Mutualistic Software Services (MSS) for Large-Scale Networks

A Project

By John P. Quan

Presented to the faculty of University of Alaska Fairbanks

In Partial Fulfillment of the Requirements of

MASTERS IN SOFTWARE ENGINEERING

Fairbanks, Alaska
April 2012

Mutualistic Software Services (MSS) for Large-Scale Networks

A Project

By John P. Quan

RECOMMENDED:

Advisory Committee Chair	Date
Advisory Committee Member	Date
Advisory Committee Member	Date

APPROVED:

Dept. Head, Computer Science Department	Date
Dean, College of Science, Engineering and Mathematics	Date
Dean of the Graduate School	Date
Date	

Abstract

Think of a large-scale network (LSN) as a geographically separated collection of servers, or server clusters, interconnected by the Internet to achieve some common purpose, such as communication, computation, or experimentation. Several experimental LSNs have taken root across the globe, which span international boundaries through joint collaboration to develop a secure, robust global Internet. For instance, the Future Internet Research and Experimentation (FIRE) program aims to boost European innovation and its competitive role in defining future Internet concepts. It creates an open research environment, which gives researchers the opportunity to conduct large-scale experiments. FIRE will eventually incorporate users at all levels. It has identified two key challenges: security and federation (which includes governance, sustenance, and incentives). Similarly, Japan's Generation Network 2plus (JGN2plus) is an overlay and service platform for network control and measurement, designed to encourage advanced applications and international collaboration. It aims to foster research partnerships with industry, academia, government, and regional organizations. Within the US, the Global Environment for Network Innovation (GENI) project is the largest effort in this area.[1] GENI has four control frameworks: ProtoGENI, PlanetLab, ORBIT, and the Open Resource Control Architecture (ORCA).[2] This project centers on creating MSS Version 1.0 on the ORCA control framework (CF) as the first step in the following successive goals.

My first goal was to create a means by which the GENI ORCA CF users, such as developers, experimenters, and researchers, can share and track software services for use within ORCA. As a proof of concept, MSS Version 1.0 focuses on the "middle-mile" and "last-mile" of service delivery. My second goal is to create an infrastructure on which all GENI CFs can share services in exchange for resources. After successfully applying MSS to GENI, my ultimate goal is to expand this concept so that other LSNs may take advantage of the mutualistic model by offering valuable services, with computer security as the flagship service, in exchange for resources.

The concept of sharing services in exchange for resources is hardly new. For example, Amazon and the Apple App Store have a centralized delivery system in which customers exchange resources (money) for services (applications and products). In addition, peer-to-peer networks such as LimeWire and Kazaa use a decentralized delivery system in which members exchange resources (videos and music) for resources (other videos and music). MSS is different from both of these systems in several ways.

First, the planned architecture for MSS uses a hierarchical tree-like distribution, which flows centrally from the root at the GENI Maintenance Operations Center (GMOC), then branches through each CF origin to its affiliates, and finally flows to the users as leaves. The second difference relates to the first in the middle-mile and last-mile of service delivery. In the middle-mile, each node in the tree is a client of its parent server, and so a server is a client of a server, is a client of a server, and so on. This means that each server must authenticate with the originating CF, and each user must authenticate with his or her parent CF in order to receive services. In the last-mile, the required number of servers at each node decreases like the diameter of a tree branch from many servers at the origin to a CF residing on a single server (e.g., laptop and desktop computers) that a single individual can donate. This single server model upon which all ORCA and MSS systems must reside did not exist before this project. Finally yet importantly, GENI experimenters "pay" for GENI by hosting a CF and sharing resources in the form of

[1] Quan, J., Nance, K., & Hay, B. (May/June 2011). *A Mutualistic Security Service Model: Supporting Large-Scale Virtualized Environments*. IT Professional (IEEE), 18-23

[2] BBN Technologies. (March 30, 2012). *GENI Spiral 4.* GENI. Retrieved on April 7, 2012, from: http://groups.geni.net/geni/wiki/SpiralFour

virtual machines, which means GENI's basic unit of currency is the resource. If one considers the software products of these experiments to be services, then offering services to the public in exchange for donated resources provides the payment GENI requires to grow.

This first version of MSS incorporates a lightweight Linux, Apache, MySQL, and PHP (LAMP) infrastructure that ORCA owners can use to advertise and deliver services. In doing so, it addresses the fundamental, mutual need between owners and users: owners require vast amounts of resources to meet their goal of conducting at-scale Internet experiments, and users, experimenters, and researches require software services to conduct experiments and business. MSS leverages the hierarchical GENI structure to establish a distributed service delivery system. When a business or university donates a portion of its resources to GENI as an affiliate, it receives all or part of its sponsor's services, which organizational members can then access.

Acknowledgements

I thank Dr. Brian Hay, Department of Computer Science, University of Alaska Fairbanks, Dr. Kara Nance, Department of Computer Science, University of Alaska Fairbanks, Dr. Jon Genetti, Chair of the Department of Computer Science, University of Alaska Fairbanks, and Dr. Peter Knoke, Department of Computer Science, University of Alaska Fairbanks for their guidance, technical knowledge, and support. Most importantly, I thank my wife Megan for her love and support, without which I could not have completed any of this.

Project Conventions

This project consists of the following volumes, in which each page has a header that reflects the volume, and which are numbered and designed to be self-contained documents:

I. Software Requirements Specification for Mutualistic Software Services (MSS)
II. Software Architecture for Mutualistic Software Services (MSS)
III. Software Design for Mutualistic Software Services (MSS)
IV. Software Testing for Mutualistic Software Services (MSS)

Table of Contents

Challenges

Mutualistic Software Services began in my mind's eye as a simple solution to two seemingly disparate problems. After reading about a series of computer attacks on the United States in the first decade of the 21st century, I began to wonder how one might curb such incidents when it seems that many Americans are oblivious to the risks of surfing the Internet. During this time, I worked as a GENI research assistant at the University of Alaska Fairbanks (UAF) developing federation incentives to increase GENI membership. This is when the idea of offering a security service in exchange for computer resource donations from the public occurred to me. After realizing that no use case exists for this arrangement, I set about devising such an infrastructure. My abstract portrays my goals for this project as beginning with a small scope, which progressively grows until finally LSNs offer computer security, among other services, in exchange for resources. In reality, I pictured the "grand vision" first, and then shrunk the scope in steps until I finally developed a plausible beginning for MSS.

This project presented several challenges throughout its course. Some of these hurdles included:

- creating a CF cluster that resides on a single server
- developing a hierarchical service delivery system in which the client has no access to the sponsor's file system and the end users only have the Secure Shell (SSH) port 22 open
- pushing database updates to client sponsors
- incorporating heartbeats into Version 1.0 with no CF support
- applying a recursive attribute tree that describes the services to the file system directory.

The single greatest challenge was developing the single-server CF cluster. A canonical ORCA installation resides on at least two, but typically several computers, and condensing all of the hardware onto one computer took me several months to accomplish. Fortunately, I was able to apply this to one of the UAF GENI goals, which is to set up a GENI CF at remote locations across Alaska. Currently, Barrow, Alaska hosts the first "remote server," on which a working ORCA installation and MSS Version 1.0 harmoniously coexist on one computer. The remote server works by virtualizing the router ORCA prescribes for Network Address Translation (NAT), and I thank Dr. Brian Hay for applying his networking expertise to modify the ORCA method of NAT in a way better suited for a single-server application.

The second difficulty was to develop a hierarchical service delivery system. In MSS, the sponsor controls service delivery by using a proxy on each client. This is a user specifically designated to push MSS services using remote synchronization (Rsync) with SSH, and the proxy serves two purposes. Primarily, sponsors will deliver new services upon a client administrator's request if the client computer is donating resources to GENI. Secondarily, this allows sponsors to apply updates to downloaded services without the client having to initiate the action, which later may provide an avenue for a security service to keep the client computer's virus definitions up to date. One might infer that sponsors also can delete services from unruly client sponsors, but this is not the intent of MSS. Once a client downloads a service, it belongs to the client. Fortunately, I also was able to use remote SSH commands to solve the third challenge of pushing database updates to the client sponsors. MSS uses the same user proxy to send MySQL statements remotely by allowing the user proxy to access the ORCA, Users, and Services databases.

The fourth barrier I faced was incorporating heartbeats, which are packets of information sent from participating ORCA clusters to the ORCA Remote Actor Registry, and that contain identifying information flagging the resources as available to GENI experimenters. Since I do not have access to the ORCA Actor Registry database, I had to develop an alternative means by which I use the Registry's Hypertext Transfer Protocol (HTTP) source code to determine whether an ORCA actor is currently donating resources.

Lastly, I required an intuitive service storage system on the sponsor and client computers so that end-users easily can find their services and that would not add overhead to MSS. My first inclination was to use MySQL Binary Large Object (BLOB) storage and deliver the service to the user with the Secure File Transfer Protocol, but this would have added maintenance and upkeep in the Services database when a perfectly good file system already exists on the computer. However, the file system still requires a method to translate the attributes of the services to the file location on the computer. Coincidentally, my friend and former co-worker Donald Kline and I wrote a paper about his brainchild, the *Attribute Description Service for Large-Scale Networks* (2011) to address this problem and others.[3]

I did not use our Attribute Description Service in Version 1.0 because the project scope would have been too great, but I did include it in the architecture for MSS Version 2.0. The main idea I applied to MSS Version 1.0 is if one can describe a service according to its attributes, one can describe a service location according to its attributes. For example, consider a ficticious new program named the Simple Border Gateway Protocol (sBGP) to be a networking protocol service within the GENI framework that one may download as the archived file *sBGP.zip*. One might consider the attribute tree for sBGP like so: *MSS* has attributes, of which one is *Software…*has attributes, of which one is *Services…*has attributes, of which one is *Network…*has attributes, of which one is *Protocol…*has attributes, of which one is the service sBGP. Therefore, MSS stores the uniquely named file sBGP.zip in the same place on all computers: [MSS root location]/MSS/Software/Services/Network/Protocol/sBGP.zip. One need only apply the attributes one used to find the service sBGP to retrieve the full directory path to the file.

[3] Kline, D., & Quan, J. (2011). *Attribute Description Service for Large-Scale Networks.* (M. Kurosu, Ed.) Lecture Notes in Computer Science, 6776 (Human Computer International Conference 2011, Human Centered Design), 519-528

Summary and Conclusions

In summary, the project successfully delivered a working system to mutualistically exchange services for resources to authorized users on authorized computers. MSS guarantees this mutualistic relationship because the sponsor only delivers services upon request to users when the sponsor is donating resources. This is true for every sponsor in the ORCA hierarchy. Moreover, the *Software Design for Mutualistic Software Services Version 1.0* (Volume IV of this document) provides the configuration changes necessary for one to set up a single computer from which one can donate a portion of its resources to the ORCA CF in exchange for services.

MSS meets its Version 1.0 objective to create a means by which ORCA CF users, such as developers, experimenters, and researchers, can share and track software services for use within ORCA. Furthermore, it lays a solid foundation to refine this process in Version 2.0 because it provides a viable option for advertising and delivering services to current ORCA donors. As experimenters and developers create more and more services, more and more people will want to donate their resources to gain services.

Volume I

Software Requirements Specification for Mutualistic Software Services (MSS) Version 1.0

1. Introduction

According to *GENI At A Glance* (June 1, 2011), the National Science Foundation sponsors GENI, which is a large-scale network based on existing infrastructure that approximately 83 academic/industrial teams and 19 corporations are developing across the United States [1]. Currently, these same entities also contribute the vast majority of experimental resources to GENI. However, the GENI Systems Overview explains, "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." It further states that GENI goals are to:

1) Provide the flexibility for researchers to experiment on programmable components.
2) Include a wide range of technologies, to include wireless, and to incorporate new technologies as they emerge.
3) Permit experiments that act as one expects to see in the real world
4) Strongly support measurement based research.
5) Remove practical research barriers.
6) Support multiple experiments on a shared infrastructure suite.
7) Support a strong isolation of slices, to which donated resources will belong.
8) Ensure a broad array of contributors can donate resources easily.
9) Provide a secure environment safe from subversion.
10) Designed for a 15-20 year lifetime [2].

1.1 MSS Envisioned

In the future, Mutualistic Software Services (MSS) will leverage the GENI need for experimental resources with the public need for useful software services. A natural benefit of GENI experimentation is that new services arise from this effort, and so delivering these services to donors in exchange for resources forms a mutualistic relationship between the two. Unlike the current system where experimenters share their resources within GENI with little public donation, this product will provide valuable services in exchange for a small portion of the subscriber's CPU cycles, hard disk space, RAM, and Internet bandwidth. MSS is available to anyone who donates a slice of his or her networked resource to GENI.

To meet this objective, MSS will require a central office to oversee the exchange of resources for services. The MSS Center will develop and maintain MSS standards, but its final incarnation may be as part of another GENI department, like the GENI Meta-Operations Center (GMOC). The Center will collect the services from developers and advertise them on a web site, though the developers will maintain the web pages for the service. In addition, the Center will use the Attribute Description Service [3] to allow service developers to describe all aspects of their service, such as its purpose, scope, uses, versions, revisions, rescissions, additions, and other such information as deemed appropriate by the MSS Center. Donors may then query MSS to choose which services are most useful to their organization and download the service. Moreover, donors may download as many services as they desire as long as they are donating to GENI.

Currently, GENI's Open Resource Control Architecture (ORCA) control framework (CF) already has a "heartbeat" mechanism in place to ensure donors are donating and resources are available for experimentation. These heartbeats can carry relevant data, such as how many users are using which services, how many resources and of which type are available for experimentation, and other useful data that the MSS Center deems appropriate. Alternatively, the CF developer may choose to separate usage statistics from heartbeats to decrease collection intervals. In either case, MSS Center will collect statistics for each service to determine whether a service should receive continued support.

Furthermore, the MSS Center will act as the root of a hierarchy that delivers a subset of MSS services to the four GENI CFs: PlanetLab, ProtoGENI, ORBIT, and ORCA. For example, MSS Center will maintain every service, but the ORCA administrator at Duke University will only download services geared towards ORCA users. The University of Alaska Fairbanks (UAF) ORCA administrator then will download services as an affiliate of Duke, but may choose a smaller subset of services that fit UAF users. UAF then may sponsor a single server in Barrow, Alaska, and the Barrow affiliate then may sponsor an even smaller subset of services that are unique to Barrow users. As the number of offered services grows, so do the resource donations, to the mutual benefit of GENI and the resource donors.

1.2 MSS at Present

This SRS identifies the requirements to create MSS Version 1.0, which must exhibit key features of the MSS vision above in order to demonstrate success. Some of these features include:

- A hierarchical services delivery, in which the affiliate may choose all or a subset of sponsored services.
- A heartbeat check at the CF level to ensure the affiliate is donating resources to its sponsor before one can download new services.
- A means to install all CF and MSS software on a single computer to allow individual public participants.
- A process to ensure only authorized users on authorized computers can receive services.
- A method for sponsors to deliver MSS services, service data, and user data to its affiliates.
- A way for sponsors to host virtual and physical clients.

Furthermore, certain elements of the future vision are not necessary in the initial offering of MSS, but are identified in this document as Version 2.0 requirements. These features are:

- A means to gauge the percentage of resources an affiliate or client is donating.
- The inclusion of the GMOC at the root of MSS.
- The Attribute Description Service.
- A heartbeat check at the virtual and physical client level of service delivery.
- Usage statistics delivery to the CF developer.

In sum, MSS Version 1.0 administrators must be able to host and deliver services to affiliates, physical resources, and virtual resources from a canonical CF cluster and from a single-server CF cluster as a proof of concept.

1.1 Purpose

This SRS describes the software functional and nonfunctional requirements for version 1.0 of MSS. The project team will implement and verify the correct functioning of the system with this document. Unless otherwise noted, all requirements specified here are high priorities and committed for release 1.0.

1.2 Document Conventions

For ease of use, rules tables within this document begin on new pages so that project team members may divide it among the teams. The software team must address all Version 1.0 rules within this document before stakeholders will consider this project complete. Appendix A contains a Data Dictionary of terms.

1.3 Intended Audience and Reading Suggestions

The audience for this document consists of all favored User Classes listed in the *2.2 User Classes and Characteristics*, and it follows the conventions from *Software Requirements* (2003, p. 172) [4]:

1.4 Project Scope and Product Features

This SRS identifies the requirements to create MSS Version 1.0, which must exhibit key features of the MSS vision above in order to demonstrate success. Some of these features include:

- A hierarchical services delivery, in which the affiliate may choose all or a subset of sponsored services.
- A heartbeat check at the CF level to ensure the affiliate is donating resources to its sponsor before one can download new services.
- A means to install all CF and MSS software on a single computer to allow individual public participants.
- A process to ensure only authorized users on authorized computers can receive services.
- Other processes to:
    - Authenticate users.
    - Validate services.
    - Encrypt and Decrypt services during service delivery.
- A method for sponsors to deliver MSS services, service data, and user data to its affiliates.
- A way for sponsors to host virtual and physical clients.

Furthermore, certain elements of the future vision are not necessary to demonstrate the functionality of MSS, and they are identified in this document as Version 2.0 requirements. These features are:

- A means to gauge the percentage of resources an affiliate or client is donating.
- The inclusion of the GMOC at the root of MSS.
- The Attribute Description Service.
- A heartbeat check at the virtual and physical client level of service delivery.
- Usage statistics delivery to the CF origin.
- Secure Socket Layer encryption of the website.

In sum, MSS Version 1.0 administrators must be able to host and deliver services to affiliates, physical resources, and virtual resources from a canonical CF cluster and from a single-server CF cluster as a proof of concept.

MSS will permit GENI contributors to receive GENI developed software as services for donating a portion of their computer resources. Project team members must familiarize themselves with the following in order to conceptualize MSS, and it is recommended that other stakeholders familiarize themselves with this material.

MSS leverages four prevailing technologies to accomplish its goal:

- Open Resource Control Architecture (ORCA), a GENI CF [5]
- Eucalyptus (Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems) [6]
- Xen [7] and Kernel-based Virtual Machines (KVM) [8]

- Linux, Apache, MySQL, and PHP (LAMP) Server [9]

As a use case example, one can find two proposed scenarios that explore the implementation of these features in:

- *A Mutualistic Security Service Model: Supporting Large-Scale Virtualized Environments* (2011) [10]
- *Attribute Description Service for Large-Scale Networks* (2011) [3]

2. Overall Description

2.1 Product Perspective

MSS is a new system that leverages the need for public computer services and the GENI need for contributed resources. The system will evolve over subsequent releases to offer automatic lookup queries based on the initiating resource type and other criteria listed by the resource owner.

It is important to note the difference between the resource owner and the resource user. For example, a resource owner may be the owner of an organization or a delegated authority, such as a systems administrator. The resource owner decides which services are appropriate for his or her organizational needs, acquires the desired services from his or her parent CF, and advertises those services to affiliate CFs and resource users within the organization. Resource owners may pare these services further based on mission needs and user groups, such as installing an affiliate CF that only hosts accounting services for accountants or clerical services for information specialists.

This arrangement forms a hierarchy of service delivery, in which the company provides a small amount of resources in exchange for valuable services. MSS Version 1.0 will meet all of the requirements identified as *Version 1.0* in this document, with design considerations geared toward meeting *Version 2.0* requirements. Research findings of exactly how many experimenters GENI supports has proved elusive; however Version 2.0 support requirements are based on figures from GENI's largest CF, Planetlab,which Princeton University sponsors. Princeton's Policy Report & Analysis document, *Understanding and Resolving Conflicts* (November 13, 2008), lists 4,700 researchers [21], and so 4,700 multiplied by four CFs and doubled for expected growth equals 37,600 MSS users.

2.2 User Classes and Characteristics

User Class	Description
Resource Donors (favored)	These are members of the public who may find GENI services to be useful. Public donors view GENI Services as practically free because experimenters only use a portion of the donor's "unused" resources. However, one might not view bandwidth, RAM, and CPU cycles as unused because they can range from 0-100% use very quickly, and so some means of regulating experimenter sharing of these resources during "slow" times will be considered by the GENI CFs.
Resource Users	Resource users do not necessarily own the resource, but use the services on behalf of the resource owner. For instance, if a small business owner donated 20% of her computer resources, then her administrative assistant, as a registered resource user, might have access to a customer billing service.
GMOC	The GENI Meta Operations Center, based out of the University of Indiana, is the central authority for CF interconnections [11]
CFs (favored)	Each CF will download the MSS services and updates, and it will deliver them to their donors. This greatly simplifies the process of deciding who should receive MSS because if a member of the public does not connect to a CF, he or she cannot receive the service. This way the parent CF only offers MSS to its own users and one or more child CFs instead of thousands of individual donors, which reduces overhead. Furthermore, the originating CF already tracks and connects to its donor resources through heartbeats, and so this arrangement adds minimal authentication overhead to the system, while enabling each originating CF to gauge the MSS impact on donor participation. Though individual CFs determine their own federation standards, donors must meet some minimum GENI federation standards before federating resources, and must maintain these standards while connected to the CF. Lastly, MSS must not burden the CF with additional work. Current CF administrators must be able to manage MSS, such as performing downloads and updates, and adding authorized actors (computers) and users as necessary to run MSS.
Developers (favored)	Service developers are also the service maintainers. Developers use MSS to share their services created during the course of research or to fill a particular need, and they advertise those services to GENI donors. Donors provide a portion of their resources in exchange for valuable services, and so the developers/researchers have more resources on which to experiment.
Experimenters	Experimenters require resources on which to experiment, and increasing experimental resources is the main reason for creating MSS. The more resources experimenters have, the more "at-scale" experiments are possible. Furthermore,

	allowing the public to donate resources provides opportunities for "in the wild" experiments that the *GENI Systems Overview* (2008) identifies as crucial to GENI success [2].
GPO (favored)	The GPO represents the NSF, which pays for MSS. Service developers must give GPO recommendations the utmost consideration when forming decisions. One key aspect of GPO concern is that donor interest and trust in MSS remain firm. Maintaining donor confidence will ensure project continuation by steadily growing CFs.
"Pay-for" Applications Companies (disfavored)	Initially, these companies may perceive MSS as a threat, and therefore exert political pressure on the NSF to end MSS both directly and indirectly. The project manager will notify the GPO of any correspondence with these groups.

2.3 Operating Environment

Rule	Version 1.0	Version 2.0	Environment
OE-1	X	X	Hardware Platforms: MSS will operate on GENI clusters and single servers. MSS will operate on large server farms, small servers, personal computers, and tablet computers.
		X	
OE-2			Operating Systems:

	X	X	MSS is intended to support:
			
			• Linux
			  ○ Debian
			  ○ Slackware
			  ○ RedHat

		X	MSS will support other operating systems, such as:
			
			• Unix
			  ○ Macintosh OS X Leopard and above
			  ○ FreeBSD
			  ○ OpenSolaris
			• Windows (in subsequent releases)
			  ○ Windows 7

			○ Windows Vista
OE-3	X X	X X	Web Browsers: MSS is intended for major web browsers, such as: • Windows ○ Internet Explorer • Mozilla ○ Firefox • Konquerer ○ Safari ○ Google Chrome In addition, MSS must work on a text-based web browser in order to support virtual machines and computers without desktop environments: • w3m
OE-4	X	X X	Geographical Locations: Release 1.0 only supports US donors due to funding. Subsequent releases may include CFs from other countries, which may have their own MSS offerings.
OE-5	X	X	Users: US public donors to GENI CFs.
OE-6	X	X	Servers: Release 1.0 support will focus on Linux server platforms.
OE-7	X	 X	Databases: MSS will maintain at least three separate databases for: • Service data • CF data • User and client data MSS will use the Attribute Description Service to track service, user, and client data.

2.4 Design and Implementation Constraints

Rule	Version 1.0	Version 2.0	Constraint
CO-1		X	This system's design, code, and maintenance documentation shall conform to the *W3C Quality Assurance Interest Group* specifications [13].
CO-2	X	X	Web design will be similar to the CF style.
CO-3	X	X	OS Versions: Design MSS only for *supported* versions of operating systems.
CO-4	X	X	Web Browser Versions: Design MSS only for *supported* versions of web browsers.

2.5 User Documentation

Rule	Version 1.0	Version 2.0	Documentation
UD-1		X	The services shall provide an online, cross-linked help system in HTML that describes all service functions.
UD-2		X	A "Quick Start" document shall provide screenshots of applicable windows and how the user shall interact with them for each major operating system. The *ORCA User Manual* (2011) serves as an example [12].
UD-3	X	X	The service shall provide the ability to download the user manual in Portable Document Format (PDF).
UD-4		X	The system shall provide CF operators with online documentation on how to download MSS additions.
UD-5		X	The system shall provide CF operators with online documentation on how to download MSS revisions.
UD-6		X	The system shall provide CF operators with online documentation on how to process MSS rescissions.
UD-7	X	X	The system shall provide CF operators with online documentation on how to maintain MSS versions.

2.6 Assumptions and Dependencies

Rule	Version 1.0	Version 2.0	Documentation
AS-1	X	X	CFs are connected in a hierarchical tree
AS-2	X	X	The child CF can synchronize MSS services with its parent CF.
AS-3	X	X	Parent CFs have the capacity to distribute services to its child CF and donors
AS-4	X	X	All CFs will send “heartbeats” to its originating CF
DE-4		X	The system will provide CF operators with online documentation on how to download MSS revisions.
DE-5		X	The system will provide CF operators with online documentation on how to process MSS rescissions.
DE-6		X	The system will provide CF operators with online documentation on how to maintain MSS versions.

3. System Features

Feature	Description	Requirements	Version 1.0
FE-1	CF driven federation standards	DESCRIPTION	Each CF decides whether donors meet federation requirements, instead of MSS acting as a central authority for making this determination. This allows for greater autonomy among CFs and LSNs because each can have stringent standards that target certain donor types or relaxed standards that accept a broad variety of resources.
		PRIORITY	HIGH
		STIMULUS	Individual donors choose a CF to which he or she contributes resources.
		RESPONSE	That CF decides whether the donor meets its federation standards, offers steps to meet the standards, or recommends another CF.
		FUNCTIONAL REQUIREMENTS	TBD by each CF.
FE-2	CF controlled service	DESCRIPTION	Each CF determines whether the donor meets its standards and receives MSS. This may be as simple as, "while you are connected to this CF, you receive this service," or the CF may limit the service, such as "if you do not maintain a connection 24 hours a day, you do not receive the service." CFs already have a heartbeat mechanism in place to determine connection times.
		PRIORITY	HIGH
		STIMULUS	An individual donor chooses a CF to which he or she contributes resources.
		RESPONSE	That CF decides whether the donor meets its federation standards, offers steps to meet the standards, or recommends another CF.
		FUNCTIONAL REQUIREMENTS	TBD by each CF.

FE-3	Software Updates	DESCRIPTION	MSS delivers MSS software updates through the MSS interface. Hierarchical delivery conserves bandwidth and streamlines service decisions (FE-1, 2)
		PRIORITY	HIGH
		STIMULUS	MSS finishes final testing of its next software update.
		RESPONSE	MSS delivers the system update to participating CFs in GENI by the end of the next business day.
		FUNCTIONAL REQUIREMENTS	MSS and the participating CF: o Must be connected to the Internet. o Must authenticate before transmission. o Must use data encryption. o Must verify valid receipt.
FE-4	Service Updates	DESCRIPTION	MSS delivers all GENI services through the MSS interface. Hierarchical delivery conserves bandwidth and streamlines service decisions (FE-1, 2)
		PRIORITY	HIGH
		STIMULUS	MSS finishes final testing of its service or service update.
		RESPONSE	MSS delivers the service or update to participating CFs in GENI by the end of the next business day.
		FUNCTIONAL REQUIREMENTS	MSS and the participating CF: o Must be connected to the Internet. o Must authenticate before transmission. o Must use data encryption. o Must verify valid receipt.
FE-5	Donor Web Interface	DESCRIPTION	MSS queries will be conducted over a web interface using an appropriate authentication mechanism.
		PRIORITY	HIGH

		STIMULUS	Donor initiates a query for a service based on his or her parent CF service list.
		RESPONSE	MSS responds with a list of services available to the donor.
		FUNCTIONAL REQUIREMENTS	The web interface must support the major browsers listed in *2.3 Operating Environment.*
Feature	**Description**	**Requirements**	**Version 2.0**
FE-6	Service Management interface	DESCRIPTION	ADS will provide "cradle-to-grave" service management by tracking service revisions, rescissions, dependencies, and other life-cycle data that an appropriate authority can add, edit, and delete.
		PRIORITY	HIGH
		STIMULUS	Manager adds a new service Manager adds a new developer.
		RESPONSE	The interface responds whether the new entry is complete.
		FUNCTIONAL REQUIREMENTS	The web interface must support the major browsers listed in *2.3 Operating Environment.*
FE-7	GENI Research Information	DESCRIPTION	GENI provides this venue for GENI experimenters to post service "tips and tricks," advise donors and other experimenters, post editorials and current research information, and any other information that GENI sees fit to post. It will be in a wiki format. The GUI and Online Users Manual will have links to this wiki.
		PRIORITY	LOW
		STIMULUS	Donor clicks GENI Research Information Wiki link
		RESPONSE	GENI Research Information Wiki link comes up.
		FUNCTIONAL REQUIREMENTS	GENI provides server space for MSS wiki. The MSS wiki shall use an appropriate database.

FE-8	GENI Notices	DESCRIPTION	GENI provides notices of upcoming events, information on GENI experiments, how contributions help GENI develop network innovation, and any other information that GENI sees fit to post.
		PRIORITY	LOW
		STIMULUS	Donor clicks GENI Notices link
		RESPONSE	GENI Notices link comes up.
		FUNCTIONAL REQUIREMENTS	GENI provides server space for GENI Notices. This may link to a web site GENI already owns.

4. Functional Requirements

Rule	Version 1.0	Version 2.0	Interface
FR-1	X	X	Resource Owner User data must be safeguarded.
FR-2	X	X	MSS will use industry approved encryption standards, such as Digital Signature Algorithm (DSA) encryption.
FR-3	X	X	Only resource donors will have access to new GENI Services. MSS will rely on heartbeats sent from the resource to the originating CF to ensure this is so.
FR-4		X	ADS will track service usage
FR-5	X	X	Resource owners will have a means to control which authorized users can download services from its parent CF
FR-6	X	X	MSS-CENTER will support service mirroring
FR-7	X	X	MSS-PARENT will support service mirroring

5. External Interface Requirements

5.1 User Interfaces

Rule	Version 1.0	Version 2.0	Interface
UI-1	X	X	MSS screen displays shall conform to W3C formatting standards [13].
UI-2		X	MSS will comply with W3C handicap accessibility standards [13].
UI-3	X	X	MSS shall permit complete navigation using the keyboard alone, in addition to using mouse and keyboard combinations.
UI-4	X	X	MSS shall comply with GENI CF style designs
UI-5		X	The system shall display a help link on each page that links to a help page. At a minimum, the help page will contain a link to: ○ The user manual ○ The online manual ○ The wiki ○ The GENI Notification web page
UI-6	X	X	The web pages shall permit complete navigation using the keyboard alone, in addition to using mouse and keyboard combinations.

5.2 Hardware Interfaces

At a minimum, MSS-CENTER will maintain these components:

Rule	Version 1.0	Version 2.0	Interfaces
HI-1		X	MSS-CENTER will support web site mirroring.
HI-2		X	A web server will host all of the web pages necessary for finding service information. The web pages will include the Home page, About, News, Wiki, Events, Member, Links, and Help with appropriate information included in each. The Links page will contain hyperlinks to helpful service information, with a GENI link prominently displayed.
HI-3		X	A DNS server will act as the authoritative name server for MSS. Authoritative name servers are assigned to be responsible for their particular domains, and in turn can assign other authoritative name servers for their sub-domains. It will also implement the recursive algorithm necessary to resolve a given name starting with the DNS root through to the authoritative name servers of the queried domain. With this function

			implemented in the name server, user applications gain efficiency in design and operation [14].
HI-4		X	A communications server shall be an enterprise real-time communications server, providing the infrastructure for enterprise instant messaging, presence, file transfer, peer-to-peer and multiparty Voice and Video calling, ad hoc and structured conferences (audio, video and web) and PSTN connectivity. These features are available within an organization, between organizations, and with external users on the public internet, or standard phones, on the PSTN as well as SIP trunking [15].
HI-5	X	X	A database server will contain the information about MSS services and Resource Owners/Users data.
HI-6		X	A systems administration server will utilize Simple Network Management Protocol (SNMP) and Cacti, among other tools. SNMP is a UDP-based network protocol. It is used mostly in network management systems to monitor network-attached devices for conditions that warrant administrative attention [16]. Cacti is a complete network graphing solution [under the GNU General Public License] designed to harness the power of RRDTool's data storage and graphing functionality. Cacti provides a fast poller, advanced graph templating, multiple data acquisition methods, and user management features out of the box. All of this is wrapped in an intuitive, easy to use interface that makes sense for LAN-sized installations up to complex networks with hundreds of devices [17].
HI-7		X	A load balancer will distribute workload evenly across two or more computers, network links, CPUs, hard drives, or other resources, in order to get optimal resource utilization, maximize throughput, minimize response time, and avoid overload. Using multiple components with load balancing, instead of a single component, may increase reliability through redundancy. The load balancing service is usually provided by a dedicated program or hardware device (such as a multilayer switch or a DNS server) [18].

At a minimum, MSS-ORIGIN/AFFILIATES will maintain these components:

Rule	**Version**		**Interfaces**
	1.0	**2.0**	
HI-8	X	X	Single-server components will connect behind a firewall
HI-9	X	X	Constant connection to the Internet through a Cable Modem, Direct Service Line, or other network connection
HI-10	X	X	A computer resource, provisioned according to a GENI CF standard.

5.3 Software Interfaces

Rule	Version 1.0	Version 2.0	Interface
SI-1			**MSS-ENTITY (CENTER, ORIGIN, AFFILIATE)**
SI-1.1		X	MSS-ENTITY shall maintain MSS in accordance with MSS-CENTER standards.
SI-1.2	X	X	MSS-ENTITY shall connect through an encrypted means
SI-1.3	X	X	MSS-ENTITY shall deliver services to its children.
SI-1.4	X	X	MSS-ENTITY shall deliver MSS information to its children.
SI-1.5	X	X	MSS-ENTITY shall maintain a hash, such as Secure Hash Algorithm (SHA), for all services it maintains as proof of the correct download.
SI-1.6	X	X	MSS-ENTITY shall maintain a subset of service distributions for itself and its children according to MSS-CENTER standards.
SI-1.7		X	MSS-ENTITY shall maintain current ADS management information, such as additions, revisions, and rescissions for all of its services.
SI-1.8		X	MSS-ENTITY shall send heartbeats to its MSS-ORIGIN.
SI-1.9		X	MSS-ENTITY shall send registered user information to its parent
SI-1.9.1		X	MSS user information will only contain what MSS-CENTER requires to determine whether software should receive continued support, be revised, rescinded, and other similar data.
SI-1.10		X	MSS-ENTITY shall be chained through its parent to MSS-CENTER in normal operation, excepting short, unplanned periods.
SI-1.11		X	MSS-ENTITY shall maintain its connection to its children in normal operation, excepting short, unplanned periods.
SI-1.12		X	MSS-ENTITY shall transmit service download statistics for itself and its children to its MSS-ORIGIN.
SI-2			**MSS-CENTER**
SI-2.1		X	MSS-CENTER shall approve ADS definitions.
SI-2.2		X	MSS-CENTER shall approve new and existing services
SI-2.3		X	MSS-CENTER shall maintain all MSS services.

SI-2.4		X	MSS-CENTER shall provide a web site for service developers to deliver service information to users (HI-2)
SI-2.5		X	The developer's web site shall have a means to log in securely and prevent tampering with its web pages.
SI-3			**MSS-ORIGIN**
SI-3.1	X	X	MSS-ORIGIN shall maintain a current GENI CF.
SI-4			**MSS-AFFILIATE**
SI-4.1	X	X	MSS-AFFILIATE shall maintain a current GENI CF.
SI-5			**MSS-RESOURCE**
SI-5.1	X	X	MSS-RESOURCE shall be authorized by its parent.
SI-6			**MSS-USER**
SI-6.1	X	X	MSS-USER shall register under his or her parent CF.
SI-7			**MSS-DEVELOPER**
SI-7.1		X	MSS-DEVELOPER shall register as a MSS developer under his or her parent CF.
SI-7.2		X	MSS-DEVELOPER shall submit service information to the developer's website according to MSS-CENTER standards.
SI-7.3		X	MSS-DEVELOPER shall submit a service to MSS according to MSS-CENTER standards.

5.4 Communications Interfaces

Rule	Version 1.0	Version 2.0	Interfaces
CI-1		X	MSS employees shall use an email application that uses these protocols at a minimum: • IMAP • POP • SMTP
CI-2		X	MSS-CENTER shall have an HTTP secure method for customers to contact a central mailbox.
CI-3	X	X	MSS-ENTITY shall transfer files using at least DSA encryption

CI-4		X	MSS-CENTER shall set up a secure FTP box for any other customer transfers.

6. Other Nonfunctional Requirements

6.1 Performance Requirements

Rule	Version 1.0	Version 2.0	Interfaces
PE-1		X	MSS-CENTER will support site mirroring and return to service within 1 minute of a hard disk failure. (HI-1)
PE-2		X	A web server will act as the primary interface between the public and the service, and it will host all of the web pages necessary for MSS-CENTER interaction. Release 2.0 will support 37,600 visitors. (HI-2)
PE-3		X	A DNS server will act as the authoritative name server for MSS. Release 2.0 will support 37,600 visitors. (HI-3)
PE-4		X	A communications server shall be an enterprise real-time communications server. Release 2.0 will support 37,600 visitors. (HI-4)
PE-5		X	A database server will contain the ADS service information loaded into MSS-CENTER. Release 2.0 will support 37,600 visitors. (HI-5)
PE-6		X	A systems administration server is used mostly in network management systems to monitor network-attached devices for conditions that warrant administrative attention. Release 2.0 will support 37,600 visitors. (HI-6)
PE-7		X	A load balancer will distribute workload evenly across MSS-CENTER. Release 2.0 will support 37,600 visitors. (HI-7)
PE-8		X	Routers will interconnect all of these components and provide a connection to the internet. Release 2.0 will support 37,600 visitors. (HI-8)

6.2 Safety Requirements

None listed

6.3 Security Requirements

List the security requirements not covered elsewhere within this document.

Rule	Version 1.0	Version 2.0	Requirement
MSS-USER			
SE-1	X	X	Must register with a password
SE-2		X	Read the End User License Agreement and electronically sign his or her understanding with an "I agree" button before using the service.
SE-3		X	Use CAPCHA when emailing MSS-CENTER over HTTP
MSS-CENTER			
SE-4		X	Will anonymize subscriber information to the greatest extent possible to guard against personal information loss.
SE-5		X	Will use strong password protection, which will be enforced by the system
SE-6		X	Must use a physical security token to access workstations and servers.
SE-7		X	Passwords must be changed every 90 days
MSS-ORIGIN/ AFFILIATE			
SE-8		X	Will anonymize subscriber information to the greatest extent possible to guard against personal information loss.
SE-9		X	Will use strong password protection, which will be enforced by the system

6.4 Quality Attributes

Rule	Version 1.0	Version 2.0	Requirement
Flexibility		X	MSS should be flexible enough to work on several manufacturer-supported operating systems such as Windows, Linux, and Macintosh.
FL-1		X	A maintenance programmer who has at least six months experience programming a Windows supported operating system shall be able to update MSS data and services within one hour.
FL-2	X	X	A maintenance programmer who has at least six months experience programming a Linux supported operating system shall be able to update MSS

			data and services within one hour.
FL-3		X	A maintenance programmer who has at least six months experience programming a Macintosh supported operating system shall be able to update MSS data and services within one hour.
Integrity		X	The MSS Chief Security Officer has the final say in what security authentication measures are implemented to ensure appropriate measures are taken. He or she should also keep a history of changes to the security authentication measures.
IN-1		X	The MSS CSO must approve any change to MSS security authentication protocols in writing.
IN-2		X	The MSS CSO must maintain records of changes to security authentication protocols for 10 years.
Reliability	X	X	MSS shall only be unavailable for 24 hours per month during peak hours, and 48 hours per month during non-peak hours.
RE-1	X	X	MSS shall be at least 96.77% available from 6:00 a.m. EST to 10:00 p.m. EST.
RE-2	X	X	MSS shall be at least 93.75% available from 10:00 p.m. EST to 6:00 a.m. EST.

7. Other Requirements

At least one GENI CF must work on a single server in order to gain public resource donations. A key requirement for MSS Version 1.0 is developing a systems design so that all software required by one CF and all MSS software works on a single server.

8. References

[1] BBN Technologies. (June 1, 2011). *GENI at a Glance.* GENI. Retrieved on April 7, 2012, from: http://www.geni.net/wp-content/uploads/2011/06/GENI-at-a-Glance-1Jun2011.pdf

[2] BBN Technologies. (2008, September 29). *GENISysOvrvw092908.pdf.* Retrieved May 30, 2010, from GENI System Overview: http://groups.geni.net/geni/attachment/wiki/

[3] Kline, D., & Quan, J. (2011). *Attribute Description Service for Large-Scale Networks.* (M. Kurosu, Ed.) Lecture Notes in Computer Science, 6776 (Human Computer International Conference 2011, Human Centered Design), 519-528

[4] Wiegers, K. (2003). *Software Requirements* (2nd ed.). Redmond: Microsoft Press.

[5] RENCI. *Open Resource Control Architecture*. (August 13, 2011). Renaissance Computing Institute. Retrieved December 8, 2011, from http://groups.geni.net/geni/wiki/ORCABEN

[6] Eucalyptus Systems. *Eucalyptus*. Eucalyptus Systems, Inc. Retrieved December 8, 2011, from http://www.eucalyptus.com/

[7] Citrix. *Xen*. (2011). Citrix Systems, Inc. Retrieved December 8, 2011, http://xen.org/

[8] RedHat. *KVM*. (2011). RedHat Emerging Technology Project. Retrieved December 8, 2011, from http://www.linux-kvm.org/page/Main_Page

[9] Wikipedia. (March 23, 2012). *LAMP (software bundle).* Wikipedia.org. Retrieved on March 23, 2012 from: http://en.wikipedia.org/wiki/LAMP_(software_bundle)

[10] Quan, J., Nance, K., & Hay, B. (May/June 2011). *A Mutualistic Security Service Model: Supporting Large-Scale Virtualized Environments*. IT Professional (IEEE), 18-23

[11] GENI Meta-Operations Center. (2010). *GMOC Concept of Operations.* Retrieved November 10, 2010, from GMOC: http://gmoc.grnoc.iu.edu/uploads/8i/Gu/8iGu80-LqQB37VU4ZE1i5g/GENI_Concept_of_Operations-final.pdf

[12] Quan, John. *ORCA 3.0 User Manual*. (June 30, 2011). Retrieved December 8, 2011, from ORCA3.0_User_Manual_20110630

[13] Worldwide Web Consortium. (2010, April 29). *W3C Quality Assurance Interest Group*. Retrieved December 8, 2010, from The Matrix of W3C Specifications: http://www.w3.org/QA/TheMatrix

[14] Wikipedia. (2010, April 16). *Domain Name System*. Retrieved December 6, 2010, from Wikipedia.org: http://en.wikipedia.org/wiki/Domain_Name_System

[15] eHow Inc. (2010). *Office Communications Server*. Retrieved Dec 6, 2010, from eHow: http://www.ehow.com/office-communications-server/

[16] Wikipedia. (2010, April 12). *Simple Network Management Protocol*. Retrieved April 17, 2010, from Wikipedia.org: http://en.wikipedia.org/wiki/Snmp

[17] The Cacti Group. (2009). *About Cacti*. Retrieved December 6, 2010, from Cacti.net: http://www.cacti.net/

[18] Wikipedia. (2010, April 14). *Load balancing (computing)*. Retrieved December 6, 2010, from Wikipedia.org: http://en.wikipedia.org/wiki/Load_balancing_(computing)

[19] DS Development Software. (2010). Email Protocols: IMAP, POP3, SMTP and HTTP. Retrieved December 10, 2010, from Email Productivity Software: http://www.emailaddressmanager.com/tips/protocol.html

[20] Wikipedia. (2010, October 27). Kernel (computing). Retrieved November 1, 2010, from Wikipedia.org: http://en.wikipedia.org/wiki/Kernel_(computing)

[21] Peterson, L. (November 13, 2008). *Understanding and Resolving Conflicts on PlanetLab. Princeton University*. Retrieved on April 11, 2012, from: http://www.cs.princeton.edu/~llp/policy.pdf

Appendix A

Data Dictionary

ADS – Attribute Description Service [3]

Aggregate – a collection of components that usually comprise a system

Bandwidth - a bit rate measure of available or consumed data communication resources expressed in Gigabits/second.

CF – Control Framework

Component – a physical computer resource, such as a router, switch, computer, phone, or copy machine

Community – MSS users identified by a GENI control framework

Contributor – an entity that donates a portion of its resources to GENI

Control framework – one of four GENI architectures used to federate computer resources

Donation – Donations include nearly all types of network resources, such as computers, network routers and switches, cell phones, computer tablets, copy machines, and so on. However, MSS is only available for computers as of this writing.

Experimenter – one who conducts network research on GENI

Federate – incorporate one's computer resource into a GENI control framework

File list – hierarchical locations of files on a resource

GENI – Global Environment for Network Innovation

GUI – Graphical User Interface

Hard Disk – Non-volatile storage device for digital media

HTTP (Hypertext Transfer Protocol) – is not a protocol dedicated for email communications, but it can be used for accessing your mailbox. In addition, called web based email, this protocol can be used to compose or retrieve emails from your account. Hotmail is a good example of using HTTP as an email protocol [19].

IMAP (Internet Message Access Protocol) – Is a standard protocol for accessing e-mail from your local server. IMAP is a client/server protocol in which e-mail is received and held for you by your Internet server. As this requires only a small data transfer this works well even over a slow connection such as a modem. Only if you request to read a specific email message will it be downloaded from the server. You can also create and manipulate folders or mailboxes on the server, delete messages etc… [19]

Insertable media – a storage device that is not native to the resource, such as a USB drive, Compact Disc, and external hard drive

ISP – Internet Service Provider

Kbps – Kilobytes per second

Kernel - In computing, the kernel is the central component of most computer operating systems; it is a bridge between applications and the actual data processing done at the hardware level. The kernel's responsibilities include managing the system's resources (the communication between hardware and software components). Usually as a basic component of an operating system, a kernel can provide the lowest-level abstraction layer for the resources (especially processors and I/O devices) that application software must control to perform its function. It typically makes these facilities available to application processes through inter-process communication mechanisms and system calls [20].

LSN – Large-Scale Network. GENI is an example of a large-scale network.

LTS – refers to a Long-Term Service agreement provided by many operating system developers, such as Ubuntu.

Malware – includes computer viruses, computer worms, Trojan horses, most rootkits, spyware, dishonest adware and other malicious and unwanted software, including true viruses.

Mbps – Megabytes per second

MSS – Mutualistic Software Services

MSS-CENTER – refers to the main MSS office, where MSS is developed and distributed to participating control frameworks.

MSS-AFFILIATE – refers to all child nodes of the MSS-ORIGIN control framework. At a minimum, MSS-AFFILIATE will contain its MSS-ORIGIN control framework and a subset of its services.

MSS-DEVELOPER – refers to a member of academia or business who submits a service to MSS.

MSS-ENTITY – refers to MSS-CENTER, MSS-ORIGIN, and MSS-AFFILIATE. These are all of the MSS entities that have a control framework installed.

MSS-ORIGIN – refers the control framework developer and maintainer. MSS-ORIGIN falls directly beneath MSS-CENTER in the hierarchy.

MSS-RESOURCE – a resource without a control framework, such as a physical computer or virtual machine.

MSS-USER – refers to an authorized user of a MSS-RESOURCE due to his or her working relationship with a resource owner.

Node – a computer

NSF – National Science Foundation

OS – Operating System

POP (Post Office Protocol 3) – provides a simple, standardized way for users to access mailboxes and download messages to their computers. When using the POP protocol all your email messages will be downloaded from the mail server to your local computer. You can choose to leave copies of your emails on the server as well. The advantage is that once your messages are downloaded you can cut the internet connection and read your email at your leisure without incurring further communication costs. On the other hand you might have transferred a lot of message (including spam or viruses) in which you are not at all interested at this point [19].

Programmable component – a network resource with a modifiable operational instruction set

RAM – Random Access Memory. For MSS, this refers to the computer's volatile memory, such as Dynamic RAM (DRAM).

Real-world user – a private citizen

Resource – a portion or an entire physical computer hardware component. MSS is only concerned with resources such as laptops, desktops, and computer tablets (such as iPad).

Resource Owner – one who is authorized to donate all or a portion of a resource to GENI

Slice – a collection of one or more aggregates and components

SMTP (Simple Mail Transfer Protocol) – is used by the Mail Transfer Agent (MTA) to deliver your email to the recipient's mail server. The SMTP protocol can only be used to send emails, not to receive them. Depending on your network / ISP settings, you may only be able to use the SMTP protocol under certain conditions (see incoming and outgoing mail servers [19].

SRS – Software Requirements Specification

Standard – the minimum hardware and software configuration required to federate into GENI

Traffic – the movement of information across the Internet

UI – User Interface

USB – universal serial bus

Wireless – radio-based Internet connection

Volume II

Software Architecture for Mutualistic Software Services (MSS) Version 1.0

1. Introduction

1.1 Document Introduction

The *Software Requirements Specification for Mutualistic Software Services Version 1.0* identifies Mutualistic Software Services (MSS) as a means for GENI to advertise and deliver services that experimenters develop in exchange for experimental resources. As Len Bass et al., recommend in *Software Architecture in Practice* (2003), this document intends to develop an architecture for the "… structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them" [1]. Therefore, this document will develop an architectural skeleton based on the functional requirements, non-functional requirements, and constraints identified in the SRS.

1.2 Business Decisions

Mutualistic Software Services (MSS) addresses several experimenter, researcher, developer, entrepreneur, and resource owner needs. Its most basic meaning derives from:

- Mutualistic – a win-win relationship in which both parties benefit.
- Software – the tools that those parties use to perform experiments, do research, sell products, run resources, and create services, to include the created services.
- Services – the virtual machine images, libraries, scripts, code, directions, and other useful products, to include the advertising of these products to stakeholders.

MSS has four different business models: trade, monetized, GENI-only, and a hybrid of the first three. All but the GENI-only model hinges on the idea that GENI requires many resources to create at-scale network experiments, and all models recognize that experimenters need a way to advertise and share experiments. In the trade model, GENI experimenters create services in the course of research that are unique and valuable, and public organizations and individuals can donate a portion of their resources in exchange for these worthy services. The monetized model is in keeping with the US Ignite program explained below, which relies on the notion that GENI experimenters will develop services that are so useful that people are willing to pay for them. The GENI-only model considers that GENI experimenters create software to create services, and that both of these products are useful to other experimenters. The hybrid model combines one or more of the first three models. However, no one will know about these products within any model unless one creates some advertising and sharing mechanisms.

The MSS architecture relies on distributed service delivery because this fits all of the models, if one considers a monetized system to have one parent with every other entity as its child. Therefore, the architectural perspective is that MSS will use the trade model because it is the most highly distributed of the three.

1.3 System Purpose and Scope

MSS leverages the GENI need for experimental resources with the public's need for useful services, and serves to meet the goals of two National Science Foundation (NSF) sponsored programs. The Computer & Information Science & Engineering (CISE) branch of the NSF sponsors U.S. Ignite:

> US Ignite is an initiative to spark the development of killer apps in areas of national priority: health, education, energy, economic development (including advanced manufacturing), transportation, and public safety on an ultra high speed (>100 Mbps up- and download), deeply programmable (not requiring internet protocol) and sliceable network. US Ignite is doing this by: 1) funding researchers and developers to create applications and services, and 2) stitching together an at-scale testbed with real users that researchers, developers, and entrepreneurs can use as a platform to develop applications and services [2].

The CISE also sponsors GENI, which is the "at scale testbed" on which U.S. Ignite will develop the "killer apps" of the future. The GENI Project Office describes GENI in this way:

> GENI, a virtual laboratory for exploring future internets at scale, creates major opportunities to understand, innovate and transform global networks and their interactions with society. Dynamic and adaptive, GENI opens up new areas of research at the frontiers of network science and engineering, and increases the opportunity for significant socio-economic impact. GENI will:
>
> - support at-scale experimentation on shared, heterogeneous, highly instrumented infrastructure;
> - enable deep programmability throughout the network, promoting innovations in network science, security, technologies, services and applications; and
> - provide collaborative and exploratory environments for academia, industry and the public to catalyze groundbreaking discoveries and innovation [3].

MSS serves as the conduit through which GENI experimenters gain resources for at-scale experiments and U.S. Ignite delivers services. Some examples of new services in development by GENI experimenters are uCap, FlowScale, OpenFlow, and MobilityFirst. All of these were highlighted in a recent GENI "News and Events" article [4], which is reproduced in its entirety in Appendix C.

1.4 Definitions, Acronyms, and Abbreviations

Please find definitions, acronyms, and abbreviations in *Software Requirements Specification for Mutualistic Software Services Version 1.0.*

- Appendix A: Data Dictionary

1.5 System Overview

MSS solves the problem of providing a vast number of real-world experimental resources to GENI experimenters by providing a means to "pay" for the resources with the services they create. This arrangement is mutually beneficial to resource donors, who must only provide a small portion of their resources in order to receive the services. MSS resides on one computer in a GENI cluster and distributes services rooted at MSS-CENTER to its subordinate CFs. MSS-ORIGINS deliver a subset of services to their MSS-AFFILIATES, and this distribution continues until services are distributed to all MSS-RESOURCES. This forms a hierarchical tree, with each parent CF acting as a node and its children as its

branches. MSS branches are capable of autonomy in that a MSS-ENTITY can continue using downloaded services even when its upstream connection to the MSS-CENTER is severed. However, the MSS-ENTITY must be sending heartbeats to its MSS-ORIGIN in order to receive new services.

