Federated XDMoD Requirements

Date	Version	Person	Change
2016-04-08	1.0 draft	XMS Team	Initial version

Summary

Federated XDMoD will support the collection and aggregation of data from individually managed HPC centers into a single federated instance of XDMoD for displaying federation-wide metrics. Data particular to an individual center will be available by applying filters. Each participating center will deploy an XDMoD instance through which local data will be collected and viewed. The Client will ship data for selected resources to a central repository for aggregation and display by a master XDMoD instance. In essence, we will create a central repository to support the display of data collected from multiple organizations (Service Providers) including jobs, users, projects (allocations), organizational hierarchies, etc. Data from individual organizations will be mapped into a common format where appropriate. This is similar to what XSEDE has done with the XDCDB but **without a single, XSEDE-provided set of resources, users, and allocations defined in common across all of the organizations**.

Definitions

Term	Definition
Federation	A set of individually managed XDMoD instances providing data to a centrally-managed repository for collection and visualization via a single shared XDMoD instance that provides cross-federation metrics.
Service Provider (SP)	An individual organization maintaining its own local XDMoD instance and providing data to the Federation Repository.
Federation Repository	The central location where Service Providers will send their local data to be included in the federation. The Federation Repository is configured as part of the Federation Master.
Federation Master	The single XDMoD instance that provides metrics across all members of the federation. It may perform some normalization, disambiguation, and aggregation of data from multiple SPs taken from the Federation Repository.
Federation Client	An XDMoD instance running locally at an SP that is configured to collect, normalize, and disambiguate data local to the SP and send it to the Federation Repository.
Federation Administrator	An administrator responsible for maintaining the Federation Master and Federation Repository. This administrator will be responsible for elevating user privileges and approving/configuring member SPs.
Client & Master Data Federation Modules	The optional XDMoD modules to be installed allowing an instance to communicate with the Federation Repository as a client or a master.
XDMoD User	An individual with an XDMoD account (i.e., a user of XDMoD). A User does not necessarily map to a Person.
Person	An individual associated with data contained in XDMoD. For example, a person is associated with individual jobs or financial data.
Local Value Space	The local view of data provided by a particular Federation Client. Values presented in this space may be specific to the Client may not be able to be compared across Clients without normalization.
Global Value Space	This is the global view of federation data provided by the Federation Master. It has been normalized across all of the clients where necessary.

Single Sign On Service Provider (SSO-SP)	An XDMoD instance when within an identity federation is considered a service provider. This will be used to differentiate between an XDMoD SP and an Identity SP

Assumptions

The design of Federated XDMoD makes the following assumptions.

1. The federation will be tightly coupled. Members of the federation will have a vested interest in a working collaboration and will cooperate to manage items such as a common organization hierarchy, fields of science, and method for disambiguating people common to multiple sites in the federation. Federated XDMoD is not designed to map or disambiguate arbitrary information from members that do not conform to the federation-established guidelines.
2. Each SP will maintain their own set of local people, local usernames, local job identifiers, fields of science, organizational hierarchy, and other data. The process of federating this information will be handled by the Federation Master and any mapping will need to be agreed upon a-priori by the Federation.
3. Each SP who is a member of the federation will install and maintain their own local instance of Open XDMoD. This instance will collect data local to the SP and deposit it into the Federation Repository.
4. Federation Clients must request membership and be added to the federation by a Federation Administrator before they can begin sending data to the Federated Repository.
5. Each SP will maintain the appropriate version of XDMoD needed to properly report their data to the Federation Repository. If their version is incompatible, membership will be rejected.
6. It is possible for the same person to hold accounts at more than one SP, each with potentially a different local username. A mechanism will be provided for these people to be disambiguated at the federated level.
7. It is possible, especially in a federated cloud environment, for a job to both span multiple resources and to migrate from one resource to another (e.g., Aristotle cloud instances).
8. Not all SPs will be required to provide the same set of data. For example, a subset of SPs may provide low level job information while another subset may provide application kernel data. The Federation Master will display data for those SPs that provide the data and charts will be annotated appropriately so that it is clear to the user what they are viewing.

9. The XDMoD Federation Client running at each SP may be a Single Sign On Service Provider (SSO-SP) using a supported institutional Identityrovider (IdP) as well as local (non-SSO) XDMoD accounts. The Federation Master can provide authentication via each supported Federation Client’s configured IdP and will also support XDMoD accounts local to the Federation Master. Authenticating to the Federation Master using accounts local to a Federation Client is not supported.
10. Data federation will be enabled by two Data Federation modules that will be provided for XDMoD. These will allow a local XDMoD instance to ship locally collected data to the Federation Repository as well as allow a Federation Master to pull data from the repository.

