Service Provider Integration
Use Cases

July 18, 2019

Version 1.7

This document outlines specific coordination mechanisms that Service Providers (SPs) use to promote and manage their resources within the public research computing environment. SPs are independent organizations that provide services to the public research computing community. For example, an academic institution might allow researchers and/or students at other institutions to use the campus's data archive system. As another example, a supercomputing center might operate a national-scale supercomputer on behalf of the National Science Foundation's eXtreme Digital (XD) program.

Note: The research computing community has varying forms and degrees of coordination among SPs. For example, the XSEDE system defines SPs as "autonomous entities that agree to coordinate with XSEDE and with each other to varying degrees." [1] The first use cases in this category came from XSEDE, but others are welcome.

SPI-01: Resource integration console

SPI-02: Guided resource discovery

SPI-03: View resource information

SPI-04: Access resource information from an application

SPI-05: Active account information

SPI-06: Emergency account suspension

SPI-07: Enable a research community to install software on a resource

SPI-08: Register availability of a specific application or service for use

References

History

SPI-01: Resource integration console

An **SP representative** (an individual representing a service provider) wants to access a console that allows him/her to view the up-to-date status of all XSEDE system integration steps for a specific resource. (A resource is a specific service offered to the community, like a computing system, a storage

service, a science gateway, etc.) If authorized, he/she will also be able to change the status for any integration step.

In most cases, the SP representative wants to experience it like the following steps. We assume the SP representative has registered and has been authorized to view (and possibly manage) the integration console for one or more resources.

1. First, the SP representative opens a web browser and navigates to a service provider management site.
2. Then, if the SP representative isn't already authenticated, the SP representative clicks the "login" link (or equivalent) and authenticates.
3. Then, the SP representative selects the resource of interest from a list.
4. Then, the SP representative views a dashboard-style display that shows a set of system integration steps for the resource, including the current status for each. If the SP representative is authorized to manage the console, an "Edit" link is displayed next to each step.
5. Finally, if the SP representative clicks an "Edit" link, he/she will be presented with an interface to edit the status details for that step. The details of the interface will vary depending on what choices are available to the SP for the given step.

It'll always be like that, except when the SP representative hasn't previously associated him/herself with the resource, in which case Step 3 won't list the resource. The representative will need to click "Add a resource" or something like that to file a request to associate his/her identity with the resource, and the request will be submitted for approval by the managers of the service provider management site.

If the SP representative has never associated him/herself with any SP resource, the SP management site may not allow him/her to access the site in Step 2. In this case, the representative will need to submit an email request for access to the SP management site.

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

1. In Step 2, the SP management site must use a community-wide login service as described in use case CAN-6.
2. The integration steps displayed in Step 4 should be based on an official/canonical set of steps for all resources, though it can be customized for the specific resource and the current status of the resource. E.g., there might be different steps for different classes of resources (e.g., storage vs. HPC vs. cloud), different coordination levels (L1, L2, L3, etc.), and previous decisions entered into the system about the resource (e.g., accepting or declining optional integration steps).
3. In Steps 4 and 5, the interface should provide links for further information/actions for each integration step. E.g., description, documentation, registry data, decisions that need to be made.

SPI-02: Guided resource discovery

A **community member** needs to identify or select resources appropriate for specific research tasks so the community member can request access to these resources in an allocation proposal. This resource discovery will likely be useful for other resource selection needs.

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

1. First, the community member opens a web browser and navigates to a resource information site.
2. The user is presented with a default/initial description of available resources.
3. Then, the user iteratively selects resource characteristics that they are interested in.
4. Finally, the user views a summary of the resources with the appropriate characteristics.

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

1. The resource information site should take advantage of all available resource status information.
2. The resource information site should cover any resources whose service providers choose to provide descriptions (see use case SPI-01), including XSEDE resources at all coordination levels and any campus resources that choose to publish their descriptions.
3. The solution should allow for iterative deployment of discovery interfaces with incremental improvements.
4. Access to the resource information site should not require registration or login.
5. It should be easy to distinguish between resources that participate in the XSEDE allocation process (XRAS) and those that do not.

SPI-03: View resource information

A **service provider (SP)** or a **community member** responsible for SP integration needs to view all community information available about a specific resource in order to verify that the information is correct and complete.

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

1. First, the user opens a web browser and navigates to the SP management site.
2. Then, the user selects the resource of interest from a list.
3. Finally, the user views a dashboard-style view providing access to all registered information about the resource.

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

1. The community information displayed in Step 3 should include all resource status information from all community sources, including both static descriptive information (see use cases SPI-01, CI-02, CAN-12) and dynamic status information (see use cases CI-03 and CAN-11).

2. The resource information site should cover any resources whose service providers choose to provide descriptions (see use case SPI-01), including XSEDE resources at all coordination levels and any campus resources that choose to publish their descriptions.
3. If the dashboard in Step 3 requires authorization for any of the information, it must use a community-wide login service as described in use case CAN-6.

SPI-04: Access resource information from an application

An **application developer** needs to add a feature to an application that retrieves and displays up-to-date description and status information about a community resource from all available sources.

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

1. First, the developer accesses the community's information services documentation and locates the material describing application programmer interfaces (APIs) for resource information.
2. Then, the developer uses one of the documented APIs to add resource information to his/her application, configuring the information retrieved and displayed to the application's needs.
3. Finally, the users of the application view the resulting resource information while using the application.