MSS favors this distributed method over a centralized service warehouse because it:

- reduces donors' operational reliance on MSS-CENTER
- distributes MSS management to the lowest levels required
- allows each MSS-ENTITY to choose only those services it finds useful or necessary
- enables donors to choose a MSS-ENTITY that delivers services most in line with their objectives
- gives donors a means to manage their remaining resources with the required CF software
- gives donors a means to virtualize their remaining resources with the required CF software

In addition, sending "heartbeats" from an MSS-ENTITY up to its MSS-ORIGIN serves a two-fold purpose. In Version 1.0, sending heartbeats from the resource ensures that the CF is operational, that donated resources are available for experimentation, and that the resource is authorized to receive services.

In Version 2.0, adding intelligence-filled heartbeats from an MSS-ENTITY to its MSS-ORIGIN provides a means to deliver information, such as which services are in use, how many users are using the service, and other data required by the MSS-CENTER. The MSS-ORIGIN may separate MSS heartbeats from its CF heartbeats, but it will deliver the MSS data to MSS-CENTER. This data is ultimately serves as usage statistics to determine which services deserve continued support.

2. Decomposing the SRS

2.1 System Description

MSS has four main components: MSS-CENTER, MSS-ORIGIN, MSS-AFFILIATE, and MSS-RESOURCE. The actual hardware and software used may vary from component to component, but each component must support service delivery to its children. As of this writing, MSS will only support the remote server as a single-server MSS/CF installation, as described in section 5.1 Remote Server Model.

Currently, the MSS-AFFILIATE architecture only supports computers with two network interface cards and a static IP address or a Fully Qualified Domain Name. ORCA clusters meet these requirements, to include the remote server. In addition, later versions of MSS will support the Windows OS; therefore, one must develop MSS with common workstations and laptop computers in mind.

2.2 Functional Attributes

The System must:

- Deliver services over an Internet connection
- Provide a management interface for MSS administrators
- Allow the administrator to choose a subset of parent services
- Allow the user to choose a service from the parent CF
- Check for heartbeats on MSS-ORIGIN

2.3 Non-functional Attributes

The System shall have:

- Portability
- Autonomy
- Securability
- Credibility
- Integrability
- Extensibility
- Interoperability
- Usability

2.4 System Constraints

2.4.1 Hardware Constraints

Mutualistic Software Services Software Requirements Specification, Version 1.0 lists:

- 5.2 Hardware Interfaces
- 6.1 Performance Requirements

The following tables list examples to help describe the minimum hardware necessary to host MSS-CENTER and a MSS-enabled CF for each component.

Table 1. MSS-CENTER System Description

Attribute	Value
Multiple Servers	These servers may be used as a platform for other MSS Center functions.
Note:	Servers will be similar to MSS-ORIGIN/AFFILIATE below.

Table 2. MSS-ORIGIN/AFFILIATE Head Node System Description

Attribute	Value [5]
Make	Dell
Model	PowerEdge C1100
Processor	Quad-Core Intel® Xeon® Processor E5620 (12M Cache, VT enabled)
Processor Clock Speed	2.40 GHz, 5.86 GT/s Intel® QPI
System RAM	12 GB
System Storage	1 TB
Operating System	Unix, Linux with hypervisor support

Table 3. MSS-ORIGIN/AFFILIATE Worker Node System Description

Attribute	Value [5]
Make	Dell
Model	PowerEdge C1100
Processor	2 Quad-Core Intel® Xeon® Processors E5620 (12M Cache, VT enabled)
Processor Clock Speed	2.40 GHz, 5.86 GT/s Intel® QPI
System RAM	48 GB
System Storage	250 GB
Operating System	Unix, Linux with hypervisor support

Table 4. MSS-AFFILIATE (and Remote Server) System Description

Attribute	Value [6]
Make	Dell
Model	PowerEdge 2850
Processor	2 Dual-core 64-bit Intel Xeon processors
Processor Clock Speed	2.8GHz
System RAM	12GB DDR-2 400 SDRAM
System Storage	> 1 TB
Operating System	Unix, Linux with hypervisor support

Table 5. MSS-RESOURCE Physical System Description

Attribute	Value [7]
Make	Apple
Model	MacBook Pro 15”
Processor	2.8GHz Intel Core 2 Duo processor with 6MB shared L2 cache
Processor Clock Speed	2.8GHz
System RAM	4GB (two 2GB SO-DIMMs) of 1066MHz DDR3 SDRAM
System Storage	> 300 GB
Operating System	Unix, Linux, (and Windows in subsequent releases)

2.4.2 Software Constraints

In addition to the Constraints listed in *Software Requirements Specification for Mutualistic Software Services Version 1.0*

- 2.3 Operating Environment
- 2.4 Design and Implementation Constraints

The System must:

- Support a CF, such as ORCA [8]

- Support a CF-approved virtualized or paravirtualized hypervisor, such as Xen [9]
- Support a CF-approved virtualization technology, such as Eucalyptus [10]
- Support a CF-approved image-passing system, such as Image Proxy [11]
- Only modify the OS in accordance with CF instructions

2.5 User Characteristics

2.5.1 User Classes and Characteristics

Please find this section in *Software Requirements Specification for Mutualistic Software Services Version 1.0*

- 2.2 User Classes and Characteristics

2.5.2 User Groups and Attributes

A typical CF has several actor roles that a member of a MSS-ENTITY fills. The System must support:

Table 6. User Groups and Attributes

Actor	Attribute
Resource Donor	Has the authority to donate resources to GENI in exchange for services
Administrator	Manages all aspects of the CF within the organization in accordance with MSS Center policies and organizational policies, to include: • Availability • Access • Control • Child CF connections • Parent CF connection • Services subset
Site	Each resource has one or more site actors. For instance, each of the following resources would have one site actor: • Eucalyptus Virtual Machine Cluster • External Network Switch Cluster • Internal Network Switch Cluster
Broker	Each Site Actor has one or more brokers, who control a portion of the Site's resources
Service Manager	Manages all aspects of the CF within his or her organizational department, to include: • Availability

	• Access • Control
User	Experimenters use the resource delegated by the Service manager to conduct experiments. For business donors, the user conducts the daily business of his or her organizational department, such as management, information technology, marketing, manufacturing, shipping, and customer service

2.6 Assumptions and Dependencies

Please find this section in *Software Requirements Specification for Mutualistic Software Services, Version 1.0*

- 2.6 Assumptions and Dependencies

2.7 Stakeholders

Table 7. Stakeholders

Stakeholder	Major Value	Attitudes	Major Interests	Constraints
NITRD	MSS helps meet its objective to increase GENI experimental resources and bolster the U.S. Ignite program	Concerns about sustained funding for MSS, no matter how useful; cost vs. gain in resources	MSS must be less than the requisite "seed" money; MSS must help meet NITRD objectives	MSS must show a proven increase in GENI donations
NSF	Increasing LSN size meets NITRD "at-scale" experimental requirement	Concern about cost vs. gain in resources; contingent on GENI leadership attitudes	MSS cost must be less than requisite NSF "seed" money	GENI must maintain a philanthropic public image
GENI PI	MSS enables "at-scale" experimentation	Relatively small risk for the reward; phased implementation lowers risk	Gain in resources must greatly exceed business/ academic contributions	MSS development should not hamper experimentation

GENI GPO	Control framework growth ensures project continuation	MSS management may overshadow GENI development of other fields	Donor's interest and perceived value in MSS remains high	MSS management must not outweigh GENI management
GENI Experimenters	"In the wild" resource availability; allows for realistic scenarios	Concern about opt-out policy allowing donors to pull resources in mid-experiment; experimental isolation	At-scale experimentation and resource variety	None
MSS Developers	Job opportunities; service sharing and potential for monetization; GPL enables code modification, open-source community support	Concerned whether version 1 features will provide enough value to gain/retain donors	Creating new opportunities for service distribution based on emerging research	CFs must deliver MSS to customers to keep a low development overhead.
Control Frameworks	Increases CF size; increases experimental value; possibly increases NSF funding	Concern about upkeep of MSS	MSS must need minimal CF upkeep; must not make CF vulnerable to cyber attack	MSS must not lose donors due to poor service
GENI Meta-Operations Center	Increases job security	Concerns about manning for additional workload	Each service must have its own maintain its own support structure	GMOC requires additional funding for additional workload
GENI Donors	Relatively free services	Concerns about slowed computer/Internet; experiment isolation	MSS should provide reasonable value for the donation	MSS should not hamper organizational operations

General Public	Provides a way to get involved in the future of the Internet; puts emerging technologies in their hands now	Concerns about increased taxes; whether increased funding is better than current "pay-for" services.	Whether MSS will charge money for services.	MSS should remain "free" to the public

3. System Capabilities, Conditions, and Constraints

Software Requirements Specification for Mutualistic Software Services Version 1.0 lists:

- 2. Overall Description
- 3. System Features
- 4. Functional Requirements
- 5. External Interface Requirements
- 6. Other Non-functional Requirements

This section addresses the system capabilities, conditions, and constraints necessary to implement the above with the functional attributes, non-functional attributes, and constraints identified in this document. The following tables contain the architectural rule, the SRS reference, the solution, the location where the software resides, the solution maintainer, and the owner who controls the solution.

3.1 Capabilities

Table 8. Software Capabilities

Rule	Version 1.0	Version 2.0	Ref.	Solution	Location	Creator	Maintainer	Owner
CP1		X	FE1 CD4	CF compares MSS-AFFILIATE to federation standards	MSS-ORIGIN	Developer	Developer	MSS-CENTER
CP2		X	FE4 CD4	CF compares MSS-AFFILIATE to connection standards	MSS-ORIGIN	Developer	Developer	MSS-CENTER
CP3	X	X	FE3 FE4 CD2 SI2.1	Service interface describes system updates and services	MSS-ENTITY	Developer	Developer	MSS-CENTER
CP4	X	X	FE1 FE2 FE3 FE4 CD4	Authentication Process authenticates/de-authenticates MSS-AFFILIATES	MSS-ENTITY	Developer	Developer	MSS-CENTER

CP5	X	X	FE3 FE4 SI1.2 CI3	Encryption/Decryption process encrypts/decrypts data	MSS-ENTITY	Developer	Developer	MSS-CENTER
CP6	X	X	FE3 FE4 SI1.5	Validation Process validates transmissions	MSS-ENTITY	Developer	Developer	MSS-CENTER
CP7	X	X	FR6 FR7 CP1 CP2 CP3 CP4 CP5 CP6 SI1.3 CI3	Service Interface delivers and receives service updates and services	MSS-ENTITY	Developer	Developer	MSS-CENTER
CP8	X	X	FE5 FE6 CP3 CP4 CS3 CS4 SI1.4	Web interface describes services subset	MSS-ENTITY	Developer	Developer	MSS-CENTER

CP9	X	X	FE6 SI6.1 SI7.1 SE1 SE2 SE5 SE9	Authentication Process authenticates a person	MSS-ENTITY	Developer	Developer	MSS-CENTER
CP10		X	FE6 CP3 CP5 CP6 CP9 SI2.2 SI2.5 SI7.2 SI7.3 CI3 SE1 SE2 RE1 RE2	NewService interface uploads new service	MSS-CENTER	Developer	Developer	MSS-CENTER
CP11	X	X	FR3 FR5 CD12	MSS interface describes users and donors	MSS-ENTITY	Developer	Developer	MSS-CENTER

3.2 Conditions

Table 9. Software Conditions

Rule	Version 1.0	Version 2.0	Ref.	Solution	Location	Creator	Maintainer	Owner
CD1	X	X	SI3.1 SI4.1 SI5.1 SE1 SE2	MSS-ENTITY has a CF installed	MSS-ENTITY	CF	MSS-ENTITY	CF
CD2	X	X	FR6 FR7 SI1.6 SI2.3 RE1 RE2	File system holds services	MSS-ENTITY	Developer	Developer	MSS-CENTER
CD3	X	X	OE7 FE3 FE4 CD1 SI1.7	Services database holds system updates and services data	MSS-ENTITY	Developer	Developer	MSS-CENTER
CD4	X	X	OE7	CF database holds MSS-ENTITY data	MSS-ENTITY	Developer	Developer	MSS-CENTER
CD5	X	X	OE7	User database holds user data	MSS-ENTITY	Developer	Developer	MSS-CENTER

CD6		X	CI1 CI2 SE1 SE2 SE3 SE4 SE5 RE1 RE2	MSS-CENTER receives emails	MSS-CENTER	Developer	Developer	MSS-CENTER
CD7		X	CI4 SE1 SE2 SE3 SE4 SE5 RE1 RE2	MSS-CENTER receives files	MSS-CENTER	Developer	Developer	MSS-CENTER

CD8		X	UD1 UD2 UD3 OE7 CO1 CO2 CO4 FE7 FE8 UI1 UI2 UI3 UI4 UI5 UI6 SI2.4 RE1 RE2	User web interface accesses GENI and service information, including: HTML File Downloads	MSS-CENTER	Developer	MSS-DEVELOPER	MSS-CENTER

CD9		X	UD4 UD5 UD6 UD7 DE4 DE5 DE6 CS3 CS4 RE1 RE2	Administrator web interface accesses management information, including: HTML File Downloads	MSS-CENTER	Developer	MSS-DEVELOPER	MSS-CENTER
CD10	X	X	AS1 AS2 AS3 AS4 SI1.11	MSS interface connects many MSS-USERS to many MSS-ENTITIES and MSS-RESOURCES	MSS-ENTITY	Developer CF	CF	CF
CD11	X	X	AS4 SI1.8 SI1.9 SI1.9.1 SI1.10 SI1.11 CI3	CF interface sends heartbeats to MSS-ORIGIN	MSS-ENTITY	Developer CF	CF	CF
CD12	X	X	AS4 SI1.8	CF interface receives heartbeats from MSS-AFFILIATE	MSS-ORIGIN	Developer CF	CF	CF

CD13	X	X	AS4 FR4 CD8 SE4 SE8	CF interface encodes heartbeats to MSS-ORIGIN	MSS-AFFILIATE	Developer CF	CF	CF
CD14		X	AS4 FR4 CD8 SI1.12 CI3	MSS interface decodes heartbeats from MSS-AFFILIATE	MSS-ORIGIN	Developer CF	CF	CF
CD15	X	X	FR1 FR2 CP5 IN1 IN2	Database holds user and donor data	MSS-ENTITY	Developer	Developer	MSS-CENTER

3.3 Constraints

Table 10. Software Constraints

Rule	Version 1.0	Version 2.0	Ref.	Solution	Location	Creator	Maintainer	Owner
CS1	X	X	OE1 OE5 OE6 CO3 SE4 SE5 SE6 SE7 SE9 FL1 FL2 FL3	MSS must operate on Linux platforms initially and port to several types of hardware platforms later	MSS-ENTITY	Developer	Developer	MSS-CENTER
CS2	X	X	OE2 OE6 CO3	MSS libraries must operate on Unix and Linux platforms initially and Windows platforms later	MSS-ENTITY	Developer	Developer	MSS-CENTER

CS3	X	X	OE3 CO1 CO2 CO4 UI1 UI2 UI3 UI4 UI5 UI6	MSS web interfaces must display correctly on major web browsers	MSS-ENTITY	Developer	Developer	MSS-CENTER
CS4	X	X	OE4 CO1 CO2 CO4 UI1 UI2 UI3 UI4 UI5 UI6 SE1 SE2	MSS web interfaces must display in English, with the ability to add other languages later	MSS-ENTITY	Developer	Developer	MSS-CENTER
CS5		X	OE5 SE1 SE2	MSS must verify donor location data to ensure US location	MSS-ENTITY	Developer	Developer	MSS-CENTER

CS6		X	SI1.1	MSS-CENTER sets MSS standards				

4. System Characteristics

4.1 Autonomy

The System shall:

- Not require MSS-CENTER interaction to use existing parent services.
- Be separable from its parent node without incurring interruption of current services

4.2 Integrability

The System shall:

- Install on a CF with no modification to the CF, excepting the remote server. For instance:
    - ORCA only requires a MSS scripts and configuration file adjustments using commonly installed software, such as Apache, MySQL, and PHP.

4.3 Extensibility

The System shall:

- Easily add child nodes and resources, for instance:
    - An administrator adds a child CF by inserting the child's IP, port, encryption key, or other identity information into a file.

4.4 Portability

The System shall:

- (Version 1.0 and 2.0) Operate on Linux
- (Version 2.0) Operate on Unix, and Windows OS
- Operate on servers

4.5 Usability

The System shall:

- Be geared towards users in a non-technical field, such as clerical.
- Be geared towards administrators in a non-technical field, for instance:
    - MSS will offer helpful instructions without technical jargon.

4.6 Securability

The System shall:

- (Version 1.0 and 2.0) At a minimum, require a Unique user name and password for login
- (Version 2.0) Provide an option for elevating secure access for different user groups, for instance:
    - User – Unique user name and password for login
    - Service Manager – add Captcha login
    - Owner/Administrator – add Common Access Card login
    - This option may itself be a service

4.7 Credibility

The System shall:

- Provide a means to verify the service as a MSS service, for instance:
    - list a MD5 hash value
    - Provide a MSS signature
- Easily connect to a parent sponsor, or change to a new parent sponsor

4.8 Interoperability

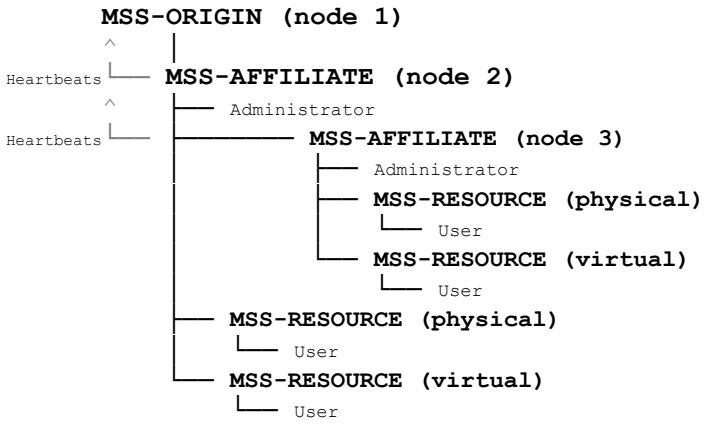
The System shall:

- Easily exchange information between parent and child sponsors, for instance:
    - MSS parent services are queriable and downloadable through a web browser

5. Architectural Plan

MSS relies on the notion that experimenters use the GENI architecture to develop at least some unique services that other organizations cannot offer. In fact, other experimenters, developers, and users find these services valuable enough to donate their resources to the experimenters in exchange for these services. With more resources available, more experimenters can create more services, and the mutualistic cycle continues.

Version 1.0 provides a skeleton architecture in preparation for Version 2.0 by creating a service advertisement and delivery system between two MSS-AFFILIATES that administrators can use to download entire subsets of services, and that resource users can use to query and download individual services. MSS checks for the heartbeats sent to the MSS-ORIGIN before allowing administrators and users to download services onto the actor. The following figure depicts the hierarchical relationships between the entities.



In Version 2.0, this architecture combines MSS Version 1 & 2 software requirements. It provides a means for the MSS-ORIGINS to plug into the MSS-CENTER and receive the services that MSS-DEVELOPERS upload and maintain. An MSS-ORIGIN (node 1) then advertises the applicable subset of services, and MSS-AFFILIATES (nodes 2 & 3) may choose all, or a subset of its parent's services. At each node, users can download individual services on to authorized MSS-RESOURCES.

In order to receive the services in Version 1.0 and 2.0, donors simply host a CF and dedicate a portion of their resources for GENI experimentation. After donors advertise their resources as available by sending heartbeats to its MSS-ORIGIN, it may download services from its parent MSS-AFFILIATE.

The hardware footprint of the system depends entirely on how many child CFs the parent owns. MSS Version 2.0 may look like this:

- MSS-CENTER will require a separate server or servers for each MSS component
- MSS-ORIGIN/AFFILIATES will require several servers: one to hold MSS, one to hold services, and several for its GENI CF

- MSS-AFFILIATES, at the lowest level, will only require a portion of one server, workstation, or laptop and serve as its own resource.
- MSS-RESOURCE, at the lowest level, will only require a portion of one server, workstation, or laptop.

"MSS-AFFILIATES, at the lowest level…" (above) forms a special case in which all software components reside on one server. To meet this goal, the following section describes an architecture that resides on one server, and that conforms to MSS-CHILD requirements.

5.1 Remote Server Model

According to *Software Requirements Specification for Mutualistic Software Services Version 1.0*, a MSS-AFFILIATE must:

- [Maintain a] Constant connection to the Internet, through a Cable Modem, Direct Service Line, or other network connection
- [Be a] A computer resource, provisioned according to CF standards.

This means that a MSS-AFFILIATE may only have a single internet connection and must reside on a single computer, which is not how one sets up a typical CF.

Normally, a CF resides on several computers, with one head node and several worker nodes. The CF may have several IP addresses, or it might have only one IP address and use a separate Network Address Translation (NAT) server to shuffle traffic between the public address space and a private address space where the CF resides. In a canonical ORCA installation, a CF must have at least two computers or several IP addresses to run all of the required software components [13], which violates the SRS minimum hardware interface requirements.

This is not an easy problem to solve because the virtualization software (e.g., Xen) requires direct access to the computer kernel to run a hypervisor, and the virtualization management software (e.g., Eucalyptus) requires direct access to the networking tables (e.g., iptables) to allocate public or pseudo-public IP addresses using Dynamic Host Control Protocol (DHCP). Therefore, one cannot use NAT on the hypervisor, and one cannot run Eucalyptus on a virtual machine because the Eucalyptus DHCP server would hand out addresses to other computers on its network that are not Eucalyptus Instances.

The remote server model creates an unexploited solution (at least in the author's research) that uses a virtual NAT server to control the network connections, shields the outside network from the Eucalyptus DHCP server, and leaves the Eucalyptus software to control virtualization on the hypervisor. Appendix B contains a component diagram that illustrates how one might implement a remote server to act as a MSS-AFFILIATE.

5.2 Context Diagrams and Data Model

Appendix B contains two context diagrams and one data model. The context diagrams CD-1 and CD-2 represent the external connections, and the data flow model DM-1 represents the generation and flow of data for each MSS entity in version 1.0.

5.2.1 MSS-CENTER

- (Version 2.0) Hosts all GENI CFs to utilize their heartbeat functionalities
- (Version 2.0) Receives heartbeats, service information requests, and service requests from its child CFs
- (Version 2.0) Delivers service information and service requests to its child CFs
- (Version 2.0) Delivers service content to resource users and administrators

5.2.2 MSS-ORIGIN

- Develops and maintains a GENI CF
- Has one or more MSS-AFFILIATES
- Receives heartbeats from its MSS-AFFILIATES
- (Version 2.0) Receives service information requests and service requests from its MSS-AFFILIATES
- (Version 2.0) Delivers service information and service requests to its MSS-AFFILIATES

5.2.3 MSS-AFFILIATE

- Hosts a GENI CF and maintains resources
- Delivers heartbeats to its MSS-ORIGIN
- Receives service information requests, and service requests from its child MSS-AFFILIATES
- Delivers service information and service requests to its child MSS-AFFILIATES

5.2.4 MSS-RESOURCE

- Authorized by an MSS-ENTITY to receive services, make service information requests, and receive service information
- Has authorized MSS-USERS

5.2.5 MSS-USER

- Authorized to use at least one MSS-RESOURCE
- Requests service information and services from his or her MSS-RESOURCE to his or her MSS-ENTITY
- Receives service information and services from his or her MSS-RESOURCE to his or her MSS-ENTITY
- (Version 2.0) Requests service content from MSS-CENTER
- (Version 2.0) Receives service content from MSS-CENTER

5.2.5 MSS-DEVELOPER

- Usually a GENI experimenter
- Creates MSS services

- (Version 2.0) Delivers service information, service content, and services to MSS-CENTER
- Requests resources from MSS-AFFILIATES for experimentation

5.3 Component Model

Appendix B contains one component diagram. The component diagram CM-1 represents the external connections, and the data flow model DM-1 represents the generation and flow of data for each MSS-ENTITY.

5.3.1 Databases

5.3.1.1 Service Data

- Holds the subset of service attribute data of its MSS-ENTITY

5.3.1.2 CF Data

- Holds the MSS-ENTITY data for itself. For instance, ORCA uses database *orca*

5.3.1.3 User Data

- Holds the authorized user's data for itself and its child MSS-AFFILIATES

5.3.2 Components

5.3.2.1 CF

- MSS-ORIGIN presents an interface to its children to receive heartbeats
- MSS-AFFILIATES deliver heartbeats to its MSS-ORIGIN
- Provides a CF interface with a list of current donors. For instance, ORCA provides the *ORCA Actor Registry* [14]
- Provides a list of available resources to GENI experimenters and users
- Enables GENI experimenters and users to request resources

5.3.2.2 User Interface

- Uses the Authentication process to verify actor and user requests
- Receives a list of MSS-ENTITY resources from the CF
- Receives user service information and service requests
- Provides service delivery information to the Service Interface
- Triggers the Service Interface to deliver services
- Checks for heartbeats on the CF interface
- Triggers the Service Interface to download services

5.3.2.3 Service Interface

- Receives a service information request from the User Interface
- Receives a service download request from the User Interface
- Uses the Encryption process to encrypt services
- Uses the Decryption process to decrypt services

- Uses the Validation process to validate services
- Receives services from parent MSS-ENTITY
- Delivers services to child MSS-AFFILIATES

5.3.2.4 User Web Site

- (Version 2.0) Delivers service content to service users
- (Version 2.0) Enables MSS-CENTER/MSS-DEVELOPERS to add service content
- (Version 2.0) Delivers GENI information as required by MSS-CENTER
- (Version 2.0) Enables users to email MSS-CENTER/MSS-DEVELOPERS

5.3.2.5 Administrator Web Site

- (Version 2.0) Delivers administrative content to MSS administrators
- (Version 2.0) Allows MSS-CENTER/MSS-DEVELOPERS to add administrative content
- (Version 2.0) Delivers GENI information as required by MSS-CENTER
- (Version 2.0) Enables users to email MSS-CENTER/MSS-DEVELOPERS

5.3.2.6 Developer Uploads

- (Version 2.0) Allows enrolled MSS-DEVELOPERS to add services
- (Version 2.0) Allows enrolled MSS-DEVELOPERS to add service descriptions
- (Version 2.0) Enables MSS-CENTER to approve services
- (Version 2.0) Enables MSS-CENTER to approve service descriptions

5.3.3 Service Repository

5.3.3.1 File System

- Holds MSS services
- (Version 2.0) Holds all MSS services for MSS-CENTER
- Holds a subset of parent services, to include the entire subset, for its MSS-AFFILIATES

6. Architecture Trade-off Analysis Method (ATAM)

6.1 Purpose

Software Architecture in Practice (2003) explains that the ATAM "… reveals how well an architecture satisfies particular quality goals, and (because it recognizes that architectural decisions tend to affect more than one quality attribute) it provides insight into how quality goals interact—that is, how they trade off" [1]. To this end, this section evaluates MSS in order to verify the architectural decisions that make up the MSS architecture.

6.2 Main Architectural Drivers

From 2.3 Non-functional Attributes

The System shall have:

- Portability
- Autonomy
- Securability
- Credibility
- Integrability
- Extensibility
- Interoperability
- Usability

6.3 Business Goals

- MSS-DEVELOPERS can advertise services
- GENI CFs can exchange services for a portion of a resource owner's resources
- An individual member of the public can donate his or her resources in exchange for services
- GENI CFs grow as a result of exchanging services for resources
- A break in heartbeats will not affect the operation of current services
- Decentralized management to reduce MSS-CENTER operational costs

6.4 Major Stakeholders

Mutualistic Software Services Software Requirements Specification, Version 1.0 lists:

- 2.2 User Classes and Characteristics

Favored User Classes:

- Resource donors
- CFs
- Developers
- GENI Project Office (GPO)

6.5 Architectural Approaches

MSS:

- separates databases according to the data held
- requires that a MSS-AFFILIATE send MSS-ORIGIN heartbeats to download new services
- requires that only authorized users on authorized MSS-AFFILIATES and MSS-RESOURCES can download new services
- CF Interface:
  - provides the CF Interface with a list of donors [14]
  - provides the User Interface with a list of currently connected MSS-ENTITIES
- User Interface:
  - authenticates MSS-ENTITIES and users, and validates service requests
  - requests service information and services from its parent
  - validates donation status from its MSS-AFFILIATE and its children
  - delivers service information to its children
  - triggers the Service Interface service transmission and reception
- Service Interface:
  - transmits and receives services
  - encrypts and decrypts services for transmission
  - validates whether services are successfully delivered or received

6.6 Utility Tree

Table 11. Utility Tree

Quality Attribute	Attribute Refinement	Case	Scenario	Rating (Importance, Difficulty)
Portability	OS Change	1.	A donor installs MSS on a new Linux OS distribution with the ORCA control framework. This OS has a newer kernel than the old OS.	(H,M)
		Best:	MSS works with the new OS	
		Worst:	MSS requires older libraries than the OS supports	
	CF Change	2.	A donor installs MSS on a Unix OS with the protoGENI control framework. The CF is compatible with Unix, but the CF does not use PHP, on which the MSS interface relies.	(H,M)
		Best:	PHP interpreter and libraries are available from the Unix package manager, such as aptitude.	
		Worst:	The administrator must install the PHP interpreter and appropriate libraries from the PHP repository.	

	Resource	3.	A donor installs MSS resources on a virtual machine.	(H,L)
		Best:	MSS services work on the virtual machine.	
		Worst:	MSS services will not work on the virtual machine.	
Autonomy	Service Delivery	4.	A user requests a *new* service from its parent MSS-ENTITY, but the MSS-ENTITY is not sending heartbeats to MSS-ORIGIN.	(H,H)
		Best:	The MSS-ENTITY *does not* deliver the *new* service.	
		Worst:	The MSS-ENTITY *does* deliver the *new* service.	
		5.	A user requests a currently owned service from its parent MSS-ENTITY that is sending heartbeats.	(H,H)
		Best:	The parent CF *does* deliver the service.	
		Worst:	The parent CF *does not* allow the service download.	
		6.	A MSS-RESOURCE requests a service from its parent MSS-ENTITY, but the MSS-ENTITY is not sending heartbeats.	(H,H)
		Best:	The CF *does not* deliver the service to the resource.	
		Worst:	The CF *does* deliver the service to the resource.	
	MSS-ENTITY Rules	7.	A MSS-ENTITY has more stringent donor rules than its parent. Its MSS-AFFILIATES are unhappy with the strict rules.	(H,M)
		Best:	Another MSS-ENTITY advertises donor rules that are in keeping with this MSS-AFFILIATE. The child switches parents by contacting another MSS-ENTITY, providing MSS-ENTITY, MSS-RESOURCE, and MSS-USER data, and sending heartbeats to MSS-ORIGIN.	
		Worst:	The children of the MSS-ENITITY with strict donor rules remove their resources permanently from GENI.	
		8.	A MSS-ENTITY has less stringent donor rules than its parent.	(M,L)

		Best:	The resource donors enjoy the less stringent rules and the MSS-ENTITY maintains the hardware necessary to support the increase in donors.	
		Worst:	This attracts more donors than the MSS-ENTITY can adequately serve. The donors permanently remove their resources from GENI	
Securability	Service Encryption	9.	A MSS-ENTITY stops using service encryption to deliver services. This allows hackers to "pluck" the services from the Internet during transmission.	(H,L)
		Best:	The parent of the MSS-ENTITY starts using encryption keys, and once again starts encrypting service transmissions.	
		Worst:	Several MSS-ENTITIES stop encrypting services, and the services are available to a large number of non-donating resource owners, which devalues resource donation in exchange for services because most of these services are now free. Donations dwindle to a point where MSS loses usefulness.	
	Service Fraud	10.	A donor connects to a CF and obeys all rules, except he or she gives GENI services away to others who are not donating.	(H,H)
		Best:	The parent notices a large number of downloads by the donor, and enforces MSS rules that require service donations for services.	
		Worst:	The parent does not notice the increased downloads from the offending donor, and the services are available to a large number of non-donating resource owners, which devalues resource donation in exchange for services because most of these services are now free. Donations dwindle to a point where MSS loses usefulness.	
	Donation Fraud	11.	A donor connects to a CF only long enough to download services and then intentionally stops sharing resources with the CF. Once a week, this same donor connects again, downloads new services, and disconnects from the parent CF.	(H,H)
		Best:	The MSS-PARENT keeps a history of donor connection times. The MSS-PARENT rules are strict	

			enough that the donor cannot download more new services until the donor meets the resource donation threshold.	
		Worst:	The parent does not notice the connection history of the offending donor, and too few resources are available to experimenters than MSS maintenance is worth.	
	MSS-ENTITY Encryption	12.	A branch of MSS-ENTITIES uses the same encryption keys for its MSS-AFFILIATES.	(H,L)
		Best:	The parent at the highest point in the branch starts changing encryption keys after a specified time, which requires its children down the chain to follow suit out of necessity.	
		Worst:	One MSS-ENTITY is hacked, which leaves all of the CFs in the branch vulnerable to attack.	
Credibility	CF Fraud	13.	A donor connects to a disreputable parent CF, and user identities, encryption keys, and other information are stolen.	(H,M)
		Best:	Encryption keys are replaced.	
		Worst:	User identity information containing sensitive personally identifiable information is in criminal hands.	
	Poor Service Operation	14.	The service does not perform as the developer described.	(M,M)
		Best:	Users voice their displeasure on the User Web Site, and the developer responds by enhancing the service operation or describing it more accurately.	
		Worst:	Poor service operation is common throughout MSS, which drives users away from donating resources.	
	Poor Service Descriptions	15.	The service does not have the functionality that the service description provides	(M,M)
		Best:	Users voice their displeasure on the User Web Site, and the developer responds by enhancing the service operation or describing it more accurately.	
		Worst:	Poor service descriptions are common throughout	

			MSS, which drives users away from donating resources.	
	Poor Service Instructions	16.	The service does not have clear, accurate directions for service use, or is missing directions.	(M,M)
		Best:	Users voice their displeasure on the User Web Site, and the developer responds by enhancing the service operation or describing it more accurately.	
		Worst:	Poor service instructions are common throughout MSS, which drives users away from donating resources.	
Integrability	CF Versions	17.	A new CF version comes out, and MSS must work with both versions.	(H,H)
		Best:	MSS requires no change to work with the new version, or the changes are minimal because MSS is developed with integrability in mind.	
		Worst:	MSS requires significant changes to work with each version of every CF, and the lack of "plug-n-play" functionality overburdens MSS developers	
	Database type	18.	CFs choose to use a different database than the design prescribes. Some database queries do not work.	(L,M)
		Best:	The database uses similar statements to the intended database, and so requires little work to integrate.	
		Worst:	Each CF decides to use different databases with dissimilar statements, and so developers must tailor software commands to each database type.	
	CF Interface Compatibility	19.	A donor decides to change CFs without changing MSS.	(H,M)
		Best:	The CF Interface requires no change to work with the new CF.	
		Worst:	The CF interface has significant compatibility problems.	
	Service Interface	20.	Database version changes format of its query structure.	(L,L)

	Compatibility	Best:	The Service Interface requires little work to re-write database queries.	
		Worst:	The Service Interface database queries require complete re-work.	
Extensibility	Parent Nodes	21.	MSS-ENTITY has poor service because it does not have the hardware to support all of its children.	(H,L)
		Best:	Resource donors easily switch to a new MSS-ENTITY that perhaps the current MSS-ENTITY recommends.	
		Worst:	MSS-ENTITY does not acquire the hardware, nor does the donor switch. Service is so poor that donors leave MSS.	
	Tree Balancing	22.	Too many child nodes create a long tree branch from MSS-CENTER, which slows down cascading service delivery.	(H,M)
		Best:	This MSS branch re-links itself to balance the sub-tree, and services flow quickly.	
		Worst:	MSS-ENTITIES continue to ignore sub-tree lengths, and service delivery slows to a point that donors stop donating resources.	

7. Cost Benefit Analysis Method (CBAM)

Len Bass et al., summarize the purpose of the CBAM in *Software Architecture in Practice* (2003):

> The CBAM is an iterative elicitation process combined with a decision analysis framework. It incorporates scenarios to represent the various quality attributes. The stakeholders explore the decision space by eliciting utility-response curves to understand how the system's utility varies with changing attributes. The consensus basis of the method allows for active discussion and clarification amongst the stakeholders. The traceability of the design decision permits updating and continuous improvement of the design process over time [1].

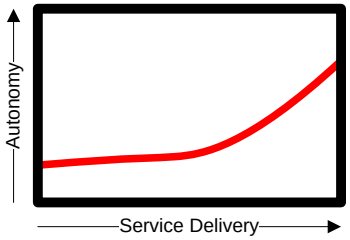
The MSS CBAM meets this goal by creating a utility response curve for each case in the Utility Tree rated (H, H), (H, M), or (M, H). These cases are ordered according to their:

- Utility response curves and ratings
- The architectural strategy to combat the negative effects of the worst-case scenarios
- The cost of implementing the strategy
- List potential risks that the strategy may not mitigate

7.1 Utility Response Curves

7.1.1 Autonomy vs. Service Delivery

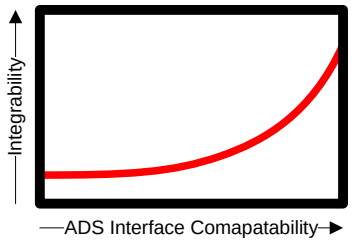
Table 12. Autonomy vs. Service Delivery

Service Delivery (H,H)	4.	A user requests a *new* service from its parent MSS-ENTITY, but the parent CF is not sending heartbeats to MSS-ORIGIN.	
	Best:	The MSS-ENTITY does not deliver the *new* service.	
	Worst:	The user can download the *new* service, even though the MSS-ENTITY is not sending heartbeats.	
	5.	A user requests a currently owned service from its parent MSS-ENTITY, but its parent MSS-ENTITY does not have a parent sponsor.	
	Best:	The parent MSS-ENTITY delivers the service because it is sending heartbeats.	
	Worst:	The parent CF does not allow the service download, even though it is sending heartbeats.	
	6.	A user requests a currently owned service from its parent MSS-ENTITY, but the MSS-ENTITY is not sending heartbeats to MSS-ORIGIN.	
	Best:	The MSS-ENTITY does not deliver the service to the	

		resource.	
	Worst:	The CF does deliver the service to the resource.	
Strategy / Cost	1.	Strictly adhere to the MSS architecture, in which heartbeats must go to MSS-ORIGIN.	LOW – Ensure developers adhere to the MSS Architecture
	2.	Strictly adhere to the MSS architecture ,in which the MSS-ENTITY must be enrolled in the ENTITY database	LOW – Ensure developers adhere to the MSS Architecture
	3.	Strictly adhere to the MSS architecture, in which the user must be enrolled in User database	LOW – Ensure developers adhere to the MSS Architecture

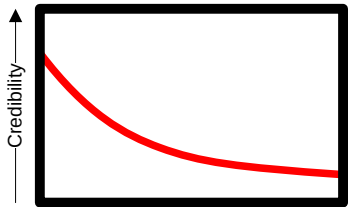
7.1.2 Integrability vs. User Interface Compatibility

Table 13. Integrability vs. User Interface Compatibility

User Interface Compatibility	19.	A donor decides to change CFs without changing MSS.	
	Best:	The MSS Interface requires no change to work with the new CF.	
(H,M)	Worst:	The MSS interface has significant compatibility problems.	
Strategy / Cost	1.	Use the same software as the CF, which is generally MySQL, PHP, and BASH scripts.	LOW – This approach fits with the architectural model
	2.	Use the same software libraries as the CF. The CFs already have authentication, decryption, and validation processes in place, and so using these causes as few compatibility problems as possible.	LOW – This approach fits with the architectural model

7.1.3 Credibility vs. CF Fraud

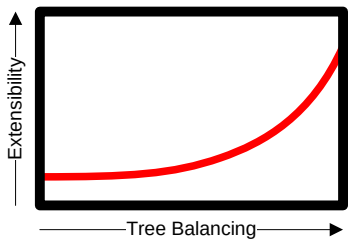
Table 14. Credibility vs. CF Fraud

CF Fraud	13.	A donor connects to a disreputable parent CF, and user identities, encryption keys, and other information are stolen.	
(H,M)	Best:	Encryption keys are replaced.	

	Worst:	User identity information containing sensitive personally identifiable information is in criminal hands.	
Strategy / Cost	1.	Trust Campaign – begin a real-world campaign, such as on the NSF, GENI, CF, and MSS-CENTER websites, explaining that a donor should only attach to a reputable organization. Avoid unknown parents.	MEDIUM – MSS-CENTER can add this content to which other agencies link, but it adds additional development and maintenance costs.
	2.	MSS Instructions have notices and warnings prominently displayed that warn the donor to only join a trustworthy MSS parent, such as a university or respected business.	MEDIUM – MSS-CENTER can add this content to which other agencies link, but it adds additional development and maintenance costs.

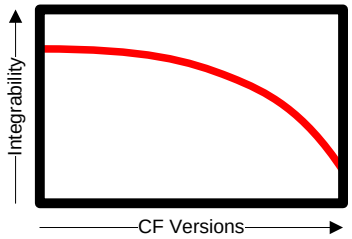
7.1.4 Extensibility vs. Tree Balancing

Table 15. Extensibility vs. Tree Balancing

Tree Balancing	22.	Too many child nodes create a long tree branch from MSS-CENTER, which slows down cascading service delivery.	
(H,M)	Best:	This MSS branch re-links itself to balance the sub-tree, and services flow quickly.	
	Worst:	MSS-ENTITIES continues to ignore sub-tree lengths, and service delivery slows to a point that donors stop donating resources.	
Strategy / Cost	1.	Self-balancing binary search tree – In computer science, a self-balancing (or height-balanced) binary search tree is any node based binary search tree that automatically keeps its height (number of levels below the root) small in the face of arbitrary item insertions and deletions [12].	HIGH – implementing this requires MSS-CENTER to know all of its nodes and MSS-ENTITY rules at each level.
	2.	Recommend to MSS-ENTITIES on the Administrator Web Site to keep balanced branches to the greatest extent possible to speed service delivery.	MEDIUM – MSS-CENTER can add this content, to which other agencies link, but it adds additional development and maintenance costs.

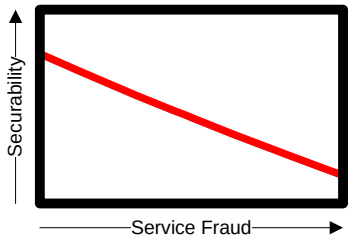
7.1.5 Integrability vs. CF Versions

Table 16. Integrability vs. CF Versions

CF Versions	17.	A new CF version comes out, and MSS must work with both versions.	
(H,H)	Best:	MSS requires no change to work with the new version, or the changes are minimal because MSS is developed with integrability in mind.	
	Worst:	MSS requires significant changes to work with each version of every CF, and the lack of "plug-n-play" functionality overburdens MSS developers	
Strategy / Cost	1.	Use the same software as is loaded on the CF, which is generally MySQL, PHP, and BASH scripts.	LOW – This approach fits with the architectural model
	2.	Use the same software libraries as the CF. The CFs already have authentication, decryption, and validation processes in place, and so using these causes as few compatibility problems as possible.	LOW – This approach fits with the architectural model

7.1.6 Securability vs. Service Fraud

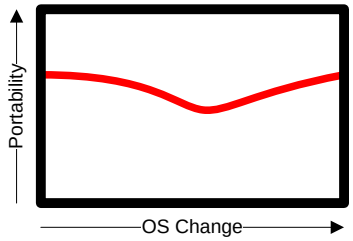
Table 17. Securability vs. Service Fraud

Service Fraud	10.	A donor connects to a CF and obeys all rules, except he or she gives GENI services away to others who are not donating.	
(H,H)	Best:	The parent notices a large number of downloads by the donor, and enforces MSS rules that require service donations for services.	
	Worst:	The parent does not notice the increased downloads from the offending donor, and the services are available to a large number of non-donating resource owners, which devalues resource donation in exchange for services because most of these services are now free. Donations dwindle to a point where MSS loses usefulness.	
Strategy / Cost	1.	Management – include color coding and other means to highlight children that download more than a given amount.	RISK – This strategy may not mitigate the worst-case scenario MEDIUM – The parent CF needs a needs to know how many downloads a child has, to recognize each child (by its

			MAC address, for instance) and to permanently exclude the offending child.
	2.	Add MAC address to MSS-ENTITY database	MEDIUM – Adding a MAC address is not difficult, but GENI CFs do not currently require donors to provide a MAC address. GENI must approve this unilaterally.

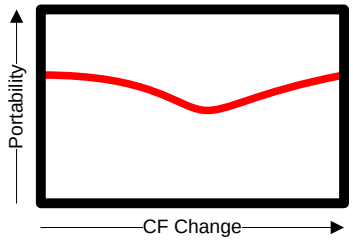
7.1.7 Portability vs. OS Change

Table 18. Portability vs. OS Change

OS Change	1.	A donor installs MSS on a new Linux OS distribution with the ORCA control framework. This OS has a newer kernel than the old OS.	
(H,M)	Best:	MSS works with the new OS	
	Worst:	MSS requires older libraries than the OS supports	
Strategy / Cost	1.	MSS Instructions stress using LTS OS distributions to avoid this problem altogether.	MEDIUM – MSS-CENTER can add this content to which other agencies link, but it adds additional development and maintenance costs. PHP, BASH, and MySQL are widely supported by all Linux distributions. The best strategy to combat this is to avoid "bleeding edge" distributions, such as beta and daily build distributions, which are generally used for testing and not for enterprise level deployments.

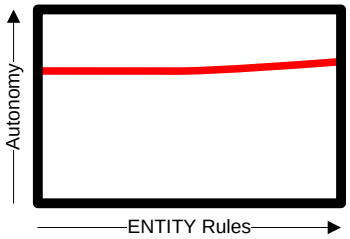
7.1.8 Portability vs. CF Change

Table 19. Portability vs. CF Change

CF Change (H,M)	2.	A donor installs MSS on a Unix OS with the protoGENI control framework. The CF is compatible with Unix, but the CF does not use PHP, on which the MSS interface relies.	
	Best:	PHP interpreter and libraries are available from the Unix package manager, such as *ports.*	
	Worst:	The administrator must install the PHP interpreter and appropriate libraries from the PHP repository.	
Strategy / Cost	1.	MSS Instructions stress using LTS OS distributions to avoid this problem altogether.	MEDIUM – MSS-CENTER can add this content to which other agencies link, but it adds additional development and maintenance costs. PHP, BASH, and MySQL are widely supported by all Unix distributions. The best strategy to combat this is to avoid "bleeding edge" distributions, such as beta and daily build distributions, which are generally used for testing and not for enterprise level deployments.

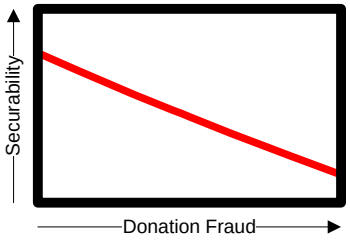
7.1.9 Autonomy vs. MSS-ENTITY Rules

Table 20. Portability vs. CF Change

MSS-ENTITY Rules (H,M)	7. Best: Worst:	A MSS-ENTITY has more stringent donor rules than its parent. The children of the MSS-ENTITY are unhappy with the strict rules. Another MSS-ENTITY advertises donor rules that are in keeping with this MSS-AFFILIATE. The child switches parents by contacting another MSS-ENTITY, providing MSS-ENTITY data and User data, and sending heartbeats to the MSS-ORIGIN. The children of the MSS-ENTITY with strict donor rules remove their resources permanently from GENI.	
Strategy / Cost	1.	MSS-AFFILIATES can choose whichever MSS-ENTITY meets their needs or can switch to a different parent. An MSS-AFFILIATE can also become a parent MSS-ENTITY. However, it still must obey the minimum connection, service distribution, and service delivery requirements of the ultimate parent, MSS-CENTER.	NONE – This scenario is in keeping with MSS.

7.1.10 Securability vs. Donation Fraud

Table 21. Securability vs. Donation Fraud

Donation Fraud (H,H)	11. Best: Worst:	A donor connects to a CF only long enough to download services and then intentionally stops sharing resources with the CF. Once a week, this same donor connects again, downloads new services, and disconnects from the parent CF. The MSS-ORIGIN keeps a history of donor connection times. The MSS-ENTITY rules are strict enough that the donor cannot download more new services until the donor meets the resource donation threshold. The parent does not notice the connection history of the offending donor, and too few resources are available to experimenters than MSS maintenance is worth.	

Strategy / Cost	1.	Management – Each parent MSS-ENTITY decides the required connection criteria for each child, such as how many hours of the day the child is actively donating resources.	NONE – In the MSS Architecture, the parent MSS-ENTITY can enforce its own connection rules. Each parent in the chain has this ability, and so an entire branch may be cut off from new services until it meets the higher parent's connection criteria.

7.2 Architectural Strategies

The following strategies are ranked according to its Utility rating

- Adhere to architecture requirement for heartbeats, and MSS-ENTITY and user data
- Use the same software and libraries that are on the CFs
- (Version 2.0) Add to the MSS web sites:
  - Trust Campaign – Reputable parent and child information
  - The importance of balancing child nodes
  - Define service fraud and how to prevent it.
  - Using LTS OS distributions
- (Version 2.0) Provide MSS-ENTITIES with MSS-AFFILIATE information highlighting high downloads or low donation times.
- (Version 2.0) Add MAC address to the ENTITY database.

8. References

[1] Bass, L., Clements, P., & Kazman, R. (2003). *Software Architecture in Practice* (2nd ed.). Boston, MA, USA: Pearson Education, Inc.

[2] CISE. (November 28, 2011). *US Ignite Gigabit Applications Workshop*. NSF. Retrieved on November 29, 2011, from: http://www.nsf.gov/cise/usignite/usignite_workshop.jsp

[3] BBN Technologies. (November 2011). *About GENI*. Global Environment for Network Innovation. Retrieved on 29 November, 2011, from: http://www.geni.net/?page_id=2

[4] BBN Technologies. (November 2011). *News and Events*. Global Environment for Network Innovation. Retrieved on 29 November, 2011, from: http://www.geni.net/

[5] Baldine, I. (June 7, 2011). *Possible Eucalyptus Hardware configurations*. RENCI. Retrieved on December 1, 2011, from: https://code.renci.org/gf/project/networkedclouds/wiki/?pagename=EucaHardwareConfiguration

[6] Dell. (2005). *Dell PowerEdge 2850 Server*. Dell.com. Retrieved on December 2, 2011, from: http://www.dell.com/downloads/global/products/pedge/en/2850_specs.pdf

[7] Apple. (October 21, 2008). *MacBook Pro (15-inch, Late 2008) - Technical Specifications*. Apple Inc. Retrieved on December 2, 2011, from: http://support.apple.com/kb/sp499

[8] RENCI. *Open Resource Control Architecture*. (August 13, 2011). Renaissance Computing Institute. Retrieved December 8, 2011, from http://groups.geni.net/geni/wiki/ORCABEN

[9] Citrix. *Xen*. (2011). Citrix Systems, Inc. Retrieved December 8, 2011, http://xen.org/

[10] Eucalyptus Systems. *Eucalyptus*. Eucalyptus Systems, Inc. Retrieved December 8, 2011, from http://www.eucalyptus.com/

[11] RENCI. *Eucalyptus/XCat image proxy*. (July 25, 2011). Renaissance Computing Institute. Retrieved December 2, 2011, from: https://code.renci.org/gf/project/networkedclouds/wiki/?pagename=ImageProxy

[12] Wikipedia (December 7, 2011).*Self-balancing binary search tree*. Wikipedia.org. Retrieved December 11, 2011, from: http://en.wikipedia.org/wiki/Self-balancing_binary_search_tree

[13] RENCI. (December 5, 2011). Deploying an Authority (Aggregate Manager/AM). Renaissance Computing Institute. Retrieved on March 28, 2012, from: https://geni-orca.renci.org/trac/wiki/deploy-am

[14] RENCI. (March 28, 2012). *ORCA Remote Actor Registry.* Renaissance Computing Institute. Retrieved on March 28, 2012, from: https://geni.renci.org:12443/registry/actors.jsp

Appendix A

Data Dictionary

Aggregate – a collection of components that usually comprise a system

Bandwidth - a bit rate measure of available or consumed data communication resources expressed in Gigabits/second.

CF – Control Framework

Component – a physical computer resource, such as a router, switch, computer, phone, or copy machine

Community – MSS users identified by a GENI control framework

Contributor – an entity that donates a portion of its resources to GENI

Control framework – one of four GENI architectures used to federate computer resources

Donation – Donations include nearly all types of network resources, such as computers, network routers and switches, cell phones, computer tablets, copy machines, and so on. However, MSS is only available for computers as of this writing.

Experimenter – one who conducts network research on GENI

Federate – incorporate one's computer resource into a GENI control framework

File list – hierarchical locations of files on a resource

GENI – Global Environment for Network Innovation

GUI – Graphical User Interface

Hard Disk – Non-volatile storage device for digital media

ISP – Internet Service Provider

Kbps – Kilobytes per second

LAMP – (Linux, Apache, MySQL, and PHP) – this is a common configuration for a database server

LSN – Large-Scale Network. GENI is an example of a large-scale network.

LTS – refers to a Long-Term Service agreement provided by many operating system developers, such as Ubuntu.

Mbps – Megabytes per second

MSS – Mutualistic Software Services

MSS-CENTER – refers to the main MSS office, where MSS is developed and distributed to participating control frameworks.

MSS-AFFILIATE – refers to a control frameworks child relationship with its MSS-ORIGIN control framework. At a minimum, MSS-AFFILIATE will contain its MSS-ORIGIN control framework and a subset of its services.

MSS-DEVELOPER – refers to a member of academia or business who submits a service to MSS.

MSS-ORIGIN – refers the control framework developer and maintainer. MSS-ORIGIN falls directly beneath MSS-CENTER in the hierarchy.

MSS-RESOURCE – a resource without a control framework, such as a physical computer or virtual machine.

MSS-USER – refers to an authorized user of a MSS-RESOURCE due to his or her working relationship with a resource owner.

Node – a computer

NSF – National Science Foundation

OS – Operating System

Programmable component – a network resource with a modifiable operational instruction set

RAM – Random Access Memory. For MSS, this refers to the computer's volatile memory, such as Dynamic RAM (DRAM).

Real-world user – a private citizen

RENCI – Rennaissance Computing Institute

Resource – a portion or an entire physical computer hardware component. MSS is only concerned with resources such as laptops, desktops, and computer tablets (such as iPad).

