity Management	GENI will include services permitting new users and components to acquire identification. The service will support changes in user status (e.g., when students graduate and lose their university affiliation), cross-organizational affiliation (e.g., a researcher from one university with access to an experiment owned by a different university), and offline interactions.
Failsafe	GENI will provide a service for rapidly bringing down experiments that are out of control or under attack. Components or regions of the infrastructure suite will be capable of being isolated to prevent damage from malevolent traffic.
Resource Allocation	Resource allocation tools will configure components and aggregates to meet user resource requests. These tools will generate high- and low-level descriptions of available resources for reservation by experimenters. The tools will be responsible for allocating resources to preserve isolation among slivers.
Policy Management	The NetSE Council will define by the component owners and usage policies. GENI will provide tools for the expression of policies in human-understandable terms and translating those policies to the authentication mechanisms used in the GENI control plane. The policy management tools will need to support local policies (e.g., “don’t allocate more than 10% of the component capacity to a single experiment”) and global ones (e.g., “a graduate student can reserve no more than 10 CPUs”).
Legacy Internet Services	A minimal set of ISP services will be provided to facilitate IP-based experimentation such as DNS, HTTP, NAT, BGP, and address management services.

9 Why Clearinghouses?

Clearinghouses provide a number of potential benefits for both the baseline GENI and the broader GENI 'ecosystem.' In this section, we briefly catalog their baseline utility plus a number of ways in which they can enable transitions to larger GENI ecosystems.

Clearinghouses are an operational convenience (particularly needed for bootstrapping the infrastructure suite) and a container for several important services, many of which are critical to the functioning of GENI. The importance of these services make clearinghouses likely to run into scaling challenges as the infrastructure suite grows and become candidates for denial of service attacks. Distributed instantiations of some clearinghouse services will likely be desirable where practical. Additionally, parts of GENI – or other facilities wishing to interoperate with GENI – may have reason to run a reduced set of the clearinghouse functions described in this document. For this reason, clearinghouses should be viewed as design elements, which are likely to evolve or even disappear as the infrastructure suite grows and the functions migrate elsewhere.

Many of the roles clearinghouses are involved in – federation, resource discovery, usage policy administration – are not yet well defined and until the mechanisms become clear one can't be sure if the clearinghouse will have a central role (e.g., operating the mechanism), a peripheral role (e.g., a well-known point of entry to the mechanism), or no role at all. Nevertheless, the functions we allocate to clearinghouses will need to be performed somewhere and it is worth exploring what they are and why they are important.

Organize Trust Relationships. The basic use for a clearinghouse is to organize and manage trust relationships – on the one hand, between a clearinghouse and research organizations, and on the other between the clearinghouse and aggregate operators. This simplifies the potential web of N x N trust relationships between research organizations and aggregate providers, who may have little a priori reason to know or trust each other, into a more scalable 2N relationships.

If a Public Key Infrastructure (PKI) approach to authentication and authorization is employed, a clearinghouse is a natural root for the necessary certificates. Since the system organization permits multiple clearinghouses, it will be possible to have many different roots – or indeed to have some clearinghouses that use PKI techniques but others that use different, non-PKI techniques. This flexibility avoids the problem of 'lock in' to a single, unchanging security framework for authentication, etc.

Support Private Versions of GENI. If a commercial company wishes to set up and operate its own GENI-compatible infrastructure suite, it can easily run its own clearinghouse which can then implement the access control, policies, etc., that make sense for that organization. The same functionality may also be useful for various branches of the Federal Government, e.g., there might be DOE and DARPA clearinghouses in addition to the NSF clearinghouse.

Provide a Framework for Federation. Clearinghouses provide a natural way to federate entities that don't share a common 'root' organization, e.g., between different governments' research organizations, different companies, etc. For example, federation between US, EU, and Japanese portions of a GENI ecosystem could be naturally accomplished via setting up a clearinghouse for each national portion, then federating these 3 clearinghouses. A similar approach could be employed for federating one commercial company's private GENI infrastructure with that of another company.

