Campus Bridging Use Cases

July 28, 2019

Version 3.1

These use cases describe how campus information technology (IT) administrators and campus-based research projects: (a) treat public research computing resources as extensions of their campus environments, (b) offer services at their campuses to others in the public research community, and (c) deliver new services to their campuses that leverage the public research community's experiences and solutions.

CB-01: InCommon-based authentication to public research resources

A **campus IT administrator** would like researchers at his/her campus to be able to login to *community resources* (services provided by other members of the public research computing community) using InCommon-based authentication mechanisms. [1][2][3][4]

In most cases, the **campus IT administrator** would like to experience it as follows.

1. The administrator operates an InCommon IDP for his/her campus.
2. The administrator directs on-campus researchers to login to community resources (e.g., XSEDE User Portal, login servers at other campuses or service providers) using their campus identity.
3. Any researcher at the administrator's campus can select their campus's InCommon IDP while logging in to a community resource, authenticate to the IDP, and then (if they are authorized) access the resource without further authentication.

It's always like the steps above, except when the user is logging in to a Science Gateway, in which case jobs may be "translated" to ownership by an account used by the gateway.

We'll accept any solution to this problem, as long as the following are true.

1. Once authentication is completed at the InCommon IDP, access to community resources doesn't take more than an additional 5 seconds.
2. Access via InCommon must be supported by XSEDE Level 1 and 2 resources.
3. Access via InCommon must be supported by XSEDE services (XSEDE User Portal, XSEDE Resource Allocation Service, etc.).
4. The InCommon authentication mechanism must be available for use by Level 3 and campus IT administrators to enable access to their own resources via InCommon.

CB-02: Share the public research computing "environment" with campus resources

A **campus IT administrator** needs to reuse documentation and tools provided by other members of the public research computing community in his/her campus environment to reduce local costs and to smooth the path to public research computing for campus researchers. We assume the community has built a consensus around "common environment" expectations.

In most cases, the **campus IT administrator** would like to experience it as follows.

1. The administrator finds an editable documentation template on a community website that he/she can reuse to document his/her campus systems.
2. The administrator finds training materials on a community website that he/she can use with minimal alterations for campus researchers.
3. The administrator finds software packages on a community website that he/she can use on campus resources to provide the open source elements of a basic community system.

We'll accept any solution to this problem, as long as the following are true.

1. Documentation and training materials provided by the community must be released with a license that allows reuse and modification, such as the CC BY 3.0 license. [5]
2. Documentation and training materials should be provided in editable, commonly used formats.
3. Software provided by the community must be available for use under a free-use license.
4. The community's "common environment" includes the following items, which we assume are already standard across community resources: directory hierarchy, locations of standard software 'kits' and optional locally-installed or user-contributed software.

CB-03: Remote desktop services for researchers

A **researcher** on a college or university campus needs to open and use a remote desktop session for a period of time potentially measured in several days. The system that provides the desktop session is a service of the public research computing community.

In most cases, the **researcher** would like to experience it as follows.

1. The researcher's campus IT administrator installs any required client software on the researcher's device.
2. The researcher uses the device and initiates a remote desktop session.
3. The session stays active as long as the researcher specifies. (This could be multiple days.)
4. The researcher may temporarily disconnect from the desktop session and reconnect to it later. The state of the remote desktop should be retained across sessions.

It will always be like this except when the researcher doesn't have a local campus IT administrator. In this case, Step 1 is: "The researcher installs any required client software on the researcher's device."

We'll accept any solution to this problem, as long as the following are true.

1. Client software required on the researcher's device is easy to obtain (free or low-cost) and easy to install (pre-installed on most systems or very simple installation). Ideally, the client software would be available for use on personal systems owned by the researcher as well as on campus systems.
2. The solution should work with a variety of research devices, including desktop computers, laptop computers, tablets, and high-end smartphones.

CB-04: Access campus research data from a community resource

A **researcher** needs access to data stored on a campus resource when using a *community resource* (a service provided by another member of the public research computing community) for analysis and/or visualization. [1][6][7][8][9]

In most cases, the **researcher** would like to experience it as follows.

1. The researcher logs in to a community resource to use its analysis/visualization capabilities.
2. The researcher accesses his/her data on a campus resource from the community resource.
3. The researcher analyzes and/or visualizes data on the community resource.
4. The researcher writes/updates/deletes data back to the campus resource.

We will accept any solution to this problem so long as the following are true.