Resource Owner – one who is authorized to donate all or a portion of a resource to GENI

Slice – a collection of one or more aggregates and components

Standard – the minimum hardware and software configuration required to federate into GENI

Traffic – the movement of information across the Internet

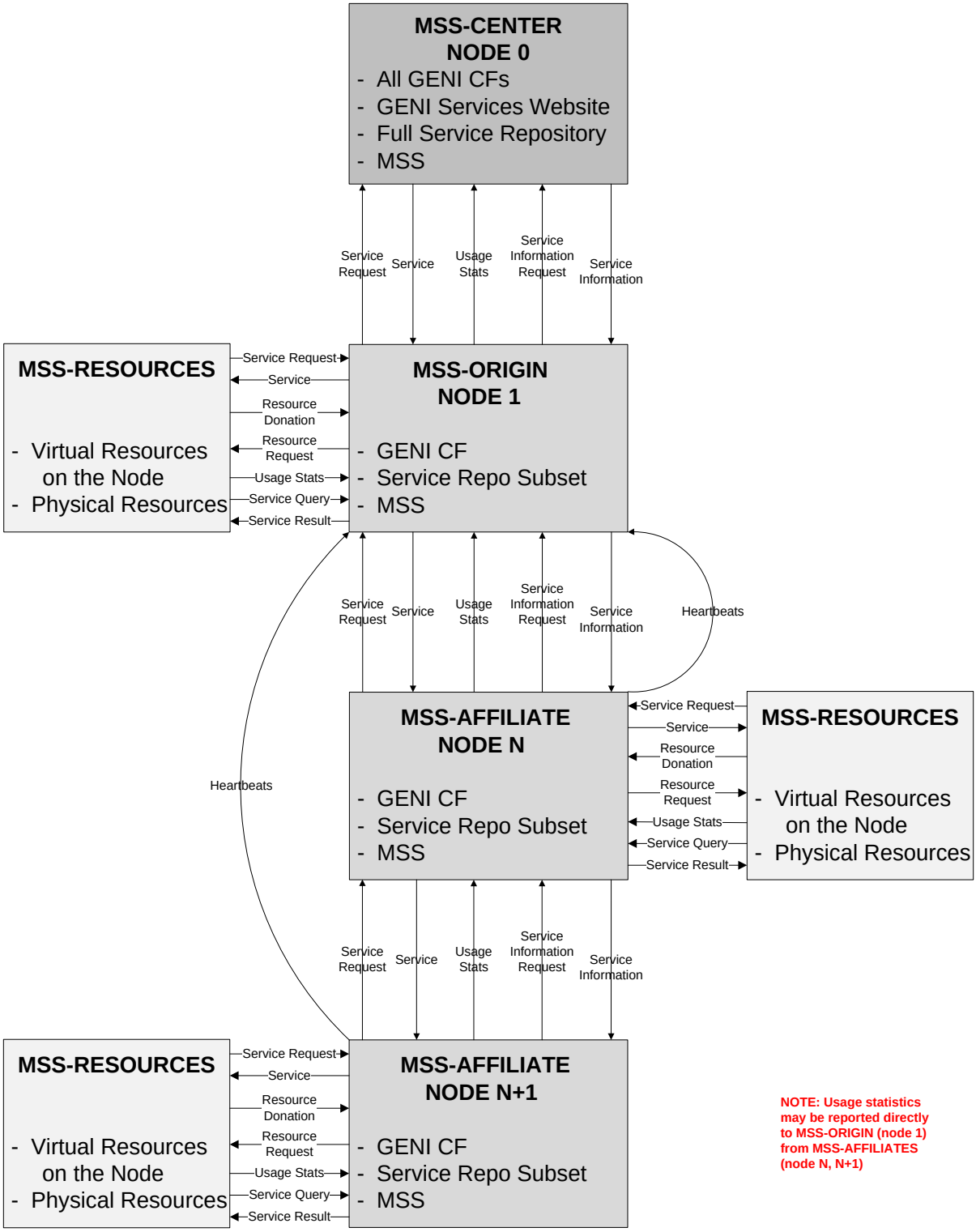
UI – User Interface

USB – universal serial bus

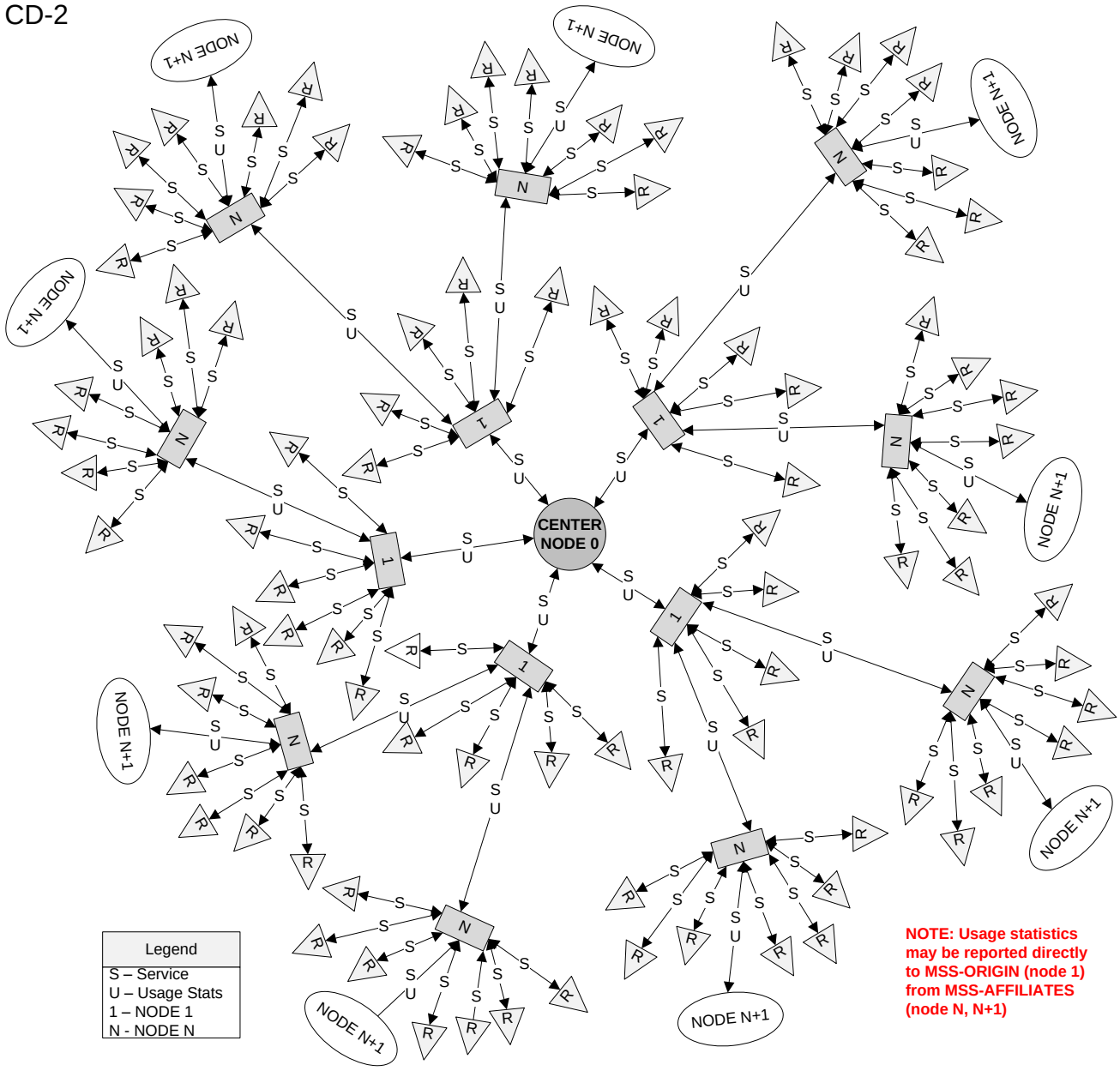
Appendix B

Context Diagrams

CD-1

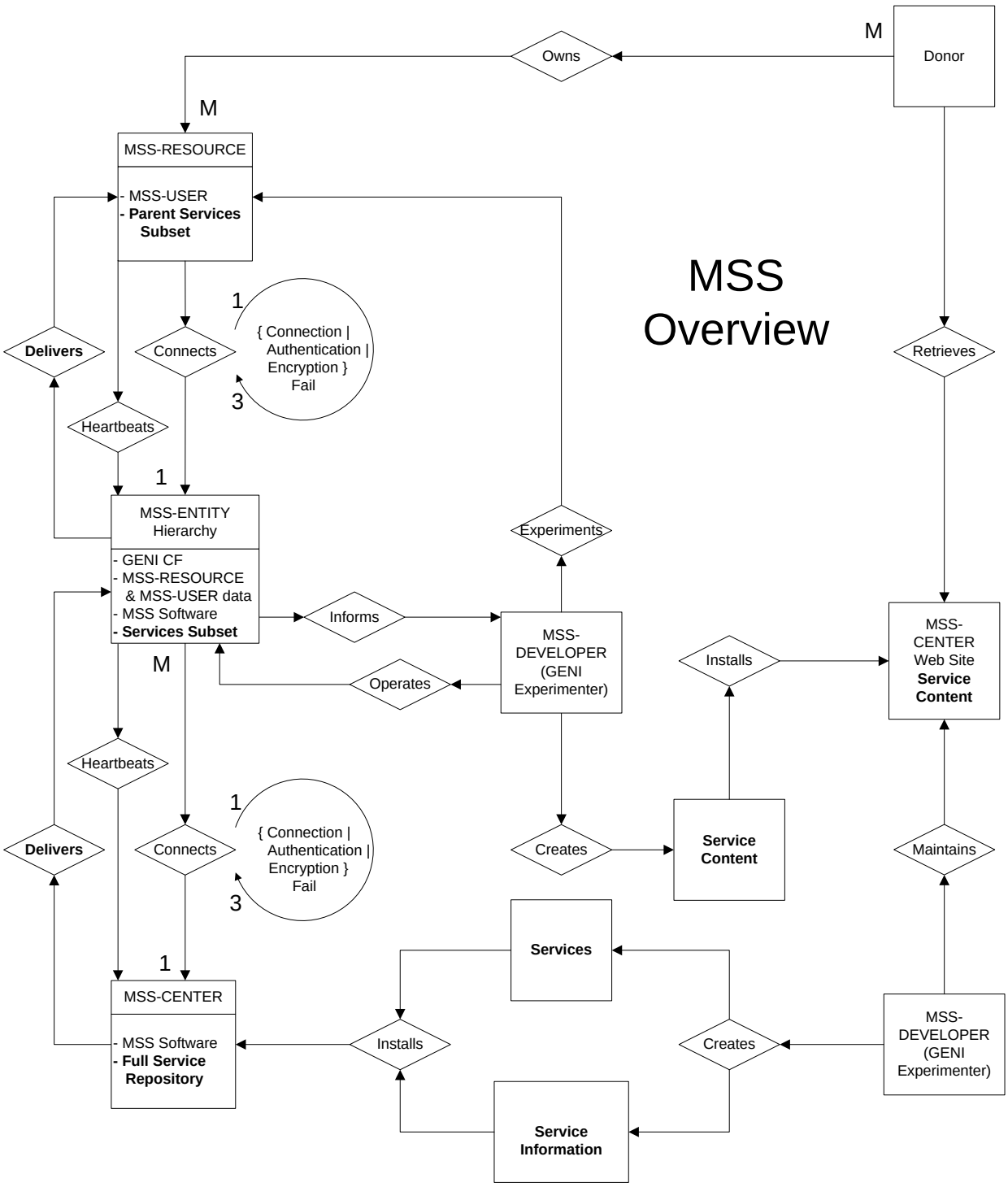


CD-2


CENTER NODE 0
NODE N+1
Legend
S – Service
U – Usage Stats
1 – NODE 1
N - NODE N
NOTE: Usage statistics may be reported directly to MSS-ORIGIN (node 1) from MSS-AFFILIATES (node N, N+1)

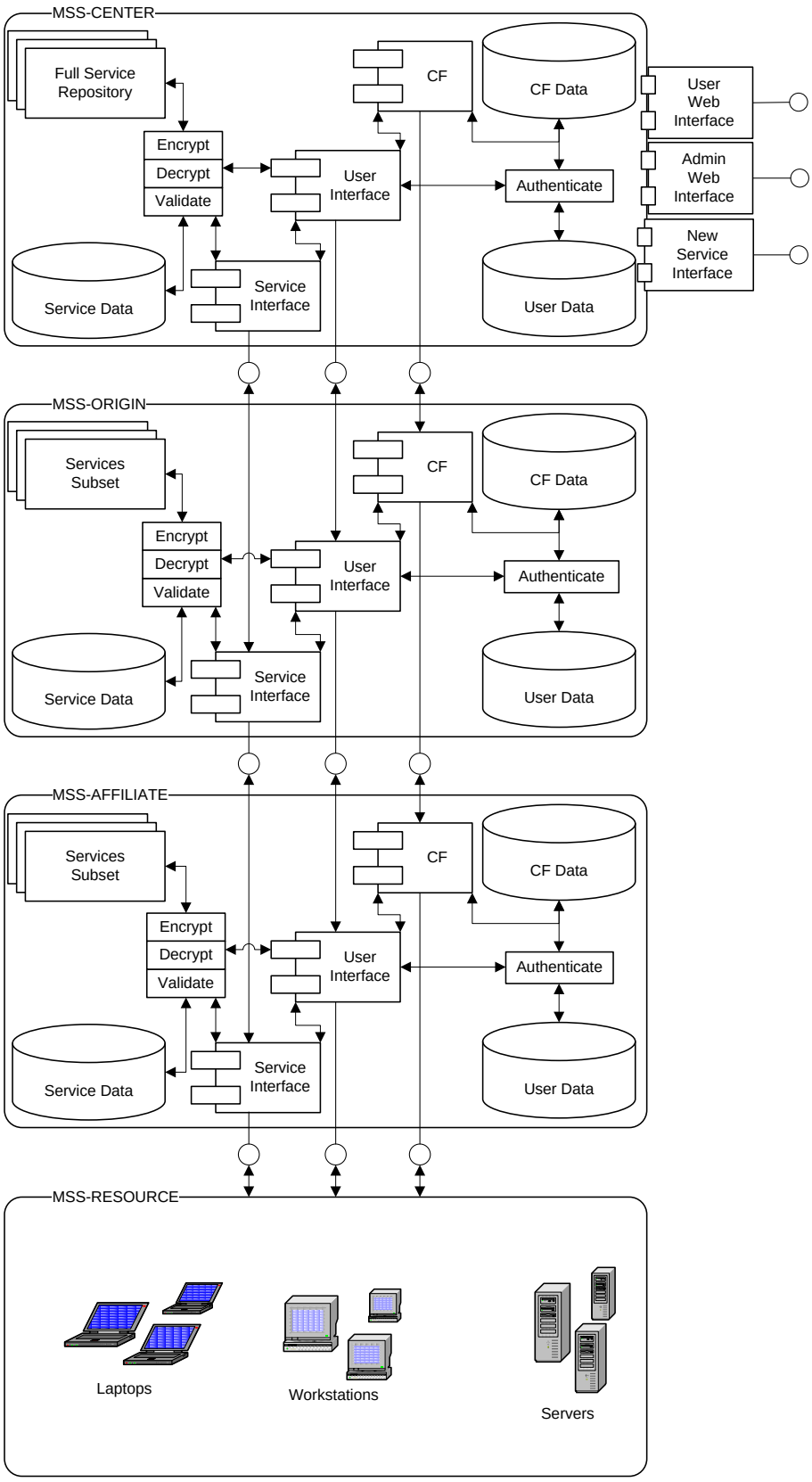
System Overview

SO-1

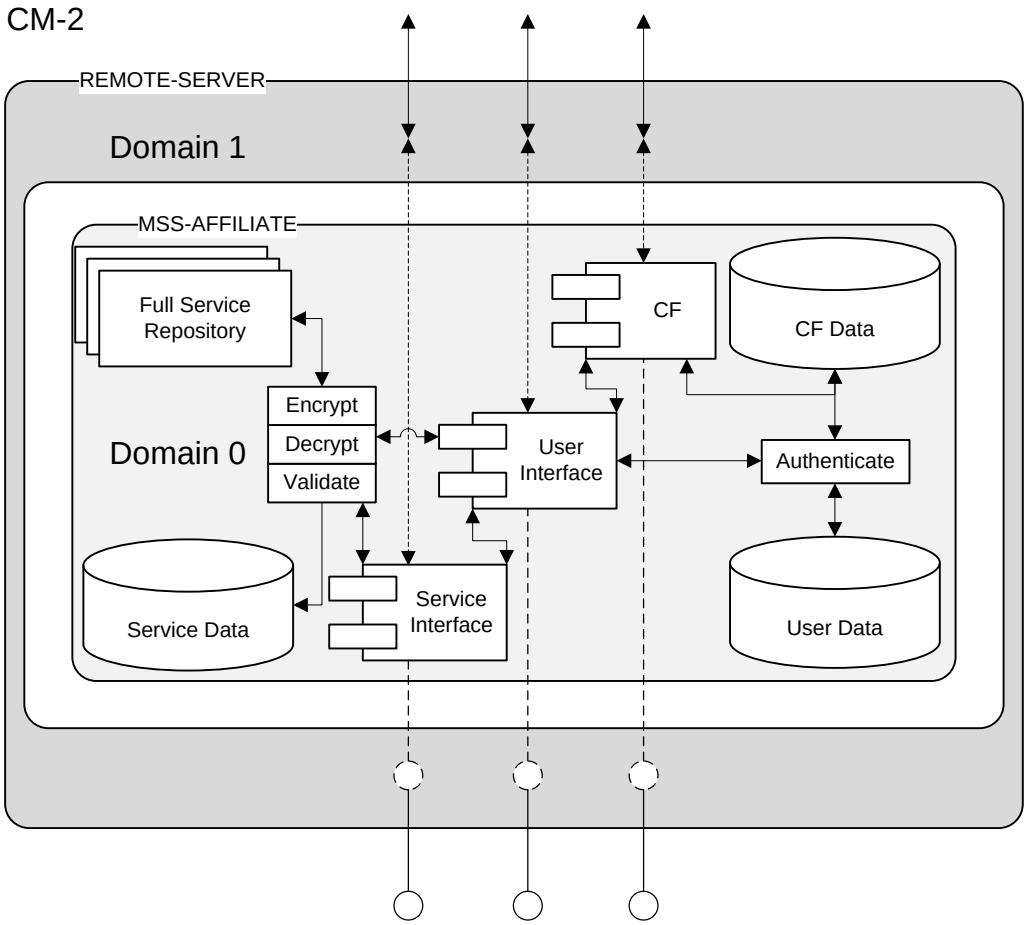


Component Models

CM-1



Remote Server Model



Appendix C

GENI News and Events (2011)

With the GENI project advancing smoothly into Spiral 4, many key projects were highlighted at GEC 12 during the Experiments Plenary at the twelfth GENI Engineering Conference (GEC12) in Kansas City on November 3, 2011.

Leading researchers presented live demonstrations of their experiments to two hundred ninety five attendees Experiments built on the unique capabilities of the GENI mesoscale deployment, a prototype distributed virtual laboratory for exploring future internets, currently spanning over a dozen university campuses and backbone points of presence across the US. Using GENI's capabilities of slicing and deep programmability, experimenters were able to deploy and validate novel services and applications, many of which are not realizable in today's internet.

One group of experiments focused on taking advantage of server and cloud resources to provide new and more efficient capabilities to end users of home networks and mobile devices. Researchers from Georgia Institute of Technology demonstrated uCap, a tool that works with a specialized home router to permit home users to manage network usage allocations across family members, applications (browsing, e-mail, video streaming), and devices. The University of Wisconsin used a suite of chess-playing smart phones to show how computationally intensive tasks can be offloaded to heterogeneous cloud resources while meeting users' goals for security and power efficiency. The Infinity project at the University of Michigan integrates energy-efficient wireless communication techniques and predictive caching to optimize performance of smart phone applications such as Facebook photo sharing.

In a collaborative effort with the SC11 SCInet Research Sandbox (SRS), another group of experiment teams highlighted novel in-network capabilities that build upon GENI's deeply programmable network resources. Researchers from Northwestern University showed how advanced programmable networks can shortcut years of custom engineering to bring ad hoc specialized networks to individuals and organizations. Indiana University gave attendees a live view of their FlowScale system, which balances multi-gigabit network loads across the campus' multiple intrusion detection system (IDS) servers, an integral part of the University's network operations. A combined team from Indiana University and the University of Delaware used their eXtensible Session Protocol (XSP) to boost performance by seamlessly connecting GENI-enabled resources at the network edge to core routers running a high-performance transfer protocol. Clemson University researchers showed off their Steroid OpenFlow Service, transparently tuning network performance to increase end-to-end TCP transfer rates by two orders of magnitude.

Finally, a team led by Rutgers WINLAB showed how they are using the GENI mesoscale deployment to deploy, test, and validate their MobilityFirst architecture. MobilityFirst is sponsored by the National Science Foundation under its Future Internet Architecture program. Using new protocols and design paradigms, MobilityFirst is developing a novel architecture for a future internet where mobility is the norm, with dynamic host and network mobility at scale. Two key MobilityFirst capabilities, Storage Aware Routing and the Global Name Resolution Service, were demonstrated with wired and two different wireless connectivity modalities across a nationwide GENI slice covering eight campuses, two national network backbones, and nine backbone points of presence.

Watch GEC12 Demonstration Videos

Calendar of upcoming GECs

Volume III

Software Design for Mutualistic Software Services (MSS) Version 1.0

1. Introduction

The software design for Mutualistic Software Services incorporates key features identified in the *Software Requirements Specification for Mutualistic Software Services (MSS) Version 1.0* and the *Software Architecture for Mutualistic Software Services (MSS) Version 1.0*. The primary goal addressed by these documents is to create a means by which the GENI Open Resource Control Architecture (ORCA) control framework (CF) users, such as developers, experimenters, and researchers, can share and track software services for use within ORCA.

The concept of sharing services in exchange for resources is hardly new. For example, Amazon and the Apple App Store have a centralized delivery system in which customers exchange resources (money) for services (applications and products). In addition, peer-to-peer networks such as LimeWire and Kazaa use a decentralized delivery system in which members exchange resources (videos and music) for resources (other videos and music). MSS is different from both of these systems in several ways.

First, MSS uses a hierarchical tree-like distribution, which flows centrally from the root at the GENI Maintenance Operations Center (GMOC), then branches through each CF origin to its affiliates, and finally flows to the resources as leaves. The second difference relates to the first in the "middle-mile" and "last-mile" of service delivery. In the middle-mile, each node in the tree is a client of its parent server, and so a server is a client of a server, is a client of a server, and so on. This means that each server must authenticate with the originating CF, and each resource and user must authenticate with his or her parent CF in order to receive services. In the last-mile, the required number of servers at each node decreases like the diameter of a tree branch from many servers at the origin to a CF residing on a single server (e.g., laptop and desktop computers) that a single individual can donate. This single server model on which all ORCA and MSS systems must reside did not exist before this project. Finally yet importantly, GENI experimenters "pay" by hosting a CF and sharing resources in the form of virtual machines, which means GENI's basic unit of currency is the resource. If one considers the software products of these experiments to be services, then offering services to the public in exchange for donated resources provides the payment GENI requires to grow.

This first version of MSS incorporates a lightweight Linux, Apache, MySQL, and PHP (LAMP) infrastructure that ORCA owners can use to advertise and deliver services. In doing so, it addresses the fundamental, mutual need between owners and users: owners require vast amounts of resources to meet their goal of conducting at-scale Internet experiments, and users, experimenters, and researches require software services to conduct experiments and business. MSS leverages the hierarchical GENI structure to establish a distributed service delivery system. When a business or university donates a portion of its resources to GENI as an affiliate, it receives all or part of its sponsor's services, which organizational members can then access.

2. Document Outline

This document incorporates the design features in the *Software Requirements Specification for Mutualistic Software Services (MSS) Version 1.0* and the *Software Architecture for Mutualistic Software Services (MSS) Version 1.0*. The reader should be thoroughly familiar with these because this document refers to them often.

This document follows this outline:

3. System Overview

MSS Version 1.0 has three main components: MSS-ORIGIN, MSS-AFFILIATE, and MSS-RESOURCE. The actual hardware and OS used on the origin and affiliates vary, and each must support service delivery to its child resources. As of this writing, MSS will only support the remote server as a single-server MSS/CF installation, as described in the *Software Architecture for Mutualistic Software Services (MSS) Version 1.0* section 5.1 Remote Server Model.

MSS-AFFILIATE does not fully support a typical computer in the personal computer domain that a member of the public might own, or that an employee of a resource owner might use because of IP address and network interface card requirements. However, later versions of MSS will support these systems. MSS-AFFILIATE does fully support the current state of CF server clusters and its virtual machines, to include the remote server. In addition, later versions of MSS will support the Windows OS; therefore, one must design MSS with common workstation and laptop computers in mind.

3.1 Functional Attributes

The System must:

- Deliver services over an Internet connection
- Provide a management interface for MSS administrators
- Allow the administrator to choose a subset of parent services
- Allow the user to choose a service from the parent sponsor
- Check for heartbeats on MSS-ORIGIN

3.2 Non-functional Attributes

The System shall have:

- Portability
- Autonomy
- Securability
- Credibility
- Integrability
- Extensibility
- Interoperability
- Usability

3.3 Components

Refer to the *Software Architecture for Mutualistic Software Services (MSS) Version 1.0* section 5.3 Component Model for a complete listing of MSS components.

4. System Architecture

The System Architecture is based on the *Software Architecture for Mutualistic Software Services (MSS) Version 1.0* section 5.3 Component Model Version 1.0 components. The following sections link the architectural requirement with the function or interface required to demonstrate adherence to the architectural strategy. Components are listed in font `Courier New`, with functions or classes listed in lower case and with parentheses like so, `courier_new()`. Components, functions, or classes in green already exist, such as `orca.sql`.

4.1 Databases

4.1.1 Services Data

- Holds the subset of service attribute data of its MSS-ENTITY
  - `connect_Services`
  - `Services.sql` [Diagram DB-1 in Appendix A depicts Services.sql]
    - `Attributes`
      - `attribute_ID`
      - `attribute`
      - `description`
      - `parent_ID`
    - `Attributes_have_Services`
      - `Attributes_attribute_ID`
      - `Services_service_ID`
    - `Services`
      - `service_ID`
      - `name`
      - `filename`
      - `description`
      - `shasum`
      - `developer`
      - `publisher`

4.1.2 CF Data

- Holds the MSS-ENTITY data for itself.
  - `connect_orca`
  - `orca.sql`
    - `Actors`

4.1.3 Users Data

- Holds the authorized user's data for itself and its child MSS-AFFILIATES (actors)
  - `connect_Users`
  - `Users.sql` [Diagram DB-2 in Appendix A depicts Users.sql]
    - `Actor`

        - guid
        - name
        - ip
        - port
        - private_key_loc
        - mss_user
        - sponsor
    - Actor_has_User
        - actor_guid
        - user_user_ID
    - Services
        - service_ID
        - totals
    - User
        - user_ID
        - name
        - password
        - guid
        - administrator
    - User_has_Services
        - User_user_ID
        - Services_service_ID

4.2 Components

4.2.1 CF Interface

- MSS-ORIGIN presents an interface to its children to receive heartbeats
    - The ORCA Actor Registry
- MSS-AFFILIATES deliver heartbeats to its MSS-ORIGIN
    - The ORCA Actor Registry
- Provides a CF interface with a list of current donors.
    - The ORCA Actor Registry
    - authorized_sponsor()
- Provides a list of available resources to GENI experimenters and users
    - The ORCA Actor Registry
- Enables GENI experimenters and users to request resources
    - The ORCA Web Portal on each CF

4.2.2 User Interface

- Uses the Authentication process to verify actor and user requests
    - System Specific Constants [constants.php]
    - authorized_user() [Index.php]

- Receives a list of MSS-ENTITY resources from the CF
  - `List Sponsor`
- Receives user service information and service requests
  - `List Users`
  - `List Actors`
  - `Delete Users`
  - `Delete Actors`
- Provides service delivery information to the Service Interface
  - `Add Client`
  - `Add Sponsor`
  - `Add User`
- Triggers the Service Interface to deliver services
  - `Choose Category` [for administrators]
  - `Choose Service`
- Checks for heartbeats on the CF interface
  - `check_heartbeats()`
- Triggers the Service Interface to download services
  - `Assign Users [`… to authorized clients]
  - `authorized_actor()`

4.2.3 Service Interface

- Receives a service information request from the User Interface
  - `List Services`
  - `Delete Services`
- Receives a service download request from the User Interface
  - `Download Services`
- Uses the Encryption process to encrypt services
  - `Secure Shell (SSH)`
    - `ssh_command()`
    - `rsync_command()`
- Uses the Decryption process to decrypt services
  - `Secure Shell (SSH)`
    - `ssh_command()`
    - `rsync_command()`
- Uses the Validation process to validate services
  - `compare_shasum()`
- Receives services from parent MSS-ENTITY
  - `Secure Shell (SSH)`
    - `ssh_command()`
    - `rsync_command()`
- Delivers services to child MSS-AFFILIATES
  - `Secure Shell (SSH)`
    - `ssh_command()`

        - `rsync_command()`

4.3 Service Repository

4.3.1 File System

- Holds MSS services
    - `service_tree()`
    - `match_attributes()` [directory structure must match attributes' recursive structure]
- Holds a subset of parent services, to include the entire subset, for its MSS-ENTITY
    - Clear the Services DB for a potentially totally different subset of services
        - `DELETE FROM Attributes`
        - `DELETE FROM Attributes_have_Services`
        - `DELETE FROM Services`
    - Make sure there is an administrator or no one can log in
        - `INSERT INTO User` […the client's administrator]
    - Add the new Services data
        - `INSERT INTO Attributes`
        - `INSERT INTO Attributes_have_Services`
        - `INSERT INTO Services`

Section *5. Detailed System Design* below expands the requirements above.

4.4 Remote Server

The document *Software Architecture for Mutualistic Software Services (MSS) Version 1.0*, 5.1 Remote Server Model, identifies the remote server as an essential element to MSS success. This is because it enables individual members of the public to donate their resources to GENI in exchange for services. Furthermore, some key components for the remote server are:

- It must reside on a single server
- It must contain all CF programs
- It must incorporate a firewall

In contrast, canonical ORCA [1] installation has at several physical computers. It has one Eucalyptus [2] head node with the program ImageProxy [3] that hosts virtual machine (VM) images, several Eucalyptus worker nodes, and an additional computer that acts as a router to control iptables by using Shorewall [4]. The ORCA cluster administrator uses Shorewall to map the public IP and a static port number to the ORCA cluster's web portal, to the Eucalyptus web portal, and to ImageProxy, and Shorewall dynamically maps the public IP and one of an assigned block of port numbers to a Eucalyptus "Instance." Eucalyptus typically uses a Xen or KVM image-based VM as an instance. The ORCA "cloud" centers at the main Duke University ORCA cluster and includes all other geographically separated ORCA clusters, such as the one at the University of Fairbanks, Alaska. The following describes a typical ORCA experimenter request for one instance.

An experimenter opens the ORCA web portal and requests one or more VMs, which are Eucalyptus instances in this case, from the ORCA "cloud." ORCA main finds an ORCA cluster with available resources in the ORCA Actor Registry, and Shorewall performs Name Address Translation (NAT) between that ORCA web portal's public IP address and port number. Then ORCA requests one instance from the Eucalyptus head node, which commands a worker node to instantiate an instance. Upon success, the worker node assigns a private IP address to the instance's virtual interface (VIF), 10.10.10.130 for example, and requests an IP address from DHCP installed on the Eucalyptus head node. The head node assigns an address, 192.168.1.11 for example, from a pre-defined block of IP addresses, and performs NAT between 10.10.10.1 and 192.168.1.11. Eucalyptus reports the address 192.168.1.11 to ORCA, which requests Shorewall perform NAT between 192.168.1.11, the public IP address, and a port number Shorewall dynamically assigns. ORCA delivers the pubic IP address and port number to the experimenter's ORCA web portal, and he or she now can access the instance using Secure Shell (SSH) from his or her remote location. In addition, ORCA uses a programmable switch to interconnect any second and subsequent VIFs that the experimenter requests so that the instances may communicate with each other across virtual local area networks (VLAN). The figure below depicts the canonical ORCA cluster.

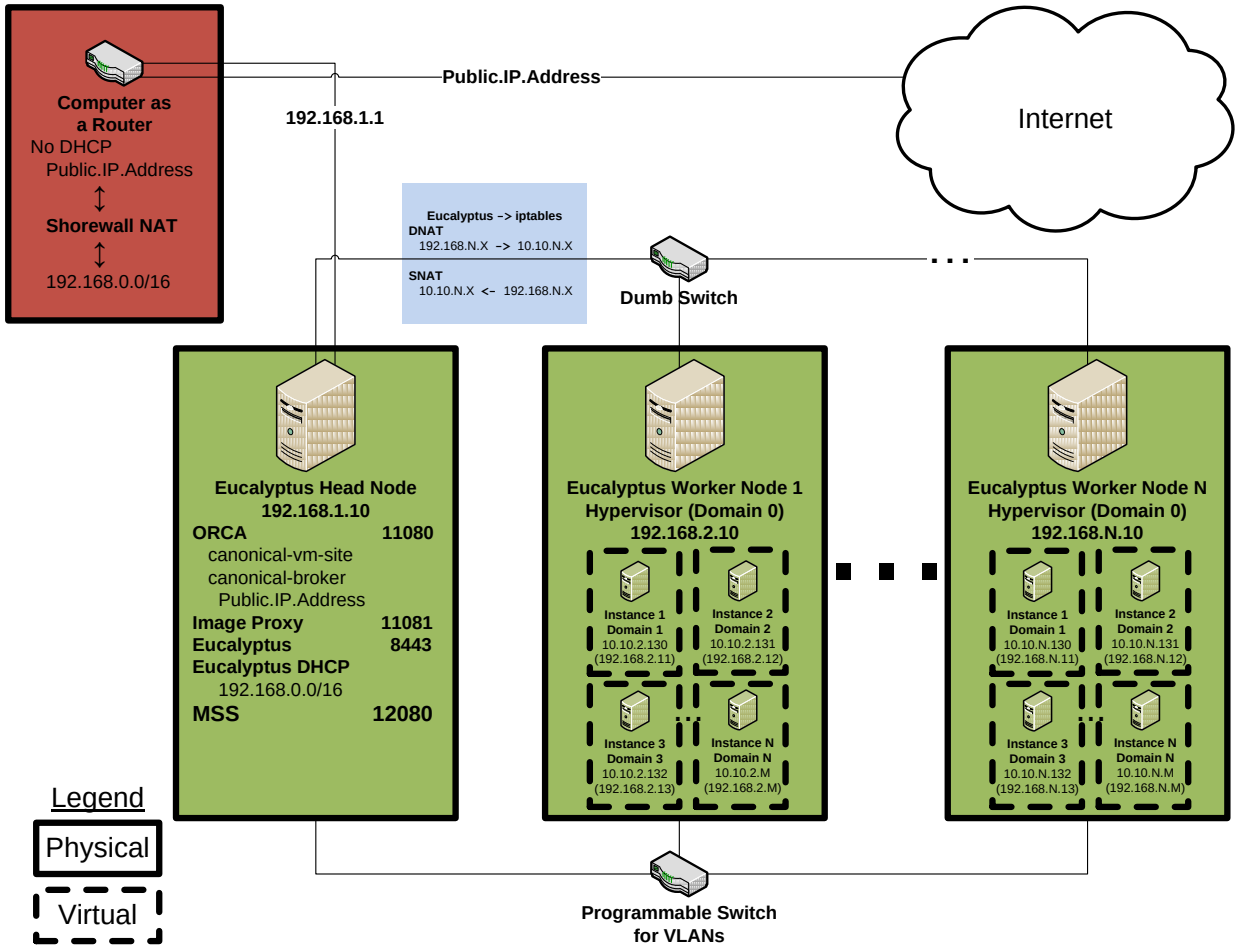


Figure 1. Canonical ORCA cluster

The remote server combines all of the canonical ORCA elements into one computer, by allocating a small portion of the hard drive using the Logical Volume Manager. For instance, hypervisor uses two logical volumes, hyper-root and hyper-swap, and Domain 1 uses two logical volumes, router-root and router-swap, while the Eucalyptus instances use image-based VMs. Furthermore, Xen and KVM allow one to create VIFs 0 & 1 as an extension of the physical Network Interface Cards (NIC) 0 & 1, and Linux includes a "dummy" module to make an internal dummy interface. The picture below exhibits these components.

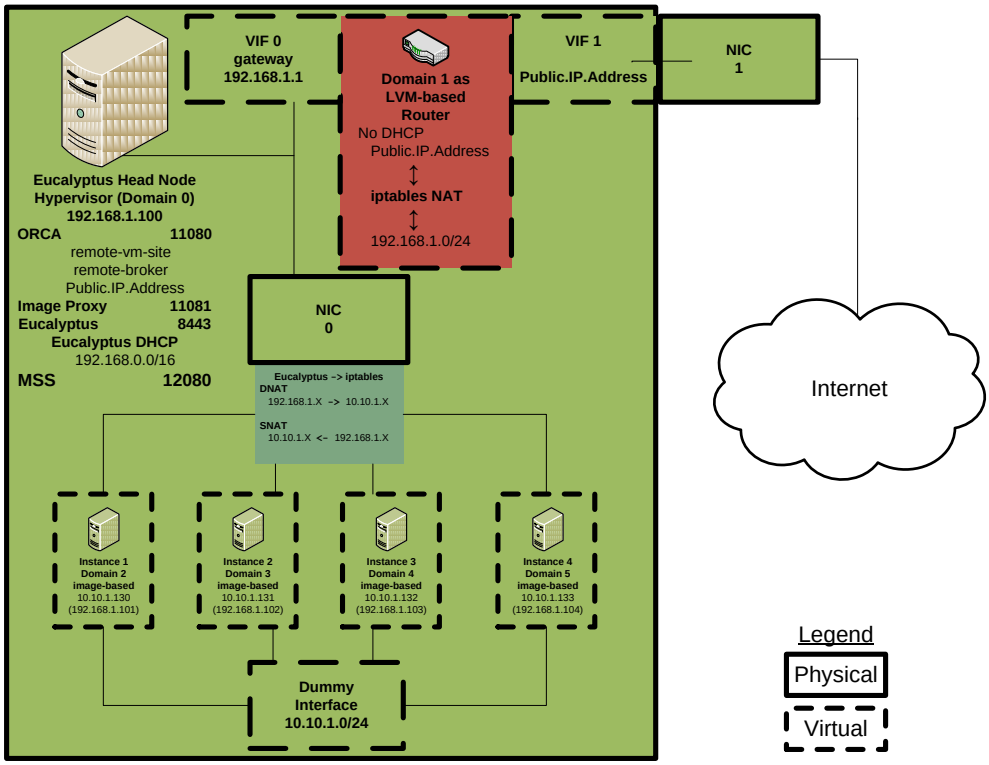


Figure 2. ORCA Remote Server

One important note is that this setup requires two NICs, but some computers owned by individuals only have one. The Dell Inspiron 570 mini-tower computer serves as an example, which one can purchase for $300 as of this writing [5]. Dell includes an option to add a wireless LAN PCIe card, and this computer has expansion slots to add additional cards [6]. Moreover, many computers have two NICs installed from the factory. For instance, laptops often have one Ethernet card and one wireless LAN card. Furthermore, suitable NICs are relatively inexpensive, with prices ranging from less than $10 for a 10/100 Mbps to less than $30 for a 10/100/1000 Mbps Ethernet card at NewEgg.com [7]. However, I believe one could change the configuration to use only one Ethernet card with a little more experimentation with the configuration files. Appendix C contains all the configuration file changes necessary to create the remote server set up.

5. Detailed System Design

The detailed system design lists the sub-interface identified in 4. System Architecture above, with pseudo-code written for the requirements. In addition, the MSS System only maintains essential files in the web-facing **MSS** directory, and so the MSS System is partitioned like so:

```
[web server root directory]
        ├── MSS
        │    ├── actors.jsp
        │    ├── add_client.php
        │    ├── add_sponsor.php
        │    ├── add_user.php
        │    ├── assign_users.php
        │    ├── choose_category.php
        │    ├── choose_service.php
        │    ├── del_actors.php
        │    ├── del_services.php
        │    ├── del_users.php
        │    ├── download_service.php
        │    ├── functions_php.php
        │    ├── functions_Services.php
        │    ├── functions_Shell.php
        │    ├── functions_User.php
        │    ├── index.php
        │    ├── list_actors.php
        │    ├── list_services.php
        │    ├── list_sponsor.php
        │    ├── list_users.php
        │    ├── menu.php
        │    ├── mssLogo.png
        │    ├── style.css
        │    └── user_logout.php
        └── scripts
             ├── connect_orca.php
             ├── connect_Services.php
             ├── connect_Users.php
             ├── constants.php
             ├── cookie.php
             ├── footer.php
             ├── header.php
             ├── logs
             │    └── rsync
             │         └── [YYYYMMDD]_mss_rsync.log
             ├── sh_check_heartbeats
             ├── sh_rsync_command
             └── sh_ssh_command
```

Appendix A contains the SQL Entity-Relationship model for the Services and Users databases and the full SQL code. Diagram CM-3 in Appendix B depicts the high-level system design for one MSS-ENTITY, which is a stretched and expanded version of diagram CM-1 in the *Software Architecture for Mutualistic Software Services (MSS) Version 1.0*. In addition, Appendix B includes the full PHP code for MSS.

Sub-Interface	Function
ALL INTERFACES	All interfaces must ensure that the user is valid and that a session has begun.
	Start a new session The session expires after 20 minutes of inactivity Destroy the session upon exit or expiry Ensure the user is valid for every page or display an error Display a menu relevant to the user's permissions
add_client	Add a new client actor, onto which its LOCAL Administrator may download services from its sponsor.

	If the user is valid and an administrator, then
	  If Submit is pressed, then
	  If the variables are not empty, then
	  Check the IP format
	  Check the Port format
	  Check Actor table to see if the Name already exists
	  Else print an error message
	  Set Sponsor to "client"
	  Give the Client a new GUID
	  INSERT name, ip, port, guid, private_key_loc, mss_user, sponsor into Actor
	  Else
	  Display Name IP PORT Private_Key_Location MSS_User_Name from Actor as textboxes
	Else print an error message and go to user_logout

add_sponsor	Add a new sponsor actor, if the user is valid and an administrator. Sponsors host a CF

	If the user is valid and an administrator, then
	  If Submit is pressed, then
	  If the variables are not empty, then
	  Check the Actor table for the GUID
	  Check the Actor table for the Name
	  Check the IP format
	  Check the Port format
	  Else print an error message
	  INSERT guid, name, ip, port, private_key_loc, mss_user, sponsor into Actor
	  Else
	  Display guid, name, ip, port, private_key_loc, mss_user from Actor as textboxes
	Else print an error message and go to user_logout

add_user	Add a new user if the adding user is valid and an administrator. Users can be a Local Admin, Client Admin, or user

	If the user is valid and an administrator, then
	  If Submit is pressed, then
	  If variables are not empty, then
	  Check the Actor table to see if the name already exists
	  Compare password1 to password2 to check for typos
	  Set the new user to Admin, client, or user
	  Else print an error message

```
                        INSERT name, password, guid, administrator into Actor
                Else
                        Display Name Password1 Password2 Local_Administrator
                        Client_Administrator from User as textboxes
        Else print an error message and go to user_logout
```

assign_users	Administrators can assign a user to an actor. This allows Local or Client Administrators to add new services to their sponsor actor, and users to download services from their sponsor onto the approved actor

```
        If the user is valid and an administrator, then
                If Submit is pressed, then
                        For each Actor
                                INSERT the checked user into Actor_has_User
                                Display a confirmation message
                Else
                        For each Actor
                                Display GUID Name IP Port Sponsor from Actor as
                                textboxes
                                For each User
                                        Display Name Administrator from User as
                                        checkboxes
        Else print an error message and go to user_logout
```

choose_category	This is the only way that a Client Administrator can add new services to his or her Client Sponsor. The user must be a Client Administrator on the parent and a Local Administrator on the child

```
        If the user is valid and on an authorized sponsor, then
                If the user is a local or client administrator and sending
                heartbeats to MSS-ORIGIN, then ON THE CLIENT
                        If Submit is pressed, then
                                If the user does not exist [special case: first install]
                                        INSERT user_ID, name, password, guid,
                                        administrator into User
                                Clear the Attributes Attributes_have_Services
                                and Services tables
                                INSERT all Attributes into Attributes
                                For each checked Attribute
                                        Make directories on the client that match
                                        the attribute tree in the System
                                        Constants location
                                        Rsync the entire directory named
                                        [Attribute]
                                        INSERT the matching Service
                                        INSERT the matching
                                        Attributes_have_Services
                                Else
                                        For each Attribute
                                        Display the Attribute indented by its level
                                        For each Attributes_have_Services
                                                Display a checkbox
                        Else print an error message
        Else print an error message and go to user_logout
```

choose_service	All users can choose services if they are valid users on an authorized computer

```
        If the user is valid and on an authorized client, then
                If the Service_ID button is pressed
                        If sending MSS-ORIGIN heartbeats
                                Display alert for download size
                                Go to download_service
                        Else
                                Display the attribute tree by its level
                                For each Attribute
                                        Display Service_ID as Submit button
                                        Display the list of matching Services
```

```
Else print an error message and go to user_logout
```

constants	Allow the administrator to tailor the installation by providing definable Constants. Separate database names, passwords, and file locations from the PHP web pages.

```
MSS_SERVICES directory
GENI_HEARTBEATS webpage
GENI_DB for GENI database name
GENI_DB_USER for GENI database access
GENI_DB_PASS for GENI database password
USERS_DB for Users database name
USERS_DB_USER for Users database access
USERS_DB_PASS for Users database password
SERVICES_DB for Services database name
SERVICES_DB_USER for Services database access
SERVICES_DB_PASS for Services database password
MSS_USER that conducts MSS activities on the host only
MSS_HOME the directory that contains MSS webpages
MSS_SCRIPTS the directory that contains MSS scripts
```

del_actors	Administrators can delete a client from the sponsor

```
If the user is valid and a local administrator, then
       If the Submit button is pressed, then
              DELETE the Actor from Actor_has_User where the checkbox
              is checked
              DELETE the Actor from Actor where the checkbox is
              checked
       Else
              Display "delete actor" checkbox, name, guid, ip, port,
              private_key_loc, mss_user, sponsor from Actor
Else print an error message and go to user_logout
```

del_services	Administrators can choose to delete services from the sponsor immediately. For everyone, only delete the service after every user deletes the service to save bandwidth and download time.

```
If the user is valid, then
       If the Submit button is pressed, then
              DELETE the service from User_has_Services where the
              "delete my service" checkbox is checked
              Decrement Services Totals
              If this is the last user to own the service, then
                     DELETE the service from Services where the
                     "delete my service" checkbox is checked
                     Delete the service from the client's file system
              If the user is a local administrator, then
                     DELETE the service from User_has_Services where
                     the "delete all service" checkbox is checked
                     DELETE the service from Services where the
                     "delete all service" checkbox is checked
                     Delete the service from the client's file system
       Else
              Display "delete my service" checkbox
              If the user is a local administrator, then
                     Display "delete all service" checkbox
              Display Name, Attribute, Filename, description from
              view_Attributes_Services
Else print an error message and go to user_logout
```

del_users	Administrators can delete a user from the sponsor

```
If the user is valid and a local administrator, then
       If the Submit button is pressed, then
              DELETE the user from User_has_Services where the
              checkbox is checked
              DELETE the user from Actor_has_User where the checkbox
              is checked
              DELETE the user from User where the checkbox is checked
```

```
        Else
                Display "delete user" checkbox, user_ID, name, guid,
                "local or client" administrator from User
Else print an error message and go to user_logout
```

download_service	Download the service if all of the choose_service requirements are met, and if the client and host IP addresses match an authorized actor.

```
If the user is valid and on an authorized client, then
        If the host IP exists and the client IP exists in Actors
                Send the service to the client
                Compare the SHA sums of the original to the downloaded
                service
                If the shasums do not match
                        Delete the downloaded service from the client
                        Print an error message
        Else print an error message
Else print an error message and go to user_logout
```

index	This is the home page for MSS, where authorized users on authorized clients log in. Administrators can log in from any computer, which also solves the "initial installation" problem where no users are assigned to any actors.

```
If username and password are blank, then
        If Submit button is pressed, then
                If username and password match a user in Users
                        valid_user == true
                If the user is a local administrator, then
                        authorized_user == true
                        Load a list of services into the session
                        Display administrator instructions
                If Actor_has_User where Actor.ip == client ip
                        authorized_user == true
                        Load a list of services into the session
                        If user is a client administrator
                                Display client administrator instructions
                        If user is a ordinary user
                                Display user instructions
                Else print an error message and deny access
        Else
                Display username and password as textboxes
Else print an error message and go to user_logout
```

list_actors	Valid and authorized users may view a list of actors authorized to download from the sponsor.

```
If the user is valid and authorized, then
        Display name, guid, ip, port, private_key_loc, mss_user from
        User
        Display "local or client" sponsor based on host IP
Else print an error message and go to user_logout
```

list_services	Valid and authorized users may view a list of services available on the sponsor

```
If the user is valid and authorized, then
        Display service_ID, name, filename, description, shasum,
        developer, publisher from Services
Else print an error message and go to user_logout
```

list_sponsor	Valid and authorized users may view a list of sponsors, such as the ORCA actors Service Manager, Broker, and Site, that are registered in the GENI database

```
If the user is valid and is authorized, then
        Display act_id, act_name, act_guid from GENI_DB.Actors
Else print an error message and go to user_logout
```

list_users	Valid users and administrators may view a list of users that are registered on the sponsor

```
If the user is valid and an administrator, then
        Display user_ID, name, guid, administrator from User
```

```
        Display "local or client" administrator
Else print an error message and go to user_logout
```

menu	Display a menu ribbon that changes based on the user type: administrator, client administrator, user

```
If the user is valid and authorized, then
        Display index, list_sponsor, list_services, choose_services
        If the user is a local or client administrator, then
                If the actor is an authorized sponsor, then
                        Display choose_category
                If the actor is an administrator, then
                        Display list_users, list_actors, add_user,
                        add_client, add_sponsor
                        If the web browser is not text-based
                                Display assign_users
                        Display del_actors, del_users
        Display del_services, user_logout
Else print an error message and go to user_logout
```

user_logout	Log the user out of the session when the link "Log Out" is chosen, time expires, or the user is not valid.

```
        Close the session write
        Unset the session
        Destroy the cookie
```

6. References

[1] RENCI. (December 5, 2011). Deploying an Authority (Aggregate Manager/AM). Renaissance Computing Institute. Retrieved on March 28, 2012, from: https://geni-orca.renci.org/trac/wiki/deploy-am

[2] Eucalyptus Systems. *Eucalyptus*. Eucalyptus Systems, Inc. Retrieved December 8, 2011, from http://www.eucalyptus.com/

[3] RENCI. *Eucalyptus/XCat image proxy*. (July 25, 2011). Renaissance Computing Institute. Retrieved December 2, 2011, from: https://code.renci.org/gf/project/networkedclouds/wiki/?pagename=ImageProxy

[4] Eastep, Thomas M. (2011). *Shorewall: IP Tables Made Easy.* Shorewall. Retrieved on April 2, 2012, from: http://shorewall.net/

[5] Dell. (April 4, 2012). *Shopping cart.* Dell Store. Retrieved on April 4, 2012 from: http://configure.us.dell.com/dellstore

[6] Dell. (April 2012). *Dell Inspiron 570 Desktop.* Dell.com. Retrieved on April 4, 2012 from: http://www.dell.com/content/products/productdetails.aspx/inspiron-570?c=us&cs=19&l=en&s=corp&~lt=popup

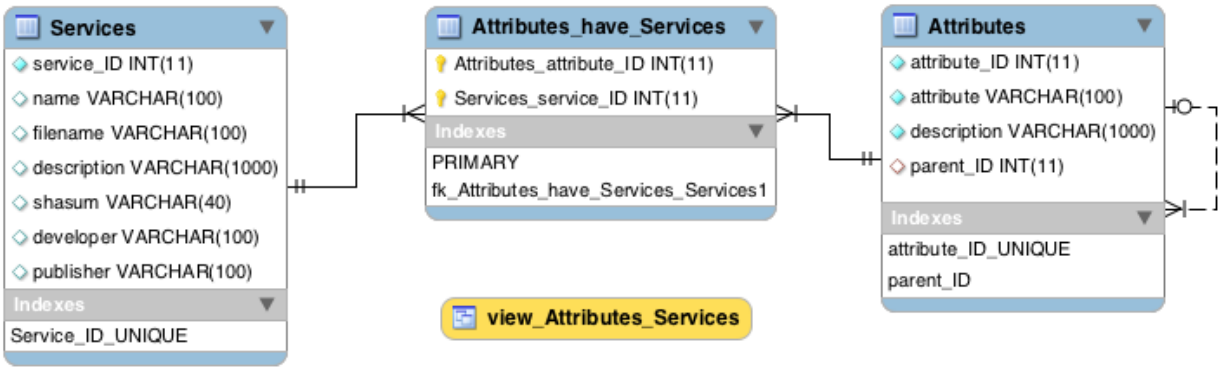
[7] NewEgg. (April 4, 2012). *Network Interface Cards.* NewEgg.com. Retrieved on April 4, 2012 from: http://www.newegg.com/Store/SubCategory.aspx?SubCategory=27&name=Network-Interface-Cards