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

1. The API should enable applications to retrieve information in a web browser format (for direct inclusion in websites) or in machine readable form (for application processing).
2. The API should allow the developer to customize the information details to be retrieved and (for web browser formats) how the details are displayed.
3. The API should enable applications to retrieve (and optionally display) all available resource description and status information from community sources. (See use cases SPI-01, CI-02, CI-03, CAN-11, and CAN-12.)
4. The API should provide access to data about any resources whose service providers choose to provide descriptions (see use case SPI-01), including XSEDE resources at all coordination levels and any campus resources that choose to publish their descriptions.

SPI-05: Active account information

A **community member** responsible for community security needs to obtain a list of all resources and services on which a specific community identity has an active account. (Examples: an XSEDE security team member is responding to an account compromise, or a user support team member is assisting someone with logging into a resource.)

In most cases, the **community member** wants to experience it as follows. We assume the community member has registered and has been authorized to access the relevant interfaces.

1. First, the community member accesses an information interface either via a remote login session or a web interface.
2. Then, the community member enters a username. (This may be a community-wide username or a local username on a specific resource.)
3. Finally, the staff member is shown a list of community usernames and, for each username, the resources and/or services on which the user has an active account, along with any resource-specific identity information (e.g., local username). (Matches may be in the community username or in the local username on a resource.)

We'll take any solution, as long as Step 1 requires the community member to authenticate him or herself and access is permitted only to authorized individuals.

SPI-06: Emergency account suspension

While responding to an account compromise, a **community member** needs to temporarily prevent a specific community user identity from being used on community resources and services.

In most cases, the **community member** wants to experience it as follows. We assume that the community member has already registered and has been authorized to access the relevant interfaces.

1. First, the community member accesses an interface either via a remote login session or a web interface.
2. Then, the community member issues a suspend request on one or more community usernames.
3. Finally, the community member is shown a list of the specific resources and services where the community username has access and the status of the suspend request for each resource and service.

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

1. Step 1 requires the community member to authenticate him or herself to the system and access is permitted only to authorized individuals.
2. When Step 2 is completed, the system must be able to take different actions for each resource or service, depending on the various agreements between the service provider (SP) and the community. For example, some SPs may give the community a mechanism to deactivate their local accounts, while others may require the community to submit a request that is acted on by SP personnel.
3. The information display in Step 3 must accurately reflect the action taken for each resource or service. (E.g., the account was disabled, a suspension request was submitted to the SP, no action was taken, etc.)

SPI-07: Enable a research community to install software on a resource

A **service provider** wants to enable research communities to install software on a resource they administer so the software can be used by the community and so community information systems

contain resource-specific instructions that researchers can find and follow. We assume the resource has already been integrated with the community. (See SPI-01.)

In most cases, the **service provider** wants to experience it as follows.

1. First, the service provider opens a web browser and navigates to the resource integration console for a specific resource. (See use case SPI-01.)
2. Then, the service provider opens the "Community Software Area" (CSA) interface within the console and is presented with a brief overview of how other resources offer this capability to their users and an interface for managing public information about the resource's CSA.
3. The service provider can use the interface to indicate that the resource offers a CSA and to provide specific instructions for researchers who wish to install software on the resource.
4. The researcher can use the interface to add information about a specific software tool that has been installed by a researcher for his/her community.
5. Finally, information entered in the console will appear on the community's public website.

We'll take any solution, as long as…

1. In Step 2, the overview text has been provided by a representative of the SP forum and accurately reflects current practices by service providers.
2. In Step 3, the CSA availability indicator is a binary "yes" or "no," but the instructions should be entirely left to the service provider without prescribed format or structure. (In practice, the instructions could be used by a service provider to explain how the resource's CSA differs from other CSAs.)
3. In Step 4, the interface should require enough structure in the information to ensure that researchers searching for a specific software tool will find a copy on this resource if it exists.
4. In Step 5, the service provider should be able to see how the new information appears on the public site before moving on to other work. This implies that information entered via the resource integration console should appear on the community's public website as soon as possible: immediately would be best, but not longer than a few hours.

SPI-08: Register availability of a specific application or service for use

A service provider or science gateway operator (hereafter referred to as **"service provider"**) needs to advertise an application or service they make available for use as part of a resource or gateway. We assume that the service provider's resource or gateway has already been registered with the system and the service provider is registered as one of its administrators (see SPI-01).

In most cases, the **service provider** wants to experience it as follows.

1. First, the service provider opens a web browser and logs in to the service provider management site.

2. Then, the service provider selects the resource of interest from a list of resources he or she administers.
3. Then, the service provider selects an option to manage the list of software and services offered by the resource.
4. Finally, the service provider uses the interface to add, remove, or change the configuration of the application or service on her resource or gateway.

It will always be like this, except when, in Step 4, the software or service to be registered hasn't been added to the system yet. In this case, the service provider must first add the resource as described in use case SPI-01.

We'll accept any solution as long as in Step 1, the service provider management site uses a community-wide login service as described in use case CAN-6.

References

[1]   https://www.xsede.org/xsede-federation

History

	Version	Date	Changes	Author
Entire document	1.0	5/18/2017	First version, described use cases (01-06)	Liming, Navarro
Entire document	1.1	6/22/2017	Minor edits based on feedback	Liming
Use cases 07-08	1.5	8/17/2018	Added use cases 07 and 08	Liming
Use case 07	1.6	10/17/2018	Remove "data" from this use case	Navarro
Entire document	1.7	07/18/2019	Remove unnecessary XSEDE terminology; remove XSEDE implementation details from SPI-02	Liming