XSEDE
Community Infrastructure
Use Cases

April 20, 2017

Version 1.1

This document outlines the typical needs of the "Community Infrastructure" area. The XSEDE system is explicitly intended to be open and extensible. Unlike a stand-alone product, XSEDE provides a framework for delivering research services and applications. In order to use this framework, community members (application developers, service providers, campus IT administrators) must be able to access details about the system's design, implementation status, and the driving user needs.

Note: In these use cases, the terms "service" and "service provider" are used broadly. Any of the following would be examples of "services" if they are shared with researchers via XSEDE: a computer system, a website or web-based application, a computer system with a specific software application installed for use, a high-capacity data storage device, a long-term data archiving service, a virtual machine (IaaS) host, a software repository, or an identity provider.

CI-1: Access system information

CI-2: Manually publish system information

CI-3: Automate changes to system information

CI-1: Access system information

An **XSEDE community member** needs to access specific information about the XSEDE system. Several specific examples of this general need follow.

a. A **researcher** has been invited to participate in a use case review.
b. An **XSEDE staff member**, an **application developer**, or a **service provider** needs to find underserved use cases in order to offer something of value to the community.
c. A researcher, an application developer, or an service provider needs to know which use cases XCI is working on in order to avoid duplicating effort.
d. A researcher, a service provider, an application developer, or a **campus IT administrator** needs to find software for a particular use.
e. A researcher needs to find a provider for a particular service.

In most cases, the **community member** wants to experience it as follows.

1. First, the community member opens a web browser and navigates to www.xsede.org.
2. Then, the community member navigates the website to locate the system information section.
3. Finally, the community member locates the specific system details he/she needs.

We'll take any solution, as long as the following are true.

1. The community member shouldn't need to have an XSEDE identity or authenticate to the XSEDE website to access the necessary information.
2. No client software, other than a standard web browser, should be needed to access or use the necessary information.
3. The community member shouldn't be able to change the system information he/she is accessing.

CI-2: Manually publish system information

An **XSEDE community member** needs to update the public record of specific system information so that others can find and use the information. Several specific examples of this general need follow.

a. An **XCI staff member** needs to revise/update the known use cases.
b. An XCI staff member needs to publish the degree to which use cases are/are not satisfied.
c. An XCI staff member needs to publish which use cases XCI is currently working on.
d. A **software provider** needs to add/update information about their software.
e. A **service provider** needs to add/update availability of a service.

In most cases, the **community member** wants to experience it as follows.

1. First, the community member opens a web browser and navigates to www.xsede.org.
2. Then, the community member navigates the website to locate the system information section.
3. Then, the community member locates the specific system details he/she needs to change (or the place he/she needs to add new information).
4. Finally, the community member makes the necessary changes.

It'll always be like this except, in Step 4, when the community member hasn't already authenticated with the system, or if the latest authentication has expired. In that case, the community member will need to authenticate before he/she will be able to change any information.

We'll take any solution, as long as the following are true.

1. No client software, other than a standard web browser, should be needed to access or change the necessary information.
2. The system must allow XSEDE staff to authorize specific individuals and/or groups to change specific types of information. For example: XCI-RACD members can change use cases; a software

provider can change details about his/her software products; a service provider can change details about his/her services.

3. XSEDE's standard authentication mechanism should be sufficient for Step 4.

CI-3: Automate changes to system information

An **XSEDE community member** needs to automate updates to system information so that updated information about one or more components can become available without human action.

In most cases, the **community member** wants to experience it as follows.

1. First, the community member registers with XSEDE's information service and obtains a credential for updating system information.
2. Then, the community member downloads and installs a lightweight "information update application:" a tool for pushing status updates to the XSEDE information service.
3. Then, the community member configures the information update application with the credential provided in Step 1, a means to gather information about the component(s), and a mechanism for periodically executing the application.
4. Finally, the community member checks the XSEDE system information service (as described in use case CI-1) to confirm that the information about the component(s) is correct and current.

It'll always be like this except in rare instances when the community member chooses to build the status update mechanism directly into a component using a software application programmer interface (API). In that case, Step 2 either won't happen at all or will be replaced by downloading a software development kit (SDK) and Step 3 will be configuring the component to update XSEDE's information service using the API.

We'll take any solution, as long as the following are true.

1. In its simplest form, the registration mechanism in Step 1 should allow registration via a Help Desk ticket.
2. The information update application in Step 2 must work on a wide variety of Linux/Unix distributions and hardware architectures.
3. The information update application in Step 3 should include configuration examples for the most commonly used (and commonly provided) services.
4. The credential provided in Step 1 should only allow the community member to update the system information for his/her component(s).