Appendix A

Services Database

Overview

DB-1



SQL Code

```
  1 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
  2 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
  3 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';
  4
  5 DROP SCHEMA IF EXISTS `Services` ;
  6 CREATE SCHEMA IF NOT EXISTS `Services` DEFAULT CHARACTER SET utf8 COLLATE
utf8_general_ci ;
  7 USE `Services` ;
  8
  9 -- -----------------------------------------------------
 10 -- Table `Services`.`Services`
 11 -- -----------------------------------------------------
 12 DROP TABLE IF EXISTS `Services`.`Services` ;
 13
 14 CREATE  TABLE IF NOT EXISTS `Services`.`Services` (
 15   `service_ID` INT(11) NOT NULL ,
 16   `name` VARCHAR(100) NULL DEFAULT NULL ,
 17   `filename` VARCHAR(100) NULL DEFAULT NULL ,
 18   `description` VARCHAR(1000) NULL DEFAULT NULL ,
 19   `shasum` VARCHAR(40) NULL DEFAULT NULL ,
 20   `developer` VARCHAR(100) NULL DEFAULT NULL ,
 21   `publisher` VARCHAR(100) NULL DEFAULT NULL ,
 22   UNIQUE INDEX `Service_ID_UNIQUE` (`service_ID` ASC) )
 23 ENGINE = InnoDB
 24 AUTO_INCREMENT = 2
 25 DEFAULT CHARACTER SET = utf8
 26 COLLATE = utf8_general_ci
 27 COMMENT = 'This table holds the MSS Service Repository.';
 28
 29
 30 -- -----------------------------------------------------
 31 -- Table `Services`.`Attributes`
 32 -- -----------------------------------------------------
 33 DROP TABLE IF EXISTS `Services`.`Attributes` ;
 34
 35 CREATE  TABLE IF NOT EXISTS `Services`.`Attributes` (
```

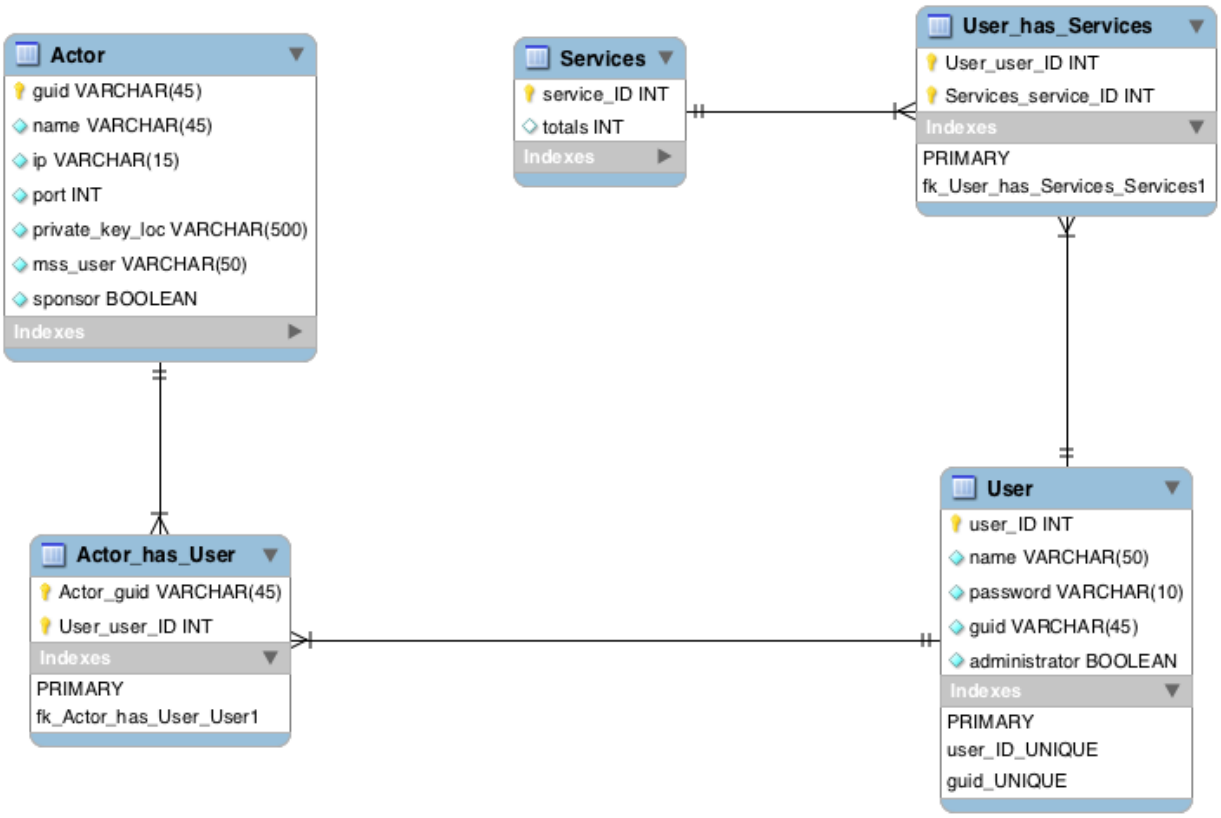
```
 36   `attribute_ID` INT(11) NOT NULL ,
 37   `attribute` VARCHAR(100) NOT NULL ,
 38   `description` VARCHAR(1000) NOT NULL ,
 39   `parent_ID` INT(11) NULL ,
 40   UNIQUE INDEX `attribute_ID_UNIQUE` (`attribute_ID` ASC) ,
 41   INDEX `parent_ID` (`parent_ID` ASC) ,
 42   CONSTRAINT `parent_ID`
 43     FOREIGN KEY (`parent_ID` )
 44     REFERENCES `Services`.`Attributes` (`attribute_ID` )
 45     ON DELETE CASCADE
 46     ON UPDATE CASCADE)
 47 ENGINE = InnoDB
 48 AUTO_INCREMENT = 10
 49 DEFAULT CHARACTER SET = utf8
 50 COLLATE = utf8_general_ci
 51 COMMENT = 'This table holds the MSS Attribute Repository.';
 52
 53
 54 -- -----------------------------------------------------
 55 -- Table `Services`.`Attributes_have_Services`
 56 -- -----------------------------------------------------
 57 DROP TABLE IF EXISTS `Services`.`Attributes_have_Services` ;
 58
 59 CREATE  TABLE IF NOT EXISTS `Services`.`Attributes_have_Services` (
 60   `Attributes_attribute_ID` INT(11) NOT NULL ,
 61   `Services_service_ID` INT(11) NOT NULL ,
 62   PRIMARY KEY (`Attributes_attribute_ID`, `Services_service_ID`) ,
 63   INDEX `fk_Attributes_have_Services_Services1` (`Services_service_ID` ASC) ,
 64   CONSTRAINT `fk_Attributes_have_Services_Ads1`
 65     FOREIGN KEY (`Attributes_attribute_ID` )
 66     REFERENCES `Services`.`Attributes` (`attribute_ID` )
 67     ON DELETE NO ACTION
 68     ON UPDATE NO ACTION,
 69   CONSTRAINT `fk_Attributes_have_Services_Services1`
 70     FOREIGN KEY (`Services_service_ID` )
 71     REFERENCES `Services`.`Services` (`service_ID` )
 72     ON DELETE NO ACTION
 73     ON UPDATE NO ACTION)
 74 ENGINE = InnoDB
 75 DEFAULT CHARACTER SET = utf8
 76 COLLATE = utf8_general_ci;
 77
 78
 79 -- -----------------------------------------------------
 80 -- Placeholder table for view `Services`.`view_Attributes_Services`
 81 -- -----------------------------------------------------
 82 CREATE TABLE IF NOT EXISTS `Services`.`view_Attributes_Services` (`service_ID`
INT, `name` INT, `filename` INT, `description` INT, `shasum` INT, `developer` INT,
`publisher` INT, `attribute_ID` INT, `attribute` INT);
 83
 84 -- -----------------------------------------------------
 85 -- View `Services`.`view_Attributes_Services`
 86 -- -----------------------------------------------------
 87 DROP VIEW IF EXISTS `Services`.`view_Attributes_Services` ;
 88 DROP TABLE IF EXISTS `Services`.`view_Attributes_Services`;
 89 USE `Services`;
 90 CREATE  OR REPLACE VIEW `Services`.`view_Attributes_Services` AS
 91 SELECT `Services`.`service_ID`, `Services`.`name`, `Services`.`filename`,
`Services`.`description`, `Services`.`shasum`, `Services`.`developer`,
`Services`.`publisher`, `Attributes`.`attribute_ID`, `Attributes`.`attribute`
 92 FROM `Services`
```

```
 93 INNER JOIN (`Attributes` INNER JOIN `Attributes_have_Services` ON
`Attributes`.`attribute_ID`=`Attributes_have_Services`.`Attributes_attribute_ID`) ON
`Services`.`service_ID`=`Attributes_have_Services`.`Services_service_ID`;
 94 ;
 95
 96
 97 SET SQL_MODE=@OLD_SQL_MODE;
 98 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
 99 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Users Database

Overview

DB-2



SQL Code

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';

DROP SCHEMA IF EXISTS `Users` ;
CREATE SCHEMA IF NOT EXISTS `Users` DEFAULT CHARACTER SET utf8 COLLATE
utf8_general_ci ;
USE `Users` ;

-- -----------------------------------------------------
-- Table `Users`.`User`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `Users`.`User` ;

CREATE  TABLE IF NOT EXISTS `Users`.`User` (
  `user_ID` INT NOT NULL AUTO_INCREMENT ,
  `name` VARCHAR(50) NOT NULL ,
  `password` VARCHAR(10) NOT NULL ,
  `guid` VARCHAR(45) NOT NULL ,
  `administrator` TINYINT(1)  NOT NULL DEFAULT 0 ,
  PRIMARY KEY (`user_ID`) ,
  UNIQUE INDEX `user_ID_UNIQUE` (`user_ID` ASC) ,
  UNIQUE INDEX `guid_UNIQUE` (`guid` ASC) )
```

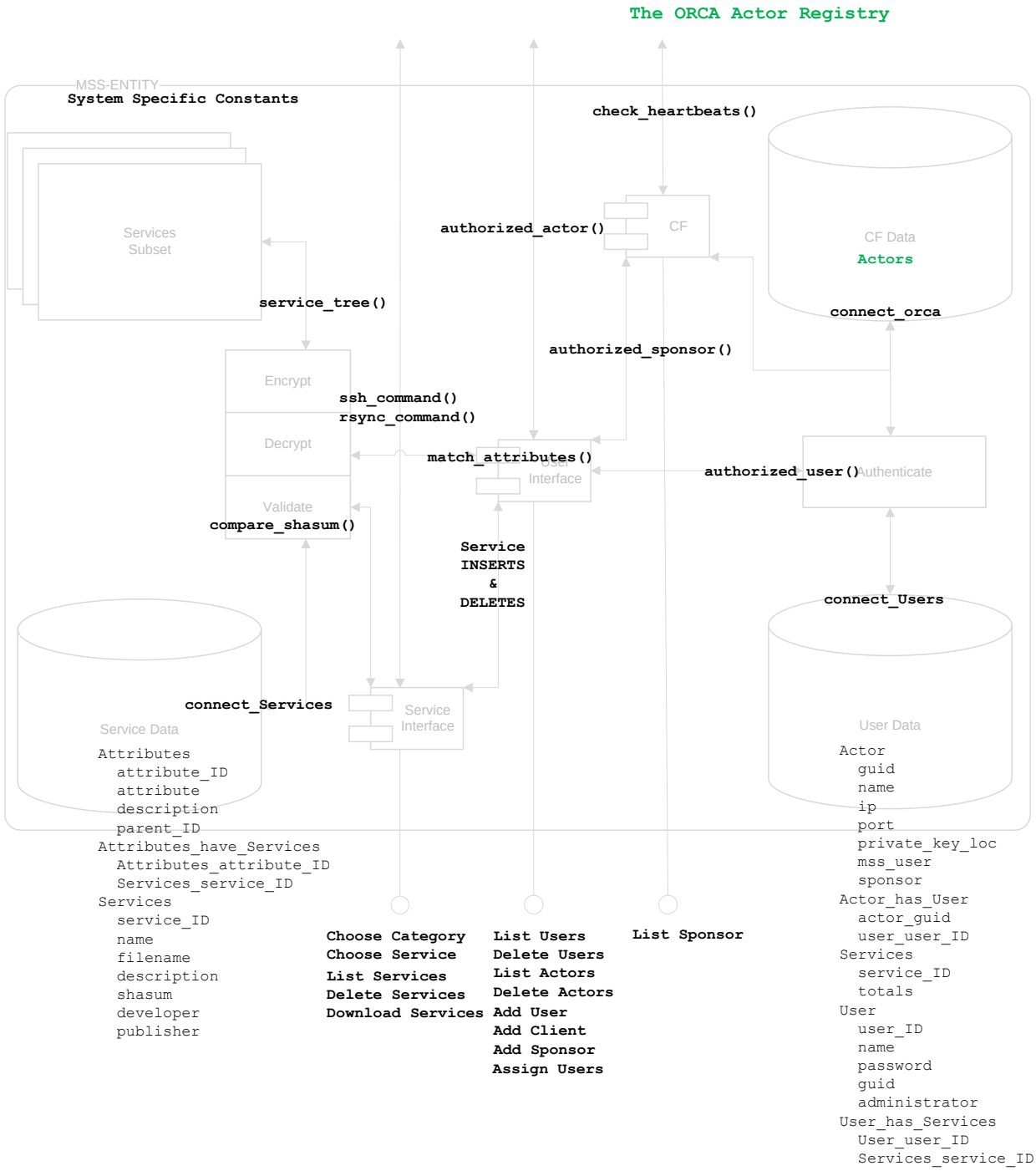
```
23 ENGINE = InnoDB
24 DEFAULT CHARACTER SET = utf8
25 COLLATE = utf8_general_ci
26 COMMENT = 'Contains the user GUID, password, name, and ssh_key_location';
27
28
29 -- -----------------------------------------------------
30 -- Table `Users`.`Actor`
31 -- -----------------------------------------------------
32 DROP TABLE IF EXISTS `Users`.`Actor` ;
33
34 CREATE  TABLE IF NOT EXISTS `Users`.`Actor` (
35   `guid` VARCHAR(45) NOT NULL ,
36   `name` VARCHAR(45) NOT NULL ,
37   `ip` VARCHAR(15) NOT NULL ,
38   `port` INT NOT NULL DEFAULT 22 ,
39   `private_key_loc` VARCHAR(500) NOT NULL ,
40   `mss_user` VARCHAR(50) NOT NULL ,
41   `sponsor` TINYINT(1)  NOT NULL DEFAULT 0 ,
42   PRIMARY KEY (`guid`) ,
43   UNIQUE INDEX `guid_UNIQUE` (`guid` ASC) )
44 ENGINE = InnoDB
45 DEFAULT CHARACTER SET = utf8
46 COLLATE = utf8_general_ci
47 COMMENT = 'The GENI actor that owns the resource the user is using.';
48
49
50 -- -----------------------------------------------------
51 -- Table `Users`.`Services`
52 -- -----------------------------------------------------
53 DROP TABLE IF EXISTS `Users`.`Services` ;
54
55 CREATE  TABLE IF NOT EXISTS `Users`.`Services` (
56   `service_ID` INT NOT NULL ,
57   `totals` INT NULL DEFAULT 0 ,
58   PRIMARY KEY (`service_ID`) )
59 ENGINE = InnoDB;
60
61
62 -- -----------------------------------------------------
63 -- Table `Users`.`User_has_Services`
64 -- -----------------------------------------------------
65 DROP TABLE IF EXISTS `Users`.`User_has_Services` ;
66
67 CREATE  TABLE IF NOT EXISTS `Users`.`User_has_Services` (
68   `User_user_ID` INT NOT NULL ,
69   `Services_service_ID` INT NOT NULL ,
70   PRIMARY KEY (`User_user_ID`, `Services_service_ID`) ,
71   INDEX `fk_User_has_Services_Services1` (`Services_service_ID` ASC) ,
72   CONSTRAINT `fk_User_has_Services_User1`
73     FOREIGN KEY (`User_user_ID` )
74     REFERENCES `Users`.`User` (`user_ID` )
75     ON DELETE NO ACTION
76     ON UPDATE NO ACTION,
77   CONSTRAINT `fk_User_has_Services_Services1`
78     FOREIGN KEY (`Services_service_ID` )
79     REFERENCES `Users`.`Services` (`service_ID` )
80     ON DELETE NO ACTION
81     ON UPDATE NO ACTION)
82 ENGINE = InnoDB
83 DEFAULT CHARACTER SET = utf8
84 COLLATE = utf8_general_ci;
85
```

```
 86
 87 -- -----------------------------------------------------
 88 -- Table `Users`.`Actor_has_User`
 89 -- -----------------------------------------------------
 90 DROP TABLE IF EXISTS `Users`.`Actor_has_User` ;
 91
 92 CREATE  TABLE IF NOT EXISTS `Users`.`Actor_has_User` (
 93   `Actor_guid` VARCHAR(45) NOT NULL ,
 94   `User_user_ID` INT NOT NULL ,
 95   PRIMARY KEY (`Actor_guid`, `User_user_ID`) ,
 96   INDEX `fk_Actor_has_User_User1` (`User_user_ID` ASC) ,
 97   CONSTRAINT `fk_Actor_has_User_Entity1`
 98     FOREIGN KEY (`Actor_guid` )
 99     REFERENCES `Users`.`Actor` (`guid` )
100     ON DELETE NO ACTION
101     ON UPDATE NO ACTION,
102   CONSTRAINT `fk_Actor_has_User_User1`
103     FOREIGN KEY (`User_user_ID` )
104     REFERENCES `Users`.`User` (`user_ID` )
105     ON DELETE NO ACTION
106     ON UPDATE NO ACTION)
107 ENGINE = InnoDB
108 DEFAULT CHARACTER SET = utf8
109 COLLATE = utf8_general_ci;
110
111
112
113 SET SQL_MODE=@OLD_SQL_MODE;
114 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
115 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Appendix B

High-Level System Design

CM-3



PHP Code

add_client.php

```php
<?php
//Purpose: Add a new client actor
//Means: Connect to the Users database as user "workers"
//Conventions: na stands for "new actor"
//Author:  John P. Quan
//Version: 1.0
//Date:    20120105
?>

<? ////////////////////////////////////////////////////
//STANDARD SESSION LIFE AND INACTIVITY CHECK
session_cache_expire(20);
session_start();

$inactive = 1200;
if (isset($_SESSION['start'])) {
    $session_life = time() - $_SESSION['start'];
    if ($session_life > $inactive) {
        header("Location: user_logout.php");

        $_SESSION['valid_user'] = 'false';
        //CLOSE PREVIOUS SESSION*
        $_SESSION;
        session_destroy();
    }
}
//Set the session start time
$_SESSION['start'] = time();

//////// START OF PAGE ////////////////

if ($_SESSION['valid_user'] == true
        AND $_SESSION['authorized_user'] == true
        AND $_SESSION['administrator'] == "1")
{
?>
<?php
include '../scripts/connect_Users.php';
include '../scripts/header.php';
include 'functions_User.php';
include 'functions_php.php';

connect_Users();

echo "<h3><center>Add Client</center></h3>";

//Display the menu ribbon
include 'menu.php';
//adjusts the header
echo "<BR>";
    ?>

<div id="wrap">

    <? ////////////////////////////////////////////////
    //  Check the posted variables after "Submit"
    if (isset($_POST['na_submit']))
    {
        //if a new user name is entered...
```

```
60          if(!empty($_POST['na_name']) and
61             !empty($_POST['na_ip']) and
62             !empty($_POST['na_port']) and
63             !empty($_POST['na_private_key_loc']) and
64             !empty($_POST['na_mss_user']))
65          {
66              // add a GUID
67              $na_guid = gen_uuid();
68
69              // see if the user already exists
70              $query=sprintf("SELECT *
71                              FROM Actor");
72
73              $result = mysql_query($query);
74
75              while($row = mysql_fetch_array($result))
76              {
77                  //Check the name
78                  if($_POST['na_name'] == $row['name'])
79                  {
80                      echo "<BR>Actor name ".$_POST['na_name']."
81                            already exists.<BR>";
82                      unset_na_vars();
83                      die(mysql_error());
84                  }
85                  else
86                  {
87                      if(!empty($_POST['na_name'])) {
88                          $na_name = $_POST['na_name'];
89                      }
90                      else
91                      {
92                          echo "<BR>Please enter your Client's
93                                Name.<BR>";
94                          unset_na_vars();
95                          die(mysql_error());
96                      }
97                  }
98
99                  ////////////////////////////////////////////////
100                 //Check the IP address format and port
101                 // and load the private key and mss_user
102                 if(filter_var($_POST['na_ip'], FILTER_VALIDATE_IP))
103                 {
104                     $na_ip =  $_POST['na_ip'];
105                     if(isset( $_POST['na_port']) and
106                               $_POST['na_port'] >= 0 and
107                               $_POST['na_port'] < 65536) {
108                         $na_port = $_POST['na_port'];
109                     }
110                     else {
111                         //default port
112                         $na_port = 22;
113                     }
114                     //Individual checks for na_private_key_loc,
115                     // mss_user, and name
116                     if(!empty($_POST['na_private_key_loc'])) {
117                         $na_private_key_loc = $_POST['na_private_key_loc'];
118                     }
119                     else
120                     {
121                         echo "<BR>Please enter your Client's
122                               Private Key.<BR>";
```

```
123                         unset_na_vars();
124                         die(mysql_error());
125                     }
126                     if(!empty($_POST['na_mss_user'])) {
127                         $na_mss_user = $_POST['na_mss_user'];
128                     }
129                     else
130                     {
131                         echo "<BR>Please enter your Client's
132                               MSS User.<BR>";
133                         unset_na_vars();
134                         die(mysql_error());
135                     }
136                 }
137                 else
138                 {
139                     echo "<BR>The IP address for ".$_POST['na_name']."
140                         appears to be invalid.<BR>";
141                     unset_na_vars();
142                     die(mysql_error());
143                 }
144             }
145             //INSERT all of the values into User db
146             $query = mysql_query(
147                     "INSERT INTO Actor (
148                             guid,
149                             name,
150                             ip,
151                             port,
152                             private_key_loc,
153                             mss_user)
154                         VALUES(
155                             '$na_guid',
156                             '$na_name',
157                             '$na_ip',
158                             '$na_port',
159                             '$na_private_key_loc',
160                             '$na_mss_user' ) ")
161                     or die(mysql_error() . "\n Query1: " . $query);
162
163             echo "New CLIENT ".$na_name." inserted.<HR><pre>
164                   <b>GUID:</b>        ".$na_guid."<BR>
165                   <b>IP and Port:</b> ".$na_ip.":".$na_port."<BR>
166                   <b>Private Key:</b> ".$na_private_key_loc."<BR>
167                   <b>MSS User:</b>    ".$na_mss_user."</pre>";
168             unset_na_vars();
169             mysql_close(connect_Users());        }
170         else
171         {
172             echo "<BR> Please fill out the form completely.";
173             unset_na_vars();
174             mysql_close(connect_Users());
175         }
176
177
178         // Set the variables to null and close the connection
179         unset_na_vars();
180         mysql_close(connect_Users());
181     }
182     else
183     {
184         //Display the texboxes in table form
185         ?>
```

```
186         <table style="margin-left:25px; table-layout: auto;"
187                border="0"
188                cellspacing="10"
189                cellpadding="1" >
190
191                 Insert the new CLIENT data:
192                 <HR>
193                 <form method="post" action="add_client.php">
194                     <tr>
195                         <th align="right">Name</th>
196                         <td align="left">
197                             <input type="text"
198                                    name="na_name"
199                                    size="40"></td>
200                     </tr>
201                     <tr>
202                          <th align="right">IP Address</th>
203                          <td align="left">
204                             <input type="text"
205                                    name="na_ip"
206                                    size="17"></td>
207                     </tr>
208                     <tr>
209                         <th align="right">Port</th>
210                         <td align="left">
211                             <input type="text"
212                                    name="na_port"
213                                    size="5">(default: 22)</td>
214                     </tr>
215                     <tr>
216                         <th align="right">Private Key Location</th>
217                         <td align="left">
218                             <input type="text"
219                                    name="na_private_key_loc"
220                                    size="50"></td>
221                     </tr>
222                     <tr>
223                         <th align="right">MSS User Name</th>
224                         <td align="left">
225                             <input type="text"
226                                    name="na_mss_user"
227                                    size="20"></td>
228                     </tr>
229                     <tr>
230                         <th><? //Placeholder ?></th>
231                         <td align="left">
232                             <input name="na_submit"
233                                 type="Submit"
234                                 value="Submit"></td>
235                     </tr>
236                 </form>
237
238             <div id="foot">
239     <?
240     }
241
242         echo "</table>";
243 echo "</div>";
244
245 include('../scripts/footer.php');
246 echo "</div>";
247 ?>
248
```

```
249 <?php
250 ///////// END OF PAGE /////////////////
251 }
252 else
253 {
254     //time expired or access denied; log in again
255     include ('../scripts/header.php');
256     ?>
257     Either you are not allowed to access this page, or your session has expired.
258     Please <A href="index.php">log in</a> again.
259
260 <?php
261 } ?>
```

add_sponsor.php

```
<?php
//Purpose: Add a new sponsor actor
//Means: Connect to the Users database as user "workers"
//Conventions: ne stands for "new actor"
//Author:  John P. Quan
//Version: 1.0
//Date:    20120105
?>

<? ////////////////////////////////////////////////////
//STANDARD SESSION LIFE AND INACTIVITY CHECK
session_cache_expire(20);
session_start();

$inactive = 1200;
if (isset($_SESSION['start'])) {
    $session_life = time() - $_SESSION['start'];
    if ($session_life > $inactive) {
        header("Location: user_logout.php");

        $_SESSION['valid_user'] = 'false';
        //CLOSE PREVIOUS SESSION*
        $_SESSION;
        session_destroy();
    }
}
//Set the session start time
$_SESSION['start'] = time();

///////// START OF PAGE ////////////////

if ($_SESSION['valid_user'] == true
        AND $_SESSION['administrator'] == 1)
{
?>
<?php
include '../scripts/connect_Users.php';
include '../scripts/header.php';
include 'functions_User.php';

connect_Users();

echo "<h3><center>Add Sponsor</center></h3>";

//Display the menu ribbon
include 'menu.php';
//adjusts the header
echo "<BR>";
    ?>

<div id="wrap">

    <? ////////////////////////////////////////////////////
    //  Check the posted variables after "Submit"
    if (isset($_POST['na_submit']))
    {
        //Set the actor as a sponsor
        $na_sponsor = 1;

        //if a new user name is entered...
        if(!empty($_POST['na_guid']) and
           !empty($_POST['na_name']) and
```

```
63              !empty($_POST['na_ip']) and
64              !empty($_POST['na_port']) and
65              !empty($_POST['na_private_key_loc']) and
66              !empty($_POST['na_mss_user']))
67          {
68              // see if the user already exists
69              $query=sprintf("SELECT *
70                              FROM Actor");
71
72              $result = mysql_query($query);
73
74              while($row = mysql_fetch_array($result))
75              {
76
77                  //Check the GUID
78                  if($_POST['na_guid'] == $row['guid'])
79                  {
80                      echo "<BR>Actor GUID ".$_POST['na_guid'].
81                              " already exists.<BR>";
82                      unset_na_vars();
83                      die(mysql_error());
84                  }
85                  else
86                  {
87                      if(!empty($_POST['na_guid'])) {
88                          $na_guid = $_SESSION['na_guid'];
89                      }
90                      else
91                      {
92                          echo "<BR>Please enter your Sponsor's GUID.<BR>";
93                          unset_na_vars();
94                          die(mysql_error());
95                      }
96                  }
97
98                  //Check the name
99                  if($_POST['na_name'] == $row['name'])
100                 {
101                     echo "<BR>Actor name ".$_POST['na_name']."
102                           already exists.<BR>";
103                     unset_na_vars();
104                     die(mysql_error());
105                 }
106                 else
107                 {
108                     if(!empty($_POST['na_name'])) {
109                         $na_name = $_POST['na_name'];
110                     }
111                     else
112                     {
113                         echo "<BR>Please enter your Sponsor's
114                               Name.<BR>";
115                         unset_na_vars();
116                         die(mysql_error());
117                     }
118                 }
119
120                 ////////////////////////////////////////////////
121                 //Check the IP address format and port
122                 // and load the private key and mss_user
123                 if(filter_var($_POST['na_ip'], FILTER_VALIDATE_IP))
124                 {
125                     $na_ip =  $_POST['na_ip'];
```

```
126                     if(isset( $_POST['na_port']) and
127                               $_POST['na_port'] >= 0 and
128                               $_POST['na_port'] < 65536) {
129                         $na_port = $_POST['na_port'];
130                     }
131                     else {
132                         $na_port = 22;
133                     }
134                     //Individual checks for na_private_key_loc,
135                     // mss_user, and name
136                     if(!empty($_POST['na_private_key_loc'])) {
137                         $na_private_key_loc = $_POST['na_private_key_loc'];
138                     }
139                     else
140                     {
141                         echo "<BR>Please enter your Sponsor's
142                               Private Key.<BR>";
143                         unset_na_vars();
144                         die(mysql_error());
145                     }
146                     if(!empty($_POST['na_mss_user'])) {
147                         $na_mss_user = $_POST['na_mss_user'];
148                     }
149                     else
150                     {
151                         echo "<BR>Please enter your Sponsor's
152                               MSS User.<BR>";
153                         unset_na_vars();
154                         die(mysql_error());
155                     }
156                 }
157                 else
158                 {
159                     echo "<BR>The IP address for ".$_POST['na_name']."
160                         appears to be invalid.<BR>";
161                     unset_na_vars();
162                     die(mysql_error());
163                 }
164             }
165             //INSERT all of the values into User db
166             $query = mysql_query(
167                     "INSERT INTO Actor (
168                             guid,
169                             name,
170                             ip,
171                             port,
172                             private_key_loc,
173                             mss_user,
174                             sponsor)
175                         VALUES(
176                             '$na_guid',
177                             '$na_name',
178                             '$na_ip',
179                             '$na_port',
180                             '$na_private_key_loc',
181                             '$na_mss_user',
182                             '$na_sponsor' ) ")
183                     or die(mysql_error() . "\n Query2: " . $query);
184
185             echo "New SPONSOR ".$na_name." inserted.<HR><pre>
186                   <b>GUID:</b>         ".$na_guid."<BR>
187                   <b>IP and Port:</b> ".$na_ip.":".$na_port."<BR>
188                   <b>Private Key:</b> ".$na_private_key_loc."<BR>
```

```
189                 <b>MSS User:</b>    ".$na_mss_user."</pre>";
190             unset_na_vars();
191             mysql_close(connect_Users());        }
192         else
193         {
194             echo "<BR> Please fill out the form completely.";
195             unset_na_vars();
196             mysql_close(connect_Users());
197         }
198
199
200         // Set the variables to null and close the connection
201         unset_na_vars();
202         mysql_close(connect_Users());
203     }
204     else
205     {
206         //Display the texboxes in table form
207         ?>
208         <table style="margin-left:25px; table-layout: auto;"
209                border="0"
210                cellspacing="10"
211                cellpadding="1" >
212
213                 Insert the new SPONSOR data.  The Sponsor's GUID is
214                 <font color="red">REQUIRED</font>:
215                 <HR>
216                 <form method="post" action="add_sponsor.php">
217                     <tr>
218                         <th align="right">GUID</th>
219                         <td align="left">
220                             <input type="text"
221                                    name="na_guid"
222                                    size="50"></td>
223                     </tr>
224                     <tr>
225                         <th align="right">Name</th>
226                         <td align="left">
227                             <input type="text"
228                                    name="na_name"
229                                    size="40"></td>
230                     </tr>
231                     <tr>
232                          <th align="right">IP Address</th>
233                          <td align="left">
234                             <input type="text"
235                                    name="na_ip"
236                                    size="17"></td>
237                     </tr>
238                     <tr>
239                         <th align="right">Port</th>
240                         <td align="left">
241                             <input type="text"
242                                    name="na_port"
243                                    size="5">(default: 22)</td>
244                     </tr>
245                     <tr>
246                         <th align="right">Private Key Location</th>
247                         <td align="left">
248                             <input type="text"
249                                    name="na_private_key_loc"
250                                    size="50"></td>
251                     </tr>
```

```
252                     <tr>
253                         <th align="right">MSS User Name</th>
254                         <td align="left">
255                             <input type="text"
256                                    name="na_mss_user"
257                                    size="20"></td>
258                     </tr>
259                     <tr>
260                         <th><? //Placeholder ?></th>
261                         <td align="left">
262                             <input name="na_submit"
263                                 type="Submit"
264                                 value="Submit"></td>
265                     </tr>
266                 </form>
267 
268             <div id="foot">
269     <?
270     }
271 
272         echo "</table>";
273 echo "</div>";
274 
275 include('../scripts/footer.php');
276 echo "</div>";
277 ?>
278 
279 <?php
280 //////// END OF PAGE ////////////////
281 }
282 else
283 {
284     //time expired or access denied; log in again
285     include ('../scripts/header.php');
286     ?>
287     Either you are not allowed to access this page, or your session has expired.
288     Please <A href="index.php">log in</a> again.
289 
290 <?php
291 } ?>
```

add_user.php

```
<?php
//Purpose: Add a new user or administrator
//Means: Connect to the Users database as user "workers"
//Conventions: nu stands for "new user"
//Author:  John P. Quan
//Version: 1.0
//Date:    20120105
?>

<? ////////////////////////////////////////////////////
//STANDARD SESSION LIFE AND INACTIVITY CHECK
session_cache_expire(20);
session_start();

$inactive = 1200;
if (isset($_SESSION['start'])) {
    $session_life = time() - $_SESSION['start'];
    if ($session_life > $inactive) {
        header("Location: user_logout.php");

        $_SESSION['valid_user'] = 'false';
        //CLOSE PREVIOUS SESSION*
        $_SESSION;
        session_destroy();
    }
}
//Set the session start time
$_SESSION['start'] = time();

///////// START OF PAGE /////////////////

if ($_SESSION['valid_user'] == true
        AND $_SESSION['administrator'] == 1)
{
?>
<?php
include '../scripts/connect_Users.php';
include '../scripts/header.php';
include 'functions_User.php';
include 'functions_php.php';

connect_Users();

echo "<h3><center>Add User</center></h3>";

//Display the menu ribbon
include 'menu.php';
//adjusts the header
echo "<BR>";
    ?>

<div id="wrap">

    <? ////////////////////////////////////////////////////
    //  Check the user name and password after "Submit"
    if (isset($_POST['nu_submit']))
    {
        //if a new user name is entered...
        if(!empty($_POST['nu_name']))
        {
            // see if the user already exists
            $query=sprintf("SELECT name
```

```
63                          FROM User
64                          WHERE name='%s'",
65                  mysql_real_escape_string($_SESSION['nu_name']));
66
67              $result = mysql_query($query);
68              $row = mysql_fetch_array($result);
69
70              // if so, null all of the variables
71              if($row['name'])
72              {
73                  echo "<BR>User ".$_POST['nu_name']." already exists.<BR>";
74                  null_nu_vars();
75                  die(mysql_error());
76              }
77              else
78              {
79                  $nu_name = $_SESSION['nu_name'];
80              }
81          }
82          else
83          {
84              echo "<BR>You must include a user name.<BR>";
85              null_nu_vars();
86              die(mysql_error());
87          }
88          ////////////////////////////////////////////////
89          //make sure the admin puts in the same password,
90          // or or do not add the user
91          if ($_POST['nu_password1'] != $_POST['nu_password2'])
92          {
93              echo "<BR>Password entries do not match!";
94              null_nu_vars();
95              die(mysql_error());
96          }
97          else
98          {
99              $nu_password = $_SESSION['nu_password1'];
100         }
101         // add a GUID
102         $nu_guid = gen_uuid();
103         // set the administrator/user field
104         if ($_SESSION['nu_client_administrator'] == 'on')
105         {
106             $nu_administrator = 2;
107         }
108         elseif ($_SESSION['nu_local_administrator'] == 'on')
109         {
110             $nu_administrator = 1;
111         }
112         else
113         {
114             $nu_administrator = 0;
115         }
116         //INSERT all of the values into User db
117         $query = mysql_query(
118                 "INSERT INTO User (
119                         name,
120                         password,
121                         guid,
122                         administrator)
123                     VALUES(
124                         '$nu_name',
125                         '$nu_password',
```

```
                        '$nu_guid',
                        '$nu_administrator') ")
                or die(mysql_error() . "\n Query3: " . $query);
        //Display upon success
        if ($nu_administrator == 1)
        {
            echo "<BR>New ADMINISTRATOR ".$nu_name." Inserted.";
        }
        else
        {
            echo "<BR>New USER ".$nu_name." Inserted.";
        }
        // Set the variables to null and close the connection
        null_nu_vars();
        mysql_close(connect_Users());
    }
    else
    {
        //Display the texboxes in table form
        ?>
        <table style="margin-left:25px; table-layout: auto;"
               border="0"
               cellspacing="10"
               cellpadding="1" >

                Insert the new USER data:
                <HR>
                <form method="post" action="add_user.php">
                    <tr>
                        <th align="right">Name</th>
                        <td align="left">
                            <input type="text"
                                   name="nu_name"
                                   size="20"></td>
                    </tr>
                    <tr>
                        <th align="right">Password</th>
                        <td align="left">
                            <input type="password"
                                   name="nu_password1"
                                   size="20"></td>
                    </tr>
                    <tr>
                        <th align="right">Enter Password Again</th>
                        <td align="left">
                            <input type="password"
                                   name="nu_password2"
                                   size="20"></td>
                    </tr>
                    <tr>
                        <th align="right">Local Administrator</th>
                        <td align="left">
                        <input type="checkbox"
                               name="nu_local_administrator"></td>
                    </tr>
                    <tr>
                        <th align="right">Client Administrator</th>
                        <td align="left">
                        <input type="checkbox"
                               name="nu_client_administrator"></td>
                    </tr>
                    <tr>
                        <th><? //Placeholder ?></th>
```

```
189                         <td align="left">
190                         <input name="nu_submit"
191                                type="Submit"
192                                value="Submit"></td>
193                     </tr>
194                 </form>
195
196             <div id="foot">
197     <?
198 }
199         echo "</table>";
200 echo "</div>";
201
202 include('../scripts/footer.php');
203 echo "</div>";
204 ?>
205
206 <?php
207 ///////// END OF PAGE /////////////////
208 }
209 else
210 {
211     //time expired or access denied; log in again
212     include ('../scripts/header.php');
213     ?>
214     Either you are not allowed to access this page, or your session has expired.
215     Please <A href="index.php">log in</a> again.
216
217 <?php
218 } ?>
```

assign_users.php

```php
<?php
//Purpose: Grant/Deny user download ability from an Actor.
//Means: Connect to the Actor, Actor_has_User, and User dbs.
// tables as user "workers".
// One must assign the user to a client and a sponsor because
// MSS checks for both by comparing session data with the databases.
//Conventions: au stands for "assign user"
//Author:  John P. Quan
//Version: 1.0
//Date:    20120105
?>

<? ////////////////////////////////////////////////////
//STANDARD SESSION LIFE AND INACTIVITY CHECK
session_cache_expire(20);
session_start();

$inactive = 1200;
if (isset($_SESSION['start'])) {
    $session_life = time() - $_SESSION['start'];
    if ($session_life > $inactive) {
        header("Location: user_logout.php");

        $_SESSION['valid_user'] = 'false';
        //CLOSE PREVIOUS SESSION*
        $_SESSION;
        session_destroy();
    }
}
//Set the session start time
$_SESSION['start'] = time();

///////// START OF PAGE /////////////////

if ($_SESSION['valid_user'] == true
        AND $_SESSION['administrator'] == 1)
{
?>
<?php
include '../scripts/connect_Users.php';
include '../scripts/header.php';
include 'functions_User.php';

connect_Users();

echo "<h3><center>Assign Users</center></h3>";

//Display the menu ribbon
include 'menu.php';
//adjusts the header
echo "<BR>";
    ?>

<div id="wrap">

    <? ////////////////////////////////////////////////////
    //  Insert user assignments after "Submit"
    if (isset($_POST['au_submit']))
    {
        //First, delete the current Actor/User relationships
        $del=  mysql_query("DELETE FROM Actor_has_User")
        or die(mysql_error() . "\n Query4: " . $del);
```

```
63
64          //Relate each checked user to an actor
65          foreach ($_POST['au_assign'] as $value) {
66
67              //Parse the "$Actor_guid." ".$User_user_ID" string
68              // into array $var
69              $var = explode(" ", $value);
70
71              //Insert the Actor_guid, user_ID into Actor_has_User
72              $ins = mysql_query(
73              "INSERT INTO Actor_has_User (
74                      Actor_guid,
75                      User_user_ID)
76                  VALUES(
77                      '$var[0]',
78                      '$var[1]' ) ")
79              or die(mysql_error() . "\n Query5: " . $ins);
80
81              //null the au variables
82              unset_au_vars();
83          }
84          mysql_close(connect_Users());
85
86      ?>  <BR>Users assigned.<BR> <?
87      }
88      else
89      {
90      ?>
91      Assign a user to his or her respective actor.
92      <font color="green">Checked</font> users are authorized.
93      <font color="red">Unchecked</font> users are not.<BR>
94      <HR>
95      <?
96          //Display the textboxes in table form
97          // of each actor with all users below
98          // with checkboxes to relate the actor
99          // to a user
100         $query=sprintf("SELECT guid,
101                               name,
102                               ip,
103                               port,
104                               sponsor
105                         FROM Actor");
106
107         $result = mysql_query($query);
108
109         while($row = mysql_fetch_array($result)) {
110
111             ?>
112             <table style="margin-left:2px"
113                    border="0"
114                    cellspacing="10"
115                    cellpadding="1">
116               <h2>
117                  <tr align="left">
118                          <th width="198"><? echo $row['name']; ?></th>
119                  <th align="left"><div style="border: solid 0 #060;
120                                               border-color: rgb(248, 180, 66);
121                                               border-left-width:2px;
122                                               padding-left:0.5ex">
123                          <li><? echo $row['ip'].":".$row['port'];
124                                 $ip_explode = explode(":", $_SERVER['HTTP_HOST']);
125                                 if($row['sponsor'] == 1 and
```

```
126                                         ($ip_explode[0] == $row['ip']))
127                                     echo "<td><b>LOCAL SPONSOR</b></td>";
128                                 elseif ($row['sponsor'] == 1)
129                                     echo "<td><b>Client Sponsor</b></td>"?></li>
130                             </div></th>
131                     </tr>
132                     <tr>
133                         <td align="right">
134                             <form id="form_assign"
135                                       method="POST"
136                                       action="assign_users.php">
137                                 <input name="au_submit"
138                                       type="Submit"
139                                       value="Submit All">
140                         </td>
141                         <td>
142                             <? get_user_assignments($row['guid']); ?>
143                         </td>
144                     </tr>
145                 </h2>
146     <?  } ?>
147
148             <div id="foot">
149                                 </form>
150     <?
151 }
152         echo "</table>";
153 echo "</div>";
154
155 include('../scripts/footer.php');
156 echo "</div>";
157 ?>
158
159 <?php
160 ///////// END OF PAGE /////////////////
161 }
162 else
163 {
164     //time expired or access denied; log in again
165     include ('../scripts/header.php');
166     ?>
167     Either you are not allowed to access this page, or your session has expired.
168     Please <A href="index.php">log in</a> again.
169
170 <?php
171 } ?>
```

choose_category

```
<?php
//Purpose: Get an entire category of services as an authorized sponsor.
//         Update the Attributes, Attributes_have_Services, and Services
//         tables.  Insert the administrator's info in the User table if
//         not already inserted, such as during a fresh install.
//Means: Connect to the Services and Users db as "workers", rsync the
//       services in the filesystem, and ssh mysql commands to the child
//       sponsor from the parent sponsor.
//Conventions: *_A stands for Attributes
//             *_A_h_S stands for Attributes_have_Services
//             *_S stands for Services
//             *_U stands for User
//Author:  John P. Quan
//Version: 1.0
//Date:    20120105
?>

<? ////////////////////////////////////////////////////
//STANDARD SESSION LIFE AND INACTIVITY CHECK
session_cache_expire(20);
session_start();

$inactive = 1200;
if (isset($_SESSION['start'])) {
    $session_life = time() - $_SESSION['start'];
    if ($session_life > $inactive) {
        header("Location: user_logout.php");

        $_SESSION['valid_user'] = 'false';
        //CLOSE PREVIOUS SESSION*
        $_SESSION;
        session_destroy();
    }
}
//Set the session start time
$_SESSION['start'] = time();

///////// START OF PAGE ////////////////

if ($_SESSION['valid_user'] == true
        AND $_SESSION['authorized_user'] == true)
{
?>
<?php

include '../scripts/header.php';
include 'functions_Shell.php';

echo "<h3><center>Choose Categories</center></h3>";

//Display the menu ribbon
include 'menu.php';
//adjusts the header
echo "<BR>";
    ?>

<div id="wrap">

    <? ////////////////////////////////////////////////////
    //  Choose a category to download
    if($_SESSION['administrator'] == 1 OR $_SESSION['administrator'] == 2
            AND sending_geni_heartbeats()) {
```

```
63
64          if(isset($_POST['cc_submit'])) {
65
66              include_once 'functions_Shell.php';
67              include_once 'functions_Services.php';
68              include_once 'functions_User.php';
69              include_once '../scripts/connect_Services.php';
70              include_once '../scripts/connect_Users.php';
71              include '../scripts/constants.php';
72
73              //Download the subset of services that the administrator of the
74              // child sponsor chooses
75              foreach($_POST['cc_category'] as $value) {
76
77                  //Get the directory tree from Services
78                  connect_Services();
79                  $dir_path = directory_tree($value);
80                  mysql_close(connect_Services());
81
82                  //download_category uses send_client_command, which requires
83                  // remote connection info in Users.Actors
84                  connect_Users();
85                  download_category($dir_path);
86                  mysql_close(connect_Users());
87              }
88
89              connect_Users();
90
91              //Get the db connection data for the child sponsor and
92              // prepare for a one-time administrator insertion
93              $query=sprintf("SELECT user_ID,
94                                     name,
95                                     password,
96                                     guid,
97                                     administrator
98                              FROM User
99                              WHERE user_ID='%s'",
100                    mysql_real_escape_string($_SESSION['user_ID']))
101             or die(mysql_error() . "\n Query25: " . $query);
102
103             $result = mysql_query($query);
104
105             $user = mysql_fetch_array($result);
106
107             //Insert the user, who must be a client administrator on the parent
108             // sponsor into the Users.User db on an authorized child sponsor,
109             // if there are no other entries, such as when MSS is first
110             // installed.  Duplicate Key Update effectually does nothing if
111             // the administrator already exists in the db.  This also gives
112             // the client admin the same credentials as on the parent sponsor,
113             // but sets the user as a local administrator with full privileges.
114             $ins_U = "\\\"INSERT INTO User
115                         VALUES (NULL,
116                                 '".$user['name']."',
117                                 '".$user['password']."',
118                                 '".$user['guid']."',
119                                 '1')
120                         ON DUPLICATE KEY UPDATE name = name;\\\"";
121
122             $command_U = "echo $ins_U
123                         | mysql -u".USERS_DB_USER."
124                                 -p".USERS_DB_PASS."
125                                 -hlocalhost ".
```

```
126                                 USERS_DB;
127
128             //Deliver the insert statement for the User table
129             send_client_command('ssh', $command_U);
130
131             //Prepare the command to delete the Services tables
132             // from the child sponsor so we can reload them.  We must
133             // delete Attributes_have_Services first becaues of foreign
134             // keys constraints.
135             $del = "\\\"DELETE FROM Attributes_have_Services;
136                         DELETE FROM Attributes;
137                         DELETE FROM Services;\\\"";
138             $command = "echo $del
139                         | mysql -u".SERVICES_DB_USER."
140                                 -p".SERVICES_DB_PASS."
141                                 -hlocalhost ".
142                                     SERVICES_DB;
143
144             //Send the DELETE command
145             send_client_command('ssh', $command);
146
147             //Close Users so we can connect to services
148             mysql_close(connect_Users());
149
150             //Now connect to services to get the parent sponsor's
151             // *updated* list of attributes
152             connect_Services();
153
154             //Grab ALL of the attributes
155             $query=sprintf("SELECT *
156                             FROM Attributes")
157             or die(mysql_error() . "\n Query26: " . $query);
158
159             $result = mysql_query($query);
160
161             while($row = mysql_fetch_array($result)) {
162
163                 //Prepare the insert statement for delivery to
164                 // the child sponsor
165                 $ins_A = "";
166                 //The root of the recursive table, MSS, requires
167                 // the word NULL with no single quotes in the
168                 // parent_ID
169                 if($row['parent_ID'] == NULL) {
170                 $ins_A = "\\\"INSERT INTO Attributes
171                               VALUES ('".$row['attribute_ID']."',
172                                       '".$row['attribute']."',
173                                       '".$row['description']."',
174                                               NULL);\\\"";
175                 }
176                 else {
177                 //Use parent_ID for everything else, and get rid of any
178                 // parentheses because they ruin the msql-over-ssh queries.
179                 $ins_A = "\\\"INSERT INTO Attributes
180                           VALUES ('".$row['attribute_ID']."',
181                                   '".$row['attribute']."',
182                                   '". addslashes(
183                                           str_replace("(", "--",
184                                               str_replace(")", "--",
185                                                   $row['description'])))."',
186                                   '".$row['parent_ID']."');\\\"";
187                 }
188                 $command_A = "echo $ins_A
```

```
189                             | mysql -u".SERVICES_DB_USER."
190                                     -p".SERVICES_DB_PASS."
191                                     -hlocalhost ".
192                                         SERVICES_DB;
193
194                 mysql_close(connect_Services());
195
196                 //We need the child sponsor connection data again for the
197                 // send_client_commands
198                 connect_Users();
199                 //Deliver the insert statement for the Attribute table
200                 send_client_command('ssh', $command_A);
201                 mysql_close(connect_Users());
202             }
203             //Now we will only update the Attributes_have_Services and the
204             // Services tables for the services we downloaded above, thus
205             // completing the MSS "subset of services" requirement.
206             foreach($_SESSION['cc_category'] as $category) {
207
208                 connect_Services();
209
210                 //Grab all of the Attributes_have_Services
211                 // that match the category
212                 $query=sprintf("SELECT *
213                                 FROM Attributes_have_Services
214                                 WHERE Attributes_attribute_ID='%s'",
215                     mysql_real_escape_string($category))
216                 or die(mysql_error() . "\n Query27: " . $query);
217
218                 $result = mysql_query($query);
219
220                 while($row = mysql_fetch_array($result)) {
221
222                     //Get only the chosen services
223                     $query_S=sprintf("SELECT *
224                                       FROM Services
225                                       WHERE service_ID='%s'",
226                         mysql_real_escape_string($row['Services_service_ID']))
227                     or die(mysql_error() . "\n Query28: " . $query_S);
228
229                     $result_S=  mysql_query($query_S);
230
231                     $row_S= mysql_fetch_array($result_S);
232                     //Prepare the insert statement for delivery to
233                     // the child sponsor
234                     $ins_S = "";
235                     //Remove any parentheses, as above
236                     $ins_S = "\\\"INSERT INTO Services
237                                 VALUES ('".$row_S['service_ID']."',
238                                         '".$row_S['name']."',
239                                         '".$row_S['filename']."',
240                                   '". addslashes(
241                                           str_replace("(", "--",
242                                               str_replace(")", "--",
243                                                   $row_S['description']))).",
'".$row_S['shasum']."',
244                                         '".$row_S['developer']."',
245                                         '".$row_S['publisher']."');\\\"";
246
247                     $command_S = "echo $ins_S
248                                   | mysql -u".SERVICES_DB_USER."
249                                           -p".SERVICES_DB_PASS."
250                                           -hlocalhost ".
```

```
                                                SERVICES_DB;

                    mysql_close(connect_Services());

                    //Deliver the INSERT statement for the Services table
                    connect_Users();
                    send_client_command('ssh', $command_S);
                    mysql_close(connect_Users());

                    //Prepare the insert statement for delivery to
                    // the child sponsor
                    $ins_A_h_S = "";

                    $ins_A_h_S = "\\\"INSERT INTO Attributes_have_Services
                            VALUES ('".$row['Attributes_attribute_ID']."',
                                    '".$row['Services_service_ID']."');\\\"";

                    $command_A_h_S = "echo $ins_A_h_S
                                        | mysql -u".SERVICES_DB_USER."
                                                -p".SERVICES_DB_PASS."
                                                -hlocalhost ".
                                                    SERVICES_DB;

                    //Deliver the insert statement for the
                    // Attributes_have_Services table
                    connect_Users();
                    send_client_command('ssh', $command_A_h_S);
                    mysql_close(connect_Users());
                }
            }
            echo "Updated the Services database.<BR>";
            //Unset the choose category session and post variables
            unset_cc_vars();
        }
        else
        {
        include 'functions_Services.php';
        include '../scripts/connect_Services.php';
        include '../scripts/constants.php';

        connect_Services();

            echo "This page allows administrators of authorized sponsors
                to select entire categories and all of the services within.
                \"Check\" the box next to the <i>Category</i> to download
                the entire directory of services.<BR>";

            // Find the top of the recursive Attributes tree to print
            // all of the categories.
            $query=sprintf("SELECT *
                            FROM Attributes
                            WHERE parent_ID IS NULL")
            or die(mysql_error() . "\n Query24: " . $query);

            $result = mysql_query($query);

            $row = mysql_fetch_array($result);
            ?>
            <hr>
            <table style="margin-left:2px"
                   border="0"
                   cellspacing="0"
                   cellpadding="4">
```

```
314
315                     <tr>
316                         <td align="left">
317                             <form id="form_assign"
318                                 method="POST"
319                                 action="choose_category.php">
320                                 <input name="cc_submit"
321                                     type="Submit"
322                                     value="Submit All">
323                         </td>
324                     </tr>
325                 <h2>
326                     <tr align="left">
327                             <th width="198">{Level} Category</th>
328                     <th><align-left><div style="border: solid 0 #060;
329                                             border-color: rgb(248, 180, 66);
330                                             border-left-width:2px;
331                                             padding-left:0.5ex">
332                             <li>Description</li>
333                             </align-left></div></th>
334                     </tr>
335                 </h2>
336             </table>
337
338             <table style="margin-left:3px"
339                    border="0"
340                    cellspacing="0"
341                    cellpadding="4">
342
343                         <tr align="left">
344                             <td width="197">
345                                 <? //print attribute at the top of the tree
346                                 echo "{1} ".$row[1]; ?>
347                             </td>
348
349                             <td align="left">
350                                     <div style="border: solid 0 #060;
351                                             border-color: rgb(248, 180, 66);
352                                             align: 'left';
353                                             border-left-width:2px;
354                                             padding-left:0.5ex">
355                                 <li><? //print description at the top of the
356                                        // tree
357                                     echo $row[2]; ?></li></div>
358                             </td>
359                         </tr>
360         <div id="foot">
361             <? //the attribute MSS is "hardwired" in at 1 because it's
362                // parent_ID=NULL, so start recursion at 2
363             category_tree($row[0], 2);
364         }
365         mysql_close(connect_Services());
366     }
367     else {
368         echo "This actor does not appear to be an authorized sponsor.
369             If this is incorrect, please add this actor as a sponsor on the
370             <i>Add Sponsor</i> page.<BR>";
371     }
372     ?>
373
374 <?php
375 ///////// END OF PAGE /////////////////
376 }
```

