GENI: Guiding Design Principles

GENI Planning Group
April 29, 2006

1 Introduction

Designing GENI is a multi-year process, with the 10-January-2006 *Project Execution Plan* (PEP) expected to be the first of a series of PEP snapshots. Even after construction begins, GENI will continue to evolve as new technologies become available and new user requirements come into focus. This naturally leads to a question of what capabilities GENI should and should not include. While the definition of GENI will continually change, we expect a core set of design principles will guide that decision process, and a set of value judgments that will define GENI's ''center of gravity.''

Intimately linked to the design process is the question of how priorities are established among the many possible uses of GENI. Understanding these priorities will be important to any policy committee charged with deciding how GENI resources are allocated to researcher projects. Such allocations are primarily based on research merit, but they are also influenced by how closely the proposed usage matches GENI's intent.

GENI's "center of gravity" cannot be characterized with a simple black-and-white statement. It is obviously in everyone's interest for GENI to be as useful to as many researchers as is possible—both in terms of functionality and in terms of capacity—but it is a matter of how to best manage and allocate limited resources (i.e., construction budget and facility capacity). This document attempts to define a clear set of principles and priorities that shape such decision processes. It is an attempt to capture the principles that have guided the planning group up to this point, and with community feedback, to serve as a working document to direct the process as we move forward.

2 Research Scope

The scope of research GENI is expected to support is outlined in introductory sections of the PEP. Broadly, GENI is an open, large-scale, realistic experimental facility that will revolutionize research in global communication networks. Its central goal is to *change the nature* of networked and distributed systems design: creating over time new paradigms that integrate rigorous theoretical understanding with compelling and thorough experimental validation. In doing so, we expect the intellectual merit of the research conducted on GENI will be of three general types:

- **Science:** GENI will allow us to experimentally answer questions about complex real-world systems, giving us an increased fundamental understanding about their dynamics, stability, evolvability, emergent behaviors, and related matters.
- **Architecture:** GENI will allow us to evaluate alternative architectural structures, and reconcile the contradictory goals a network architecture must meet.
- **Engineering:** GENI will help us evaluate engineering tradeoffs, and test theories about how different architectural elements might be designed.

GENI will also have broad impact by leading to artifacts that provide value to society. This might result in a better Internet (e.g., one that is more secure, available, manageable, usable, and suitable for computing in the next decades). It might also serve to catalyze the distributed digitized world (e.g., help provide personal control of your personal data, or support real-time sensing of the physical world). GENI might also provide value to science, for example, by producing enhanced services that improve the scientific process.

Note that giving a full scientific rationale for GENI is beyond the scope of this document. This note outlines a value proposition for GENI—based on widely accepted criteria for evaluating research on networked and distributed systems—but it does not anchor its arguments in a specific set of research questions. Again, we refer the reader to the first two sections of the PEP for example questions that researchers should be able to address on GENI. While the exact research questions will likely drive the details of what technology to include in GENI, we believe the discussion presented in this document is orthogonal to such questions: this document is more focused on capabilities that bring value to the research methodology.

3 Core Requirements

GENI comprises a collection of substrate hardware resources, including nodes, links and edge subnets. Each experiment using GENI will run on some subset of the GENI resources. We call the substrate resources bound to a particular experiment a *slice.* GENI includes management software that is used to allocate resources to slices, embed slices in these resources, and ensure that slices do not interfere with each other. The PEP identifies the core set of capabilities that GENI must support, which we summarize here:

- **Sharability and Reusability:** To be cost-effective, GENI must be a shared facility that can simultaneously be used to support multiple experiments running on behalf of many independent research groups. Virtualization is a key technology that supports this goal, although its use is not intended to exclude the possibility that other slicing strategies may also be employed, or that “bare metal'' can be allocated to some slices for a period of time.
- **Experimental Flexibility:** GENI should give each experimenter the flexibility needed to perform the desired experiment. First and foremost, this means that each component should be programmable, so that researchers are not limited to experimenting with small changes to pre-existing functionality. It also means that components must provide an appropriate level of abstraction, or said another way, must be virtualized at the right level.
- **Ease of User Access:** GENI must make it easy for a broad mix of users to ''opt in'' to the services that it provides, thereby bringing real user workloads to the facility. Doing this includes providing mechanisms to make it easy for users to join one or more experimental services running in GENI, and to transparently fall back to the legacy Internet whenever the experimental network cannot provide the requested service. It also means experiments must be able to run continuously, as no user will want to use a service that is up for only a limited period of time each day.
- **Controlled Isolation and Managing Collaboration:** GENI must provide boundaries *between* experiments. An obvious starting point would be to assume that different experiments embedded within GENI are isolated from each other. However, this is too simple. GENI must also support controlled *interconnection* of experiments, so that researchers can build on each other’s work.
- **Facility Extensibility:** GENI must be extensible so that it can incorporate a wide class of networking technologies, including hardware that *does not yet exist*. This capability will allow GENI to remain useful over a much longer lifespan, support GENI's role as a low-friction vehicle for deployment of new technologies, and foster close collaboration between “device researchers” and “systems researchers.”