Support Mixed Public / Private Resources for GENI. Some organizations may wish to provide some portion of their total resources for use by NSF-sponsored researchers but retain another portion

for their own use (only), or for use by themselves and their own selected partners. For example, a private corporation might support 10,000 CPUs in GENI cluster, and make 2,500 of them available for public research on a not-to-interfere basis.

This could be easily achieved by setting up two different clearinghouses (NSF, private), and linking the relevant resource aggregates into these clearinghouses. An aggregate might publish some fraction of its resources to the public NSF clearinghouse, but publish the full set of resources to its own private clearinghouse; thus any NSF-sponsored researcher could use the public resources, but any private researcher could use any of the resources. Note that this arrangement would support two different sets of authenticated researchers, and perhaps two different authentication techniques (which could be handy for organizations that deploy high-assurance authentication mechanisms to authorized users).

Provide Multiple Resource Allocation Mechanisms. The basic GENI clearinghouse is expected to manage resource scarcity by NetSE Council-mandated policy. However one can easily envision many different ways to allocate resources, e.g., by a variety of market mechanisms including use of tokens, actual cash payments, etc. These different mechanisms may all be supported at the same time, for the same set of underlying resources, by simply establishing multiple clearinghouses. Each can then employ its own resource allocation mechanism: for example, one might use policy, another might use a credit scheme, etc.

Provide a Graceful Transition Path to 'Commercial GENI'. As mentioned above, clearinghouses provide a good transition path towards market-based resource allocation. Thus it appears relatively easy for a commercial entity to set up and operate its own, commercial clearinghouse that charges users and in turns pays aggregate operators. The GENI clearinghouse approach also makes it possible for multiple commercial entities to simultaneously be open to the same sets of users and aggregates, thus enabling competition. Finally we note that such commercial activities could take place in parallel with NSF-sponsored activities, that is, researchers could obtain resources for free while at the same time commercial users had to pay for them. In short, this approach provides a graceful transition path to 'commercial GENI' should the desire ever arise.

10 GENI Instrumentation and Measurement

GENI will support multiple, simultaneous, diverse experiments from physical to application layer with measurements required throughout the stack.

10.1 GIMS

The GENI instrumentation and measurement system (GIMS) is a native GENI resource enabling data collection, storage and analysis. GDD-06-12 discusses a high-level framework for describing basic data types and access methods as well as system design considerations.

Requirements for measurement in GENI:

- Ubiquitous deployment,
- No (or at least measurable) impact on experiments,
- Extensibility (*i.e.,* the ability to add new instrumentation and/or new measurement synthesis capability),
- High availability (at least as available as GENI systems on which experiments are conducted),
- Large capacity (*i.e.,* the ability to support a diverse set of simultaneous activities from a large number of experiments),
- The ability to measure detailed activity with high accuracy and precision from physical layer through application layer (includes the ability to calibrate measurements),
- The ability to specify required measurements for an experiment in a slice (using either standard measurement types from a library or defining user specific measurements) and then having these measurements initialized in the infrastructure when an experiment is activated,
- Access control (*i.e.,* the ability to specify what data is available from a particular device or collection of devices, to whom, and for how long),
- A large, secure central repository in which collected data can be anonymized and made available to users,
- A "data analysis warehouse" where tools for visualizing, interpreting and reporting measurement data can be developed and made openly available.

It should be clear from the list above that the measurement and instrumentation needs for GENI will be quite demanding. Future design activities in GENI will include defining a measurement architecture that meets the requirements above *and* includes the ability to take advantage of existing instrumentation wherever possible.

10.2 Space, Time, and GENI

Two particularly interesting challenges having to do with instrumentation and measurements in GENI are those of *space* and *time*. GENI experiments will require knowledge of the location of components and the time when things occur. The need of precision time and space information will vary based on the application.

Precision in location information may be particularly important when components are mobile. Analogously, precision in timing information may be important when events in two locations need to be correlated.