```
377 else
378 {
379     //time expired or access denied; log in again
380     include ('../scripts/header.php');
381     ?>
382     Either you are not allowed to access this page, or your session has expired.
383     Please <A href="index.php">log in</a> again.
384
385 <?php
386 } ?>
387
388 <?
389     echo "</table>";
390 echo "</div>";
391
392 include('../scripts/footer.php');
393 echo "</div>";
394 ?>
```

choose_service.php

```php
<?php
//Purpose: Get a service as an authorized user on the local machine
//Means: Connect to the Services database as user "workers"
//       Use functions match_service_attributes()
//                     service_tree()
//Conventions:
//Author:  John P. Quan
//Version: 1.0
//Date:    20120105
?>

<? ////////////////////////////////////////////////////
//STANDARD SESSION LIFE AND INACTIVITY CHECK
session_cache_expire(20);
session_start();

$inactive = 1200;
if (isset($_SESSION['start'])) {
    $session_life = time() - $_SESSION['start'];
    if ($session_life > $inactive) {
        header("Location: user_logout.php");

        $_SESSION['valid_user'] = 'false';
        //CLOSE PREVIOUS SESSION*
        $_SESSION;
        session_destroy();
    }
}
//Set the session start time
$_SESSION['start'] = time();

//////// START OF PAGE ////////////////

if ($_SESSION['valid_user'] == true
        AND $_SESSION['authorized_user'] == true)
{
?>
<?php
//choose the service to download and
// add to the user's list
include '../scripts/connect_Services.php';
include '../scripts/header.php';
include 'functions_Services.php';
include '../scripts/constants.php';

connect_Services();

echo "<h3><center>Choose Service</center></h3>";

//Display the menu ribbon
include 'menu.php';
//adjusts the header
echo "<BR>";
    ?>

<div id="wrap">

    <? ////////////////////////////////////////////////////
    //  Choose a service to download

    echo "This page allows users to select services.
          \"Click\" the <i>Service ID</i> button to download new services.<BR>";
```

```
63
64      // Find the top of the recursive Attributes tree
65      $query=sprintf("SELECT *
66                      FROM Attributes
67                      WHERE parent_ID IS NULL")
68      or die(mysql_error() . "\n Query6: " . $query);
69
70      $result = mysql_query($query);
71
72      $row = mysql_fetch_array($result);
73      ?>
74      <hr>
75       <table style="margin-left:2px" border="0" cellspacing="0" cellpadding="4">
76           <h2>
77               <tr align="left">
78                      <th width="198">{Level} Category</th>
79               <th><align-left><div style="border: solid 0 #060;
80                                           border-color: rgb(248, 180, 66);
81                                           border-left-width:2px;
82                                           padding-left:0.5ex">
83                        <li>Description</li>
84                      </align-left></div></th>
85              </tr>
86          </h2>
87       </table>
88
89       <table style="margin-left:3px" border="0" cellspacing="0" cellpadding="4">
90
91                  <tr align="left">
92                      <td width="197">
93                          <? //print attribute at the top of the tree
94                             echo "{1} ".$row[1]; ?>
95                      </td>
96
97                      <td align="left">
98                               <div style="border: solid 0 #060;
99                                           border-color: rgb(248, 180, 66);
100                                          align: 'left';
101                                          border-left-width:2px;
102                                          padding-left:0.5ex">
103                         <li><? //print description at the top of the tree
104                                echo $row[2]; ?></li></div>
105                     </td>
106                 </tr>
107 <div id="foot">
108     <? //the attribute MSS is "hardwired" in at 1 because it's parent_ID=NULL,
109        // so start recursion at 2
110     service_tree($row[0], 2);
111     ?>
112
113 <?php
114 ///////// END OF PAGE ////////////////
115 }
116 else
117 {
118     //time expired or access denied; log in again
119     include ('../scripts/header.php');
120     ?>
121     Either you are not allowed to access this page, or your session has expired.
122     Please <A href="index.php">log in</a> again.
123
124 <?php
125 } ?>
```

```
126
127 <?
128     echo "</table>";
129 echo "</div>";
130
131 include('../scripts/footer.php');
132 echo "</div>";
133 ?>
```

connect_orca.php

```
 1 <?php
 2 //Purpose: Connect to the orca database as user "workers"
 3 //Author:  John P. Quan
 4 //Version: 1.0
 5 //Date:    20120105
 6
 7 include 'constants.php';
 8
 9 function connect_orca()
10     {
11
12     $username=GENI_DB_USER;
13     $password=GENI_DB_PASS;
14     $database=GENI_DB;
15
16     $con = mysql_connect(localhost,$username,$password);
17         @mysql_select_db($database) or die( "Unable to select GENI database");
18
19         return $con;
20     }
21 ?>
```

connect_Services.php

```php
<?php
//Purpose: Connect to the Services database as user "workers"
//Author:  John P. Quan
//Version: 1.0
//Date:    20120105

include 'constants.php';

    function connect_Services()
    {

    $username=SERVICES_DB_USER;
    $password=SERVICES_DB_PASS;
    $database=SERVICES_DB;

    $con = mysql_connect(localhost,$username,$password);
        @mysql_select_db($database) or die( "Unable to select database Services");

        return $con;
    }
?>
```

connect_Users.php

```php
<?php
//Purpose: Connect to the Users database as user "workers"
//Author:  John P. Quan
//Version: 1.0
//Date:    20120105

include 'constants.php';

    function connect_Users()
    {

    $username=USERS_DB_USER;
    $password=USERS_DB_PASS;
    $database=USERS_DB;

    $con = mysql_connect(localhost,$username,$password);
        @mysql_select_db($database) or die( "Unable to select database Users");

        return $con;
    }
?>
```

constants.php

```
<?php
//Purpose: Define Global variables for this sponsor
//Means: One can put MSS files in /var/www/ , or one can put them elsewhere
//       in the filesystem, such as in the MSS_USER's home directory.  Then,
//       create a soft link to the MSS/ directory.  For example:
//       ln -s /home/work/MSS /var/www/MSS
//Conventions:
//Author:  John P. Quan
//Version: 1.0
//Date:    20120105
?>

<?
//Where to store MSS services in the filesystem
define('MSS_SERVICES', '/usr/share');

//The location of the ORCA Actor Registry
define('GENI_HEARTBEATS', 'https://geni.renci.org:12443/registry/actors.jsp');

//The mysql db for the GENI control framework
define('GENI_DB', 'orca');

//The mysql db user for the GENI control framework
define('GENI_DB_USER', 'workers');

//The mysql password for the GENI control framework
define('GENI_DB_PASS', 'InsertPasswordHere');

//The mysql db for the MSS Users db
define('USERS_DB', 'Users');

//The mysql db user for the MSS Users db
define('USERS_DB_USER', 'workers');

//The mysql password for the MSS Users db
define('USERS_DB_PASS', 'InsertPasswordHere');

//The mysql db for the MSS Services
define('SERVICES_DB', 'Services');

//The mysql db user for the MSS Services
define('SERVICES_DB_USER', 'workers');

//The mysql password for the MSS Services
define('SERVICES_DB_PASS', 'InsertPasswordHere');

//The actor that performs MSS tasks on the client's behalf, such as
// connecting to the databases.
// This is a different user than Apache's user 'www-data',
// which performs SSH and SCP tasks on the sponsor's behalf.
define('MSS_USER', 'workers');

//The location of MSS PHP files and scripts in the filesystem
define('MSS_HOME', '/home/work');

//The universal location of the MSS functions and
// shell scripts in the filesytem. Apache user www-data
// must be able to access the directory and files
define('MSS_SCRIPTS', '/scripts/');
?>
```

cookie.php

```php
<?php
//Start the session, then make session and post variables equal.
session_start();
//Check if there are any POST variables set
if(isset($_POST)) {
    //If there are POST variables set, then set a SESSION variable
    //With the same key name
    foreach($_POST as $key=>$value) {
    $_SESSION[$key] = $value;
    }
}
//Check if there are any SESSION variables set
if(isset($_SESSION)) {
    //If there are SESSION variables set, then set a POST variable
    //With the same key name, so you don't have to change your
    //Existing code to reflect '$_SESSION' instead of '$_POST'
    foreach($_SESSION as $key=>$value) {
    $_POST[$key] = $value;
    }
}
?>
```

del_actors.php

```php
<?php
//Purpose: Delete a sponsor or client
//Means: Connect to the Users.Actor table as user "workers"
//Conventions: da stands for "delete actor"
//Author:  John P. Quan
//Version: 1.0
//Date:    20120105
?>

<?
//STANDARD SESSION LIFE AND INACTIVITY CHECK
session_cache_expire(20);
session_start();

$inactive = 1200;
if (isset($_SESSION['start']))
{
    $session_life = time() - $_SESSION['start'];
    if ($session_life > $inactive)
    {
        header("Location: user_logout.php");

        $_SESSION['valid_user'] = 'false';
        //CLOSE PREVIOUS SESSION*
        $_SESSION;
        session_destroy();
    }
}
//Set the session start time
$_SESSION['start'] = time();

///////// START OF PAGE /////////////////

if ($_SESSION['valid_user'] == true
    AND $_SESSION['administrator'] == 1)
{
    ?>

    <?
    include '../scripts/connect_Users.php';
    include '../scripts/header.php';
    include 'functions_User.php';

    connect_Users();

    echo "<h3><center><font color=\"red\">Delete Actors</font></center></h3>";
    //Display the menu ribbon
    include 'menu.php';
    //adjusts the header
    echo "<BR>";
    ?>

    <div id="wrap">

    <? ////////////////////////////////////////////////////
    //  Check the user name and password after "Submit"
    if (isset($_POST['da_submit']))
    {
        echo "Deleted the following actors:<BR>";
        echo "<HR>";
        //Return a count of deleted actors.
        foreach( $_POST['del_actor'] as $value) {
```

```
63
64              $del = explode(" ", $value, 2);
65
66              echo "<pre>          $del[1]</pre>";
67
68              //Delete the Actor from Actor_has_User first
69              // because of foreign keys
70              $query=sprintf("DELETE FROM Actor_has_User
71                              WHERE Actor_guid='%s'",
72                     mysql_real_escape_string($del[0]));
73              $result=  mysql_query($query);
74
75              //Delete from Actor next
76              $query=sprintf("DELETE FROM Actor
77                              WHERE guid='%s'",
78                     mysql_real_escape_string($del[0]));
79              $result=  mysql_query($query);
80          }
81          //unset the delete actor variables
82          unset_da_vars();
83          mysql_close(connect_Users());
84      }
85      else {
86
87          // Retrieve the Actor List
88          $query = "SELECT * FROM Actor";
89          $result = mysql_query($query);
90
91          echo "<i>Check</i> the actors to delete
92                and <i>Click</i> the \"Submit All\" button.<BR>" ?>
93                  <HR>
94              <!--Create a table of current Services-->
95              <table style="margin-left:25px; table-layout: auto;"
96                     border="0"
97                     cellspacing="10"
98                     cellpadding="1" >
99                  <tr>
100                     <td align="left">
101                         <form id="form_assign"
102                               method="POST"
103                               action="del_actors.php">
104                            <input name="da_submit"
105                                   type="Submit"
106                                   value="Submit All">
107                     </td>
108                 </tr>
109                 <h2>
110                 <tr>
111                     <th><font color="red">DELETE</font></th>
112                     <th align="left">Name</th>
113                     <th>Actor GUID</th>
114                     <th align="left">IP Address</th>
115                     <th align="left">Port</th>
116                     <th align="left">Private Key Location</th>
117                     <th align="left">MSS User Name</th>
118                     <th align="left">Sponsor</th>
119                 </tr>
120         <?
121         while ($row=  mysql_fetch_array($result))
122         {
123             ?>
124                 <tr>
125             <?      //List the user names with a checkbox and whether
```

```
126                     // he or she is an administrator
127                     //Display the current assignments upon opening,
128                     // Insert reassignments upon "Submit All". ?>
129                     <td align="center" width="100">
130                         <input type="checkbox"
131                                name="del_actor[]"
132                                value="<? echo $row['guid']." "
133                                           .$row['name']; ?>">
134                     </td>
135                     <td><?               echo $row['name']; ?></td>
136                     <td><?               echo $row['guid'];   ?></td>
137                     <td align="right"><? echo $row['ip']; ?></td>
138                     <td align="right"><? echo $row['port']; ?></td>
139                     <td><?               echo $row['private_key_loc']; ?></td>
140                     <td><?               echo $row['mss_user']; ?></td>
141                     <td align ="center">
142                         <? $ip_explode = explode(":", $_SERVER['HTTP_HOST']);
143                             if($row['sponsor'] == 1 and
144                                     ($ip_explode[0] == $row['ip']))
145                                 echo "LOCAL";
146                             elseif ($row['sponsor'] == 1)
147                                 echo "Client";
148                             else echo "No"; ?></td>
149                 </tr>
150             <?
151         }   ?>
152                 <tr>
153                     <td align="left">
154                         <input name="da_submit"
155                                type="Submit"
156                                value="Submit All">
157                     </td>
158                 </tr>
159                         </form>
160                 </h2>
161         <div id="foot">
162
163
164 <?      echo "</table>";
165         echo "</div>";
166
167         include('../scripts/footer.php');
168         echo "</div>";
169         ?>
170
171     <?php
172     ///////// END OF PAGE /////////////////
173     }
174 }
175 else
176 {
177
178     include ('../scripts/header.php');
179     ?>
180     <!--Print error message and offer to log in again-->
181     Either you are not allowed to access this page, or your session has expired.
182     Please <A href="index.php">log in</a> again.
183
184 <?php
185 } ?>
```

del_services.php

```php
<?php
//Purpose: Delete a user service client
//Means: Connect to the Users and Services db's as user "workers"
//Conventions: Notice the "includes embedded within the two main
//             if/else statements. "ds" stands for delete service.
//Author:  John P. Quan
//Version: 1.0
//Date:    20120105
?>

<?
//STANDARD SESSION LIFE AND INACTIVITY CHECK
session_cache_expire(20);
session_start();

$inactive = 1200;
if (isset($_SESSION['start']))
{
    $session_life = time() - $_SESSION['start'];
    if ($session_life > $inactive)
    {
        header("Location: user_logout.php");

        $_SESSION['valid_user'] = 'false';
        //CLOSE PREVIOUS SESSION*
        $_SESSION;
        session_destroy();
    }
}
//Set the session start time
$_SESSION['start'] = time();

///////// START OF PAGE /////////////////

if ($_SESSION['valid_user'] == true
    AND $_SESSION['authorized_user'] == true)
{
    ?>

    <?

    include '../scripts/header.php';

    echo "<h3><center><font color=\"red\">Delete Services</font></center></h3>";
    //Display the menu ribbon
    include 'menu.php';
    //adjusts the header
    echo "<BR>";
    ?>

    <div id="wrap">

    <? ////////////////////////////////////////////////////
    //  Check to see if the user clicked "Submit"
    if (isset($_POST['ds_submit']))
    {
        //Now include User functions
        include '../scripts/connect_Users.php';
        include 'functions_User.php';
        include 'functions_Shell.php';

        //Connect to the Users db to delete service_ID from
```

```
63          // the user and change the service totals
64          connect_Users();
65          echo "Deleted the following services:<BR>";
66          echo "<HR>";
67
68          if(isset($_POST['del_service'])) {
69              echo "For ".$_SESSION['username'].":<BR>";
70          }
71          //Get the values from the checkboxes, which are a four-part string.
72          foreach( $_POST['del_service'] as $value) {
73
74              $del = explode(" ", $value, 4);
75
76              $svc_ID = $del[0];
77              $svc_path = $del[1];
78              $svc_filename = $del[2];
79              $svc_name = $del[3];
80
81              echo "<pre>          $svc_name</pre>";
82
83              //Delete from User_has_Services first
84              // because of foriegn keys
85              $query=sprintf("DELETE FROM User_has_Services
86                              WHERE Services_service_ID='%s' AND
87                                    User_user_ID='%s'",
88                     mysql_real_escape_string($svc_ID),
89                     mysql_real_escape_string($_SESSION['user_ID']))
90              or die(mysql_error() . "\n Query29: " . $query);
91
92              $result=  mysql_query($query);
93
94              //Delete the service from Services next
95              $query=sprintf("SELECT * FROM Services
96                              WHERE service_ID='%s'",
97                     mysql_real_escape_string($svc_ID))
98              or die(mysql_error() . "\n Query23: " . $query);
99
100             $result=  mysql_query($query);
101
102             while($row= mysql_fetch_array($result)) {
103
104                 //Leave the service on the Actor unless no one wants the
105                 // service anymore.  This is necessary to save the expense
106                 // of re-downloading the service.
107                 if($row['totals'] < 2) {
108
109                     $query=sprintf("DELETE FROM Services
110                                     WHERE service_ID='%s'",
111                         mysql_real_escape_string($svc_ID));
112                     $result=  mysql_query($query);
113
114                     del_downloaded_service($svc_path, $svc_filename);
115                     echo "<pre>                    Removed from the Client.
116                           </pre>";
117                 }
118                 else {
119                     decrement_service_totals ($svc_ID);
120                 }
121             }
122
123         }
124         //Give administrators the option to delete the service for everyone
125         if($_SESSION['administrator'] == 1
```

```
126                 AND isset($_POST['del_all_service'])) {
127             echo "For Everyone: <BR>";
128         }
129         //Get the values from the checkboxes, which are a four-part string.
130         foreach( $_POST['del_all_service'] as $value) {
131 
132             $del = explode(" ", $value, 4);
133 
134             $svc_ID = $del[0];
135             $svc_path = $del[1];
136             $svc_filename = $del[2];
137             $svc_name = $del[3];
138 
139             echo "<pre>                $svc_name</pre>";
140 
141 
142             //Delete from User_has_Services first
143             // because of foriegn keys
144             $query=sprintf("DELETE FROM User_has_Services
145                             WHERE Services_service_ID='%s'",
146                    mysql_real_escape_string($svc_ID))
147             or die(mysql_error() . "\n Query20: " . $query);
148 
149             $result=  mysql_query($query);
150 
151             //Delete the service from Services next
152             $query=sprintf("DELETE FROM Services
153                             WHERE service_ID='%s'",
154                    mysql_real_escape_string($svc_ID));
155             $result=  mysql_query($query);
156 
157             //delete the service from the client
158             del_downloaded_service($svc_path, $svc_filename);
159         }
160         //unset the delete actor variables
161         unset_ds_vars();
162         //Get the updated list of user services
163         get_user_services();
164         //Close the connection to the User db
165         mysql_close(connect_Users());
166     }
167     else {
168 
169         //Connect to the Services db to get the service information
170         // and to set the value to the service_ID and name.
171         include '../scripts/connect_Services.php';
172         include 'functions_Services.php';
173 
174         connect_Services();
175 
176         echo "<i>Check</i> the services to delete from this client
177               and <i>Click</i> the \"Submit All\" button.<BR>" ?>
178                 <HR>
179             <!--Create a table of current Services-->
180             <table style="margin-left:25px; table-layout: auto;"
181                    border="0"
182                    cellspacing="10"
183                    cellpadding="1"
184                    width="80%">
185                 <tr>
186                     <td align="left">
187                         <form id="form_assign"
188                               method="POST"
```

```
189                             action="del_services.php">
190                         <input name="ds_submit"
191                                type="Submit"
192                                value="Submit All">
193                     </td>
194                 </tr>
195                 <tr>
196                 <? //Administrators get an extra "DELETE ALL" column
197                 if($_SESSION['administrator'] == 1) { ?>
198                     <th valign="top"><font color="red">DELETE MINE</font></th>
199                     <th><font color="red">DELETE ALL</font></th>
200              <? }
201                 else { ?>
202                     <th valign="top"><font color="red">DELETE</font></th>
203              <? } ?>
204                     <th align="left" width ="150">Service</th>
205                     <th>Attribute</th>
206                     <th align="left">File Name</th>
207                     <th>Description</th>
208                 </tr>
209                 <h2>
210         <?
211
212             //Retrieve the location of the service in the filesystem
213             $query = sprintf("SELECT * FROM view_Attributes_Services")
214                      or die(mysql_error() . "\n Query21: " . $query);
215
216             $result = mysql_query($query);
217
218             while ($row=  mysql_fetch_array($result))
219             {
220                 ?>
221                     <tr>
222                 <? if(in_array($row['service_ID'], $_SESSION['svc'])) {
223                         //Display user's delete option.
224                         // This only deletes from the filesystem when the
225                         // last user deletes the service. ?>
226                         <td align="center" width="100">
227                             <input type="checkbox"
228                                 name="del_service[]"
229                                 value="<? echo   $row['service_ID']." "
230                                                 .directory_tree(
231                                                  $row['attribute_ID'])." "
232                                                 .$row['filename']." "
233                                                 .$row['name']; ?>">
234                         </td>
235                 <? } else echo "<td> </td>"; ?>
236
237
238                 <? if($_SESSION['administrator'] == 1) {
239                         //Display a permanent delete option for administrators
240                         // This deletes the service regardless of the
241                         // service_totals count. ?>
242                         <td align="center" width="100"
243                             style="background-color:red">
244                             <input type="checkbox"
245                                 name="del_all_service[]"
246                                 value="<? echo $row['service_ID']." "
247                                               .directory_tree(
248                                                $row['attribute_ID'])." "
249                                               .$row['filename']." "
250                                               .$row['name']; ?>">
251                         </td>
```

```
252                         <td valign="top">
253                             <? echo $row['name']; ?></td>
254                         <td valign="top"><i>
255                             <? echo $row['attribute']; ?></i></td>
256                         <td valign="top">
257                             <? echo $row['filename']; ?></td>
258                         <td valign="top">
259                             <? echo $row['description']; ?></td>
260                     </tr>
261 
262                 <? }
263                    if(in_array($row['service_ID'], $_SESSION['svc'])
264                            AND $_SESSION['administrator'] == 0
265                            OR $_SESSION['administrator'] == 2) { ?>
266                         <td valign="top">
267                             <? echo $row['name']; ?></td>
268                         <td valign="top"><i>
269                             <? echo $row['attribute']; ?></i></td>
270                         <td valign="top">
271                             <? echo $row['filename']; ?></td>
272                         <td valign="top">
273                             <? echo $row['description']; ?></td>
274                     </tr>
275                 <? } ?>
276                 <?
277             }
278             ?>
279                 <tr>
280                     <td align="left">
281                         <input name="ds_submit"
282                                type="Submit"
283                                value="Submit All">
284                     </td>
285                 </tr>
286                         </form>
287                 </h2>
288         <div id="foot">
289 
290 <?      echo "</table>";
291         echo "</div>";
292 
293         include('../scripts/footer.php');
294         echo "</div>";
295 
296         //Close the connection to Services db
297         mysql_close(connect_Services()); ?>
298 
299     <?php
300     ///////// END OF PAGE /////////////////
301     }
302 }
303 else
304 {
305 
306     include ('../scripts/header.php');
307     ?>
308     <!--Print error message and offer to log in again-->
309     Either you are not allowed to access this page, or your session has expired.
310     Please <A href="index.php">log in</a> again.
311 
312 <?php
313 } ?>
```

del_users.php

```php
<?php
//Purpose: Delete a user or administrator
//Means: Connect to the Users.User table as user "workers"
//Conventions: du stands for "delete actor"
//Author:  John P. Quan
//Version: 1.0
//Date:    20120105
?>

<?
//STANDARD SESSION LIFE AND INACTIVITY CHECK
session_cache_expire(20);
session_start();

$inactive = 1200;
if (isset($_SESSION['start']))
{
    $session_life = time() - $_SESSION['start'];
    if ($session_life > $inactive)
    {
        header("Location: user_logout.php");

        $_SESSION['valid_user'] = 'false';
        //CLOSE PREVIOUS SESSION*
        $_SESSION;
        session_destroy();
    }
}
//Set the session start time
$_SESSION['start'] = time();

//////// START OF PAGE /////////////////

if ($_SESSION['valid_user'] == true
    AND $_SESSION['administrator'] == 1)
{
    ?>

    <?
    include '../scripts/connect_Users.php';
    include '../scripts/header.php';
    include 'functions_User.php';

    connect_Users();

    echo "<h3><center><font color=\"red\">Delete Users</font></center></h3>";
    //Display the menu ribbon
    include 'menu.php';
    //adjusts the header
    echo "<BR>";
    ?>

    <div id="wrap">

    <? ////////////////////////////////////////////////////
    //  Check the user name and password after "Submit"
    if (isset($_POST['du_submit']))
    {
        echo "Deleted the following users:<BR>";
        echo "<HR>";
        //Return a count of deleted actors.
        foreach( $_POST['del_user'] as $value) {
```

```
63
64              $del = explode(" ", $value, 2);
65
66              echo "<pre>          $del[1]</pre>";
67
68              //Delete the User from User_has_Services first
69              // because of foreign keys
70              $query=sprintf("DELETE FROM User_has_Services
71                          WHERE User_user_ID='%s'",
72                     mysql_real_escape_string($del[0]));
73              $result=  mysql_query($query);
74
75              //Delete the User from Actor_has_User next
76              // because of foreign keys
77              $query=sprintf("DELETE FROM Actor_has_User
78                          WHERE User_user_ID='%s'",
79                     mysql_real_escape_string($del[0]));
80              $result=  mysql_query($query);
81
82              //Delete from User last
83              $query=sprintf("DELETE FROM User
84                          WHERE user_ID='%s'",
85                     mysql_real_escape_string($del[0]));
86              $result=  mysql_query($query);
87          }
88          //unset the delete actor variables
89          unset_du_vars();
90          mysql_close(connect_Users());
91      }
92      else {
93
94          // Retrieve the User List
95          $query = "SELECT * FROM User";
96          $result = mysql_query($query);
97
98          echo "<i>Check</i> the users to delete
99                and <i>Click</i> the \"Submit All\" button.<BR>" ?>
100               <HR>
101             <!--Create a table of current Services-->
102             <table style="margin-left:25px; table-layout: auto;"
103                    border="0"
104                    cellspacing="10"
105                    cellpadding="1" >
106                 <tr>
107                     <td align="left">
108                         <form id="form_assign"
109                               method="POST"
110                               action="del_users.php">
111                            <input name="du_submit"
112                                   type="Submit"
113                                   value="Submit All">
114                     </td>
115                 </tr>
116                 <h2>
117                 <tr>
118                     <th><font color="red">DELETE</font></th>
119                     <th align="left">Name</th>
120                     <th>User ID</th>
121                     <th align="left">GUID</th>
122                     <th align="left">Administrator</th>
123                 </tr>
124         <?
125         while ($row=  mysql_fetch_array($result))
```

```
126         {
127             ?>
128                 <tr>
129             <?      //List the user names with a checkbox and whether
130                     // he or she is an administrator.
131                     //Display the current assignments upon opening,
132                     // Insert reassignments upon "Submit All". ?>
133                     <td align="center" width="100">
134                         <input type="checkbox"
135                                name="del_user[]"
136                                value="<? echo $row['user_ID']." "
137                                             .$row['name']; ?>">
138                     </td>
139                     <td><?               echo $row['name']; ?></td>
140                     <td align="center"><?echo $row['user_ID'];   ?></td>
141                     <td align="left"><? echo $row['guid']; ?></td>
142                     <td align ="center">
143                         <? if ($row['administrator'] == 0) echo 'No';
144                            elseif ($row['administrator'] == 1) echo 'LOCAL';
145                            elseif ($row['administrator'] == 2) echo'Client';
146                            else echo 'User type unknown!
147                                Contact your Local Administrator.'; ?></td>
148                 </tr>
149             <div id="foot">
150             <?
151         }   ?>
152                 <tr>
153                     <td align="left">
154                         <input name="du_submit"
155                                type="Submit"
156                                value="Submit All">
157                     </td>
158                 </tr>
159                         </form>
160                 </h2>
161         <div id="foot">
162
163 <?      echo "</table>";
164         echo "</div>";
165
166         include('../scripts/footer.php');
167         echo "</div>";
168         ?>
169
170     <?php
171     ///////// END OF PAGE /////////////////
172     }
173 }
174 else
175 {
176
177     include ('../scripts/header.php');
178     ?>
179     <!--Print error message and offer to log in again-->
180     Either you are not allowed to access this page, or your session has expired.
181     Please <A href="index.php">log in</a> again.
182
183 <?php
184 } ?>
```

download_service.php

```
<?php
//Purpose: Get a service from an authorized server as an authorized
// user on an authorized client machine.
//Means: Connect to the Services database as user "workers"
//       Use functions match_service_attributes()
//                     service_tree()
//Conventions:
//Author:  John P. Quan
//Version: 1.0
//Date:    20120105
?>

<? ////////////////////////////////////////////////////
//STANDARD SESSION LIFE AND INACTIVITY CHECK
session_cache_expire(20);
session_start();

$inactive = 1200;
if (isset($_SESSION['start'])) {
    $session_life = time() - $_SESSION['start'];
    if ($session_life > $inactive) {
        header("Location: user_logout.php");

        $_SESSION['valid_user'] = 'false';
        //CLOSE PREVIOUS SESSION*
        $_SESSION;
        session_destroy();
    }
}
//Set the session start time
$_SESSION['start'] = time();

//////// START OF PAGE /////////////////

if ($_SESSION['valid_user'] == true
        AND $_SESSION['authorized_user'] == true)
{
?>
<?php
//choose the service to download and
// add to the user's list
include '../scripts/connect_Users.php';
include '../scripts/header.php';
include 'functions_User.php';
include 'functions_Shell.php';
include '../scripts/constants.php';

connect_Users();

echo "<h3><center>Download Complete</center></h3>";

//Display the menu ribbon
include 'menu.php';
//adjusts the header
echo "<BR>";
    ?>

<div id="wrap">

<? ////////////////////////////////////////////////////
   //If the remote user and host exist in the Actor table,
   // then add the service to the user, increment the service
```

```
63    // totals, and download the service
64 if(actor_ip_exists($_SERVER['HTTP_HOST'])
65         AND actor_ip_exists($_SERVER['REMOTE_ADDR'])) {
66
67     //Increment the total number of users using the service
68     // first due to foriegn key constraints
69     increment_service_totals(
70             $_SESSION['svc_download']);
71     // Then add the service to the user's list
72     // of services.
73     add_user_service(
74             $_SESSION['svc_download']);
75 ?>
76     <hr>
77        <table style="margin-left:25px; table-layout: auto;"
78                border="0"
79                cellspacing="10"
80                cellpadding="1">
81              <h2>
82                <tr>
83                  <th>Location</th>
84                  <th>File</th>
85                </tr>
86              </h2>
87                <tr>
88                  <td>
89                      <? echo $_SESSION['svc_path']; ?>
90                  </td>
91                  <td>
92                      <? echo $_SESSION['svc_filename']; ?>
93                  </td>
94                </tr>
95 <?
96     ////Download the service to the client.
97     download_service(
98             $_SESSION['svc_path'],
99             $_SESSION['svc_filename']);
100    //Compare the shasums to ensure the download was successful
101    if(compare_shasum($_SESSION['svc_path'],
102            $_SESSION['svc_filename'])) {
103        echo "<font color=\"green\">Download Successful!</font><BR><HR>";
104    }
105    else {
106        echo "<font color=\"red\">Downloaded SHA sum does
107              not match Sponsor's SHA Sum!
108              Download FAILED!</font><BR><HR>";
109        del_downloaded_service($_SESSION['svc_path'],
110                $_SESSION['svc_filename']);
111    }
112 }
113 else {
114    echo "<font color='red'>
115            Service Downloading Disabled. You are not downloading from
116            an authorized address.<BR>
117            Please contact your adminstrator.
118            <color>";
119 }
120 //update the user services list in session
121 get_user_services();
122 ?>
123    <div id="foot">
124    </table>
125
```

```
126 <?php
127 ///////// END OF PAGE /////////////////
128 }
129 else
130 {
131     //time expired or access denied; log in again
132     include ('../scripts/header.php');
133     ?>
134     Either you are not allowed to access this page, or your session has expired.
135     Please <A href="index.php">log in</a> again.
136
137 <?php
138 } ?>
139 <?
140     echo "</div>";
141 include('../scripts/footer.php');
142 echo "</div>";
143 ?>
```

footer.php

```
1 <end><p1>
2 <br>Web Administrator:
3 <br>John Quan</p1></end>
4 </body>
5 </html>
```

functions_php.php

```php
<?php
//Purpose: Extended or general-purpose PHP functions
//Means:
//Conventions: gen_uuid() generates a GUID
//Author:  John P. Quan
//Version: 1.0
//Date:    20120105
?>

<? ////////////////////////// GEN_GUID ////////////////////////////////////////
   // Generate a guid to distinguish objects
function gen_uuid() {
    return sprintf( '%04x%04x-%04x-%04x-%04x-%04x%04x%04x',
        // 32 bits for "time_low"
        mt_rand( 0, 0xffff ), mt_rand( 0, 0xffff ),

        // 16 bits for "time_mid"
        mt_rand( 0, 0xffff ),

        // 16 bits for "time_hi_and_version",
        // four most significant bits holds version number 4
        mt_rand( 0, 0x0fff ) | 0x4000,

        // 16 bits, 8 bits for "clk_seq_hi_res",
        // 8 bits for "clk_seq_low",
        // two most significant bits holds zero and one for variant DCE1.1
        mt_rand( 0, 0x3fff ) | 0x8000,

        // 48 bits for "node"
        mt_rand( 0, 0xffff ), mt_rand( 0, 0xffff ), mt_rand( 0, 0xffff )
    );
}
?>

<?php ////////////// KEY_ARRAY_SEARCH ////////////////////////////////////////
      //Searches for even a partial match.
      // e.g., searching for 199.165.76.84 in http://199.165.76.84:11080/orca
      // returns the key it was found in, -1 if not found
function key_array_search($needle = null, $haystack_array = null, $skip = 0)
{
    if($needle == null || $haystack_array == null)
        die('$needle and $haystack_array are mandatory');
    foreach($haystack_array as $key => $eval)
    {
        if($skip != 0)$eval = substr($eval, $skip);
        if(stristr($eval, $needle) !== false) return $key;
    }
    return FALSE;
}
?>
```

functions_services.php

```php
<?php
//Purpose: Service related functions
//Author:  John P. Quan
//Version: 1.0
//Date:    20120105

include_once '../scripts/constants.php';
include_once 'functions_Shell.php';

?>

<? ////////////////// SERVICE_TREE ////////////////////////////
   //breadth-first recursion through Services.Attributes
   // Print the tree level and attribute
 function service_tree($ID, $level) {

    $query = sprintf("SELECT attribute_ID,
                             attribute,
                             description,
                             parent_ID
                      FROM Attributes
                      WHERE parent_ID=$ID")
            or die(mysql_error() . "\n Query30: " . $query);

    $result = mysql_query($query);
    while ($row = mysql_fetch_array($result)) {
        ?>
            <table border="0" cellspacing="0" cellpadding="4">
            <tr>
            <td width="200">
                <?  //add [one space]*[attribute level]
                    // for readability
                for ($i = 0; $i < $level; $i++) {
                    echo " ";
                }
                //print the level, attribute name, and description for each
                // attribute
                echo "{".$level."} ".$row['attribute'];
                ?>
            </td>
            <td><div style="border: solid 0 #060;
                            border-color: rgb(248, 180, 66);
                            border-left-width:2px;
                            padding-left:0.5ex">
                <li><? echo $row['description']; ?>
                    </li>
                </div>
            </td>
            </tr></table>
        <?
        //if the attribute has services, print them
        match_service_attributes($row['attribute_ID']);
        //recurse through the attribute tree and add a level each time
        service_tree($row['attribute_ID'], $level + 1);
    }
}
?>

<? ////////////////// CATEGORY_TREE ////////////////////////////
   //breadth-first recursion through Services.Attributes
   // Print the tree level, attribute, and checkbox
 function category_tree($ID, $level) {
```

```
63
64      $query = sprintf("SELECT attribute_ID,
65                               attribute,
66                               description,
67                               parent_ID
68                        FROM Attributes
69                        WHERE parent_ID=$ID")
70              or die(mysql_error() . "\n Query22: " . $query);
71
72      $result = mysql_query($query);
73      while ($row = mysql_fetch_array($result)) {
74          ?>
75              <table border="0" cellspacing="0" cellpadding="4">
76              <tr>
77              <td width="200">
78                  <?  //add [one space]*[attribute level]
79                      // for readability
80                  for ($i = 0; $i < $level; $i++) {
81                      echo " ";
82                  }
83                  //print the level, attribute name, and description for each
84                  // attribute
85                  echo "{".$level."} ".$row['attribute'];
86                  ?>
87              <?      //List the user names with a checkbox and whether
88                      // he or she is an administrator
89                      //Display the current assignments upon opening,
90                      // Insert reassignments upon "Submit All".
91                      echo display_category_checkbox ($row['attribute_ID']); ?>
92              </td>
93              <td><div style="border: solid 0 #060;
94                              border-color: rgb(248, 180, 66);
95                              border-left-width:2px;
96                              padding-left:0.5ex">
97                  <li><? echo $row['description']; ?>
98                      </li>
99                  </div>
100             </td>
101             </tr></table>
102         <?
103         //if the attribute has services, print them
104 //        match_service_attributes($row['attribute_ID']);
105         //recurse through the attribute tree and add a level each time
106         category_tree($row['attribute_ID'], $level + 1);
107     }
108 }
109 ?>
110
111 <? ////////////// DISPLAY_CATEGORY_CHECKBOX ///////////////////////
112    //Return matching services for an attribute
113  function display_category_checkbox ($attribute_ID) {
114
115          //find out which attributes have associated services
116     $query = sprintf("SELECT Services_service_ID
117                       FROM Attributes_have_Services
118                       WHERE Attributes_attribute_ID=$attribute_ID");
119     $result = mysql_query($query);
120
121     while ($row = mysql_fetch_array($result)) {
122             //create a table for each service to display
123             // below the attribute
124         return "<input type=\"checkbox\"
125
```

```
126                         name=\"cc_category[]\"
127                         value=\"".$attribute_ID."\" />";
128     }
129     return NULL;
130  }
131  ?>
132
133 <? ////////////// MATCH_SERVICE_ATTRIBUTES ///////////////////////
134    //print matching services for an attribute
135  function match_service_attributes($ID)
136 {
137     //find out which attributes have associated services
138     $query = sprintf("SELECT Services_service_ID
139                       FROM Attributes_have_Services
140                       WHERE Attributes_attribute_ID=$ID");
141     $result = mysql_query($query);
142
143     while ($row = mysql_fetch_array($result)) {
144             //create a table for each service to display
145             // below the attribute    ?>
146         <table border="0" cellspacing="0" cellpadding="4">
147           <tr>
148               <td valign="top">
149         <?
150         //print each matching service
151         display_service($row['Services_service_ID']);
152
153         //if the user clicks the service_ID, load SESSION
154         // svc_download with the service_ID
155         // svc_path with the filesystem full path
156         // svc_filename with Services.filename
157         if(isset($_SESSION['svc_download']) and
158                 sending_geni_heartbeats()) {
159             submit_download($_SESSION['svc_download']);
160
161             //get the file size to give the client some idea
162             // of how long the download will take.
163             $svc_path = $_SESSION['svc_path'];
164             $svc_filename = $_SESSION['svc_filename'];
165             $file_size = `ls -hal $svc_path \
166                           | grep $svc_filename \
167                           | awk '{print $5}'`;
168             $message = "Downloading: $svc_filename\nSize: $file_size";
169             //format for JavaScript
170             $message = preg_replace("/\r?\n/", "\\n",
171                     addslashes($message));
172             //display the alert box with filename and size
173             echo "<script type=\"text/javascript\">\n";
174             echo "    alert(\"$message\");\n";
175             echo "</script>\n\n";
176
177             //redirect to download_service.php
178             echo '<META HTTP-EQUIV="Refresh"
179                     Content="0;
180                     URL=download_service.php">';
181             exit;
182             ?>
183
184         <?
185         }
186         ?>
187             </td>
188           </tr>
```

```
189         </table>
190     <?
191     }
192 }
193 ?>
194
195 <?/////////////// DISPLAY_SERVICE ////////////////////////////////
196   //Display the service that matches the service_ID
197 function display_service($service_ID) {
198
199     //pull from the Services database
200     $query = sprintf("SELECT service_ID,
201                              name,
202                              filename,
203                              description,
204                              developer,
205                              publisher
206                       FROM Services
207                       WHERE service_ID=$service_ID");
208     $result = mysql_query($query);
209
210     //print the service as a table.  This table is uniquely formatted
211     ?>
212     <table style="font-family: monospace;"
213            border="0"
214            cellspacing="10"
215            cellpadding="4" >
216         <tr>
217             <th></th>
218             <th></th>
219             <th align="left">Service</th>
220             <th align="left">Description</th>
221             <th align="left">Developer</th>
222             <th align="left">Publisher</th>
223         </tr>
224         <?
225         while ($row = mysql_fetch_array($result)) {
226             ?>
227             <tr>
228
229                 <?
230                 $printed = FALSE;
231                 //If the service has not been printed already,
232                 // print the INSTALLED service array
233                 foreach ($_SESSION['svc'] as $svc_ID) {
234
235                     //if the service_ID matches the one in the row
236                     if (!$printed AND $svc_ID == $row['service_ID']) {
237                         ?>
238
239                         <td width="190">
240                             <? //Placeholder cell ?>
241                         </td>
242
243                         <td valign="top" align="center" width="100">
244                             <font color="green"><b>Installed</b></font>
245                         </td>
246
247                         <td valign="top" width="100">
248                             <? echo $row['name']; ?>
249                         </td>
250
251                         <td valign="top" width="700">
```

```
252                             <? echo $row['description']; ?>
253                         </td>
254
255                         <td valign="top" width="125">
256                             <? echo $row['developer']; ?>
257                         </td>
258
259                         <td valign="top" width="150">
260                             <? echo $row['publisher']; ?>
261                         </td>
262                 <? //The table and row either end here or in
263                     // if(!printed) below ?>
264                     </tr>
265                 </table>
266
267                 <?
268                 //set $printed to true, otherwise this prints
269                 // the same service times the number of total
270                 // services the user has
271                 $printed = TRUE;
272             }
273         }
274         //if it is not printed, then it is not installed
275         // Give the user the option to download the service
276         if (!$printed) {
277
278             ?>
279                 <td width="190">
280                     <? //Placeholder ?>
281                 </td>
282                 <? //Display the service_ID in the submit button
283                 // as the value. ?>
284                 <td valign="top" align="center" width="100">
285                     <form id="form_download"
286                         method="POST"
287                         action="choose_service.php">
288                         <input type="submit"
289                             value="<? //When clicked, set the button text to
290                                       // "Downloading"
291                                     if(isset($_SESSION['svc_path'])) {
292                                         echo "Downloading";
293                                     }
294                                     else echo $row['service_ID']; ?>"
295                             name="svc_download">
296                     </form>
297                 </td>
298
299                 <td valign="top" width="100">
300                 <? echo $row['name']; ?>
301                 </td>
302
303                 <td valign="top" width="700">
304                 <? echo $row['description']; ?>
305                 </td>
306
307                 <td valign="top" width="125">
308                 <? echo $row['developer']; ?>
309                 </td>
310
311                 <td valign="top" width="150">
312                 <? echo $row['publisher']; ?>
313                 </td>
314                 </tr>
```

```
315             </table>
316             <? //Now this service's table is definitely printed,
317                // if it exists
318             $printed = TRUE;
319         }
320     }
321 }
322 ?>
323
324 <? ///////////////// DIRECTORY_TREE ////////////////////////////////////
325    //Depth-first recursion to find the full path of the file
326    // based on the MSS_SERVICES and it's attribute description
327 function directory_tree($attribute_ID, $full_path = '') {
328
329     $query = sprintf("SELECT attribute_ID,
330                              attribute,
331                              parent_ID
332                       FROM Attributes
333                       WHERE attribute_ID=$attribute_ID");
334     $result = mysql_query($query);
335     $row=  mysql_fetch_array($result);
336     //This recurses backwards to the top of the tree, so
337     // put the next attribute BEFORE the last.
338     if( $attribute_ID == NULL) {
339         $temp = $row['attribute']."/".$full_path;
340         $full_path = MSS_SERVICES.$temp;
341         return (string)$full_path;
342     }
343     else {
344         $temp = $row['attribute']."/".$full_path;
345         $full_path = $temp;
346         return directory_tree($row['parent_ID'], $full_path);
347     }
348 }
349 ?>
350 <? ////////////////// SUBMIT_DOWNLOAD ////////////////////////////////
351    //load SESSION svc_path with the full path name
352 function submit_download($service_ID) {
353
354     //match the attribute to the service
355     $query = sprintf("SELECT service_ID,
356                              attribute_ID,
357                              filename,
358                              shasum
359                       FROM view_Attributes_Services
360                       WHERE service_ID=$service_ID");
361     $result = mysql_query($query);
362     while($row =  mysql_fetch_array($result)) {
363         //Recursive directory tree + filename = full path
364         $_SESSION['svc_path'] =
365             directory_tree($row['attribute_ID'], "");
366         $_SESSION['svc_filename'] = $row['filename'];
367         $_SESSION['svc_shasum'] = $row['shasum'];
368     }
369 }
370 ?>
```

functions_shell.php

```
<?php
//Purpose: Runs shell scripts on the host
//Means: uses the backtick operator as user 'www-data' (The default
//       apache user) on the host, scp's and ssh's as user MSS_USER
//       on the client
//Conventions: THIS RUNS AS THE APACHE USER 'www-data', SO
//              ASSUMING MSS_USER = 'workers'
//            ON THE SERVER:
//             -change www-data's default shell to bash
//               in /etc/passwd
//             -add these groups:
//               usermod -a -G workers www-data
//                (allows www-data to run the shell scripts)
//               usermod -a -G mss www-data
//                (all MSS users belong to group mss)
//             -set the UID/GID on /home/work/MSS directory to:
//               chmod ug+s /home/work/MSS
//               chmod 0770 /home/work/MSS
//             -add the private key and change ownership to www-data, e.g.,
//              -copy the Eucalyptus private key into the /home/workers/.ssh
//               directory, then:
//                chown www-data:www-data barrowkey.private
//                chmod 0600 barrowkey.private
//            ON A CLIENT EUCALYPTUS INSTANCE (Ubuntu 11.04):
//             -as root...
//             -add group mss
//               addgroup mss
//               usermod -a -G mss ubuntu
//             -add directory /usr/share/MSS
//               mkdir /usr/share/MSS
//               chown ubuntu:mss /usr/share/MSS
//            ON A PHYSICAL CLIENT:
//             -add group mss
//               addgroup mss
//             -add user workers
//               adduser workers
//               usermod -a -G mss workers
//             -add directory /usr/share/MSS
//               mkdir /usr/share/MSS
//               chown workers:mss /usr/share/MSS
//            ON A PHYSICAL SERVER FOR A EUCALYPTUS INSTANCE:
//             -for instance, on the MSS "Add Clent" webpage, add:
//               User Name [unique name, such as its IP, or IP-PORT]
//               IP
//               PORT
//               PRIVATE KEY [the eucalyptus key, such as
//                 /home/orca/mykey.private]
//               MSS USER [ubuntu]
//            ON A PHYSICAL SERVER:
//               su workers
//             -use passwordless ssh by placing the public key in the
//              authorized_users file.  For instance:
//               ssh-keygen
//               ssh-copy-id '-p XXXX -i ~/.ssh/id_dsa.pub workers@client.ip.'
//                (the public key must be in /home/workers/.ssh/authorized_keys)
//Author:  John P. Quan
//Version: 1.0
//Date:    20120105
?>

<?
include_once 'functions_php.php';
```

```
63 include_once 'functions_User.php';
64 ?>
65
66 <? ////////////////// SENDING_GENI_HEARTBEATS ////////////////////////////////
67    //wget the ORCA Registry and check to see if the MSS_PARENT (sponsor)
68    // is in it.
69    // return TRUE || FALSE
70 function sending_geni_heartbeats() {
71
72     //sh_check_heartbeats wgets the ORCA Actor Registry, greps for
73     //     url: [url of all actors]
74     //     amdiff: [if this is > 0, you're not actively donating
75     // and returns an array( [0]=> url_0
76     //                       [1]=> amdiff_0
77     //                       [2]=> url_1
78     //                       [3]=> amdiff_1 ... )
79     $heartbeats = GENI_HEARTBEATS;
80     $script = MSS_HOME.MSS_SCRIPTS;
81
82     //Download the ORCA Actors Registry
83     // amdiff is greater than 0 if the actor is NOT donating,
84     // or if the actor does not exist on the page.
85     $h_arr = explode('/', $heartbeats);
86
87     //Get the page name at the end of GENI_HEARTBEATS.  As of
88     // this writing, the name of the page is "actors.jsp"
89     $page = end($h_arr);
90
91     //Run the shell script to get all actors
92     // and check for sponsor's heartbeats
93     $data = `/bin/bash $script/sh_check_heartbeats $heartbeats $page`;
94     $actors = explode(" ", $data);
95
96     //In HTTP_HOST IP:PORT, exclude the port because the host is
97     // serving the GENI control framework on a different one, so
98     // it will not find a match with HTTP_HOST in the Actors Registry
99     // e.g., GENI CF: 11080, MSS: 12080
100    $ip = explode(":", $_SERVER['HTTP_HOST']);
101
102    //I know I (the sponsor) am sending heartbeats
103    // because I $found myself in the $actors array...
104    $found = key_array_search($ip[0], $actors);
105
106    if($found) {
107        //...and the next value in $found + 1 equals 0...
108        if($actors[$found + 1] == 0) {
109            //...so the child (user) can download the service
110            return TRUE;
111        }
112        //...else I must not be sending heartbeats, so the user cannot
113        // download new services.
114        else {
115            echo "<font color='red'>
116                    Service Downloading Disabled. No heartbeats present.<BR>
117                    Please contact your adminstrator.
118                <color>";
119            return FALSE;
120        }
121    }
122    //...else I am not in the Actor Registry, so the user cannot
123    // download new services.
124    else {
125        echo "<font color='red'>
```

```
126                 Service Downloading Disabled. No heartbeats present.<BR>
127                 Please contact your adminstrator.
128               <color>";
129         return FALSE;
130     }
131 }
132 ?>
133
134 <?/////////////// DOWNLOAD_SERVICE ////////////////////////////////////
135    // Download the service to the client.
136 function download_service($svc_path, $svc_filename) {
137
138     //Only download if the service does not exist on the client.
139     // We can do this because new releases of a service will be
140     // named differently.
141     $command = "ls -hal $svc_path
142                     | grep $svc_filename
143                     | awk '{print $5}'";
144
145     $return = send_client_command('ssh', $command);
146
147     //file_size returns the file size or a single character. I was
148     // not able to figure out what this character is in a timely
149     // manner.
150     if(strlen($return) < 2) {
151
152         //Make the MSS directory in MSS_SERVICES if it doesn't exist
153         $command = "mkdir -p $svc_path";
154         send_client_command('ssh', $command);
155
156         //Rsync the file to the client
157         send_client_command('file', NULL, $svc_path, $svc_filename);
158     }
159     else echo "File $svc_filename previously installed.  ";
160 }
161 ?>
162
163 <? ////////////////// DOWNLOAD_SERVICE ////////////////////////////////////
164    // Download the service to the client.
165 function download_category($dir_path) {
166
167     //Prepare to take the last directory name off of
168     // the string
169     $tmp = explode("/", $dir_path);
170     //The dir_path ends in a "/", so you have to pop
171     // it once for the copy directory...
172     array_pop($tmp);
173     //Put the shortened path back together
174     $copy_this_dir = implode("/", $tmp);
175     //...and a second time for the copy location
176     $category = array_pop($tmp);
177     //Put the shortened path back together
178     $loc = implode("/", $tmp);
179
180     //Echo the results
181     echo "Copied the ".$category." category to     ".$loc."<BR>";
182
183     //Make the MSS directory in MSS_SERVICES if it doesn't exist
184     $command = "mkdir -p ".$dir_path;
185
186     send_client_command('ssh', $command);
187
188     //Recursive secure copy the directory to the client
```

```
189     send_client_command('dir', NULL, $loc, $copy_this_dir);
190 }
191 ?>
192
193 <? ////////////////// COMPARE_SHASUMS ////////////////////////////
194    // Compare the SHASUM on the client to the session svc_shasum.
195 function compare_shasum($svc_path, $svc_filename) {
196
197     //Get the shasum from the client
198     $command = "shasum ".$svc_path.$svc_filename;
199     $shasum = send_client_command('ssh', $command);
200
201     if(strcmp($shasum, $_SESSION['svc_shasum'])) return TRUE;
202     else return FALSE;
203 }
204 ?>
205
206 <? ////////////////// DEL_DOWNLOADED_SERVICE ////////////////////////////
207    // Delete the downloaded service from the client.
208 function del_downloaded_service($svc_path, $svc_filename) {
209
210     //Remove the service from the client
211     $command = "rm ".$svc_path.$svc_filename;
212     send_client_command('ssh', $command);
213 }
214 ?>
215
216 <? ////////////////// SEND_CLIENT_COMMAND ////////////////////////////////
217 //Type is either 'rsync' or the default 'ssh'
218 // ssh commands require ('ssh', [command])
219 // rsync commands require ('file', NULL, path, file_to_copy) or
220 // rsync commands require ('dir', NULL, path, dir_to_copy)
221 //Use SSH to connect rsync and download the service because this
222 // is often the only port open for Eucalyptus instances (VMs).
223 function send_client_command($c_type='ssh',
224                              $command=NULL,
225                              $path=NULL,
226                              $copy_this=NULL) {
227
228     $script = MSS_HOME.MSS_SCRIPTS;
229
230     //The MSS user that performs MSS transactions.  For instance,
231     // physical clients may add user 'workers', but Eucalyptus instances
232     // often require user 'root' or 'ubuntu' to log in.
233     $user=(string)get_client_mss_user();
234
235     //The address you are sending the service to...
236     $address=(string)$_SERVER['REMOTE_ADDR'];
237
238         //... and the port
239     $port= (string)get_client_port();
240
241         //...and use the client's private key.
242     $private_key = (string)get_client_private_key();
243
244
245     //Run ssh commands
246     if($c_type === 'ssh') {
247
248         //Add quotes around the command for clarity...
249         $ssh_command = "\"".$command."\"";
250
251             //...and send using the BASH script sh_ssh_command
```