- **Global reach:** GENI must have as wide a reach as possible. This is necessary to support experimentation at scale, and to maximize the opportunity to attract real users. Access cannot be limited to only those few sites that host backbone nodes.
- **Instrumentation and Data Analysis:** The GENI substrate, along with all the architectures and services deployed on it, must be heavily instrumented. The generated data must be collected and archived, and analysis tools developed.
- **Federation and Sustainability:** To ensure the sustainability of GENI, it should be possible for participating institutions to contribute resources in return for access to the resources of the GENI as a whole. In general, it should be possible for new research communities to "opt-in" by connecting their purpose-built networks (including dedicated transmission pipes and sensor networks) into the GENI substrate and running their applications and services in a slice of GENI.

Note that there are many other general requirements that are not specific to GENI, but apply equally well. For example, GENI must be as secure, robust, easy-to-use, and efficient as possible.

4 Tensions

Few of the requirements outlined in the previous section are absolutes; there are intrinsic tensions among most of them. This section identifies several of these tensions, and offers guidance as to how conflicts should be resolved.

4.1 Sharability vs Flexibility

Balancing sharability and flexibility is one of the most fundamental challenges facing GENI. On the one hand, virtualizing the underlying hardware allows many researchers to share a common set of resources, and can increase flexibility by synthesizing multiple and/or higher-function virtual environments from a single physical resource. On the other hand, virtualization potentially hides certain capabilities and properties of the underlying hardware, making the facility less ``realistic'' than if a researcher had the resource all to him or herself. In some cases, certain resources may be hard (expensive) to virtualize.

On the surface, this particular conflict is easy to resolve—GENI should provide the lowest level of virtualization that the technology allows and users require. Any given component may not provide the desired level on day one, but advancing the state-of-art in virtualization over GENI's lifetime is an ongoing objective. Higher levels of abstraction should also be retained for those experiments that do not want to be exposed to low-level details, but virtualization should be pushed as "low" as possible.

However, there will be those that argue that any amount of virtualization is too much, and that their research requires access to "bare metal." This might be because of the need for access to a component-specific feature, or because virtualization introduces too much unpredictability in timing measurements. GENI does not preclude the possibility that raw hardware elements can be allocated to some slices—as long as there are mechanisms consistent with the management framework for allocating and reclaiming such hardware—but doing so is likely to happen only if a compelling cost-benefit argument can be made. This is a question of how many researchers will benefit from such access, and how central is the research to GENI's core value proposition (as discussed next).

4.2 Realism vs Reproducability

A second important tension exists between making GENI as realistic as possible—where realism includes factors like physical scale, being subjected to real user workload, spanning organizational boundaries, and so on—and creating an environment in which researchers can run reproducible experiments. On this point, GENI is clearly being designed to favor realism over reproducibility, as the former is the capability most lacking in today's experimental toolset. However, this does not preclude the possibility that a researcher might request a slice that includes only a set of nodes and links that can be fully isolated from both other traffic and users, and then subjected to experimental control (e.g., selectively taking a link up or down). GENI's isolation mechanisms should be sufficiently robust to make this possible, and to the extent they are not, GENI should provide enough feedback about what resources a slice actually receives to enable researchers to evaluate the validity of their results.

Thus, realism vs reproducibility is not so much a question of what functionality GENI can provide, as it is an issue of resource allocation. We expect a certain fraction of GENI's resources will be set aside for users interested in reproducible results, but that the lion's share will go to experiments that value GENI's realism.

A similar tension exists between realism and containment, where containment is roughly equivalent to reproducibility, except focusing on the experiment running in a slice not being able to affect the larger Internet rather than the other way around. For example, containment would be important to a researcher experimenting with new approaches to security threats. As with reproducibility, it should be possible to cordon off a subset of GENI resources for the sake of certain experiments.

We note, however, that "realism" is in the eye of the beholder. In the previous subsection realism was used as an argument to bypass virtualization, where here it is used to denote the unpredictable nature of communication networks. The critical question is what aspects of realism are most important. Our position is that any realism that could be achieved in a more limited facility (e.g., in a single researcher's lab or a smaller more special-purpose testbed) is of less priority than realism that is unique to GENI, where GENI's uniqueness primarily corresponds to realism achieved by (1) wide-spread deployment, (2) a diverse collection of network technologies, and (3) support for real user traffic. These three properties effectively define GENI's value proposition.