1. In the event of a transient failure (e.g., network glitch, client or server fault, expired credentials), the system should be able to restart transfers and notify the user once the transfer has successfully completed.
2. A graphical user interface should be available to initiate and manage file transfers.
3. It shouldn't take longer than one working day for a researcher with proper permissions to install the necessary software on their campus system.
4. The mechanism should allow access to flat files, file archives, and database files.

CB-05: Workflow automation combining community and campus resources

A **researcher** needs to perform an analysis via a distributed workflow in which tasks automatically execute on and access data from various *community resources* (services provided by members of the public research computing community) and campus resources. [10] We assume the community has selected, documented, tested, deployed, and configured at least one distributed workflow service.

In most cases, the **researcher** would like to experience it as follows.

1. The researcher opens an interactive session with the workflow system, specifies the workflow to be executed, and initiates the workflow.
2. The workflow system carries out the tasks specified by the workflow.
3. The workflow system notifies the researcher when the workflow has completed.

It'll always be like that, except in the following cases.

1. The researcher may prefer or need to use a batch (non-interactive) interface to the workflow system, in which case the user must submit the workflow specification and delegated user credentials (to be used by the workflow system when submitting tasks or accessing data on other systems) to the system via a batch interface. The user can check the status of the workflow (queued, executing, completed). The system will notify the researcher when the workflow has completed.
2. The workflow to be executed is submitted on the researcher's behalf by a science gateway, in which case the workflow system must employ a delegation mechanism that allows the researcher to delegate credentials to the gateway for use on both community and campus resources. The gateway will notify the researcher when the workflow has completed.

We'll accept any solution to this problem so long as the following are true.

1. The workflow system must be supported by XSEDE Level 1 resources.
2. The workflow system must be available for XSEDE Level 2 and Level 3 resource SPs to install and configure on their resources.
3. The workflow system must be available for campus IT administrators to install and configure on their campus systems.
4. The workflow system must support tasks that execute on community resources or campus resources, consistent with the researcher's authorization(s).

5. The workflow system must support tasks that access data on community resources or campus resources--via a stage in/out model or via a CRUD (create, read, update, delete) model--consistent with the researcher's authorization(s).
6. After the workflow is initiated, the workflow system should not require further user interaction to complete the workflow. Transient issues (e.g., system downtime, network glitches, expired credentials, system misconfiguration) should be handled automatically by the workflow system or by support personnel.

CB-06: Sharing computational facilities among campuses

A **researcher** needs to share a computational facility on his/her campus with other researchers--possibly at other institutions--using features of the public research computing community to mediate the sharing and establish a familiar environment. [1][6][7][8] We assume the community supports use cases CB-01 and CB-02.

In most cases, the **researcher** would like to experience it as follows.

1. The researcher builds a cluster, Condor flock, cloud, or other sort of computational resource using campus computing resources.
2. The researcher installs and configures the InCommon authentication mechanism from CB-01 on his/her resource.
3. The researcher uses the documentation, training, and/or software packages from CB-02 to make his/her resource similar to other community resources.
4. The researcher manages his/her own accounting, "value exchange," policy compliance, and security responses.
5. The researcher defines user groups and sets access control policies for his/her resource based on these groups.

It'll always be like this except when the researcher elects to become an XSEDE service provider (SP).[1] In that case, XSEDE would be able to manage exchange rates among the campus-contributed resources, Level 1 resources, and Level 2 resources, and would be able to return service units to campus contributors in accordance with a negotiated Service Level Agreement. In that case, XSEDE would also provide security notifications in the event of a security breach related to accounts or services that use campus-based authentication mechanisms, and the researcher would be expected to participate in XSEDE's integrated help desk function.

We'll accept any solution to this problem as long as the InCommon authentication mechanism from CB-01 does not require users of the researcher's computational facility to have community allocations or participate in a community allocation process.

CB-07: Support for commercial service providers

An **independent resource provider** wants to offer a privately operated computational facility for use by researchers at colleges and universities in a manner consistent with the expectations of users of public

[1] XD Service Providers Forum: Charter, Membership, and Governance (v10.1_120228). (https://www.xsede.org/documents/10157/281380/SPF_Definition_v10.1_120228.pdf)

research computing resources in exchange for a fee or other form of payment. [1] We assume the community supports use cases CB-01 and CB-02.