Information about location may range from which city to which lamppost. GPS coordinates will do for many instances but are unlikely to be sufficient for all cases, for example, logical locations such as rack number or locations not on the surface of the earth such as satellites.

Precision on timing may vary from milliseconds (e.g., to coordinate with human activity) to much finer (e.g., to coordinate high-speed packet transmissions). GPS can provide a foundation for timing but the precision of GPS timing is limited. Future work will be needed to evaluate whether GENI will need to include high-precision clocks and, if so, how many and where.

Appendix A What should an Aggregate do to fit into GENI?

As GENI is still early in its design and prototyping stage, many aspects of the system are not yet defined. GENI prototype and integration work will be performed in parallel with design work, and practical experience with early GENI aggregates and will have a strong influence on GENI"s design as it evolves.

At this early stage of development, the following guidance is suggested for how aggregates should fit into GENI. Note that **all mechanisms must be documented online** in sufficient detail so that they can be used by remote researchers without aid!

- **Programmability.** Aggregates must include components that are configurable or, ideally, programmable, i.e., into which researcher-provided software may be installed, debugged, and run. Since researchers will in general be far away from the hardware, these operations must be supported remotely. Since researchers may download almost arbitrary software, it must not be possible for software to damage the hardware on which it runs. There must be some mechanism by which access can be secured, i.e., only duly authorized researchers are allowed to install software.
- **Virtualization.** Aggregates must support virtualization or other ways of sharing, so that multiple researchers may run separate experiments simultaneously within the aggregate. This may be accomplished by a variety of techniques, such as providing virtualization in every component, being able to give each experiment its own dedicated sets of components, etc. The end result, however, should be that multiple experiments can run in parallel on the aggregate, each with its own software image(s) and each with good isolation from the others so the activities of one experiment do not affect the operation of any other experiment.
- **Internal Topology Management.** Many aggregates will be able to establish internal topologies on demand, so that a slice can be created with its own, slice-specific topology. There must be some means by which this function can be controlled remotely, and a pathway by which this functionality can be merged into GENI's control framework as that part of the system becomes defined.
- **Federation.** In the near future, the aggregate must be able to publish its available resources to a clearinghouse, and to participate in researcher authentication and authorization. These interfaces are currently undefined, so at present aggregates cannot comply with this requirement. We recommend that the aggregate set aside a dedicated computer that will eventually perform this function, e.g., a PC running Linux, and ensure that the aggregate at least have some private, internal way of determining which resources are currently available, allocating resources, etc. As this interface becomes defined, we expect a free reference implementation of aggregate manager software to be published. It can then be installed on the Linux PC, and linked in some fashion to the aggregate's internal resource allocation mechanisms.
- **Connectivity.** The exact means by which an aggregate will be "stitched into" GENI will vary greatly, depending on the aggregate itself, and on GENI's current stage of prototyping. For example, a regional optical network acting as an aggregate may well connect into GENI at an optical layer, while a cluster of computers might connect via a conventional Internet link. For GENI's earliest days, we recommend that all aggregates provide a means by which they can be "stitched into" the earliest form of GENI through Internet connectivity, e.g., via a Virtual Private Network, Network Address Translation (NAT) functionality, etc. This mechanism must include some way that this interface can be multiplex data transport for a number of simultaneous experiments.
- **Instrumentation and Measurement.** Every aggregate must include some form of instrumentation and measurement, with the near-term possibility of making per-experiment measurements available to remote researchers as this part of GENI becomes defined.

- **Operations and Management.** Every aggregate must include some form of Operations and Management (O&M) system by which the status of its components, etc., may be monitored and managed. Ideally this system will be able to interact with a GENI-wide O&M system, at least by publishing its current status information to the GENI O&M system that can then make it visible to researchers. This system must include an "emergency shutdown" mechanism by which a slice can be immediately brought to a known, benign state (e.g. within 1 second). Possible mechanisms include shutdown of connectivity to the outside world, computer reset, etc. In the early stages of GENI it is acceptable if emergency shutdown also affects other slices within the aggregate (e.g. terminates them abruptly).