```
252         $data=`/bin/bash $script/sh_ssh_command $user \
253                                                 $address \
254                                                 $port \
255                                                 $private_key \
256                                                 $ssh_command`;
257         return $data;
258     }
259     //Run rysnc to download a service to the client
260     elseif($c_type === 'file') {
261
262         $full_path = $path.$copy_this;
263         $loc = $path.".";
264
265         //...and send using the BASH script sh_rsync_command
266         $data=`/bin/bash $script/sh_rsync_command $user \
267                                                   $address \
268                                                   $port \
269                                                   $private_key \
270                                                   $full_path $loc`;
271         return $data;
272     }
273     //Run rsync to download a directory to the client
274     elseif($c_type === 'dir') {
275
276         //This is the full path of the directory
277         $full_path = $copy_this;
278         //This is the directory just above the one you
279         // want to copy
280         $loc = $path;
281
282         //...and send using the BASH script sh_rsync_command
283         $data=`/bin/bash $script/sh_rsync_command $user \
284                                                   $address \
285                                                   $port \
286                                                   $private_key \
287                                                   $full_path $loc`;
288         return $data;
289     }
290     else {
291         echo "Type must be 'file', 'dir', or 'ssh'.<BR>";
292         return -1;
293     }
294 }
295 ?>
```

functions_User.php

```php
<?php
//Purpose: User related functions
//Author:  John P. Quan
//Version: 1.0
//Date:    20120105

include_once '../scripts/connect_Users.php';
include_once "../scripts/connect_orca.php";

connect_Users();
?>

<? ///////////////// AUTHORIZED_USER ////////////////////////////////////
   //Compare the session actors array with the server IP
   // to determine whether the user can access the website
function check_authorized_user() {

    //Allows for special case of no actor/user assignments,
    // such as when MSS is first installed.
    //This always allows local administrators in to the sponsor, even
    // when no users are assigned to actors.
    //The site will not allow access for others
    // unless a local administrator assigns the user to the sponsor and
    // at least one actor.
    if($_SESSION['administrator'] == 1) {
        $_SESSION['authorized_user'] = "true";

        //See if the client IP is a sponsor to set whether the
        // administrator can download to it.  The local admin can
        // always choose to make this IP a sponsor, and the act of
        // doing so cements which IPs the admin wants to allow to download.
        $ip_explode = explode(":", $_SERVER['REMOTE_ADDR']);

        foreach($_SESSION['actors'] as $value) {

            $query = sprintf("SELECT sponsor,
                                     ip,
                                     guid
                              FROM Actor
                              WHERE ip='%s'
                              AND guid='%s'",
                mysql_real_escape_string($ip_explode[0]),
                mysql_real_escape_string($value))
            or die(mysql_error() . "\n Query7: " . $query);
            $result = mysql_query($query);

            //If the server IP address is in the user's assigned
            // session actors, return TRUE, else return FALSE.
            while($actor =  mysql_fetch_array($result)) {

                //Find out if one of the authorized sponsors is this server
                if($actor['sponsor'] == 1
                        and authorized_sponsor($actor['guid'])) {
                    $_SESSION['authorized_sponsor'] = "true";
                }
            }
        }
        return TRUE;
    }
    else {

        //The person who logged on is not an adminstrator, but
```

```
63          // he or she must be a client administrator or a user, so
64          // find out if he or she is allowed access.
65          $ip_explode = explode(":", $_SERVER['HTTP_HOST']);
66
67          foreach($_SESSION['actors'] as $value) {
68
69              $query = sprintf("SELECT sponsor,
70                                       ip,
71                                       guid
72                                FROM Actor
73                                WHERE ip='%s'
74                                AND guid='%s'",
75                  mysql_real_escape_string($ip_explode[0]),
76                  mysql_real_escape_string($value))
77              or die(mysql_error() . "\n Query31: " . $query);
78              $result = mysql_query($query);
79
80              //If the server IP address is in the user's assigned
81              // session actors, return TRUE, else return FALSE.
82              while($actor =  mysql_fetch_array($result)) {
83
84                  //Find out if one of the authorized sponsors is this server
85                  if($actor['sponsor'] == 1
86                          and authorized_sponsor($actor['guid'])) {
87                      $_SESSION['authorized_sponsor'] = "true";
88                  }
89
90                  //See if the user is authorized on this actor
91                  $query = sprintf("SELECT Actor_guid,
92                                           User_user_ID
93                                    FROM Actor_has_User
94                                    WHERE Actor_guid='%s'",
95                      mysql_real_escape_string($actor['guid']))
96                  or die(mysql_error() . "\n Query8: " . $query);
97                  $result = mysql_query($query);
98
99                  while( $A_h_U = mysql_fetch_array($result)) {
100
101                     if($A_h_U['User_user_ID'] == $_SESSION['user_ID']) {
102                         $_SESSION['authorized_user'] = "true";
103
104                         return TRUE;
105                     }
106                 }
107             }
108         }
109         $_SESSION['authorized_user'] = "false";
110         return FALSE;
111     }
112 }
113 ?>
114
115 <? function authorized_sponsor($guid) {
116
117     //Close the Users connection so authorized_sponsor() can
118     // connect to the GENI db
119     mysql_close(connect_Users());
120
121     //If the actor guid matches a sponsor guid
122     // return TRUE, else return FALSE.
123
124     connect_orca();
125
```

```
126     $query = sprintf("SELECT act_guid
127                       FROM Actors
128                       WHERE act_guid='%s'",
129         mysql_real_escape_string($guid))
130     or die(mysql_error() . "\n Query18: " . $query);
131     $geni_result = mysql_query($query);
132 
133     while($geni_row = mysql_fetch_array($geni_result)) {
134 
135         //Find out if one of the authorized sponsors is this server
136         if(isset($geni_row['act_guid'])) {
137 
138             mysql_close(connect_orca());
139 
140             connect_Users();
141 
142             return TRUE;
143         }
144     }
145 
146     mysql_close(connect_orca());
147 
148     connect_Users();
149 
150     return FALSE;
151 }
152 ?>
153 
154 <? ////////////////// ADD_USER_SERVICE ////////////////////////////////////
155    // Add the service to the user's list
156    // of services and update the total
157    // number of users using the service
158 function add_user_service($service_ID) {
159     //Insert the user_ID and service_ID into
160     // User_has_Services
161     $user_ID = $_SESSION['user_ID'];
162     $query = mysql_query(
163                 "INSERT INTO User_has_Services (
164                             User_user_ID,
165                             Services_service_ID)
166                  VALUES(    '$user_ID',
167                             '$service_ID' ) ")
168                 or die(mysql_error() . "\n Query9: " . $query);
169 }
170 ?>
171 
172 <? ////////////////// INCREMENT_SERVICE_TOTALS ////////////////////////////////
173    //Increment the service totals if the user downloads the service
174 function increment_service_totals($service_ID) {
175 
176 
177     if(!service_ID_exists($service_ID)) {
178         $query = mysql_query(
179         "INSERT INTO Services (service_ID)
180                 VALUES(      '$service_ID' ) ")
181         or die(mysql_error() . "\n Query10: " . $query);
182     }
183     //find the current service total
184     $query=sprintf("SELECT service_ID,
185                            totals
186                     FROM Services
187                     WHERE service_ID=$service_ID")//,
188     or die(mysql_error() . "\n Query11: " . $query);
```

```
189
190     $result = mysql_query($query);
191     $i=0;
192     while($row = mysql_fetch_array($result)) {
193         $i = $row['totals'];
194         $i++;
195         $update = sprintf("UPDATE Services
196                            SET    totals=$i
197                            WHERE  service_ID=$service_ID")
198                     or die(mysql_error() . "\n Query12: " . $update);
199
200         mysql_query($update);
201     }
202 }
203 ?>
204
205 <? ///////////////// DECREMENT_SERVICE_TOTALS ////////////////////////////////
206    //Decrement the service totals if the user downloads the service
207 function decrement_service_totals($service_ID) {
208
209     //find the current service total
210     $query=sprintf("SELECT service_ID,
211                            totals
212                     FROM Services
213                     WHERE service_ID=$service_ID")//,
214     or die(mysql_error() . "\n Query13: " . $query);
215
216     $result = mysql_query($query);
217
218     while($row = mysql_fetch_array($result)) {
219         //Delete the row if the last user deletes the service
220         if($row['totals'] == 1) {
221             $query = mysql_query(
222             "DELETE FROM Services
223                     WHERE service_ID=$service_ID")
224             or die(mysql_error() . "\n Query14: " . $query);
225         }
226         else {
227             //decrement totals by 1
228             $i = $row['totals'];
229             $i--;
230             $update = sprintf("UPDATE Services
231                                SET    totals=$i
232                                WHERE  service_ID=$service_ID")
233             or die(mysql_error() . "\n Query15: " . $update);
234
235             mysql_query($update);
236         }
237     }
238 }
239 ?>
240
241 <? ////////////////// ACTOR_IP_EXISTS ////////////////////////////////////////
242    //Return TRUE for matching IP address, FALSE OTHERWISE
243 function actor_ip_exists($ip) {
244
245     //Remove the port from the ip
246     $ip_explode = explode(":", $ip);
247
248     //Check to see if the parent or child ip is in the Users.Actor database
249     $query = sprintf("SELECT ip
250                       FROM Actor
251                       WHERE ip='%s'",
```

```
252                     mysql_real_escape_string($ip_explode[0]))
253     or die(mysql_error() . "\n Query16: " . $query);
254
255     $result = mysql_query($query);
256     while($row = mysql_fetch_array($result)) {
257         if(isset($row['ip'])) return TRUE;
258     }
259     return FALSE;
260 }
261 ?>
262
263 <? ////////////////// SERVICE_ID_EXISTS ////////////////////////////////////////
264    //Return TRUE for matching service, FALSE OTHERWISE
265 function service_ID_exists($service_ID) {
266
267     //Check to see if the service is in the Users.Service database
268     $query = sprintf("SELECT service_ID
269                       FROM Services
270                       WHERE service_ID='%s'",
271                       mysql_real_escape_string($service_ID))
272     or die(mysql_error() . "\n Query17: " . $query);
273
274     $result = mysql_query($query);
275     while($row = mysql_fetch_array($result)) {
276         if(isset($row['service_ID'])) return TRUE;
277     }
278     return FALSE;
279 }
280 ?>
281
282 <? ////////////////// GET_CLIENT_PORT ////////////////////////////////////////
283    //Return the port number for matching IP address, FALSE OTHERWISE
284 function get_client_port() {
285
286     foreach($_SESSION['actors'] as $value) {
287
288         //Get the client port from the Users.Actor database
289         $query = sprintf("SELECT port
290                           FROM Actor
291                           WHERE ip='%s'
292                           AND guid='%s'",
293             mysql_real_escape_string($_SERVER['REMOTE_ADDR']),
294             mysql_real_escape_string($value))
295         or die(mysql_error() . "\n Query32: " . $query);
296
297         $result = mysql_query($query);
298         while($row = mysql_fetch_array($result)) {
299
300             if(isset($row['port'])) return $row['port'];
301         }
302     }
303     return FALSE;
304 }
305 ?>
306
307 <? ////////////////// GET_CLIENT_PRIVATE_KEY ////////////////////////////////
308    //Return the port number for matching IP address, FALSE OTHERWISE
309 function get_client_private_key() {
310
311     foreach($_SESSION['actors'] as $value) {
312
313         //Get the client private key from the Users.Actor database
314         $query = sprintf("SELECT private_key_loc
```

```
315                         FROM Actor
316                             WHERE ip='%s'
317                             AND guid='%s'",
318                 mysql_real_escape_string($_SERVER['REMOTE_ADDR']),
319                 mysql_real_escape_string($value))
320             or die(mysql_error() . "\n Query33: " . $query);
321 
322         $result = mysql_query($query);
323         while($row = mysql_fetch_array($result)) {
324 
325             if(isset($row['private_key_loc'])) return $row['private_key_loc'];
326         }
327     }
328     return FALSE;
329 }
330 ?>
331 
332 <? ////////////////// GET_CLIENT_MSS_USER ////////////////////////////////////
333    //Get the mss_user for the matching IP address, FALSE OTHERWISE
334 function get_client_mss_user() {
335 
336     foreach($_SESSION['actors'] as $value) {
337 
338         //Get the client username from the Users.Actor database
339         $query = sprintf("SELECT mss_user
340                         FROM Actor
341                             WHERE ip='%s'
342                             AND guid='%s'",
343                 mysql_real_escape_string($_SERVER['REMOTE_ADDR']),
344                 mysql_real_escape_string($value))
345             or die(mysql_error() . "\n Query19: " . $query);
346 
347         $result = mysql_query($query);
348         while($row = mysql_fetch_array($result)) {
349 
350             if(isset($row['mss_user'])) return $row['mss_user'];
351         }
352     }
353     return FALSE;
354 }
355 ?>
356 
357 <? ////////////////// GET_USER_SERVICES ////////////////////////////////////
358 function get_user_services() {
359 
360     unset_svc_vars();
361     if(!isset($_SESSION['svc'])) {
362 
363         $_SESSION['svc'] = array();
364     }
365 
366     //List the user's services as a array in SESSION
367     $query = sprintf("SELECT User_user_ID,
368                             Services_service_ID
369                     FROM User_has_Services
370                     WHERE User_user_ID='%s'",
371     mysql_real_escape_string($_SESSION['user_ID']));
372     $result = mysql_query($query);
373 
374     while($row=mysql_fetch_array($result)) {
375         array_push($_SESSION['svc'], $row['Services_service_ID']);
376     }
377 }
```

```
?>

<? ////////////////////// GET_USER_SERVICES ////////////////////////////////////
function get_user_actors($user_ID) {

    //List the user's services as a array in SESSION
    $_SESSION['actors'] = array();

    $query = sprintf("SELECT Actor_guid,
                             User_user_ID
                      FROM Actor_has_User
                      WHERE User_user_ID='%s'",
    mysql_real_escape_string($user_ID));
    $result = mysql_query($query);

    while($row=mysql_fetch_array($result)) {
        array_push($_SESSION['actors'], $row['Actor_guid']);
    }
}
?>

<? ////////////////////// GET_USER_ASSIGNMENTS ////////////////////////////////////
function get_user_assignments($guid) {

    //List all users under each actor
    $query = sprintf("SELECT user_ID,
                             name,
                             administrator
                      FROM User");
    $result = mysql_query($query);

    while($row=mysql_fetch_array($result)) {

?>      <table border="0" cellspacing="10" cellpadding="1">
          <tr>
    <?      //List the user names with a checkbox and whether
            // he or she is an administrator
            //Display the current assignments upon opening,
            // Insert reassignments upon "Submit All". ?>
            <td align="center" width="100">
    <?    echo '<input type="checkbox"
                      name="au_assign[]" '.
                      set_au_value($guid, $row['user_ID']).' '.
                      get_au_checked($guid, $row['user_ID']). '/>'; ?>
            </td>

            <td width="200">
                <? echo $row['name']; ?>
            </td>

            <td style="font-family: monospace;">
            <? if ($row['administrator'] == 0) echo '';
               elseif ($row['administrator'] == 1) echo 'LOCAL Admin';
               elseif ($row['administrator'] == 2) echo'Client Admin';
               else echo 'User type unknown!
                   Contact your Local Administrator.'; ?>
            </td>
          </tr>
        </table>
<?  }
}
?>
```

```
441 <? ////////////////////// SET_VALUE ////////////////////////////////////////
442    //Returns a space separated string of Actor_guid, user_ID
443 function set_au_value($guid, $user_ID) {
444
445     return "value=\"".$guid." ".$user_ID."\"";
446 }
447 ?>
448
449 <? ////////////////////// GET_CHECKED ////////////////////////////////////////
450 function get_au_checked($guid, $user_ID) {
451
452     //Check the checkbox if the user currently can download from
453     // the actor
454     $query = sprintf("SELECT Actor_guid,
455                              User_user_ID
456                       FROM Actor_has_User
457                       WHERE Actor_guid='%s'",
458                 mysql_real_escape_string($guid));
459     $result = mysql_query($query);
460
461     while($row=mysql_fetch_array($result)) {
462         if($row['User_user_ID'] == $user_ID) {
463             return "checked=\"checked\"";
464         }
465     }
466 }
467 ?>
468
469 <? ////////////////////// NULL_NU_VARS ////////////////////////////////////////
470    // Unset only the NEW USER variables
471 function null_nu_vars()
472 {
473     //Unset session variables
474     unset($_SESSION['nu_name']);
475     unset($_SESSION['nu_password1']);
476     unset($_SESSION['nu_password2']);
477     unset($_SESSION['nu_administrator']);
478     unset($_SESSION['nu_mysql_user']);
479     unset($_SESSION['nu_mysql_password1']);
480     unset($_SESSION['nu_mysql_password2']);
481     unset($_SESSION['nu_mysql_host']);
482     unset($_SESSION['nu_submit']);
483
484     //Unset page variables
485     unset($_POST['nu_name']);
486     unset($_POST['nu_password1']);
487     unset($_POST['nu_password2']);
488     unset($_POST['nu_administrator']);
489     unset($_POST['nu_mysql_user']);
490     unset($_POST['nu_mysql_password1']);
491     unset($_POST['nu_mysql_password2']);
492     unset($_POST['nu_mysql_host']);
493     unset($_POST['nu_submit']);
494 }
495 ?>
496
497 <? ////////////////////// UNSET_SVC_VARS ////////////////////////////////////////
498    // Unset only the SVC variables
499 function unset_svc_vars()
500 {
501     //Unset session variables
502     unset($_SESSION['svc']);
503     unset($_SESSION['svc_download']);
```

```
504     unset($_SESSION['svc_path']);
505     unset($_SESSION['svc_filename']);
506     unset($_SESSION['svc_shasum']);
507
508     //Unset page variables
509     unset($_POST['svc']);
510     unset($_POST['svc_download']);
511     unset($_POST['svc_path']);
512     unset($_POST['svc_filename']);
513     unset($_POST['svc_shasum']);
514 }
515 ?>
516
517 <? ////////////////////// UNSET_NA_VARS ////////////////////////////////////
518    // Unset only the NEW ACTOR variables
519 function unset_na_vars()
520 {
521     //Unset session variables
522     unset($_SESSION['na_guid']);
523     unset($_SESSION['na_name']);
524     unset($_SESSION['na_ip']);
525     unset($_SESSION['na_port']);
526     unset($_SESSION['na_private_key_loc']);
527     unset($_SESSION['na_mss_user']);
528     unset($_SESSION['na_sponsor']);
529     unset($_SESSION['na_submit']);
530
531
532     //Unset page variables
533     unset($_POST['na_guid']);
534     unset($_POST['na_name']);
535     unset($_POST['na_ip']);
536     unset($_POST['na_port']);
537     unset($_POST['na_private_key_loc']);
538     unset($_POST['na_mss_user']);
539     unset($_POST['na_sponsor']);
540     unset($_POST['na_submit']);
541 }
542 ?>
543
544 <? ////////////////////// UNSET_AU_VARS ////////////////////////////////////
545    // unset only the ASSIGN USER variables
546 function unset_au_vars()
547 {
548     //Unset session variables
549     unset($_SESSION['au_assign']);
550     unset($_SESSION['au_submit']);
551
552     //Unset page variables
553     unset($_POST['au_assign']);
554     unset($_POST['au_submit']);
555 }
556 ?>
557
558 <? ////////////////////// UNSET_DA_VARS ////////////////////////////////////
559    // unset only the DEL_ACTORS variables
560 function unset_da_vars()
561 {
562     //Unset session variables
563     unset($_SESSION['del_actor']);
564     unset($_SESSION['da_submit']);
565
566     //Unset page variables
```

```
567     unset($_POST['del_actor']);
568     unset($_POST['da_submit']);
569 }
570 ?>
571
572 <? ////////////////////// UNSET_DU_VARS ////////////////////////////////////
573    // unset only the DEL_USERS variables
574 function unset_du_vars()
575 {
576     //Unset session variables
577     unset($_SESSION['del_user']);
578     unset($_SESSION['du_submit']);
579
580     //Unset page variables
581     unset($_POST['del_user']);
582     unset($_POST['du_submit']);
583 }
584 ?>
585
586 <? ////////////////////// UNSET_DS_VARS ////////////////////////////////////
587    // unset only the DEL_SERVICES variables
588 function unset_ds_vars()
589 {
590     //Unset session variables
591     unset($_SESSION['del_service']);
592     unset($_SESSION['del_all_service']);
593     unset($_SESSION['ds_submit']);
594
595     //Unset page variables
596     unset($_POST['del_service']);
597     unset($_POST['del_all_service']);
598     unset($_POST['ds_submit']);
599 }
600 ?>
601
602 <? ////////////////////// UNSET_CC_VARS ////////////////////////////////////
603    // unset only the CHOOSE_CATEGORY variables
604 function unset_cc_vars()
605 {
606     //Unset session variables
607     unset($_SESSION['cc_category']);
608     unset($_SESSION['cc_submit']);
609
610     //Unset page variables
611     unset($_POST['cc_category']);
612     unset($_POST['cc_submit']);
613 }
614 ?>
```

header.php

```
 1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
 2 <html xmlns="http://www.w3.org/1999/xhtml">
 3
 4
 5 <head>
 6         <link href="style.css" rel="stylesheet" type="text/css"/>
 7     <title>GENI: Mutualistic Software Services</title>
 8 </head>
 9 <body>
10 <table cellspacing="20"><tr><td><img src="mssLogo.png"/></td><td><h1>
11 Mutualistic Software Services (MSS)</h1></td></tr></table>
12 <p></p>
13 <?php include('cookie.php')?>
```

index.php

```
 1 <?php
 2
 3 //Purpose: Connect to Users with username and password.
 4 //         Log out after 20 minutes of inactivity.
 5 //Author:  John P. Quan
 6 //Version: 1.0
 7 //Date:    20120105
 8
 9 //Start a new Login SESSION
10 ?>
11 <? //STANDARD SESSION LIFE AND INACTIVITY CHECK
12 session_cache_expire( 20 );
13 session_start();
14 $inactive = 1200;
15 if(isset($_SESSION['start']) )
16 {
17     $session_life = time() - $_SESSION['start'];
18     if($session_life > $inactive)
19     {
20         header("Location: user_logout.php");
21
22         $_SESSION['valid_user']  = 'false';
23         //CLOSE PREVIOUS SESSION*
24         $_SESSION;
```

```
25          session_destroy();
26      }
27  }
28
29  $_SESSION['start'] = time();
30  //BELOW IS UNNECESSARY FOR THE INDEX.PHP PAGE
31  //if($_SESSION['valid_user'] == true
32  //         AND $_SESSION['authorized_user'] == true)
33  //{
34  ?>
35
36  <? //STANDARD AREA TO INCLUDE FILES AND INITIALIZE DATA
37
38  include '../scripts/header.php';
39  include '../scripts/connect_Users.php';
40  include 'functions_User.php';
41
42  connect_Users();
43
44  ////initialize user data
45  $_SESSION['valid_user']    = "false";
46  $_SESSION['username']      = "";
47  $_SESSION['user_ID']       = 0;
48  $_SESSION['password']      = "";
49  $_SESSION['guid']          = "0";
50  $_SESSION['administrator'] = "";
51  //initialize the user's service list
52
53  echo "<h3><center>Welcome</center></h3>";
54
55  ?>
56
57  <?php //If there is no username/password, enter them
58
59  if ($_POST['username'] == "")
60  {
61  ?>
62
63  <div id="wrap">
64
65      <? echo "Provide your MSS Credentials:<BR>" ?>
66              <HR>
67      <form method="post" action="index.php">
68          <table style="margin-left:3px"
69                 border="0"
70                 cellspacing="10"
71                 cellpadding="1">
72              <tr align="left">
73
74                  <td width="10">
75                      <? //Placeholder ?>
76                  </td>
77                  <th>
78                      Username:
79                  </th>
80                  <td>
81                      <input type="text"
82                             name="username"
83                             size="20">
84                  </td>
85              </tr>
86              <tr>
87                  <td width="10">
```

```
88                      <? //Placeholder ?>
89                  </td>
90                  <th>
91                      Password:
92                  </th>
93                  <td>
94                      <input type="password"
95                             name="password"
96                             size="20"><BR>
97
98                  </td>
99              <tr>
100                 <td width="10">
101                     <? //Placeholder ?>
102                 </td>
103                 <th>
104                     <? //Placeholder ?>
105                 </th>
106                 <td>
107                     <input type="Submit"
108                            value="Submit">
109                 </td>
110             </tr>
111         </table>
112       </form>
113
114 <?php
115 }
116 else
117 {
118     //grab the input
119     $username = $_POST['username'];
120     $password = $_POST['password'];
121
122     //compare the input to the users in the Users database
123     $query = sprintf("SELECT user_ID,
124                              name,
125                              password,
126                              guid,
127                              administrator
128                       FROM User
129                       WHERE name='%s'
130                         AND password='%s'",
131                       mysql_real_escape_string($username),
132                       mysql_real_escape_string($password));
133
134     $result= mysql_query($query);
135
136     //Print on error...
137     if (!$result)
138     {
139         $message  = 'Invalid query: ' . mysql_error() . "\n";
140         $message .= 'Whole query: '    . $query;
141         die($message);
142         // ...and make sure invalid users can't get in to other pages
143         $_SESSION['valid_user'] = "false";
144         session_unset();
145         ?>
146         Error in username or password.
147         Please <A href="index.php">log in</a> again.
148         <?
149     }
150     else
```

```
151     {
152         //Load the rest of the user's data into the session...
153         $row =  mysql_fetch_array($result);
154         //...if they give the right credentials
155         if($row['name'] == $username AND $row['password'] == $password)
156         {
157             $_SESSION['user_ID']       = $row['user_ID'];
158             $_SESSION['username']      = $row['name'];
159             $_SESSION['password']      = $row['password'];
160             $_SESSION['guid']          = $row['guid'];
161             $_SESSION['administrator'] = $row['administrator'];
162             // allow a valid_user to view the website
163             $_SESSION['valid_user'] = "true";
164             $valid_user = "true";
165         }
166         else $_SESSION['valid_user'] = "false";
167     }
168     //Get the user's actors for which he or she is authorized to
169     // download services.
170     get_user_actors($_SESSION['user_ID']);
171     //Find out if the server address matches the list of sponsors the
172     // user is allowed to download from.
173     check_authorized_user();
174     $authorized_user = $_SESSION['authorized_user'];
175     //Show the available web pages if you are a valid, authorized user
176     if ($valid_user == "true" and $authorized_user == "true")
177     {
178         include 'menu.php';
179         //adjusts the header
180         echo "<BR>"; ?>
181
182         <?
183         get_user_services();
184         $mss_locaton=MSS_SERVICES;
185         ?>
186
187 <div id="wrap">
188
189     <? // Give the instructions for each page
190     echo "Here are the directions for using MSS:<BR>" ?>
191         <hr>
192     <table style="margin-left:3px"
193            border="0"
194            cellspacing="0"
195            cellpadding="4">
196
197             <tr>
198                 <td valign="top" width="197">
199                     <?
200                     echo "<b>Sponsor</b>";
201                     ?>
202                 </td>
203                 <td><div style="border: solid 0 #060;
204                                 border-color: rgb(248, 180, 66);
205                                 border-left-width:2px;
206                                 padding-left:0.5ex">
207                     <li><? //
208                         echo "This is the actor in your Global Environment for
209                               Network Innovation (GENI) control framework.
210                               Your sponsor must be actively donating to GENI in
211                               order to download new services.";
212                         ?>
213                     </li></div>
```

```
214                 </td>
215             </tr>
216
217             <tr>
218                 <td valign="top" width="197">
219                     <?
220                     echo "<b>Sponsored Services</b>";
221                     ?>
222                 </td>
223                 <td><div style="border: solid 0 #060;
224                                 border-color: rgb(248, 180, 66);
225                                 border-left-width:2px;
226                                 padding-left:0.5ex">
227                     <li><? //
228                         echo "The subset of GENI services your sponsor
229                               provides.";
230                         ?>
231                     </li></div>
232                 </td>
233             </tr>
234
235             <tr>
236                 <td valign="top" width="197">
237                     <?
238                     echo "<b>Choose Services</b>";
239                     ?>
240                 </td>
241                 <td><div style="border: solid 0 #060;
242                                 border-color: rgb(248, 180, 66);
243                                 border-left-width:2px;
244                                 padding-left:0.5ex">
245                     <li><? //
246                         echo "Use this page to download services to your
247                               computer.";
248                         ?>
249                     </li></div>
250                 </td>
251             </tr>
252
253         <? if($_SESSION['administrator'] == 1
254                 OR $_SESSION['administrator'] == 2) {
255             if($_SESSION['authorized_sponsor'] == "true") { ?>
256
257                 <tr>
258                     <td valign="top" width="197">
259                         <?
260                         echo "<b>Choose Categories</b>";
261                         ?>
262                     </td>
263
264                     <td><div style="border: solid 0 #060;
265                                     border-color: rgb(248, 180, 66);
266                                     border-left-width:2px;
267                                     padding-left:0.5ex">
268                         <li><? //
269                             echo "This option is only available to an authorized
270                                   sponsor, which is a server that hosts a subset
271                                   of its parent sponsor.  The sponsor must be
272                                   identified as such by using the <i>Add
273                                   Sponsor</i> page.  You must use the GUID of a
274                                   currently donating GENI control framework
275                                   actor in order to download new services from
276                                   your parent sponsor.  For instance, Duke
```

```
277                                 University might serve ORCA-related services
278                                 and have many child sponsors, which in turn
279                                 would be a parent sponsor of many child
280                                 sponsors, and so on.  A high level view of
281                                 this arrangement might look like this:
282
283     <pre>
284
285
286 GENI MSS CENTER --> PlanetLab --> ..
287                 --> ProtoGENI --> ..
288                 --> ORCA --> physical client1
289                          --> physical client2
290                          --> physical client..N
291                          --> virtual client1
292                          --> virtual client2
293                          --> virtual client..N
294                          --> orca-uaf-2 --> physical client1
295                                         --> physical client2
296                                         --> physical client..N
297                                         --> virtual client1
298                                         --> virtual client2
299                                         --> virtual client..N
300                                         --> orca-barrow-0 --> physical client1
301                                                           --> physical client2
302                                                           --> physical client..N
303                                                           --> virtual client1
304                                                           --> virtual client2
305                                                           --> virtual client..N
306    </pre>";
307                             ?>
308                         </li></div>
309                     </td>
310                 </tr>
311         <?      }
312            }?>
313
314         <? if($_SESSION['administrator'] == 1) {  ?>
315             <tr>
316                 <td valign="top" width="197">
317                     <?
318                     echo "<b>List Users</b>";
319                     ?>
320                 </td>
321
322                 <td><div style="border: solid 0 #060;
323                                 border-color: rgb(248, 180, 66);
324                                 border-left-width:2px;
325                                 padding-left:0.5ex">
326                     <li><? //
327                         echo "List all of the users authorized to log on to
328                               this sponsor";
329                         ?>
330                     </li></div>
331                 </td>
332             </tr>
333
334             <tr>
335                 <td valign="top" width="197">
336                     <?
337                     echo "<b>List Actors</b>";
338                     ?>
339                 </td>
```

```
340
341                 <td><div style="border: solid 0 #060;
342                                 border-color: rgb(248, 180, 66);
343                                 border-left-width:2px;
344                                 padding-left:0.5ex">
345                     <li><? //
346                         echo "List all of the computers authorized to
347                               connect to this sponsor.  You must also load
348                               your sponsor into this list by using <i>Add
349                               Sponsor</i>.  You will find your sponsor's
350                               Globally Unique Identifier (GUID) on the
351                               <i>Sponsor</i> page.";
352                         ?>
353                     </li></div>
354                 </td>
355             </tr>
356
357             <tr>
358                 <td valign="top" width="197">
359                     <?
360                     echo "<b>Add User</b>";
361                     ?>
362                 </td>
363
364                 <td><div style="border: solid 0 #060;
365                                 border-color: rgb(248, 180, 66);
366                                 border-left-width:2px;
367                                 padding-left:0.5ex">
368                     <li><? //
369                         echo "List all of the users authorized on this sponsor";
370                         ?>
371                     </li></div>
372                 </td>
373             </tr>
374
375             <tr>
376                 <td valign="top" width="197">
377                     <?
378                     echo "<b>Add Client</b>";
379                     ?>
380                 </td>
381
382                 <td><div style="border: solid 0 #060;
383                                 border-color: rgb(248, 180, 66);
384                                 border-left-width:2px;
385                                 padding-left:0.5ex">
386                     <li><? //
387                         echo "Follow these steps to authorize a client to
388                               download services:<BR>
389                               (1) Add the client's unique <b>Name</b>.
390                               For instance, the name could be \"John Laptop\"
391                               or a combination of its IP and port, such as
392                               \"1.2.3.4-6001\".<BR>
393                               (2) Add the client's <b>IP address</b>.<BR>
394                               (3) Add the client's <b>Port</b>.  The default
395                               port is 22, but virtual machine (VM) hosts often
396                               forward a port to the VM's port 22, such as port
397                               6001.<BR>
398                               (4) Add the sponsor's <b>Private Key Location</b>
399                               for this actor.  VMs often have a specific private
400                               key to use that is different than the sponsor's
401                               private key.<BR>
402                               (5) Add the <b>MSS User Name</b>, which is the
```

```
403                             name one would use to remotely connect to the
404                             computer.  For instance, one must often log in
405                             to VMs as user \"root\" or \"ubuntu\".<BR>
406                             (6) In addition:<BR><BR>
407                             <pre>
408             MSS DELIVERS SERVICES USING SECURE SHELL (SSH) AND
409             SECURE COPY (SCP) AS THE APACHE USER 'www-data', SO
410             ASSUMING YOUR MSS USER IS NAMED 'workers'
411             <u>ON THE SPONSOR</u>:
412             -change www-data's default shell to bash
413              in /etc/passwd
414             -add these groups:
415                 usermod -a -G workers www-data
416                     (allows www-data to run the shell scripts)
417                 usermod -a -G mss www-data
418                     (all MSS users belong to group mss)
419             -set the UID/GID on /home/work/MSS directory to:
420                 chmod ug+s /home/work/MSS
421                 chmod 0770 /home/work/MSS
422             -set the ssh key directory so www-data can access:
423                 chmod 0750 /home/work/.ssh
424             <u>ON A CLIENT EUCALYPTUS INSTANCE</u> (such as Ubuntu 11.04):
425             -as root...
426             -add group mss
427                 addgroup mss
428                 usermod -a -G mss ubuntu
429             -add directory \"".$mss_locaton."/MSS\"
430                 mkdir \"".$mss_locaton."/MSS\"
431                 chown ubuntu:mss \"".$mss_locaton."/MSS\"
432             <u>ON A PHYSICAL CLIENT</u>:
433             -add group mss
434                 addgroup mss
435             -add user workers if the user does not exist
436                 adduser workers
437                 usermod -a -G mss workers
438             -add directory \"".$mss_locaton."/MSS\"
439                 mkdir /usr/share/MSS
440                 chown workers:mss /usr/share/MSS
441             <u>ON THE SPONSOR</u>:
442                 su workers
443             -use passwordless ssh by placing the public key in the
444              authorized_users file.  For instance:
445                    ssh-keygen
446                    ssh-copy-id '-p XXXX -i ~/.ssh/id_dsa.pub workers@client.ip.'
447                     (the public key must be in /home/workers/.ssh/authorized_keys)
448                  </pre>";
449                             ?>
450                         </li></div>
451                     </td>
452                 </tr>
453
454                 <tr>
455                     <td valign="top" width="197">
456                         <?
457                         echo "<b>Add Sponsor</b>";
458                         ?>
459                     </td>
460
461                     <td><div style="border: solid 0 #060;
462                                     border-color: rgb(248, 180, 66);
463                                     border-left-width:2px;
464                                     padding-left:0.5ex">
465                         <li><? //
```

```
466                         echo "Follow these steps to establish a sponsor, from
467                               which clients will download services:<BR>
468                               (1) Add the sponsor's <b>GUID</b>, which one can
469                               find on the <i>Sponsor</i> page.<BR>
470                               (2) Add the sponsors's unique <b>Name</b>.
471                               For instance, the name could be its hostname, such
472                               as \"orca-barrow-0\" or its IP
473                               address, such as \"1.2.3.4\".<BR>
474                               (3) Add the sponsors's <b>IP address</b>.<BR>
475                               (4) Add the sponsor's <b>Port</b>.  The default
476                               port is 22, but virtual machine (VM) hosts often
477                               forward a port to the VM's port 22, such as port
478                               6001.<BR>
479                               (5) Add the sponsor's <b>Private Key Location</b>
480                               for this actor.  VMs often have a specific private
481                               key to use that is different than the sponsor's
482                               private key.<BR>
483                               (6) Add the <b>MSS User Name</b>, which is the
484                               name one would use to remotely connect to the
485                               computer.  For instance, one must often log in
486                               to VMs as user \"root\" or \"ubuntu\".<BR>
487                               (7) In addition:<BR><pre>
488        MSS SPONSOR IS A TYPICAL LINUX APACHE MYSQL & PHP (LAMP) SERVER
489        PLEASE REFER TO THE DOCUMENTATION ON THE SPONSOR SETUP</pre>";
490            ?>
491                    </li></div>
492                </td>
493            </tr>
494
495            <tr>
496                <td valign="top" width="197">
497                    <?
498                    echo "<b>Assign Users</b>";
499                    ?>
500                </td>
501
502                <td><div style="border: solid 0 #060;
503                                border-color: rgb(248, 180, 66);
504                                border-left-width:2px;
505                                padding-left:0.5ex">
506                    <li><? //
507                        echo "Assign a specific user or users to a client or
508                              clients.  Assign the user to the sponsor
509                              from which he or she is authorized to download
510                              services.";
511                        ?>
512                    </li></div>
513                </td>
514            </tr>
515
516            <tr>
517                <td valign="top" width="197">
518                    <?
519                    echo "<b>Delete Actors</b>";
520                    ?>
521                </td>
522
523                <td><div style="border: solid 0 #060;
524                                border-color: rgb(248, 180, 66);
525                                border-left-width:2px;
526                                padding-left:0.5ex">
527                    <li><? //
528                        echo "Choose the actor or actors to delete from this
```

```
                                sponsor.";
                        ?>
                    </li></div>
                </td>
            </tr>

            <tr>
                <td valign="top" width="197">
                    <?
                    echo "<b>Delete Users</b>";
                    ?>
                </td>

                <td><div style="border: solid 0 #060;
                                border-color: rgb(248, 180, 66);
                                border-left-width:2px;
                                padding-left:0.5ex">
                    <li><? //
                        echo "Choose the user or users to delete from this
                                sponsor.";
                        ?>
                    </li></div>
                </td>
            </tr>

        <? }?>

            <tr>
                <td valign="top" width="197">
                    <?
                    echo "<b>Delete Services</b>";
                    ?>
                </td>

                <td><div style="border: solid 0 #060;
                                border-color: rgb(248, 180, 66);
                                border-left-width:2px;
                                padding-left:0.5ex">
                    <li><? //
                        echo "Choose the service or services to delete from your
                              computer. MSS will only completely remove the
                              service if all users on this computer choose to
                              delete the service.  This helps to ensure that
                              several users do not overwrite the same service by
                              choosing to download it at different times, and
                              thus saves bandwidth.
                              In addition, administrators have the
                              option to delete the service immediately.";

                        ?>
                    </li></div>
                </td>
            </tr>

            <tr>
                <td valign="top" width="197">
                    <?
                    echo "<b>Log Off</b>";
                    ?>
                </td>

                <td><div style="border: solid 0 #060;
                                border-color: rgb(248, 180, 66);
```

```
592                                 border-left-width:2px;
593                                 padding-left:0.5ex">
594                         <li><? //
595                             echo "Closes the current session and allows one
596                                   to log in again or as a different user.";
597
598                             ?>
599                         </li></div>
600                     </td>
601                 </tr>
602
603             <div id="foot">
604
605 <?  }
606     if ($_SESSION['authorized_user'] == "false") {
607     ?>
608         You are not authorized to download from this Actor.
609         Please contact your administrator.<BR>
610         You may also <A href="user_logout.php">log out</a>
611         and try to log in again.
612     <?  break;
613     }
614     if ($_SESSION['valid_user'] == "false")
615     {
616         //You must not be a registered user!
617         session_unset();
618         ?>
619         Error in username or password.
620         Please <A href="index.php">log in</a> again.
621 <?php
622     }
623     //refresh the page to load user services into the svc array
624     if(!isset($_SESSION['svc']))
625     {
626         //Create an array to hold the user's services in SESSION
627         $_SESSION['svc'] = array();
628         //Get the user's services
629         get_user_services();
630     }
631 } ?>
632
633 <?
634     echo "</table>";
635 echo "</div>";
636
637 include('../scripts/footer.php');
638 echo "</div>";
639 ?>
```

list_actors.php

```
<?php
//Purpose: Connect to the Services database as user "workers"
//Author:  John P. Quan
//Version: 1.0
//Date:    20120105
?>

<?
//STANDARD SESSION LIFE AND INACTIVITY CHECK
session_cache_expire(20);
session_start();

$inactive = 1200;
if (isset($_SESSION['start']))
{
    $session_life = time() - $_SESSION['start'];
    if ($session_life > $inactive)
    {
        header("Location: user_logout.php");

        $_SESSION['valid_user'] = 'false';
        //CLOSE PREVIOUS SESSION*
        $_SESSION;
        session_destroy();
    }
}
//Set the session start time
$_SESSION['start'] = time();

///////// START OF PAGE /////////////////

if ($_SESSION['valid_user'] == true
    AND $_SESSION['authorized_user'] == true)
{
    ?>

    <?
    include '../scripts/connect_Users.php';
    include '../scripts/header.php';

    connect_Users();

    // Retrieve the Actor List
    $query = "SELECT * FROM Actor";
    $result = mysql_query($query);

    echo "<h3><center>Current Actors</center></h3>";
    //Display the menu ribbon
    include 'menu.php';
    //adjusts the header
    echo "<BR>";
    ?>

    <div id="wrap">

<? echo "List of actors that are authorized to download
         from the sponsor:<BR>" ?>
            <HR>
        <!--Create a table of current Services-->
        <table style="margin-left:25px; table-layout: auto;"
               border="0"
               cellspacing="10"
```

```
63                cellpadding="1" >
64
65            <h2>
66            <tr>
67                <th align="left">Actor</th>
68                <th>Actor GUID</th>
69                <th>IP Address</th>
70                <th align="left">Port</th>
71                <th align="left">Private Key Location</th>
72                <th align="left">MSS User Name</th>
73                <th align="left">Sponsor</th>
74            </tr>
75    <?
76    while ($row=  mysql_fetch_array($result))
77    {
78        ?>
79            <tr>
80                <td><?               echo $row['name']; ?></td>
81                <td><?               echo $row['guid'];   ?></td>
82                <td align="right"><? echo $row['ip']; ?></td>
83                <td align="right"><? echo $row['port']; ?></td>
84                <td><?               echo $row['private_key_loc']; ?></td>
85                <td><?               echo $row['mss_user']; ?></td>
86                <td align ="center">
87                    <? $ip_explode = explode(":", $_SERVER['HTTP_HOST']);
88                       if($row['sponsor'] == 1 and
89                               ($ip_explode[0] == $row['ip']))
90                           echo "LOCAL";
91                       elseif ($row['sponsor'] == 1)
92                           echo "Client";
93                       else echo "No"; ?></td>
94            </tr>
95            </h2>
96        <div id="foot">
97
98        <?
99    }
100   echo "</table>";
101   echo "</div>";
102
103
104   mysql_close(connect_Users());
105
106   include('../scripts/footer.php');
107   echo "</div>";
108   ?>
109
110 <?php
111 ///////// END OF PAGE /////////////////
112 }
113 else
114 {
115
116    include ('../scripts/header.php');
117    ?>
118    <!--Print error message and offer to log in again-->
119    Either you are not allowed to access this page, or your session has expired.
120    Please <A href="index.php">log in</a> again.
121
122 <?php
123 } ?>
```

list_services.php

```php
<?php
//Purpose: Connect to the Services database as user "workers"
//Author:  John P. Quan
//Version: 1.0
//Date:    20120105
?>

<?
//STANDARD SESSION LIFE AND INACTIVITY CHECK
session_cache_expire(20);
session_start();

$inactive = 1200;
if (isset($_SESSION['start'])) {
    $session_life = time() - $_SESSION['start'];
    if ($session_life > $inactive) {
        header("Location: user_logout.php");

        $_SESSION['valid_user'] = 'false';
        //CLOSE PREVIOUS SESSION*
        $_SESSION;
        session_destroy();
    }
}
//Set the session start time
$_SESSION['start'] = time();

///////// START OF PAGE /////////////////

if ($_SESSION['valid_user'] == true
        AND $_SESSION['authorized_user'] == true)
{
?>

    <?
    include '../scripts/connect_Services.php';
    include '../scripts/header.php';

    connect_Services();

    // Retrieve the Services List
    $query = "SELECT * FROM Services";
    $result = mysql_query($query);

    $num = mysql_numrows($result);

    echo "<h3><center>Current Services</center></h3>";
    //Display the menu ribbon
    include 'menu.php';
    //adjusts the header
    echo "<BR>";
    ?>

    <div id="wrap">

<? echo "List of services on the sponsor:<BR>" ?>
            <HR>
        <table style="margin-left:25px; table-layout: auto;"
               border="0"
               cellspacing="10"
               cellpadding="1"
               width="97%">
```

```
63              <h2>
64                <tr>
65                  <th>Service ID</th>
66                  <th align="left">Service</th>
67                  <th align="left">File Name</th>
68                  <th>Description</th>
69                  <th>SHA Sum</th>
70                  <th align="left">Developer</th>
71                  <th align="left">Publisher</th>
72                </tr>
73      <?
74      $i = 0;
75      while ($i < $num) {
76
77          $service_ID = mysql_result($result,   $i, "service_ID");
78          $name = mysql_result($result,         $i, "name");
79          $filename = mysql_result($result,     $i, "filename");
80          $description = mysql_result($result,  $i, "description");
81          $shasum = mysql_result($result,       $i, "shasum");
82          $developer = mysql_result($result,    $i, "developer");
83          $publisher = mysql_result($result,    $i, "publisher");
84          ?>
85                <tr>
86                      <td valign="top"
87                          align="center"
88                          width="90">
89                              <? echo $service_ID; ?></td>
90                      <td valign="top">
91                              <? echo $name; ?></td>
92                      <td valign="top">
93                              <? echo $filename; ?></td>
94                      <td valign="top">
95                              <? echo $description; ?></td>
96                      <td valign="top">
97                              <? echo $shasum; ?></td>
98                      <td valign="top">
99                              <? echo $developer; ?></td>
100                     <td valign="top">
101                             <? echo $publisher; ?></td>
102               </tr>
103                 </h2>
104     <div id="foot">
105
106         <?
107         $i++;
108     }
109         echo "</table>";
110     echo "</div>";
111
112
113     mysql_close(connect_Services());
114
115     include('../scripts/footer.php');
116     echo "</div>"
117     ?>
118
119 <?php
120 ///////// END OF PAGE /////////////////
121 }
122 else
123 {
124
125     include ('../scripts/header.php');
```

```
126     ?>
127     <!--Print error message and offer to log in again-->
128     Either you are not allowed to access this page, or your session has expired.
129     Please <A href="index.php">log in</a> again.
130
131 <?php
132 } ?>
```

list_sponsor

```
  1 <?php
  2 //Purpose: Connect to the GENI database as user "workers"
  3 //Author:  John P. Quan
  4 //Version: 1.0
  5 //Date:    20120105
  6 ?>
  7
  8 <?
  9 //STANDARD SESSION LIFE AND INACTIVITY CHECK
 10 session_cache_expire(20);
 11 session_start();
 12
 13 $inactive = 1200;
 14 if (isset($_SESSION['start']))
 15 {
 16     $session_life = time() - $_SESSION['start'];
 17     if ($session_life > $inactive)
 18     {
 19         header("Location: user_logout.php");
 20
 21         $_SESSION['valid_user'] = 'false';
 22         //CLOSE PREVIOUS SESSION*
 23         $_SESSION;
 24         session_destroy();
 25     }
 26 }
 27 //Set the session start time
 28 $_SESSION['start'] = time();
 29
 30 ///////// START OF PAGE /////////////////
 31
 32 if ($_SESSION['valid_user'] == true
 33         AND $_SESSION['authorized_user'] == true)
 34 {
 35     ?>
```

```
36
37      <?
38      include '../scripts/connect_orca.php';
39      include '../scripts/header.php';
40
41      connect_orca();
42
43      // Retrieve the ORCA Sponsor
44      $query = "SELECT act_id,
45                       act_name,
46                       act_guid
47                  FROM Actors";
48      $result = mysql_query($query);
49
50      echo "<h3><center>Sponsor</center></h3>";
51      //Display the menu ribbon
52      include 'menu.php';
53      //adjusts the header
54      echo "<BR>";
55      ?>
56
57      <div id="wrap">
58
59      <? echo "List of sponsors through which local users can download
60               new services:<BR>" ?>
61              <HR>
62          <!--Create a table of current Services-->
63          <table style="margin-left:25px; table-layout: auto;"
64                 border="0"
65                 cellspacing="10"
66                 cellpadding="1" >
67
68              <h2>
69              <tr>
70                  <th>Actor ID</th>
71                  <th align="left">Sponsor</th>
72                  <th>GUID</th>
73              </tr>
74      <?
75      while ($row=  mysql_fetch_array($result))
76      {
77          ?>
78              <tr>
79                  <td><center><? echo $row['act_id'];   ?></center></td>
80                  <td><?         echo $row['act_name']; ?></td>
81                  <td><center><? echo $row['act_guid']; ?></center></td>
82              </tr>
83              </h2>
84          <div id="foot">
85
86          <?
87      }
88      echo "</table>";
89      echo "</div>";
90
91
92      mysql_close(connect_orca());
93
94      include('../scripts/footer.php');
95      echo "</div>";
96      ?>
97
98  <?php
```

```
 99 ///////// END OF PAGE /////////////////
100 }
101 else
102 {
103
104     include ('../scripts/header.php');
105     ?>
106     <!--Print error message and offer to log in again-->
107     Either you are not allowed to access this page, or your session has expired.
108     Please <A href="index.php">log in</a> again.
109
110 <?php
111 } ?>
```

list_users.php

```
  1 <?php
  2 //Purpose: Connect to the Users database as user "workers"
  3 //Author:  John P. Quan
  4 //Version: 1.0
  5 //Date:    20120105
  6 ?>
  7
  8 <?
  9 //STANDARD SESSION LIFE AND INACTIVITY CHECK
 10 session_cache_expire(20);
 11 session_start();
 12
 13 $inactive = 1200;
 14 if (isset($_SESSION['start'])) {
 15     $session_life = time() - $_SESSION['start'];
 16     if ($session_life > $inactive) {
 17         header("Location: user_logout.php");
 18
 19         $_SESSION['valid_user'] = 'false';
 20         //CLOSE PREVIOUS SESSION*
 21         $_SESSION;
 22         session_destroy();
 23     }
 24 }
```

```
25 //Set the session start time
26 $_SESSION['start'] = time();
27
28 ///////// START OF PAGE /////////////////
29
30 if ($_SESSION['valid_user'] == true
31         AND $_SESSION['administrator'] == 1)
32 {
33
34 ?>
35     <?
36     include '../scripts/connect_Users.php';
37     include '../scripts/header.php';
38
39     connect_Users();
40
41     // Retrieve the Services List
42     $query = "SELECT * FROM User";
43     $result = mysql_query($query);
44
45     $num = mysql_num_rows($result);
46
47     echo "<h3><center>Current Users</center></h3>";
48     //Display the menu ribbon
49     include 'menu.php';
50     //adjusts the header
51     echo "<BR>";
52     ?>
53
54     <div id="wrap">
55
56 <? echo "List of users who are authorized to download
57          from the sponsor:<BR>" ?>
58            <HR>
59         <table style="margin-left:25px; table-layout: auto;"
60                border="0"
61                cellspacing="10"
62                cellpadding="1" >
63             <h2>
64                 <tr>
65                 <th align="left">User ID</th>
66                 <th align="left">Name</th>
67                 <th>GUID</th>
68                 <th align="left">Administrator</th>
69                 </tr>
70     <?
71     $i = 0;
72     while ($i < $num) {
73
74         $user_ID       = mysql_result($result, $i, "user_ID");
75         $name          = mysql_result($result, $i, "name");
76         $guid          = mysql_result($result, $i, "guid");
77         $administrator = mysql_result($result, $i, "administrator");
78         if ($administrator == 0) $administrator = 'No';
79         elseif ($administrator == 1) $administrator='LOCAL';
80         elseif ($administrator == 2) $administrator='Client';
81         else $administrator='User type unknown!
82             Contact your Local Administrator.';
83         ?>
84                     <tr>
85                         <td align="center">
86                             <? echo $user_ID;       ?></td>
87                         <td><? echo $name;          ?></td>
```

```
                        <td><? echo $guid;          ?></td>
                        <td align="center">
                            <? echo $administrator; ?></td>
                    </tr>
                </h2>

    <div id="foot">

        <?
        $i++;
    }
        echo "</table>";
    echo "</div>";


    mysql_close(connect_Users());

    include('../scripts/footer.php');
    echo "</div>"
    ?>

<?php
//////// END OF PAGE /////////////////
}
else
{

    include ('../scripts/header.php');
    ?>
    <!--Print error message and offer to log in again-->
    Either you are not allowed to access this page, or your session has expired.
    Please <A href="index.php">log in</a> again.