4.3 Technology Development vs Architectural Design

We expect an on-going tension between researchers wanting to use GENI to test and evaluate new networking technologies, and those wanting to evaluate new architectural designs that (among other things) take the capabilities of new technologies into account. The former tend to focus on single components, while the latter must take a more comprehensive (end-to-end) perspective. GENI's policies should favor architectural research (broadly defined) that takes advantage of the fact that it spans a diverse collection of hardware resources.

We note, however, that there is value to component developers in being able to evaluate their technology in the context of end-to-end architectures and under the realistic workloads GENI is expected to generate. GENI should allow such technologies to be plugged into the facility once they are mature enough to support GENI users, but we expect early-stage technology development (both hardware and software) to happen outside of GENI. (There is also likely to be a transition path whereby a new technology is made available to early adopters in a subset of GENI.) To make a case for adding a new component to GENI, it will need to support the interfaces defined by the management framework, be sufficiently programmable to give researchers the flexibility they need, and to the extent possible (see the above discussion) be sharable by multiple slices.

4.4 Networking vs Applications Research

GENI is neutral about what level of the protocol stack researchers focus their efforts, and so does not draw sharp lines between network "architectures," "services," and "applications." Any research that benefits from wide-spread deployment, diverse network technologies, and support for realistic network conditions should be supported.

The critical point-of-tension is that GENI is designed to support research in networking and distributed systems—as opposed to simply providing bandwidth to end users—yet it also benefits from traffic generated by real users. It will be necessary to evaluate the research value of traffic generated by a given slice to decide if allocating resources to that slice is warranted. We can imagine three ways in which a research group justifies the value of traffic they are carrying: (1) making traffic traces available to other researchers, (2) providing a novel network service whose efficacy needs to be evaluated, and (3) offering to run on top of a novel network architecture.

Note that new communities that find value in some capability of GENI—or some innovative service deployed on GENI—are free to augment GENI with enough capacity to carry their user traffic, independent of other research considerations.

5 Engineering Principles

In addition to offering guidance to design and resource allocation decisions, we identify a set of engineering design principles that we consider essential to the successful construction of GENI. Many of these are related to the significant role software development is expected to play in GENI.

- **Start with a well-crafted system architecture**. The more complex the factorization of the system into a set of component building blocks, the greater the risk that the inter-dependencies among components will become unmanageable. The success of the Internet itself can be traced in large part to the fact that its architecture allowed components to evolve independently of each other. The GENI architecture is guided by the same design principle, whereby independent technologies can be plugged into the management framework with virtually no dependency on each other, and independent distributed services to be developed without heavy-weight coordination.
- **Leverage existing software**. While some aspects of GENI will need to be implemented from scratch, we expect to be able to leverage significant amounts of existing software. It is essential that we take advantage of such software, and to the extent possible, do so in a way that allows us to also leverage the support systems already in place to keep this software up-to-date. Even adapting, rather than directly using an off-the-shelf software package takes time, and raises the question of who now supports the modified package. Similar arguments favor commercially available hardware.
- **Build only what you know how to build**. Because software is plastic, there is a tendency towards feature creep; it is easier to specify the features a system "must" have, than it is to make those features work together. Left unchecked, this can result in systems that are simply too complex to work. There will be those who will complain that we are doing too little, beyond what we already understand. Our answer is, exactly, but the *synthesis* of these elements is revolutionary.

- **Build incrementally, taking experience and user feedback into account**. It is a well known result of computer science research that in software or hardware construction efforts, errors are cheapest to fix when they are caught early. The best way to do that is to put the system into active use at the earliest possible moment, gain live experience with the system, and incrementally evolve the system based on what you learn.
- **Design open protocols and software, not stovepipes**. A huge point of leverage for us, versus other examples of large scale software systems construction, is that the users of the facility—the computer science research community—are themselves capable of fixing and enhancing the system, if we give them the right tools. This is unique to the case where we build systems for ourselves, versus building systems for other people; project meltdown is much more likely if the result is take it or leave it. We aim to build a system that continues to evolve in meaningful ways after GENI construction is complete. All of the successful examples of large-scale systems being successfully delivered by the computer science research community have the property that they continued to be modified by their user community, well after initial delivery.

Authors

This statement was prepared by the GENI Planning Group:

Larry Peterson, Princeton (Chair)

Tom Anderson, Washington

Dan Blumenthal, UCSB

Dean Casey, Ngenet Research

David Clark, MIT

Deborah Estrin, UCLA

Joe Evans, Kansas

Dipankar Raychaudhuri, Rutgers

Mike Reiter, CMU

Jennifer Rexford, Princeton University

Scott Shenker, Berkeley

John Wroclawski, USC/ISI