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

1. The resource provider builds a cluster, Condor flock, cloud, or other sort of computational resource using privately owned and operated computing resources.
2. The resource provider installs and configures the InCommon authentication mechanism from CB-01 on his/her resource.
3. The researcher uses the documentation, training, and/or software packages from CB-02 to make his/her resource similar to other community resources.
4. The resource provider manages his/her own accounting, payment mechanism, and security responses.
5. The resource provider advertises the mechanism by which to contract for time (credit card or Purchase Order).
6. A **researcher** sets up an account with the resource provider.
7. The researcher uses the private resource, in ways that are convenient because they leverage system similarity with community resources, and the resource provider receives payment from the researcher.

We'll accept any solution to this problem as long as the following are true

1. The InCommon authentication mechanism from CB-01 does not require users of the researcher's computational facility to have community allocations or participate in a community allocation process.
2. The resources from CB-02 are available under licenses that permit commercial use.

CB-08: Use a community login service with campus login servers

A **campus IT administrator** wants to allow researchers registered with the public research computing community to login to campus login servers (remote command shell) using their community usernames/passwords.

In most cases, the **campus IT administrator** wants to experience it as follows.

1. First, the administrator registers his/her campus login service(s) with the community. The result should be that the community login service becomes aware of the campus login service(s) and allows end users to select it/them as login targets.
2. Then, the administrator configures his/her login services to authorize specific community users.
3. Then, the administrator notifies users that they can use the community login service to access the campus service(s).
4. Finally, an end user authorized by the campus administrator uses the community login service, selects a campus service for login, and is given a login shell (under his/her local ID) on the campus service.

It'll always be like that, except when the end user hasn't registered with the community, in which case the end user will need to register with the community before Step 2 can be completed.

We'll take any solution, as long as...

1. The solution should not require the campus services to participate in a community allocation process.
2. The solution should not require the researcher to have an active or prior community allocation.
3. The administrator must retain control over who can/cannot login to his/her service.
4. The solution should use the standard community registration mechanism.
5. The solution should use the standard community login service interface in Step 4.

CB-09: Access a community data collection from campus

A **researcher** needs access to data from a community data collection (a collection hosted by another member of the public research computing community) when using a campus resource for analysis and/or visualization. [1][6][7][8][9]

In most cases, the **researcher** would like to experience it as follows.

1. The researcher searches or browses a directory of community-hosted data collections to find the one he/she will be using.
2. The researcher specifies the data to be moved from the community-hosted data collection to a campus resource.
3. The system moves the data and lets the researcher know when it's ready.

We'll accept any solution to this problem so long as the following are true.

1. In the event of a transient failure (e.g., network glitch, client or server fault, expired credentials), the system should be able to restart transfers and notify the user once the transfer completes successfully.
2. A graphical user interface should be available to initiate and manage file transfers.
3. It shouldn't take longer than one working day for a researcher with proper permissions to install the necessary software on their campus system.
4. The mechanism should allow access to flat files, file archives, and database files.

CB-10: Synchronize research data between campus and community resources

A **researcher** needs to keep data synchronized when copies are kept on a campus resource and another resource in the public research computing community. [1][6][7][8][9]

In most cases, the **researcher** would like to experience it as follows.

1. The researcher specifies a data set that s/he wishes to maintain, in a synchronized fashion, on a campus resource and one or more community resources.
2. When the researcher makes a change to one copy of the data, the other copies are automatically updated.

We'll accept any solution to this problem so long as the following are true.

1. When data needs to be moved between systems, in the event of a transient failure (e.g., network glitch, client or server fault, expired credentials), the system is able to restart transfers and notify the user once the transfer completes successfully.

2. A graphical user interface should be available to specify the data set to be synchronized.
3. It shouldn't take longer than one working day for a researcher with proper permissions to install the necessary software on their campus system.
4. The mechanism should allow synchronization of flat files, file archives, and database files.

CB-11: Archive research data on a community resource

A **researcher** needs to create an archival copy of research data on a community resource (a storage resource provided by another member of the public research computing community). [1][6][7][8][9] We assume the public research computing community provides at least one data archive service and we assume the researcher has been granted access to use the resource.

In most cases, the **researcher** would like to experience it as follows.

1. The researcher authenticates with an archive service or resource.
2. The researcher identifies the data set (stored on a campus system or another community system) and provides appropriate metadata describing the data set.
3. The researcher specifies a license to govern use of the data.
4. The system makes an archival copy of the data set on the archive resource and adds the metadata and license information so that the data appears in appropriate search results and is accessible per the license.

We'll accept any solution to this problem so long as the following are true.