<?php } ?>
```

menu.php

```
<?php

//Purpose: Acts as a menu bar for Available pages
//Author:  John P. Quan
//Version: 1.0
//Date:    20120105

?>

<?
//STANDARD SESSION LIFE AND INACTIVITY CHECK
session_cache_expire(20);
session_start();

$inactive = 1200;
if (isset($_SESSION['start'])) {
    $session_life = time() - $_SESSION['start'];
    if ($session_life > $inactive) {
        header("Location: user_logout.php");

        $_SESSION['valid_user'] = 'false';
        //CLOSE PREVIOUS SESSION*
        $_SESSION;
        session_destroy();
    }
}
//Set the session start time
$_SESSION['start'] = time();

include_once 'functions_php.php';

//////// START OF PAGE ////////////////

if ($_SESSION['valid_user'] == true
        AND $_SESSION['authorized_user'] == true)
{
?>

        <? //everyone sees these pages
        echo "  ";?>
        <A HREF="index.php"          >Instructions</A> -
        <A HREF="list_sponsor.php" >Sponsor</A> -
        <A HREF="list_services.php" >Sponsored Services</A> -
        <A HREF="choose_service.php">Choose Services</A> -
        <? //only show the administrators these pages
        if($_SESSION['administrator'] > 0)
        {
        ?>
            <? //Only show choose_category for authorized_sponsors
            if($_SESSION['authorized_sponsor'] == "true") { ?>
                <A HREF="choose_category.php"  >Choose Categories</A> -
        <?  }
        }
        //Only the local administrators see these pages
        if ($_SESSION['administrator'] == 1) { ?>
<!--            <A HREF="phpinfo.php"   >phpinfo-FOR TESTING ONLY!</A> - -->
            <A HREF="list_users.php">List Users</A> -
            <A HREF="list_actors.php">List Actors</A> -
            <A HREF="add_user.php"  >Add User</A> -
            <A HREF="add_client.php"  >Add Client</A> -
            <A HREF="add_sponsor.php"  >Add Sponsor</A> -
            <? //Do not show assign_users.php for the text-based browser W3M
```

```
63              // because it does not interpret the page
64              // correctly. Administrators can log in from his or her computer
65              // with a full-featured browser to change these relationships.
66              if(key_array_search("w3m", $_SERVER) == FALSE) { ?>
67                  <A HREF="assign_users.php"  >Assign Users</A> -
68          <? } ?>
69              <A HREF="del_actors.php"  >Delete Actors</A> -
70              <A HREF="del_users.php"  >Delete Users</A> -
71      <? }
72          //Show the Logoff and Delete Services links last
73          ?>
74          <A HREF="del_services.php"  >Delete Services</A> -
75          <A HREF="user_logout.php"   >Log Off</A>
76
77          <BR>
78 <?php
79 ///////// END OF PAGE /////////////////
80 }
81 else
82 {
83
84     include ('../scripts/header.php');
85     ?>
86     <!--Print error message and offer to log in again-->
87     Either you are not allowed to access this page, or your session has expired.
88     Please <A href="index.php">log in</a> again.
89
90 <?php
91 } ?>
```

style.css

```
 1 /*
 2     Document   : style.css
 3     Created on : Jan 8, 2012, 7:17:41 PM
 4     Author     : johnpquan
 5     Description: Provides a style for web pages.
 6 */
 7
 8 * {
 9 margin:0;
10 padding:0;
11 }
12 html, body {
13 height:100%;
14 background:rgb(128, 187, 213);
15 }
```

```
16 h1 {
17 color:rgb(248, 180, 66);
18 font-size:40px;
19 }
20 h2 {
21 text-align:center;
22 color:rgb(248, 180, 66);
23 font-size:40px;
24 padding:-5px 0 0px; /* padding-bottom equals height of #foot */
25 }
26 h3 {
27 color:rgb(248, 180, 66);
28 font-size:40px;
29 }
30 p1 {
31 color:#000000;
32 margin-left:20px;
33 font-family:"Monospace";
34 font-size:20px;
35 }
36 #wrap {
37 min-height:100%;
38 width:100%;
39 margin-top:0 auto;
40 margin-bottom: 0 auto;
41 background:#ddd;
42 border:solid;
43 border-width: 0 0px;
44 }
45 #wrap:before { /* Opera and IE8 "redraw" bug fix */
46 content:"";
47 float:left;
48 height:100%;
49 margin-top:-999em;
50 }
51 * html #wrap { /* IE6 workaround */
52 height:10px;
53 }
54 #foot {
55 text-align:right;
56 font-family:"Times New Roman";
57 font-size:10px;
58 height:15px;
59 width:1000px;
60 margin:-15px auto 0; /* negative margin-top equals height of #foot */
61 }
62
63 root {
64     display: block;
65 }
```

user_logout.php

```php
<?php

//Purpose: Show the user as logged out of the session
//Author:  John P. Quan
//Version: 1.0
//Date:    20120105
//SESSION has ended


include '../scripts/header.php';

echo "User logged out of session.";

session_start();
session_unset();
session_destroy();
session_write_close();
setcookie(session_name(),'',0,'/');
session_regenerate_id(true);

?>
    Please <A href="index.php">log in</a> again.
```

BASH Scripts

sh_check_heartbeats

```
#! /bin/bash

#Download the ORCA Actors Registry
# amdiff is greater than 0 if the actor is NOT donating,
# or if the actor does not exist.

rm actors.jsp

ACTORS=`wget --no-check-certificate $1; egrep '(url\:|amdiff\:)' $2 | cut -d\' -f2; chown workers:workers $2`

echo $ACTORS
```

sh_rsync_command

```
#! /bin/bash
#Use SSH to send rsync commands, such as downloading the service,
# because this is often the only port open for Eucalyptus
# instances (VMs).

OPTS="-q -o PreferredAuthentications=publickey -o HostbasedAuthentication=no -o PasswordAuthentication=no -o StrictHostKeyChecking=no"
USER=$1
IP=$2
PORT=$3
KEY=$4
FILE=$5
LOC=$6

RETURN=`rsync --delete --log-file=../scripts/logs/rsync/$(date +%Y%m%d)_mss_rsync.log -e "ssh ${OPTS} -i ${KEY} -p ${PORT} -l ${USER}" -avzp ${FILE} ${IP}:${LOC}`

echo $RETURN
```

sh_ssh_command

```
#! /bin/bash
#Use SSH to send commands, such as downloading the service,
# because this is often the only port open for Eucalyptus
# instances (VMs).

OPTS="-q -o PreferredAuthentications=publickey -o HostbasedAuthentication=no -o PasswordAuthentication=no -o StrictHostKeyChecking=no"
USER=$1
IP=$2
PORT=$3
KEY=$4
COMMAND=$5

RETURN=`ssh ${OPTS} -i ${KEY} -p ${PORT} ${USER}\@${IP} ${COMMAND}`

echo $RETURN
```

Appendix C

Hypervisor

The following sections provide the files from UAF's remote server, `orca-barrow-0`, with only the uncommented lines displayed because the commented lines have no affect on the system's configuration. In addition, any passwords, encryption keys, or other security information is masked or changed. `orca-barrow-0` is a single server with the Linux distribution Debian 6 and paravirtualized Xen installed, and that resides in a server room in Barrow, AK. The files consider its public IP address to be "`public.IP.address`," and it is private, internal IP address ranges as displayed in Figure 2. ORCA Remote Server.

Networking

/etc/network/interfaces

```
auto lo
iface lo inet loopback

allow-hotplug eth0
iface eth0 inet static
        address 192.168.1.100
        netmask 255.255.255.0
       gateway 192.168.1.1
        network 192.168.1.0
        broadcast 192.168.1.255

allow-hotplug eth1
iface eth1 inet static
       address 192.168.1.10
       netmask 255.255.255.0
       broadcast 192.168.1.255

auto dummy0
iface dummy0 inet manual
        pre-up ifconfig $IFACE up
        post-down ifconfig $IFACE down

auto xenbrdummy0
iface xenbrdummy0 inet static
        address 10.10.10.1
        network 10.10.10.0
        netmask 255.255.255.0
        broadcast 10.10.10.255
        bridge_ports dummy0
        bridge_stp off
        bridge_fd 0
        bridge_maxwait 0
```

Xen

/etc/xen-tools/xen-tools.conf

```
lvm = orca-barrow-0
install-method = debootstrap

size   = 3Gb      # Disk image size.
memory = 512Mb    # Memory size
```

```
swap   = 1Gb    # Swap size
fs     = ext3     # use the EXT3 filesystem for the disk image.
dist   = `xt-guess-suite-and-mirror --suite` # Default distribution to install.
image  = sparse   # Specify sparse vs. full disk images.

 gateway    = public.IP.address
 netmask    = 255.255.255.0
 broadcast  = public.IP.broadcast.address

 passwd = 1

kernel = /boot/vmlinuz-`uname -r`
initrd = /boot/initrd.img-`uname -r`

 arch =amd64

mirror = `xt-guess-suite-and-mirror --mirror`

ext3_options     = noatime,nodiratime,errors=remount-ro
ext2_options     = noatime,nodiratime,errors=remount-ro
xfs_options      = defaults
reiserfs_options = defaults
btrfs_options    = defaults
```

/etc/xen/xend-config.sxp

```
(xend-http-server yes)
(xend-unix-server yes)
(xend-unix-path /var/lib/xend/xend-socket)
(xend-address localhost)

(network-script network-ORCA)
(vif-script vif-ORCA)

(dom0-min-mem 196)
(enable-dom0-ballooning yes)
(total_available_memory 0)
(dom0-cpus 0)
(vncpasswd '')
```

/etc/xen/orca-barrow-0.cfg

```
kernel      = '/boot/vmlinuz-2.6.32-5-xen-amd64'
ramdisk     = '/boot/initrd.img-2.6.32-5-xen-amd64'
vcpus       = '1'
memory      = '512'
root        = '/dev/xvda2 ro'
disk        = [
                  'phy:/dev/orca-barrow-0/dnat-server-disk,xvda2,w',
                  'phy:/dev/orca-barrow-0/dnat-server-swap,xvda1,w',
              ]
name        = 'orca-barrow-0'
vif = [ 'ip=192.168.1.1,   mac=00:16:3E:7B:6E:12, bridge=eth0' ,
'ip=public.IP.address, mac=00:16:3E:7B:6E:11, bridge=eth1']
on_poweroff = 'destroy'
on_reboot   = 'restart'
on_crash    = 'restart'
on_xend_start = 'start'
on_xend_stop = 'shutdown'
```

/etc/xen/scripts/vif-ORCA

```
#! /bin/sh
dir=$(dirname "$0")
```

```
IFNUM=$(echo ${vif} | awk -F. '{ print $2 }')

if [[ "$IFNUM" == "0" ]] ; then
 "$dir/vif-route"  "$@"
else
 "$dir/vif-bridge" "$@"
fi
```

/etc/xen/scripts/network-ORCA

```
#!/bin/sh
/etc/xen/scripts/network-bridge "$@" vifnum=0 netdev=eth0
/etc/xen/scripts/network-bridge "$@" vifnum=1 netdev=eth1
```

ORCA

/opt/Camano-3.1/config/ec2.site.properties

```
ec2.img.proxy.use=true
ec2.img.proxy.url=http://192.168.1.100/axis2/services/IMAGEPROXY
ec2.img.proxy.timeout=3600


ec2.ami.name=emi-499D16C3

ec2.aki.name=eki-A3B01BDD
ec2.ari.name=eri-991917F8

ec2.instance.type=m1.small

ec2.ssh.key=barrowkey

ec2.use.public.addressing=true

ec2.connection.timeout=60
ec2.request.timeout=120

ec2.ping.retries=60

ec2.ssh.retries=10

ec2.startup.retries=5

ec2.use.proxy=true

proxy.type=SHOREWALL-DNAT
proxy.proxy.ip=192.168.1.1
proxy.user=orca
proxy.ssh.key=/opt/Camano-3.1/config/orca-proxy-ssh-key
proxy.script.path=/opt/iptables-scripts
```

Initialization Scripts

/etc/init.d/start_ORCA.sh

```
case "$1" in
  start)
       echo -n "Starting orca-barrow-0 in 2 minutes"
       #To run it as root:
       sleep 10
       xm destroy orca-barrow-0
       sleep 4
       xm create /etc/xen/orca-barrow-0.cfg
```

```
        sleep 10
        /opt/eucalyptus-2.0/etc/init.d/eucalyptus-cc start
        /opt/eucalyptus-2.0/etc/init.d/eucalyptus-nc start
        /etc/init.d/eucalyptus-cloud start
        cd /opt/Camano-3.1/tomcat-7; ./start.sh
        cd /opt/imageproxy/axis2-1.5.4/bin; nohup ./axis2server.sh &
        cd /opt/imageproxy/bin; nohup ./seeding /opt/imageproxy
/opt/imageproxy/axis2-1.5.4/imageproxy.db > /opt/imageproxy/logs/seeding.log &
        echo "."
        ;;
  stop)
        echo -n "Stopping orca-barrow-0"
        #To run it as root:
        cd /opt/Camano-3.1/tomcat-7; ./stop.sh
        sleep 4
        /etc/init.d/eucalyptus-cloud stop
        sleep 6
        /opt/eucalyptus-2.0/etc/init.d/eucalyptus-cc stop
        sleep 3
        /opt/eucalyptus-2.0/etc/init.d/eucalyptus-nc stop
        sleep 10
        xm destroy orca-barrow-0
        echo "."
        ;;

  *)
        echo "Usage: /sbin/service start_ORCA.sh {start|stop}"
        exit 1
esac

exit 0
```

Virtual Router

The following sections provide the files from UAF's remote server, `orca-barrow-0`, with only the uncommented lines displayed because the commented lines have no affect on the system's configuration. In addition, any passwords, encryption keys, or other security information is masked or changed. `orca-barrow-0` is a single server that resides in a server room in Barrow, AK. The files consider its public IP address to be "`public.IP.address`," and it is private, internal IP address ranges as displayed in Figure 2. ORCA Remote Server.

ORCA

A single-server remote server installation is very similar to a canonical ORCA cluster behind a DNAT Server [1], except that it does not use Shorewall to control iptables on the DNAT Server. Instead, the remote server uses modified RENCI scripts to control iptables on the virtual DNAT Server directly. Thank you to Dr. Brian Hay, University of Alaska Fairbanks, for modifying the /opt/iptables-scripts files.

/opt/iptables-scripts/README.txt

```
These scripts were created to circumvent using Shorewall DNAT server, as RENCI
prescribes.  In order to "fit" an entire ORCA installation on one server, UAF
ORCA architecture uses a Virtual Machine named "dnat-server" as a router, which
is turned on in the host server's startup sequence.  This set up seems to
create problems for Shorewall, and so we created this work-around by modifying
/opt/shorewall/dnat-scripts created by RENCI.  Instead of using shorewall to
modify the dnat-server iptables, these scripts do so directly.  Like Shorewall,
iptables-scripts write a rule into iptables and then restarts iptables.
```

```
Unlike Shorewall, iptables is first populated with DNAT and SNAT rules so that
Dom0, which contains ORCA, Eucalyptus, and Image Proxy, can communicate with
the dnat-server DomU.  The virtual dnat-server then routes these communications
to the Internet.  Please refer to /etc/iptables.rules on the dnat-server for an
example.
```

/opt/iptables-scripts/execCmd.sh

```
set -e

SCRIPT=`readlink -f $0`
INSTALLPATH=`dirname $SCRIPT`

. $INSTALLPATH/lib/helpers.sh
. $INSTALLPATH/lib/iptables-helpers.sh

if [ "$1" = "" ] || [ "$2" = "" ] || [ "$3" = "" ]; then
       echo "STATUS=ERROR; MSG=\"Expecting [ADD|DEL] ip port\" "
       exit 1
fi

CMDCONT="$1 $2 $3"
RES=$(cmdHelper $CMDCONT) || {
       echo "STATUS=ERROR; $RES"
```

/opt/iptables-scripts/lib/iptables-helpers.sh

```
DNAT_FIRST_PORT=6001
DNAT_LAST_PORT=6999
PUBIP="public.IP.address"
SHOREWALL=`which shorewall-wrapper`
DNAT_CHAIN="PREROUTING"

RULESFILE="/tmp/rules"

function cmdHelper ()
{
        CMD=$1
        HOST=$2
        HOSTPORT=$3
        RES="NOTSET"

        if [ "$CMD" = "ADD" ]; then
                RES=$(addDNATRule $HOST $HOSTPORT) || {
                        echo $RES
                        return 1
                }
                echo $RES
                return 0
        fi
        if [ "$CMD" = "DEL" ]; then
                RES=$(delDNATRule $HOST $HOSTPORT) || {
                        echo $RES
                        return 1
                }
                echo $RES
                return 0
        fi
        echo "Invalid syntax"
        return 1
}

function getPorts ()
{
```

```
        if [ "$1" == "" ]; then
                return 1
        fi
        PORTS=`sudo /sbin/iptables -t nat -n -L $1 2>/dev/null| gawk 'BEGIN { FS
= "[ \t:]+" }
$7 ~ /dpt/ {print $8}'` || {
                return 1
        }
        echo $PORTS
        return 0
}

function findFreeDnatPort ()
{
        USEDPORTS=$(getPorts $DNAT_CHAIN) || {
                return 1
        }
        FREEPORT=`echo "" | gawk -v used="$USEDPORTS" -v first="$DNAT_FIRST_PORT"
-v last="$DNAT_LAST_PORT" 'BEGIN { RS=" "; }
{
        n=split(used, arr, " ")
        asort(arr)
        port=first
         found=-1
         i=1
         while(found<first && port<=last && i<=n) {
             if (arr[i]<port) {
                 i++
             } else if (port==arr[i]) {
                 i++
                 port++
             } else {
                 found=port
             }
         }
         print found
}'`
        test $FREEPORT -eq -1 && {
              echo ""
              return 1
        }
        echo $FREEPORT
        return 0
}

function checkIP ()
{
        RES=`echo $1 | gawk 'BEGIN { RS=" " }
$0 ~ /[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+/ { print $0 }'`
        if [ "$RES" = "" ]; then
                return 1
        fi
        echo $RES
        return 0
}

function checkPort ()
{
        RES=`echo "" | gawk -v port=$1 'BEGIN { RS=" " }
{
        if (port>0 && port < 65535)
                print port
}'`
```

```
        if [ "$RES" = "" ]; then
                return 1
        fi
        echo $RES
        return 0
}

function checkDNATRule ()
{
        #FN=$RULESFILE
        HOST=$1
        R=$(checkIP $HOST) || {
                echo "Invalid host $HOST"
                return 2
        }
        HOSTPORT=$2
        R=$(checkPort $HOSTPORT) || {
                echo "Invalid port $HOSTPORT"
                return 2
        }
        #PAT="DNAT\t\tnet\t\t$DNAT_CHAIN:$HOST:$HOSTPORT"
        #RES=`gawk -v pat="$PAT" '$0 ~ pat { print $5}' $FN`
         PAT="DNAT.*dpt:[0-9]* to:$HOST:$HOSTPORT "
         RES=`sudo /sbin/iptables -t nat -L $DNAT_CHAIN -n | gawk -v pat="$PAT"
'$0 ~ pat { print $7 }' | gawk -F ":" '{ print $2 }' `

        if [ "$RES" = "" ]; then
                return 0
        else
                echo $RES
                return 1
        fi
}

function addDNATRule ()
{
        FN=$RULESFILE
        HOST=$1
        R=$(checkIP $HOST) || {
                echo "MSG=\"Invalid host $HOST\""
                return 2
        }
        HOSTPORT=$2
        R=$(checkPort $HOSTPORT) || {
                echo "Invalid port $HOSTPORT"
                return 2
        }
        lock_file $FN
        # check if rule exists
        RES=$(checkDNATRule $HOST $HOSTPORT) || {
                echo "MSG=\"RULE EXISTS\"; HOST=$HOST; PORT=$HOSTPORT;
PUBIP=$PUBIP; FWDPORT=$RES"
                unlock_file $FN
                return 0
        }
        # find free port
        LOCPORT=$(findFreeDnatPort) || {
                echo "MSG=\"Unable to find free port\""
                unlock_file $FN
                return 1
        }
        #echo "DNAT         net           $DNAT_CHAIN:$HOST:$HOSTPORT      tcp
        $LOCPORT" >> $FN
```

```
        sudo /sbin/iptables -t nat --append $DNAT_CHAIN --destination $PUBIP --
proto tcp --dport $LOCPORT -j DNAT --to-destination $HOST:$HOSTPORT
       unlock_file $FN
       echo "MSG=\"ADDED\"; HOST=$HOST; PORT=$HOSTPORT; PUBIP=$PUBIP;
FWDPORT=$LOCPORT"
       return 0
}
function delDNATRule ()
{
       FN=$RULESFILE
       HOST=$1
       R=$(checkIP $HOST) || {
              echo "MSG=\"Invalid host $HOST\""
              return 2
       }
       HOSTPORT=$2
       R=$(checkPort $HOSTPORT) || {
              echo "MSG=\"Invalid port $HOSTPORT\""
              return 2
       }

        lock_file $FN
        RULES_TO_DELETE=`sudo /sbin/iptables -t nat -L $DNAT_CHAIN -n | grep -v
"^Chain" | grep -v "^target" | grep -n "DNAT.*$PUBIP.*dpt:[0-9]*
to:$HOST:$HOSTPORT " | gawk -F ":" '{ print $1 }' | sort -r`
        #echo $RULES_TO_DELETE
        if [ "$RULES_TO_DELETE" = "" ]; then
           echo "MSG=\"NO RULE TO DELETE\"; HOST=$HOST; PORT=$HOSTPORT"
       else
            for r in $RULES_TO_DELETE; do
                sudo /sbin/iptables -t nat --delete $DNAT_CHAIN $r
                echo "MSG=\"DELETED\"; HOST=$HOST; PORT=$HOSTPORT"
            done
        fi
        unlock_file $FN
       #PAT="DNAT\t\tnet\t\t$DNAT_CHAIN:$HOST:$HOSTPORT"
       #lock_file $FN
       #FNTMP=/tmp/`basename $FN`.new
       #cat $FN | sed '/^'"$PAT"'/ d' > $FNTMP
       #cp $FNTMP $FN
       #rm -f $FNTMPS
       #unlock_file $FN
       #echo "MSG=\"DELETED\"; HOST=$HOST; PORT=$HOSTPORT"
       return 0
}

function kickShorewall ()
{
       #$SHOREWALL restart > /dev/null 2>&1 || {
       # note that refresh does not invalidate existing connections,
       # so even the deleted connection stays on
       #      return 1
       #}
       return 0
}
```

Networking

/etc/network/interfaces

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
       address 192.168.1.1
       broadcast 192.168.1.255
       netmask 255.255.255.0

auto eth1
iface eth1 inet static
       # networking for Barrow, AK
       address public.IP.address
       netmask public.IP.netmask
       broadcast public.IP.broadcast.address
       gateway public.IP.gateway
```

/etc/iptables.rules

```
*nat
:PREROUTING ACCEPT [5:440]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A PREROUTING -d public.IP.address/32 -p tcp -m tcp --dport 12080 -j DNAT --to-
destination 192.168.1.100:12080
-A PREROUTING -d public.IP.address/32 -p tcp -m tcp --dport 11080 -j DNAT --to-
destination 192.168.1.100:11080
-A PREROUTING -d public.IP.address/32 -p tcp -m tcp --dport 11443 -j DNAT --to-
destination 192.168.1.100:11443
-A PREROUTING -d public.IP.address/32 -p tcp -m tcp --dport 11081 -j DNAT --to-
destination 192.168.1.100:11081
-A PREROUTING -d public.IP.address/32 -p tcp -m tcp --dport 8443 -j DNAT --to-
destination 192.168.1.100:8443
-A PREROUTING -d public.IP.address/32 -p tcp -m tcp --dport 12443 -j DNAT --to-
destination 192.168.1.100:12443
-A PREROUTING -d public.IP.address/32 -p tcp -m tcp --dport 222 -j DNAT --to-
destination 192.168.1.100:22
-A POSTROUTING -s 192.168.1.100/32 -j SNAT --to-source public.IP.address
-A POSTROUTING -s 192.168.1.101/32 -j SNAT --to-source public.IP.address
-A POSTROUTING -s 192.168.1.102/32 -j SNAT --to-source public.IP.address
-A POSTROUTING -s 192.168.1.103/32 -j SNAT --to-source public.IP.address
-A POSTROUTING -s 192.168.1.104/32 -j SNAT --to-source public.IP.address
COMMIT
*filter
:INPUT ACCEPT [3083:279267]
:FORWARD ACCEPT [96899:11149697]
:OUTPUT ACCEPT [18425:2683358]
-A INPUT -m state --state ESTABLISHED -j ACCEPT
-A INPUT -d public.IP.address/32 -p tcp -m tcp --dport 22 -j ACCEPT
COMMIT
```

Initialization scripts

/etc/init.d/start_iptables.sh

```
case "$1" in
  start)
        echo -n "Starting iptables"
        sleep 10
       iptables-restore /etc/iptables.rules
        echo "."
```

```
        ;;
  stop)
        echo -n "Stopping dnat-server"
        /sbin/iptables --flush
        /sbin/iptables --delete-chain
        /sbin/iptables -t nat --flush
        /sbin/iptables -t nat --delete-chain
        echo "."
        ;;

  *)
        echo "Usage: /sbin/service start_iptables.sh {start|stop}"
        exit 1
esac

exit 0
```

/etc/init.d/send_mailPowerloss.sh

```
case "$1" in
  start)
       echo -n "Mailing Administrator about power interruption"
       sendmail -v jquan2@alaska.edu < /etc/mail/ORCA_RESTARTED_MSG.txt

       echo "."
       ;;
  stop)
       echo -n "Stopping send_mailPowerLoss.sh"
       /etc/init.d/sendmail stop
        echo "."
        ;;

  *)
        echo "Usage: /sbin/service send_mailPowerLoss.sh {start|stop}"
        exit 1
esac

exit 0
```

Volume IV

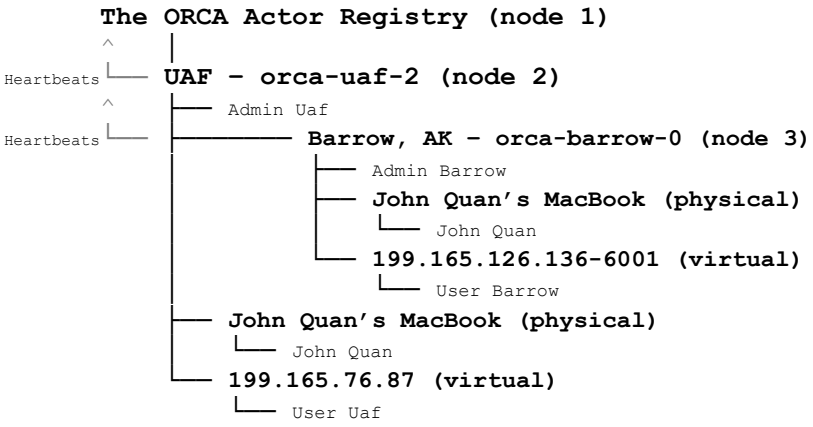
Software Testing for Mutualistic Software Services (MSS) Version 1.0

1. Introduction

This volume tests MSS according to the requirements in *Software Architecture for Mutualistic Software Services (MSS) Version 1.0,* which were synthesized from the *Software Requirements Specification for Mutualistic Software Services (MSS) Version 1.0.* The tests compare the intended actions listed in 5.3 Component Model of the Software Architecture document with the actual outcome when using MSS. Section 3. Tested Components below presents screenshots where appropriate as an example of the test case.

2. Test Criteria

Testing will use the relationship tree provided in the software architecture's section 5. Architectural Plan as a testing template. Here is the testing relationship tree:



In addition, testing must evaluate the following basic criteria:

- Work on multiple browsers, to include the text-based browser w3m
- Work on a canonical ORCA cluster
- Work on a Remote Server
- Support virtual resources
- Support physical resources
- Check for heartbeats at the MSS-ORIGIN
- Only allow MSS access to authorized users on authorized computers

The *Software Architecture for Mutualistic Software Services (MSS) Version 1.0*, Appendix B contains the component diagram. The component diagram CM-1 represents the external connections, and the data flow model DM-1 represents the generation and flow of data for each MSS-ENTITY.

3. Tested Components

3.1 Databases

3.1.1 Service Data

- Holds the subset of service attribute data of its MSS-ENTITY
    - See the Services Database in the *Software Design for Mutualistic Software Services (MSS) Version 1.0*.

3.1.2 CF Data

- Holds the MSS-ENTITY data for itself. For instance, ORCA uses database *orca*

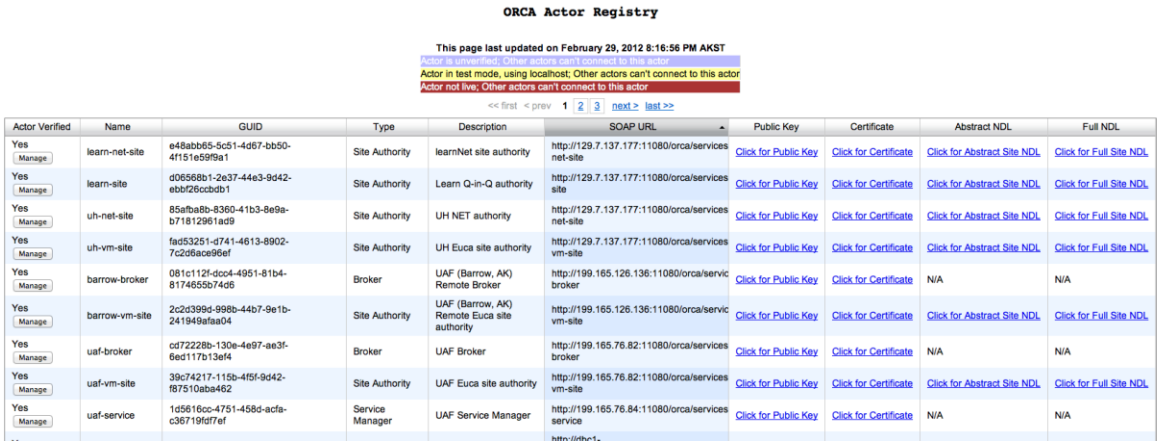
3.1.3 User Data

- Holds the authorized user's data for itself and its child MSS-AFFILIATES (actors)
    - See the Users Database in the *Software Design for Mutualistic Software Services (MSS) Version 1.0*.

3.2 Components

3.2.1 CF

- MSS-ORIGIN presents an interface to its children to receive heartbeats



ORCA Actor Registry

This page last updated on February 29, 2012 8:16:56 PM AKST

Actor in test mode, using localhost; Other actors can't connect to this actor

Actor not live; Other actors can't connect to this actor

<< first < prev 1 2 3 next > last >>

Actor Verified	Name	GUID	Type	Description	SOAP URL	Public Key	Certificate	Abstract NDL	Full NDL
Yes Manage	learn-net-site	e48abb65-5c51-4d67-bb50-4f151e59f9a1	Site Authority	learnNet site authority	http://129.7.137.177:11080/orca/services net-site	Click for Public Key	Click for Certificate	Click for Abstract Site NDL	Click for Full Site NDL
Yes Manage	learn-site	d06568b1-2e37-44e3-9d42-ebbf26ccbdb1	Site Authority	Learn Q-in-Q authority	http://129.7.137.177:11080/orca/services site	Click for Public Key	Click for Certificate	Click for Abstract Site NDL	Click for Full Site NDL
Yes Manage	uh-net-site	85afba8b-8360-41b3-8e9a-b71812961ad9	Site Authority	UH NET authority	http://129.7.137.177:11080/orca/services net-site	Click for Public Key	Click for Certificate	Click for Abstract Site NDL	Click for Full Site NDL
Yes Manage	uh-vm-site	fad53251-d741-4613-8902-7c2d6ace96ef	Site Authority	UH Euca site authority	http://129.7.137.177:11080/orca/services vm-site	Click for Public Key	Click for Certificate	Click for Abstract Site NDL	Click for Full Site NDL
Yes Manage	barrow-broker	081c112f-dcc4-4951-81b4-8174655b74d6	Broker	UAF (Barrow, AK) Remote Broker	http://199.165.126.136:11080/orca/servic broker	Click for Public Key	Click for Certificate	N/A	N/A
Yes Manage	barrow-vm-site	2c2d399d-998b-44b7-9e1b-241949afaa04	Site Authority	UAF (Barrow, AK) Remote Euca site authority	http://199.165.126.136:11080/orca/servic vm-site	Click for Public Key	Click for Certificate	Click for Abstract Site NDL	Click for Full Site NDL
Yes Manage	uaf-broker	cd72228b-130e-4e97-ae3f-6ed117b13ef4	Broker	UAF Broker	http://199.165.76.82:11080/orca/services broker	Click for Public Key	Click for Certificate	N/A	N/A
Yes Manage	uaf-vm-site	39c74217-115b-4f5f-9d42-f87510aba462	Site Authority	UAF Euca site authority	http://199.165.76.82:11080/orca/services vm-site	Click for Public Key	Click for Certificate	Click for Abstract Site NDL	Click for Full Site NDL
Yes Manage	uaf-service	1d5616cc-4751-458d-acfa-c36719fdf7ef	Service Manager	UAF Service Manager	http://199.165.76.84:11080/orca/services service	Click for Public Key	Click for Certificate	N/A	N/A

Figure 3. The ORCA Actor Registry

- MSS-AFFILIATES deliver heartbeats to its MSS-ORIGIN
    - See Figure 3. The ORCA Actor Registry.
- Provides a CF interface with a list of current donors. For instance, ORCA provides the *ORCA Actor Registry* [18]
    - See Figure 3. The ORCA Actor Registry.
- Provides a list of available resources to GENI experimenters and users
    - See Figure 3. The ORCA Actor Registry.
- Enables GENI experimenters and users to request resources

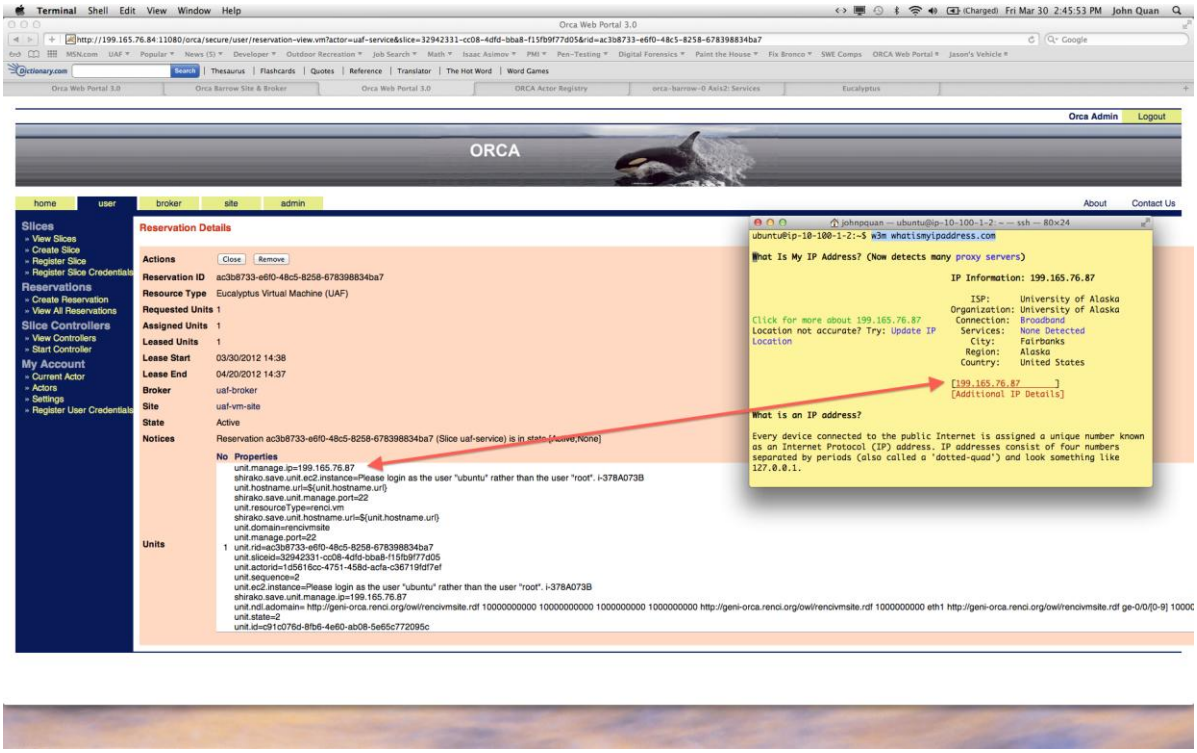


Figure 4. MSS does not interfere with CF functionality.

3.2.2 User Interface

- Uses the Authentication process to verify actor and user requests

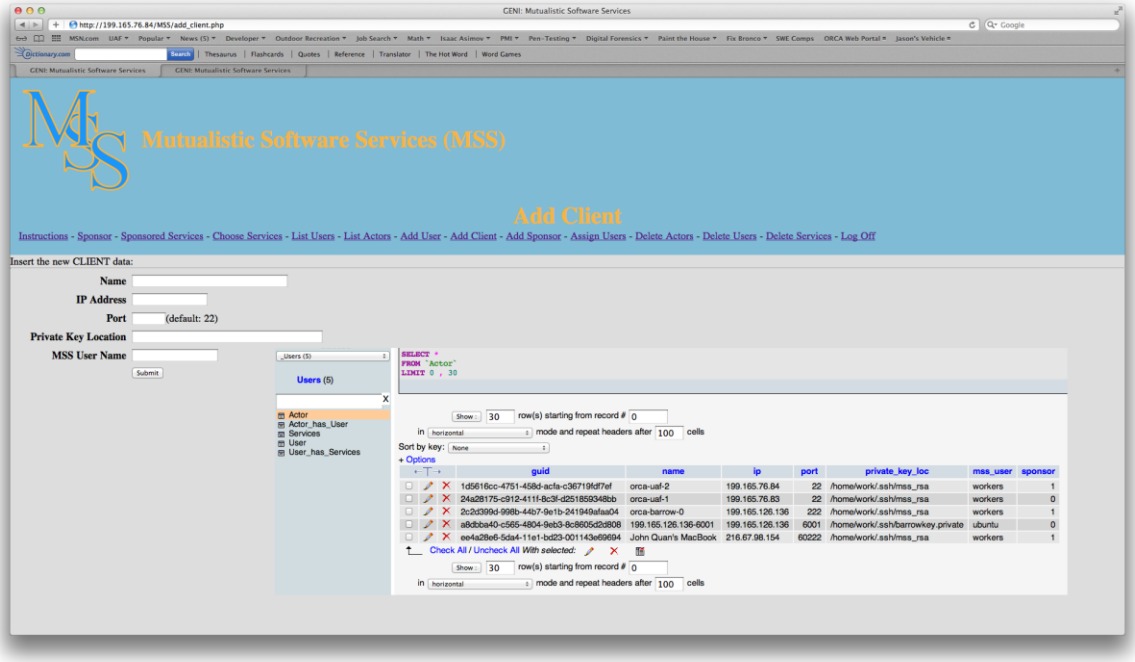


Figure 5. The administrator must authorize clients

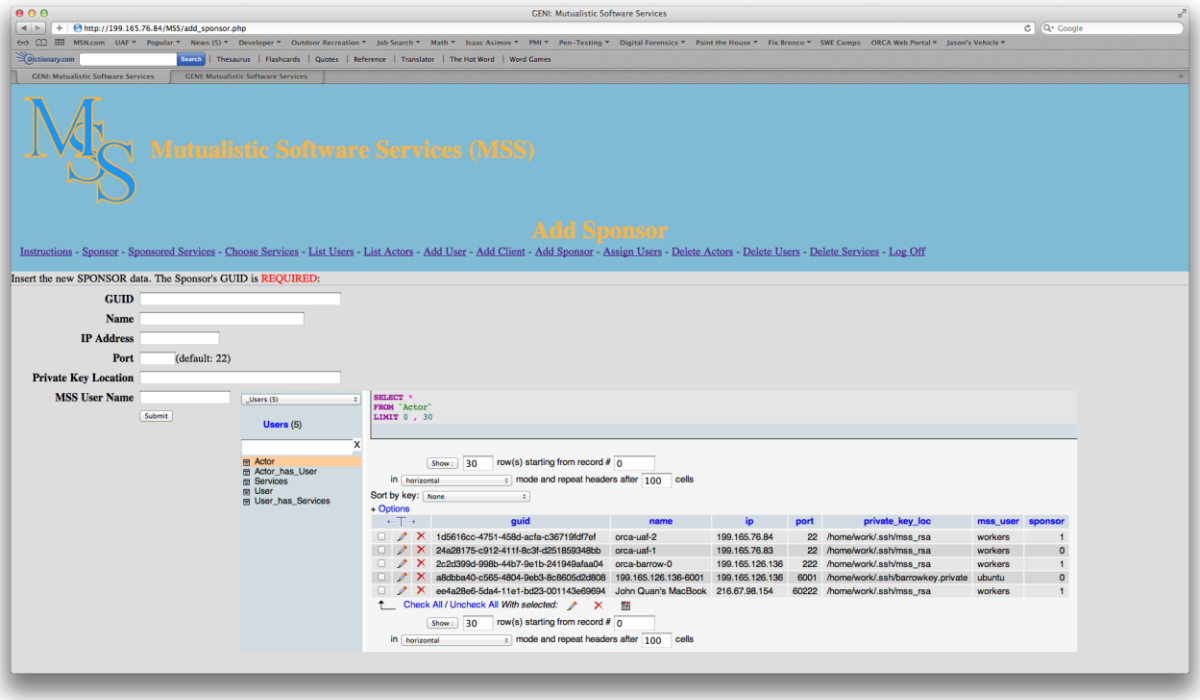


Figure 6. The administrator must specify a sponsor

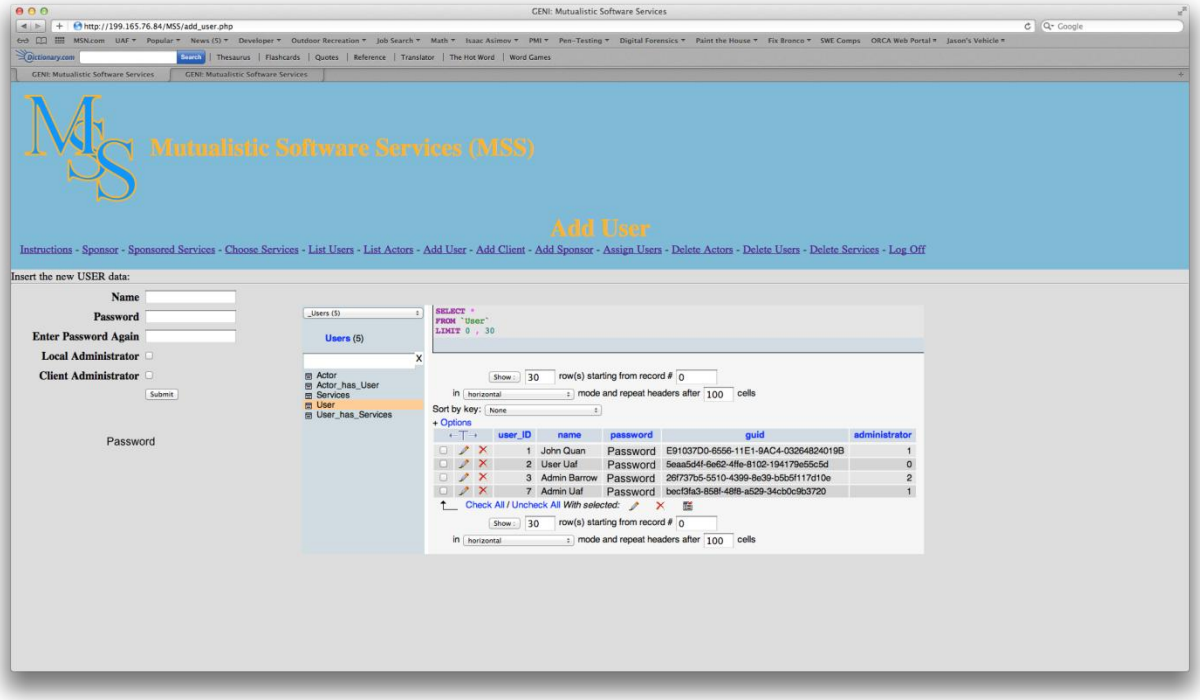


Figure 7. The administrator must authorize users.

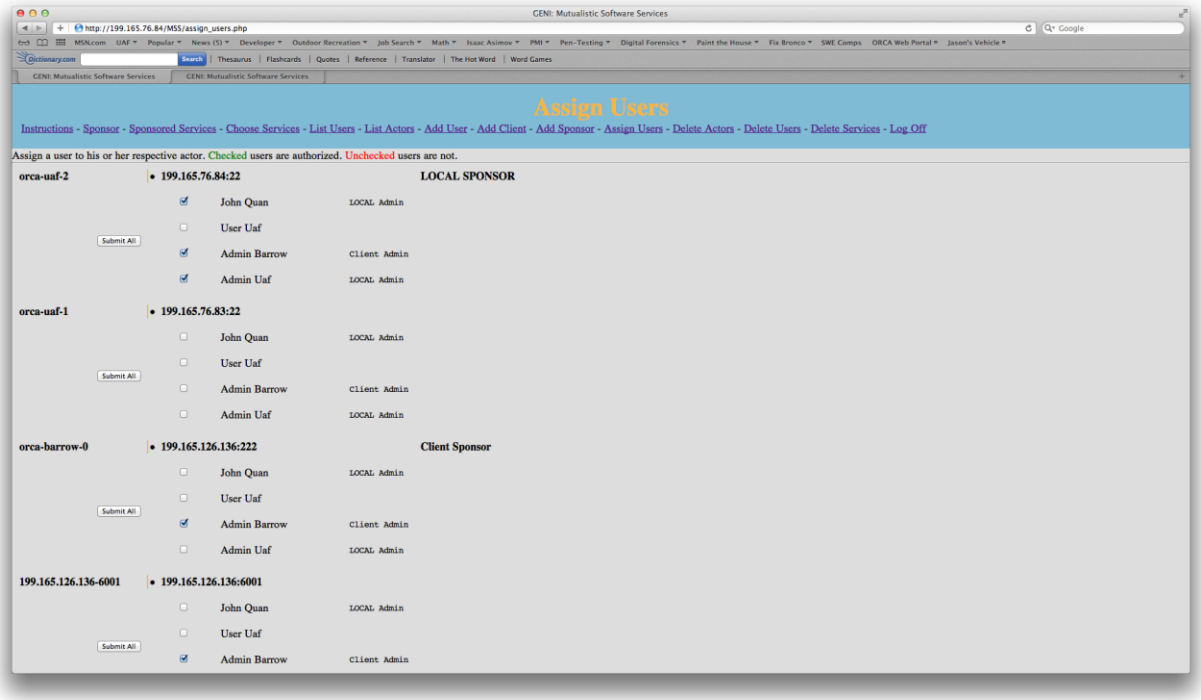


Figure 8. The administrator assigns users to actors.

- Assign Users does not display for text-based web browsers, but it does for graphical web browsers. This is only a partial fulfillment of the requirement, but it does not seem to detract from the functionality of MSS. MSS incorporates text-based web browsing functionality for Eucalyptus Instance users and not administrators.

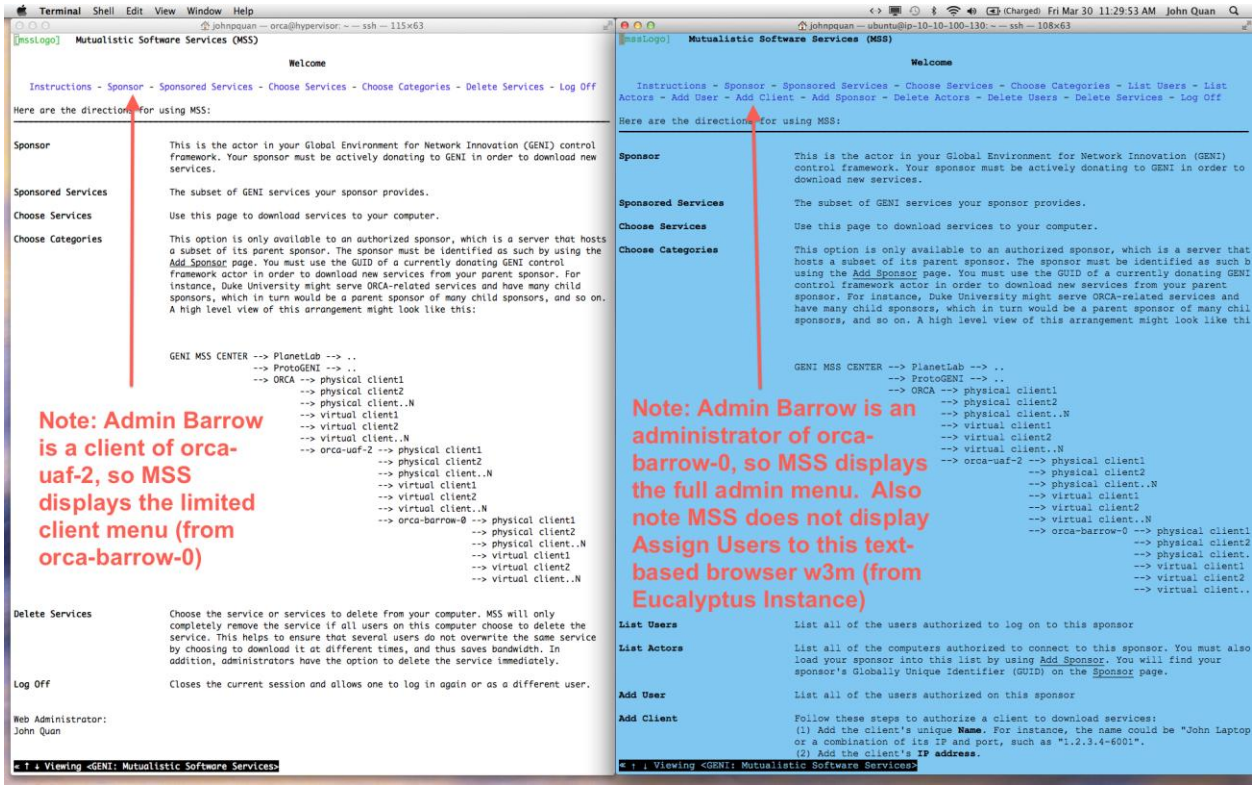


Figure 9. "Assign Users" does not display for w3m.

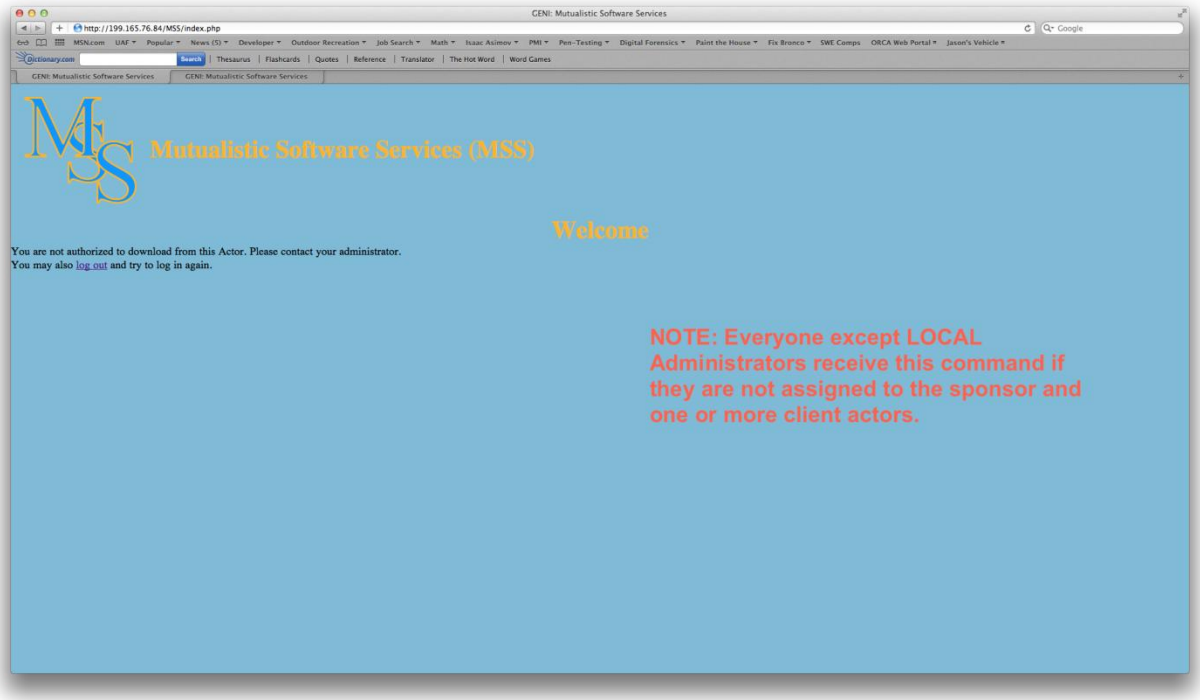


Figure 10. MSS only allows authorized users on authorized computers.

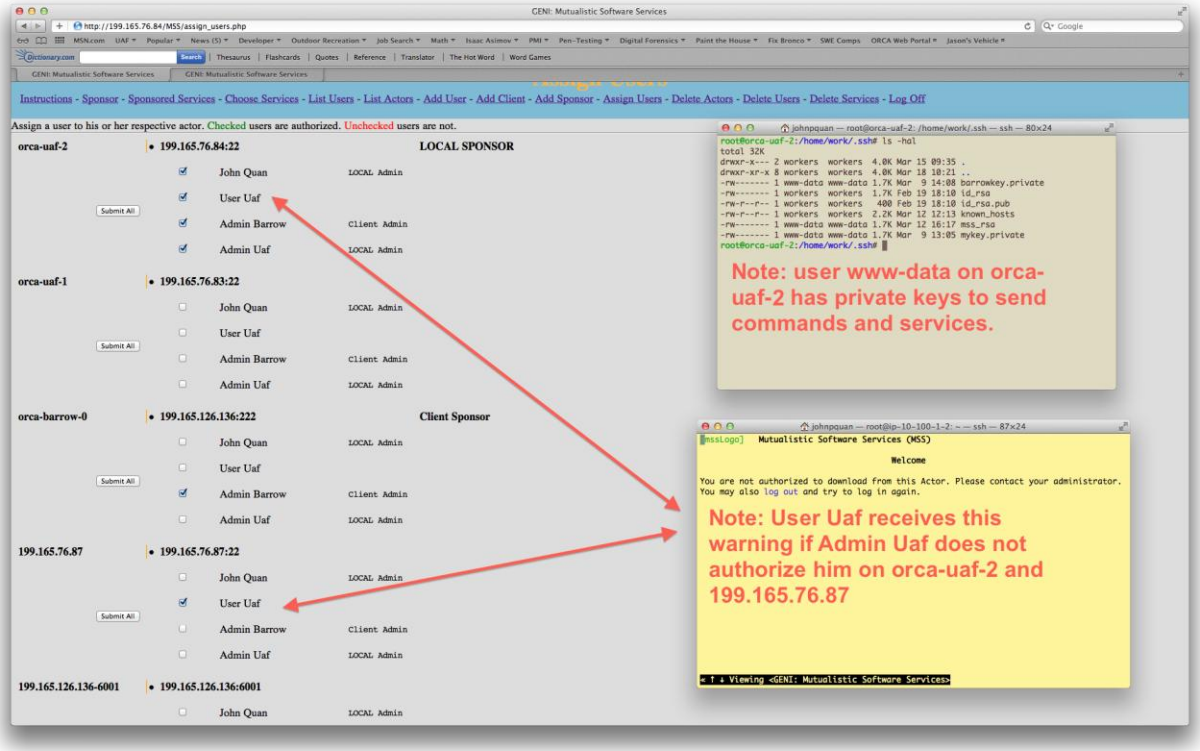


Figure 11. w3m version of "not authorized" response.