1. In the event of a transient failure (e.g., network glitch, client or server fault, expired credentials), the system should be able to restart transfers and notify the user once the transfer completes successfully.
2. A graphical user interface should be available to initiate and manage file transfers.
3. It shouldn't take longer than one working day for a researcher with proper permissions to install the necessary software on their campus system.
4. The mechanism should support flat files, file archives, and database files.

CB-12 Setup monitoring and usage reporting for a campus HPC resource

A **campus IT administrator** wants to reuse community practices concerning compute resource monitoring and reporting needs. The administrator would like to monitor load on an HPC environment and generate detailed usage reports based on resource manager logs. We assume the community has built a consensus around its own "common environment" expectations, for HPC-specific monitoring and metric-gathering.

In most cases, the **campus IT administrator** would like to experience it as follows.

1. The administrator finds software packages on a community website to install standard monitoring solutions in an HPC environment.
2. The administrator finds documentation on a community website that he/she can use with minimal alterations for the local environment.
3. After installation and configuration, the administrator is able to see load and job activity over

time of the cluster environment under administration

We'll accept any solution to this problem, as long as the following are true.

1. Documentation and training materials provided by the community must be released with a license that allows reuse and modification, such as the CC BY 3.0 license. [5]
2. Documentation and training materials should be provided in editable, commonly used formats.
3. Software provided by the community must be available for use under a freeuse license.
4. This solution does not require participation in a community allocation process.
5. It does not take longer than 1 day to set up these services on local resources.
6. The solution should provide an option to aggregate monitoring data from multiple systems. At this time, doesn't matter which format or mechanism is used.

CB-13: Provide a simple interface to access any of the high-throughput computing queues available to a campus

A campus IT administrator (the **"administrator"**) needs to provide a simple interface that allows campus researchers to submit tasks to any of the high-throughput computing (HTC) resources available to their campus. This service is hereafter referred to as a "Campus Queue."

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

1. First, the administrator visits a campus bridging website and finds the Campus Queue section. This section provides a description of the Campus Queue and associated downloadable software and documentation.
2. Then, the administrator uses the documentation to identify and assemble the necessary local resources (hardware, software, network, etc.) to set up a Campus Queue and downloads the Campus Queue software.
3. Then, using the documentation, the administrator configures the Campus Queue software to integrate with appropriate campus systems.
4. Then, the administrator configures the Campus Queue software to allow researchers to submit tasks to any HTC queues available for use by campus researchers.
5. Finally, the administrator uses examples or templates from the Campus Queue documentation to advertise and document the Campus Queue within the campus research community.

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

1. In Step 2, the Campus Queue software and documentation must be licensed for use via a Creative Commons or similar license that permits free use by research computing systems.
2. In Step 3, "appropriate campus systems" may include: user authentication, data storage service, accounting (to support department "charge backs" or allocations).
3. In Step 3, once configured, campus researchers must be able to access the Campus Queue service using their existing campus identity and credentials. They should not need to register a new identity.

4. In Step 4, the configuration settings should allow configurations with queues on campus systems (using the user's campus identity for submission) and on other systems (using a shared/group identity corresponding to the campus research organization).
5. In Step 4, for queues that are not administered by the campus and don't use campus identities for submission, the software must provide a mechanism for recording the identity of the researcher for each submission, allowing traceability from each submission to the individual who submitted it.
6. In Step 5, the examples or templates for advertising/documenting the Campus Queue must be licensed for use via a Creative Commons or similar license that permits free use by research computing systems.

CB-14: Submit tasks to any of the high-throughput computing queues available to a campus

A campus **researcher** needs to submit tasks to one of the high-throughput computing (HTC) resources available to his or her campus.

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

1. First, the researcher visits the campus's research computing website and finds the Campus Queue section. This section provides a description of the Campus Queue and associated documentation, including the list of available HTC queues.
2. Then, the researcher constructs a task that can be submitted to the Campus Queue.
3. Then, the researcher logs in to the Campus Queue system and accesses a command line interface (CLI).
4. Then, the researcher uses the CLI to submit the task to a specific queue.
5. Finally, the researcher uses the CLI to check the status of the task and receive results when the task completes.

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

1. In Step 1, the list of available HTC queues is configured by the campus IT administrator and may include on- and/or off-campus systems, so long as they are available to campus researchers.
2. In Step 3, the login should use the campus's authentication service and the researcher's existing campus identity.
3. In Step 3, the login interface may require a remote login (aka terminal emulator) application on the researcher's local system.
4. In Step 4, researchers do not need to supply additional credentials to submit to off-campus queues.