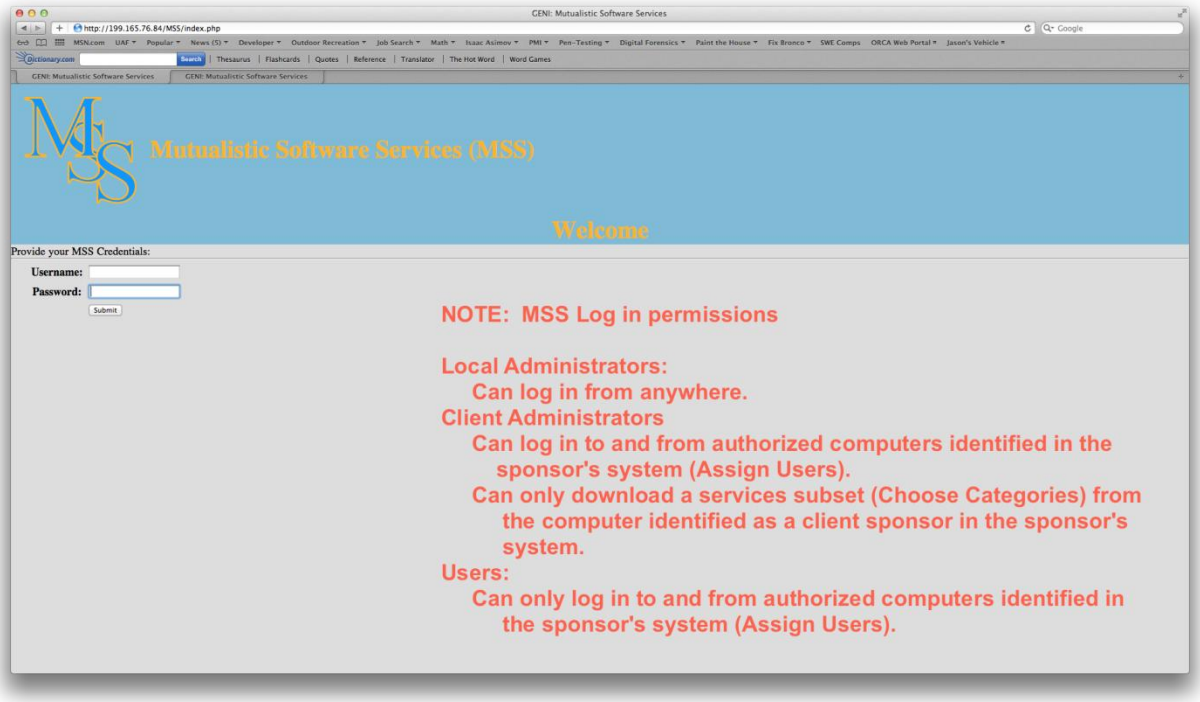


Figure 12. MSS only accepts authorized users.

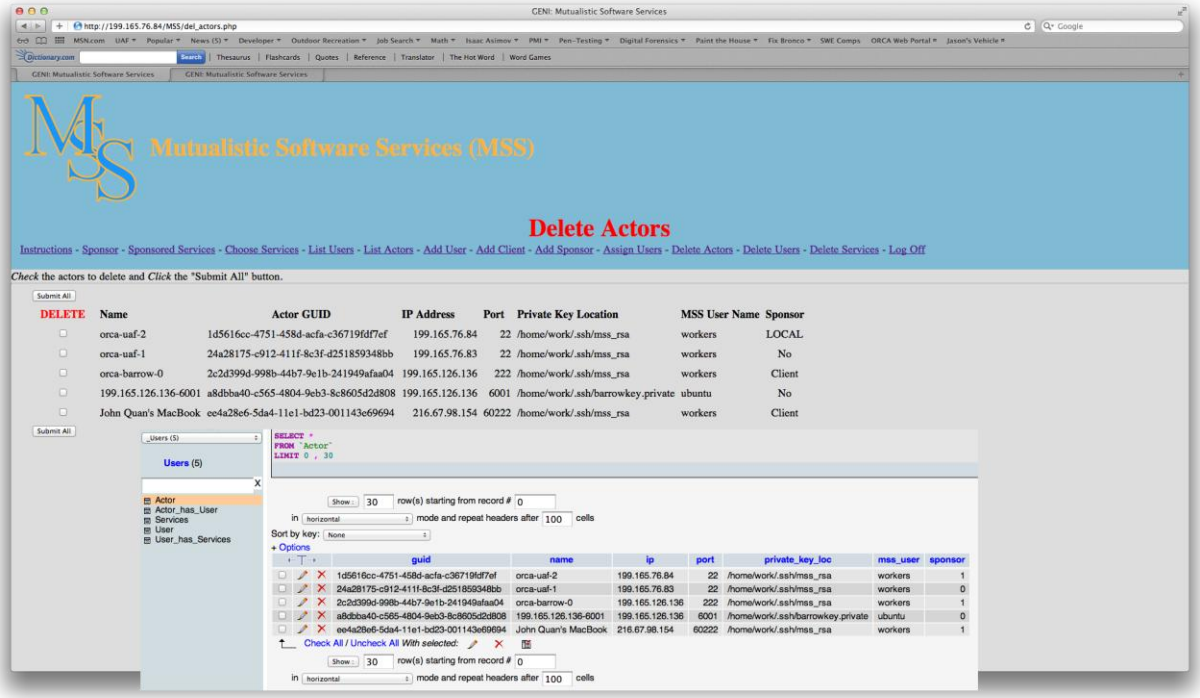


Figure 13. The administrator can remove actor authorization.

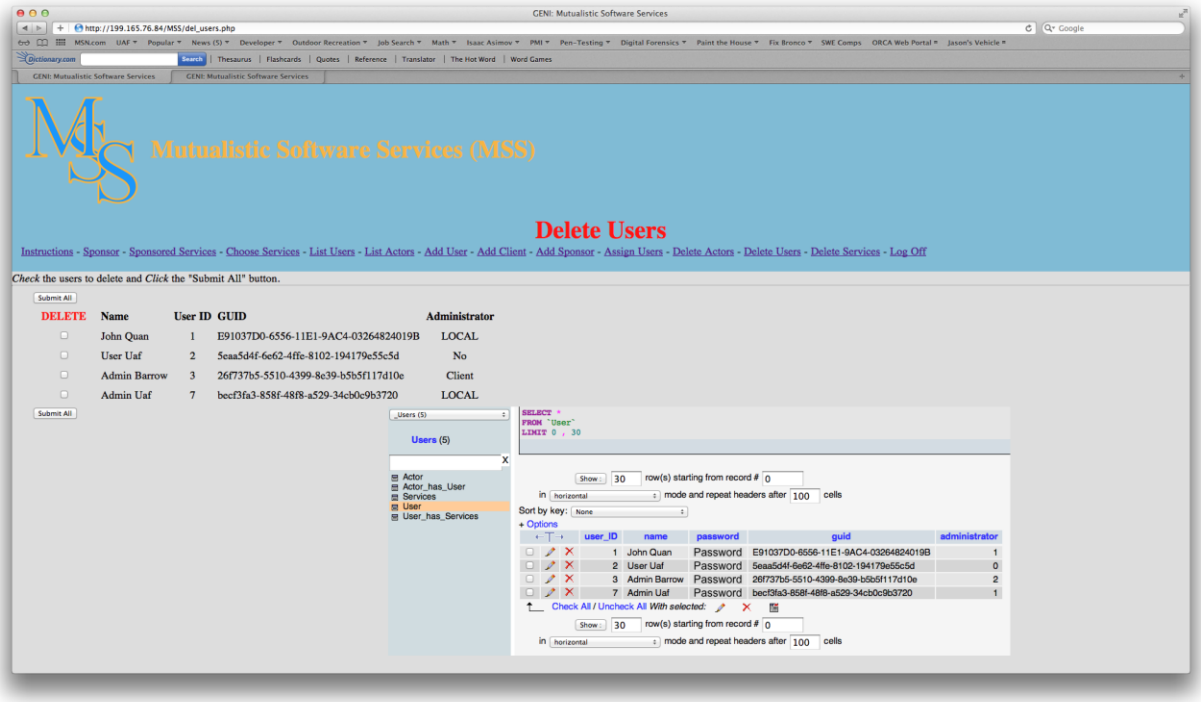


Figure 14. The administrator can remove user authorization.

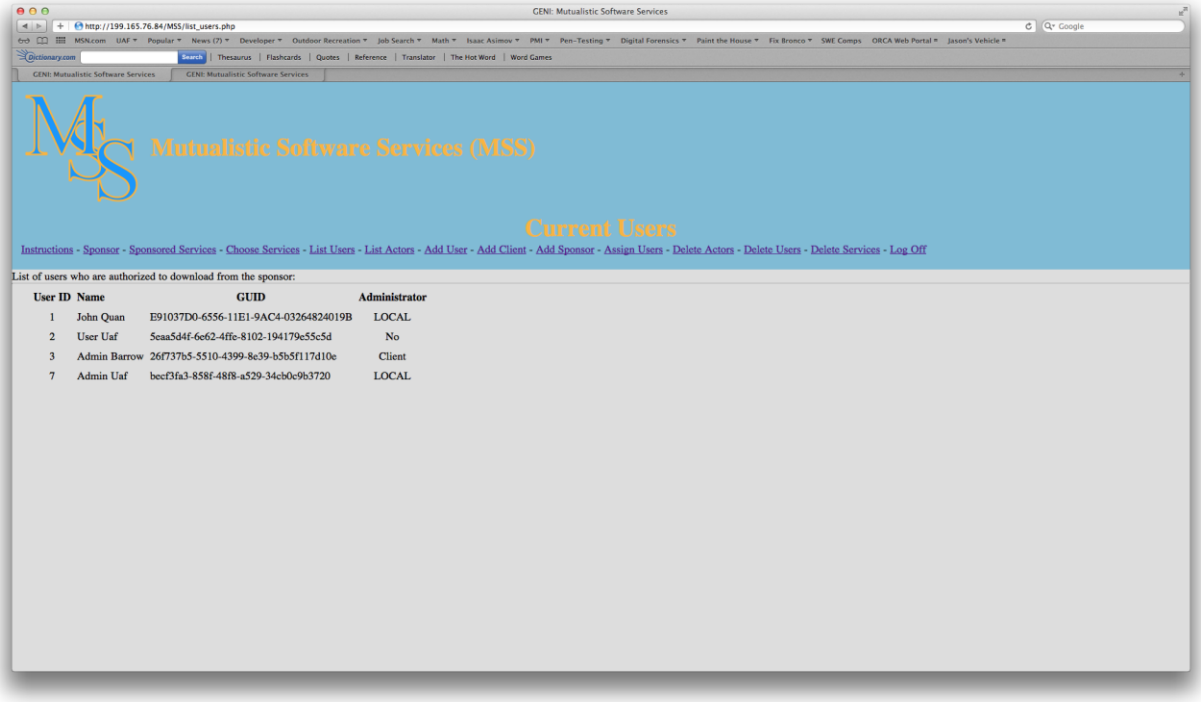


Figure 15. MSS displays authorized users to the administrator.

- Receives a list of MSS-ENTITY resources from the CF

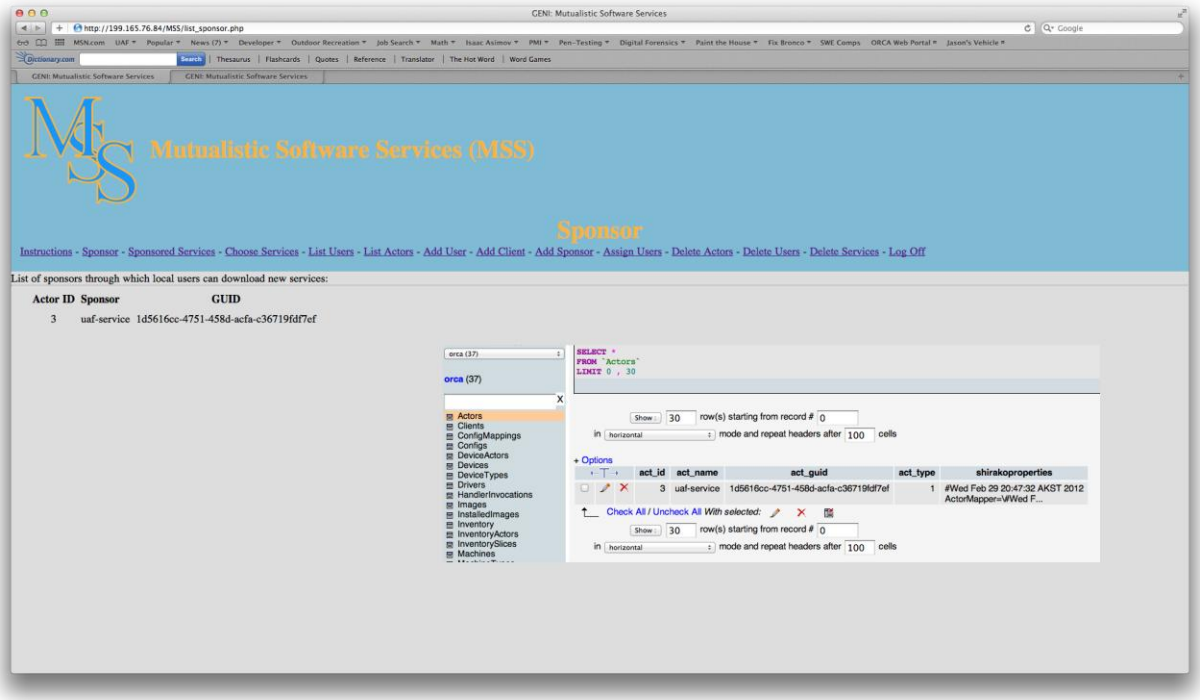


Figure 16. MSS provides ORCA actor information.

- Receives user service information and service requests

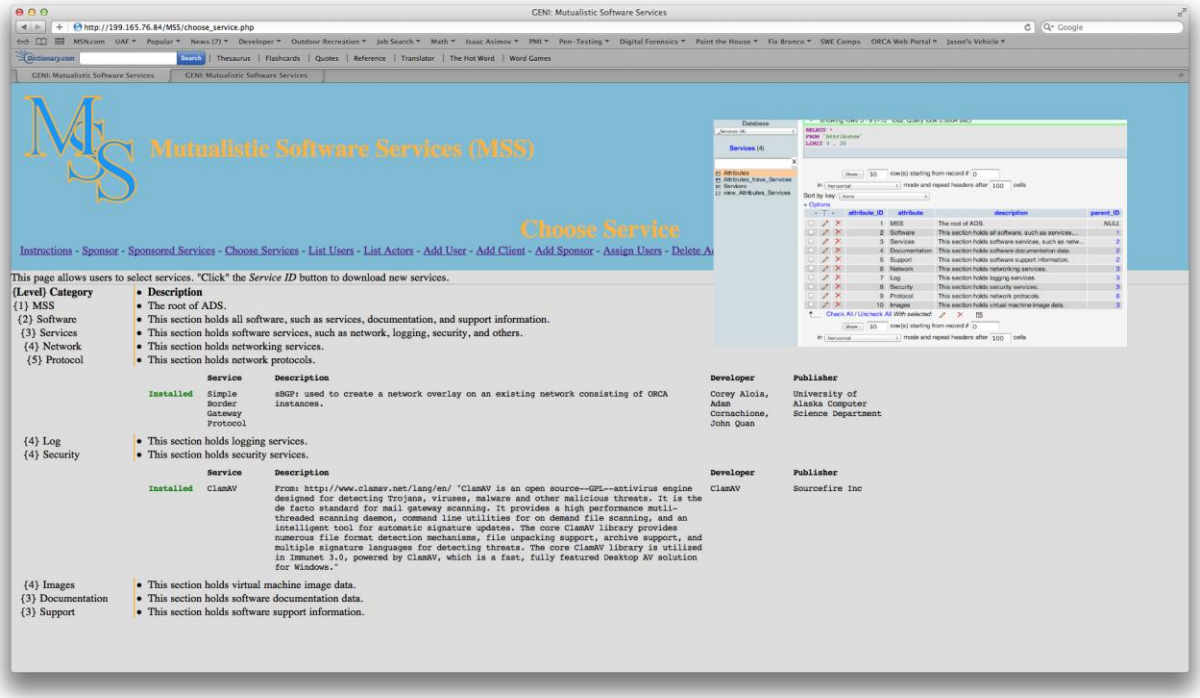


Figure 17. User service information shows available vs installed services.

Figure 18. MSS incorporates user session expiration.

- Provides service delivery information to the Service Interface

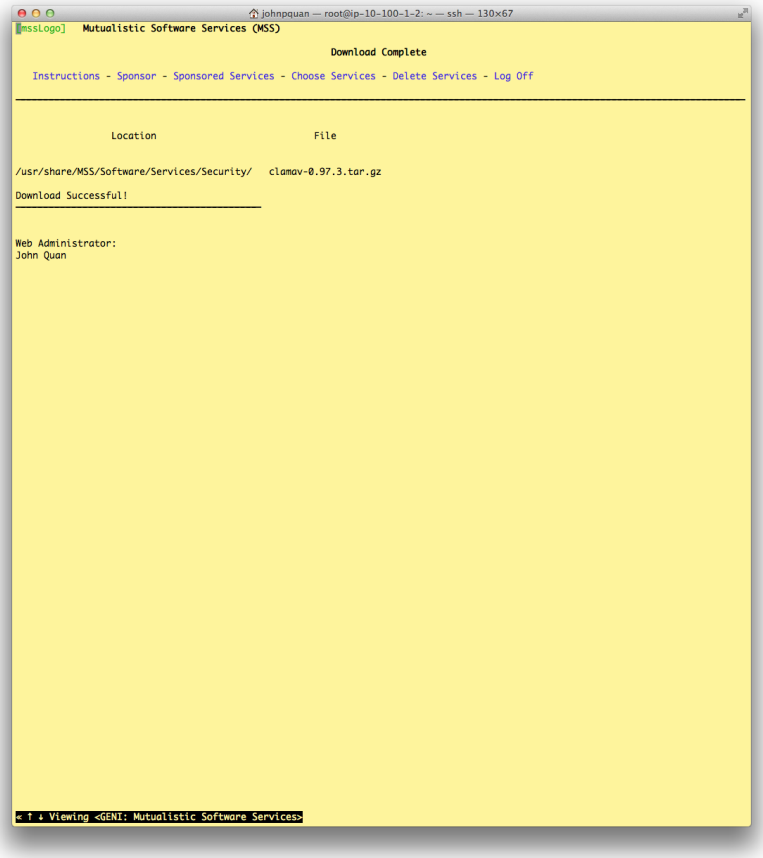


Figure 19. The Download Service page provides success/failure information.

- Triggers the Service Interface to deliver services
  - See Figure 21. User can request to download service ID 1
- Checks for heartbeats on the CF interface
  - See the sh_check_heartbeats command in the *Software Design for Mutualistic Software Services (MSS) Version 1.0.*
- Triggers the Service Interface to download services.
  - See Figure 21. User can request to download service ID 1

3.2.3 Service Interface

- Receives a service information request from the User Interface

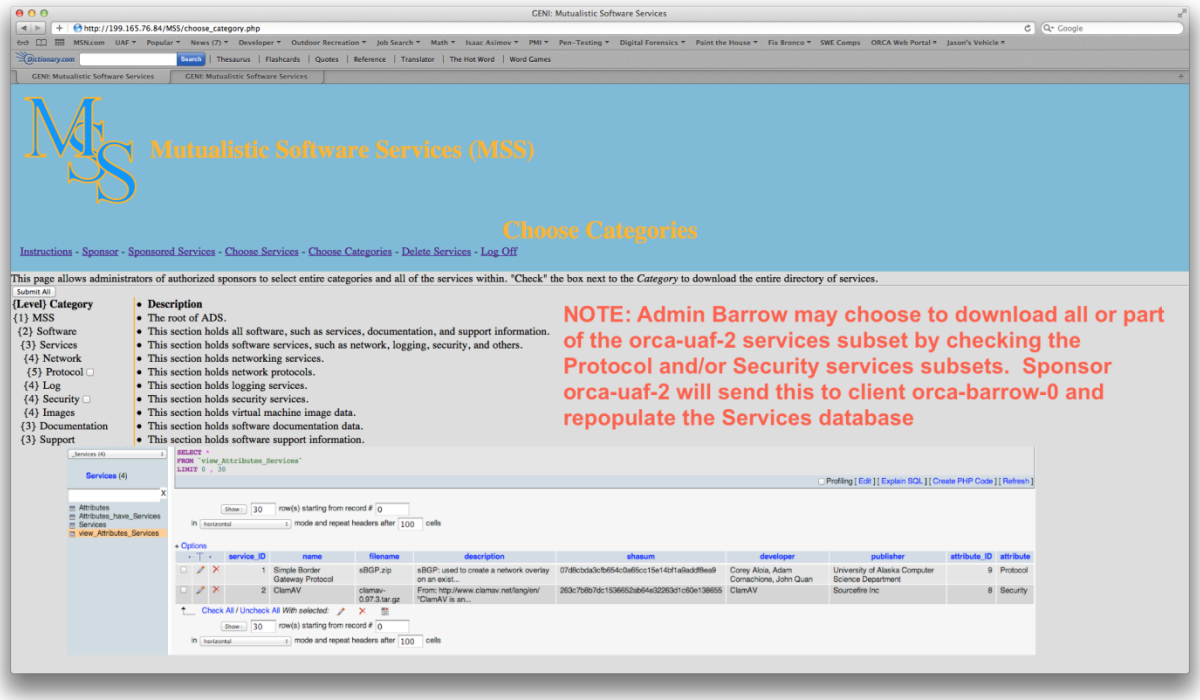


Figure 20. Administrators choose a subset of services to advertise.

- Receives a service download request from the User Interface

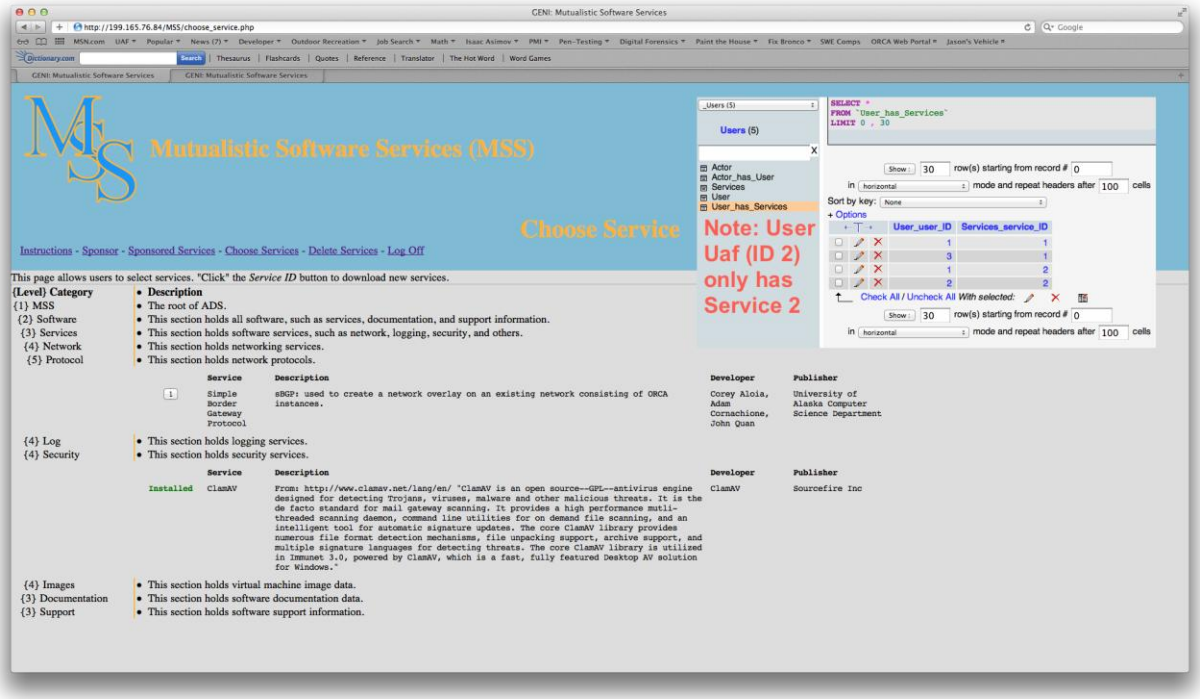


Figure 21. User can request to download service ID 1

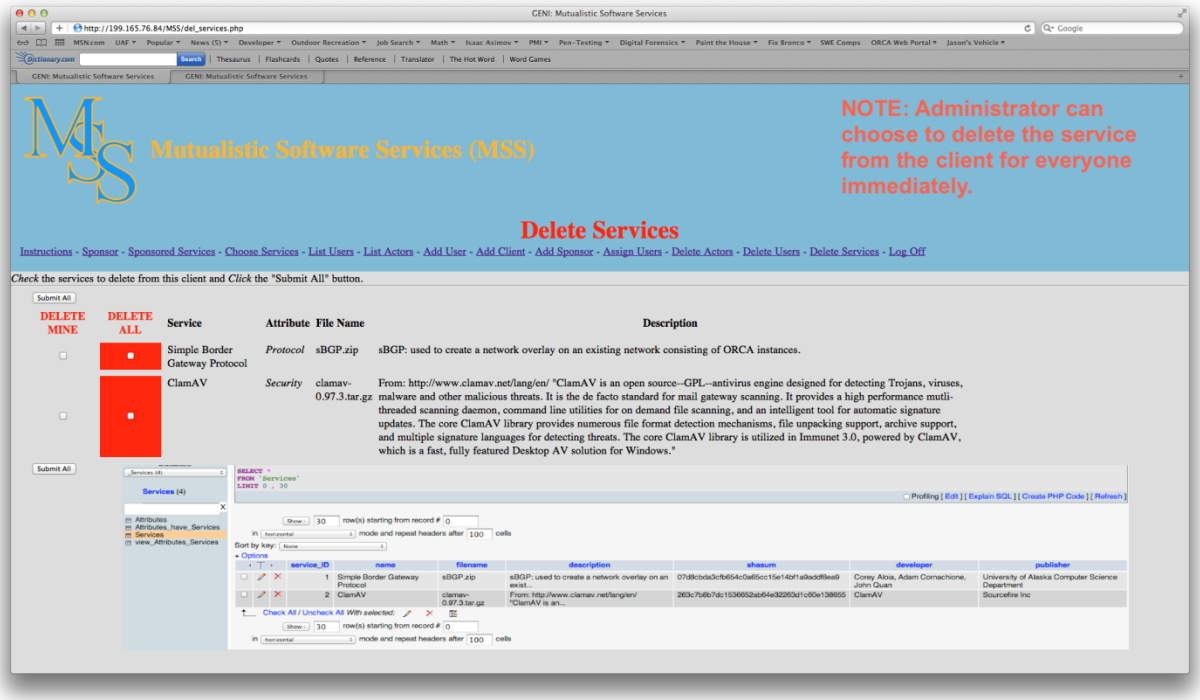


Figure 22. All users can uninstall services, but only administrators can do so immediately.

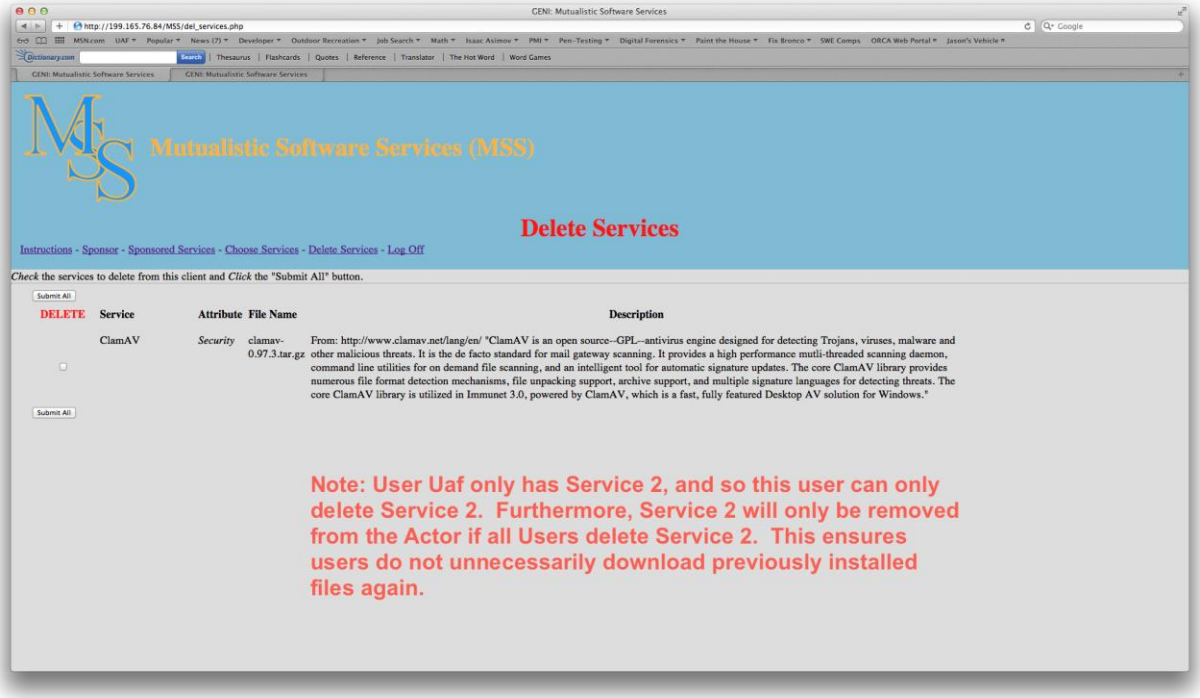


Figure 23. MSS maintains permissions on "Delete Services"

- Uses the Encryption process to encrypt services

    - See the Bash Scripts sh_rsync_command and sh_ssh_command in the *Software Design for Mutualistic Software Services (MSS) Version 1.0.*
- Uses the Decryption process to decrypt services
    - See the Bash Scripts sh_rsync_command and sh_ssh_command in the *Software Design for Mutualistic Software Services (MSS) Version 1.0.*
- Uses the Validation process to validate services
    - See the compare_shasum() function in functions_shell.php of the *Software Design for Mutualistic Software Services (MSS) Version 1.0.*
- Receives services from parent MSS-ENTITY

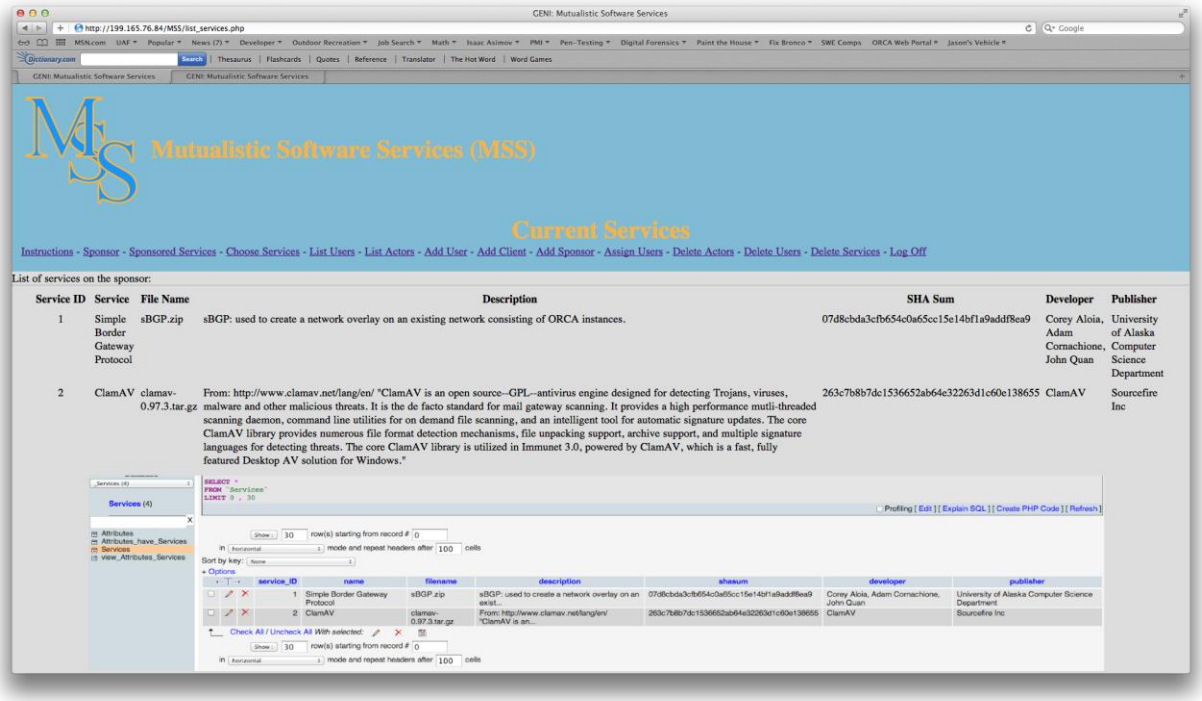


Figure 24. Service query of the sponsor.

- Delivers services to child MSS-AFFILIATES

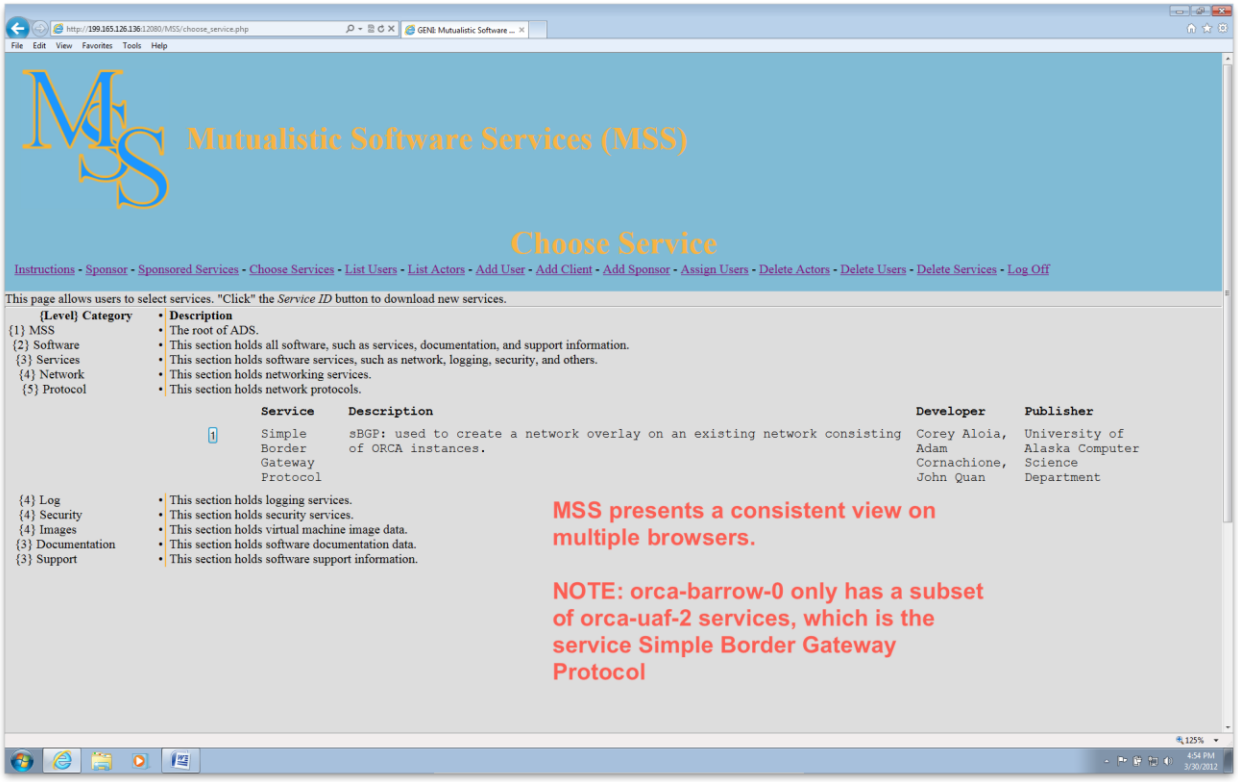


Figure 25. "Choose Service" on Internet Explorer

3.2.4 User Web Site

- (Version 2.0) Delivers service content to service users
- (Version 2.0) Enables MSS-CENTER/MSS-DEVELOPERS to add service content
- (Version 2.0) Delivers GENI information as required by MSS-CENTER
- (Version 2.0) Enables users to email MSS-CENTER/MSS-DEVELOPERS

3.2.5 Administrator Web Site

- (Version 2.0) Delivers administrative content to MSS administrators

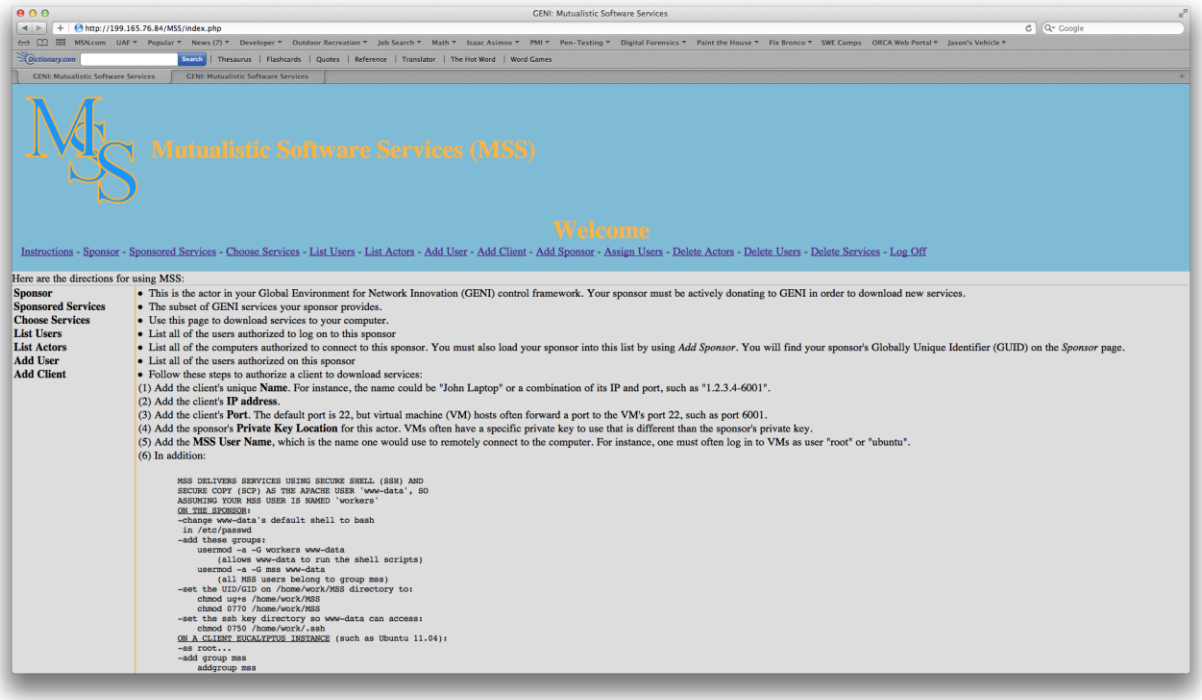


Figure 26. Delivers tailored instructions according to user permissions.

- (Version 2.0) Allows MSS-CENTER/MSS-DEVELOPERS to add administrative content
- (Version 2.0) Delivers GENI information as required by MSS-CENTER
- (Version 2.0) Enables users to email MSS-CENTER/MSS-DEVELOPERS

3.2.6 Developer Uploads

- (Version 2.0) Allows enrolled MSS-DEVELOPERS to add services
- (Version 2.0) Allows enrolled MSS-DEVELOPERS to add service descriptions
- (Version 2.0) Enables MSS-CENTER to approve services
- (Version 2.0) Enables MSS-CENTER to approve service descriptions

3.3 Service Repository

3.3.1 File System

- Holds MSS services

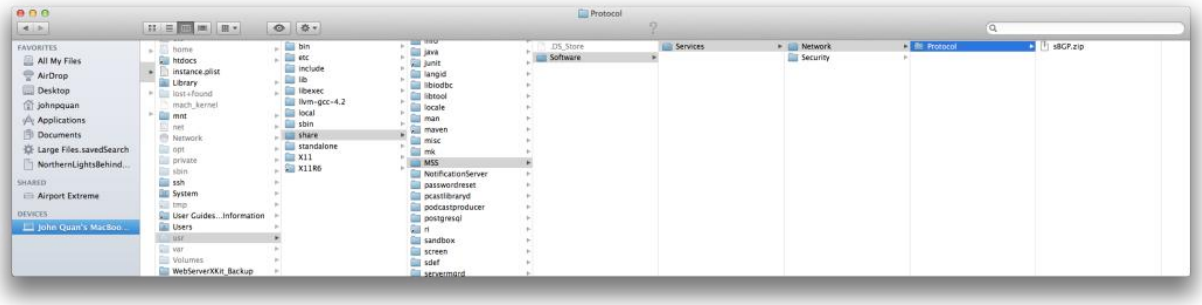


Figure 27. MSS Services subset downloaded to an Apple MacBook.

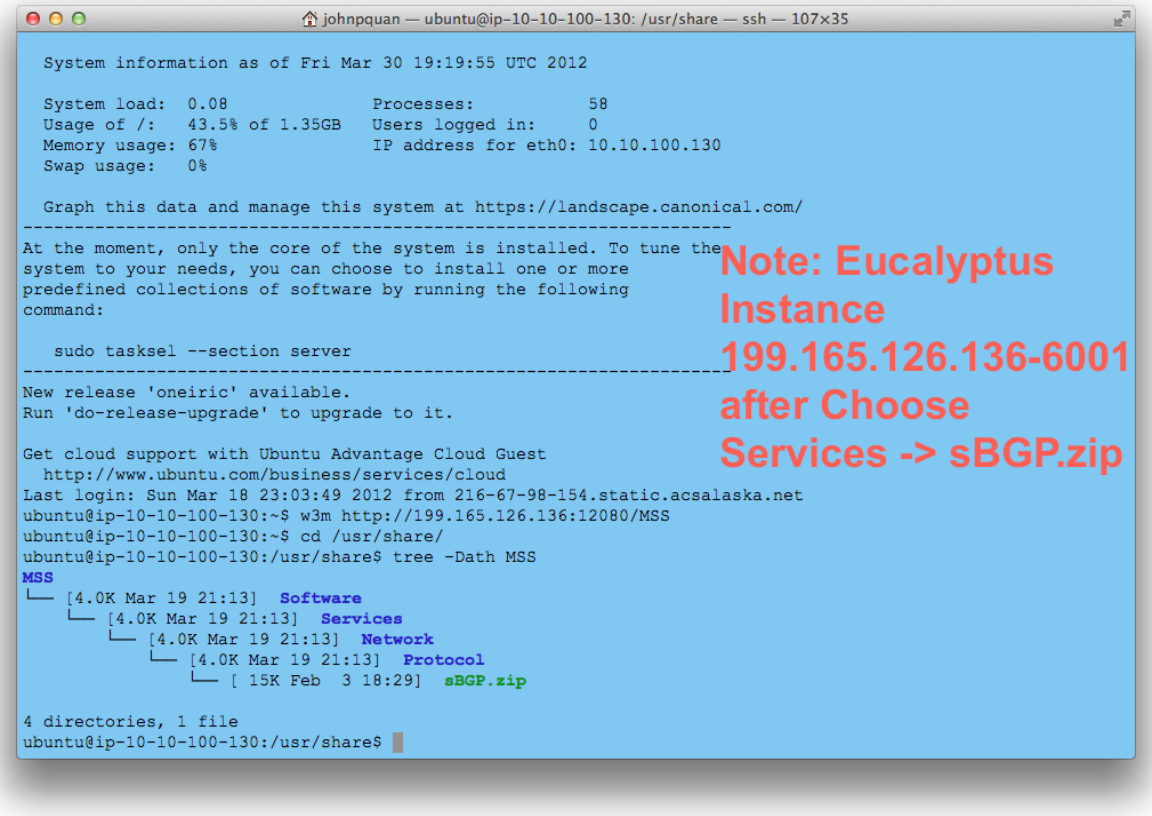


Figure 28. The services subset matches the Attribute Tree.

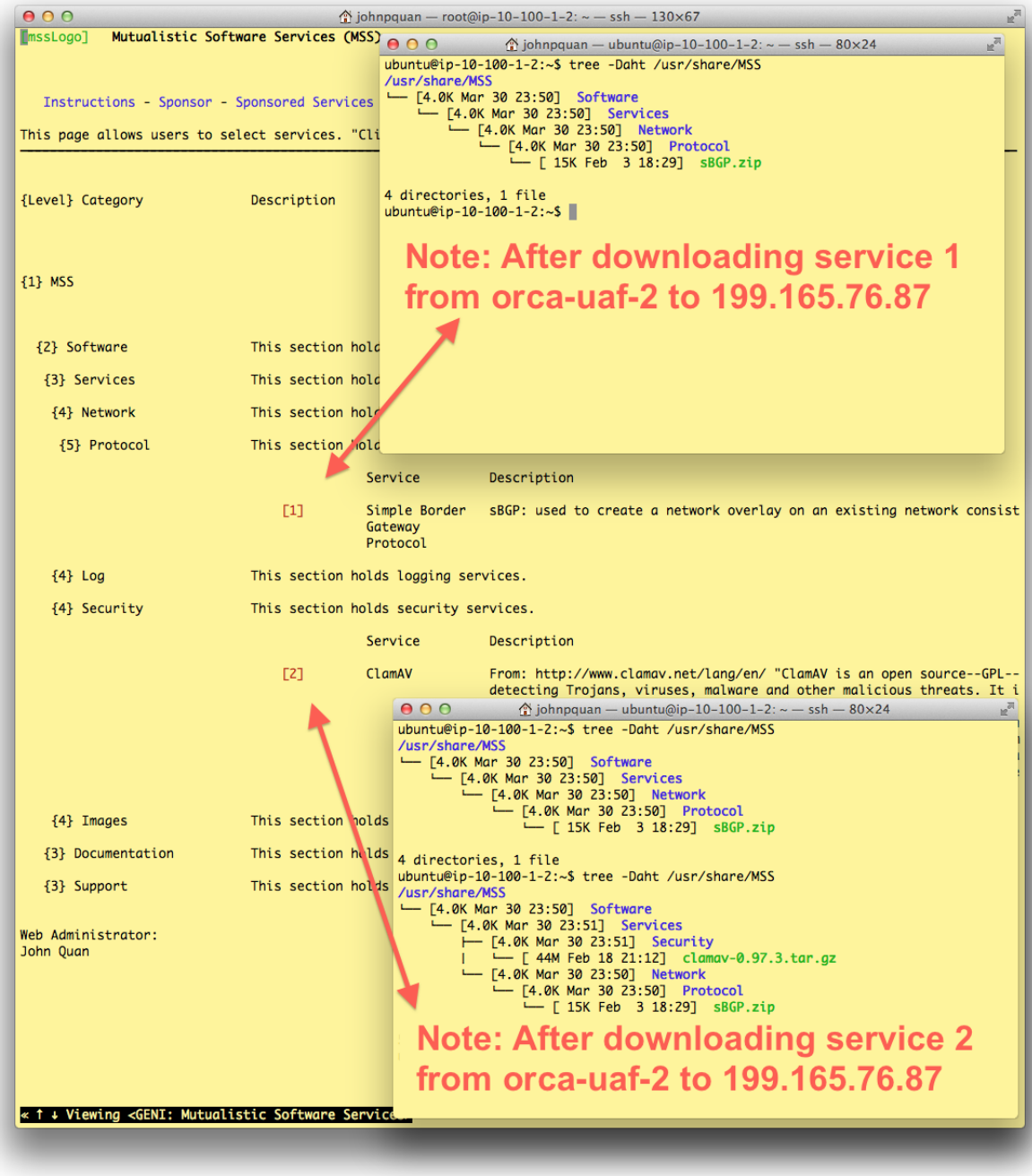


Figure 29. Eucalyptus Instance downloads from MSS on a canonical ORCA cluster.

- (Version 2.0) Holds all MSS services for MSS-CENTER
- Holds a subset of parent services, to include the entire subset, for its MSS-AFFILIATES

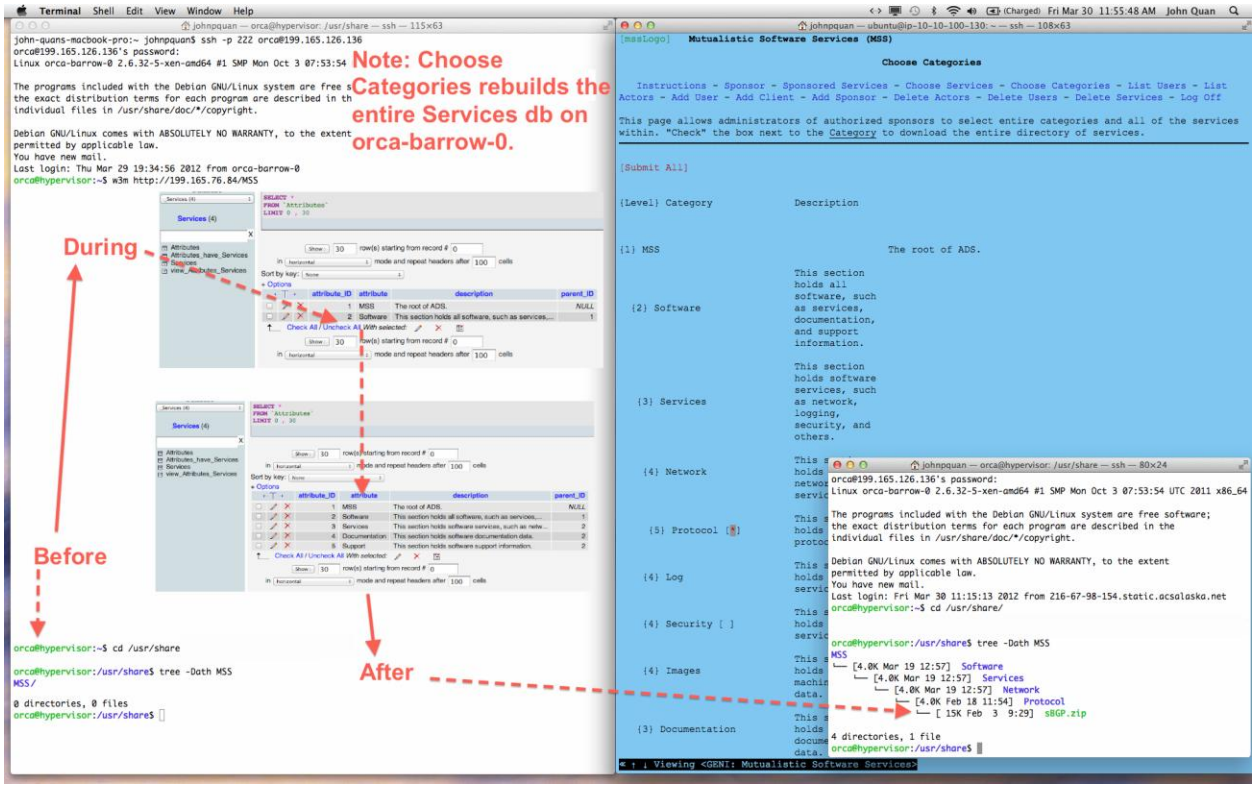


Figure 30. MSS delivers a services subset and rebuilds the Services DB.

4. Evaluation

The MSS system meets all criteria listed for Version 1.0, excepting that the page *Assign Users* does not display in the text-based browser w3m. However, the testing committee believes that this is an unnecessary requirement because only administrators use this page and not Eucalyptus Instance users, for which the w3m requirement exists. *Assign Users* works well in graphical web browsers, and this committee doubts that an administrator would prefer using a text-based browser with a limited feature set to a full-featured graphical web-browser, such as Internet Explorer, Firefox, Safari, or Chrome.

The testing committee did find several areas for improvement in MSS, which are listed below in order of importance:

Feature	Problem	Solution
No separate `password()` function	`add_user.php` has the administrator type in a new user's password twice to verify a correct entry, but it performs no strong password checking. One could make the user's password as simple as "a" or even blank.	Version 1.0 does not include a strong password-checking requirement, but we believe every web-facing server should perform this. This code is widely available, and a web search for "php strong password checker" listed many links.
Rebuilding the client's Services database	This functionality is effective, but incredibly slow. For instance, rebuilding the `orca-barrow-0` database from `orca-uaf-2` took 2 minutes 40 seconds, even with a very small attribute tree. This is mostly due to a very slow network connection, but the system is designed to open an SSH connection for each entry in all of the Services tables.	Connect only twice from sponsor to client when rebuilding the Services database, instead of connecting for each entry (14 separate connections in testing). Developers can accomplish this by writing the rebuild to a SQL file, passing the file to the client (connection one) and then sending an SSH command (connection two) to load the file into SQL. The SQL code should combine all of the SQL statements into one file. Since each connection to Barrow takes about 5 seconds, this would reduce the total Services database rebuild to about 10 seconds.
Firefox display malfunction	Firefox does not display quotes in the instructions correctly.	Define the format between the <pre></pre> tags in the user instruction set.
Attribute tree display malfunction	w3m does not display "{1} MSS The root of MSS" correctly. The words "The root of MSS" are aligned all the way to the right of the page. All graphical web browsers present the attribute tree properly.	By viewing the source HTML, one sees that there are too many </table> end tags. Format this section so that the attribute tree does not contain unnecessary tags. The testing committee was unable to repair this malfunction after a short time of troubleshooting.