CB-15: Develop an application or gateway that submits tasks to any of the high-throughput computing queues available to a campus

An application developer or science gateway developer (hereafter referred to as **"developer"**) needs to write an application that submits computation tasks to one or more of the high-throughput computing (HTC) resources available to his or her campus.

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

1. First, the developer visits the campus's research computing website and finds the Campus Queue section. This section provides a description of the Campus Queue and associated documentation, including an API (application programmer interface) and the list of available HTC queues.
2. Then, the developer uses the API documentation to write the application. The application's logic determines which queue(s) to use, which tasks to submit, and how to authenticate to the Campus Queue service.
3. Finally, the developer makes the application available for use by campus researchers and the application submits jobs to the Campus Queue as needed.

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

1. In Step 1, the API documentation should include information about authentication options. (The specific options should be based on the campus' research computing policies.)
2. In Step 2, if the application uses an identity other than the application user's identity for authentication to the Campus Queue, the campus's research computing policies should address the application developer's responsibility to properly observe the campus's IT/cybersecurity policies. For example, most campuses require that all uses of their research computing services be traceable to individual human users, which implies that the application developer must record or report the identity of the application's user when submitting a task to the Campus Queue.

References

[1] NSF Advisory Committee for Cyberinfrastructure Task Force on Campus Bridging. Final Report. March 2011. http://www.nsf.gov/od/oci/taskforces/TaskForceReport_CampusBridging.pdf. Available as print-on-demand book from: https://www.createspace.com/3597300

[2] Barnett, W., V. Welch, A. Walsh and C.A. Stewart. A Roadmap for Using NSF Cyberinfrastructure with InCommon. 2011. http://hdl.handle.net/2022/13024 or http://www.incommon.org/nsfroadmap.html. Available as print-on-demand book from https://www.createspace.com/3630011

[3] Barnett, W., V. Welch, A. Walsh and C.A. Stewart. A Roadmap for Using NSF Cyberinfrastructure with InCommon: Abbreviated Version. 2011. http://hdl.handle.net/2022/13025 or http://www.incommon.org/nsfroadmap.html

[4] Jim Basney, Terry Fleury, and Von Welch, "Federated Login to TeraGrid," 9th Symposium on Identity and Trust on the Internet (IDtrust 2010), Gaithersburg, MD, April 2010. http://dx.doi.org/10.1145/1750389.1750391

[5] http://creativecommons.org/licenses/by/3.0/ Attribution 3.0 Unported license (CC BY 3.0)

[6] Almes, G.T.; Jent, D.; Stewart, C.A. 2011. Campus Bridging: Data and Networking Issues Workshop Report. http://hdl.handle.net/2022/13200. Available as print-on-demand book from: https://www.createspace.com/3592681

[7] Dreher, P., S.C. Ahalt, G. Almes, M. Mundrane, J. Pepin and C.A. Stewart, (eds.), 2011. Campus Bridging: Campus Leadership Engagement in Building a Coherent Campus Cyberinfrastructure Workshop Report. 2011. http://hdl.handle.net/2022/13194

[8] McGee, J.; V. Welch; G.T. Almes. 2011. Campus Bridging: Software & Software Service Issues Workshop Report. http://hdl.handle.net/2022/13070

[9] Open Data Commons. ODC Public Domain Dedication and License (PDDL). http://www.opendatacommons.org/licenses/pddl/1-0/

[10] Scott Michael, Stephen Simms, W. B. Breckenridge, III, Roger Smith, and Matthew Link. 2010. A compelling case for a centralized filesystem on the TeraGrid: enhancing an astrophysical workflow with the data capacitor WAN as a test case. In Proceedings of the 2010 TeraGrid Conference (TG '10). ACM, New York, NY, USA, , Article 13 , 7 pages. DOI=http://dx.doi.org/10.1145/1838574.1838587

History

	Version	Date	Changes	Author
Entire Document	1.4	3/09/2012	Formatted	Stewart, Knepper, Grimshaw, et al.
Entire Document	1.5	3/14/2012	Hopefully penultimate version before public release	Stewart
Entire document	2.0	11/3/2016	Use cases 1-7 rewritten using XSEDE-2 format; added use case 8; new use cases 9-11 extracted from use case 4	Liming
CB-12 through CB-15	3.0	8/6/2018	Added CB-12 (drafted in 9/17) and CB-13 - 15 (7/18) from the XSEDE-2 XCRI program	J.E.Coulter, L.Liming

Entire document	3.1	7/28/2019	Removed unnecessary XSEDE terminology	L.Liming