Data Collection

Data will be collected locally at each Federation Client (SP) and transmitted to the Federation Repository, as shown in Figure 1. From there it will be ingested and aggregated into the local data warehouse of the XDMoD Federation Master where cross-federation data will be presented for visualization. Note that an XDMoD federation can have a **single** Federation Master for an unlimited number of Federation Clients while a Client can be a member of **more than one federation** (e.g., CCR DIBBS may be a member of the aristotle federation and the CCR federation). A client can configure one or more resources as part of a federation. *A single XDMoD instance cannot serve as both a Master and Client.*

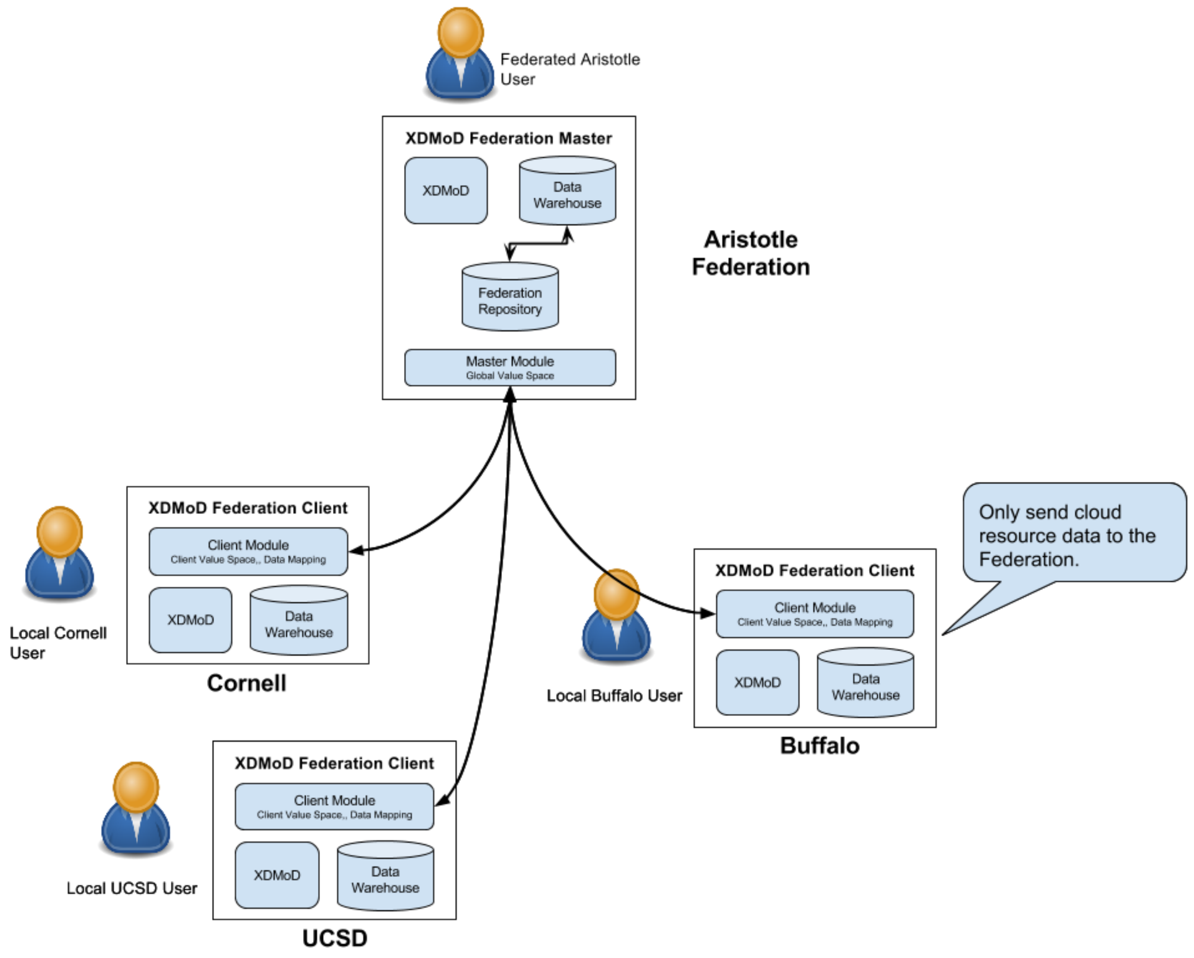

Figure 1: Example data flow for Aristotle Cloud Federation

Local XDMoD Installation

Each participating SP will install and maintain a fully-functional local instance of XDMoD and install and configure the Data Federation Client module. Data (jobs, people, local usernames, accounts/projects, FoS, organizational hierarchy, etc.) will be collected locally using whatever means appropriate for their specific configuration, and stored in the local XDMoD data warehouse. This local installation will be capable of displaying metrics for the SP. If an SP wishes to configure only the portion that collects data and not the web portal, that configuration will be supported as well. The Federation Master will also be a fully-functional XDMoD instance, however, rather than obtaining data from local sources it will obtain data from the Federation Repository. If the local installation (Client) has installed any optional XDMoD modules such as Application Kernels or XDMoD Value Analytics, these modules must also be installed on the Master in order for it to display information specific to these modules.

Module Support

In order to support federation of data collected and presented by a particular XDMoD module, the module must be installed on both the Master and at least one Client and *the module must* *be federation-aware.* This means that the module must specify which dimensions that it will make available to the global space and be able to map any keys in those dimensions from the Client's Local Value Space to the federation Global Value Space based on information provided by the Master prior to sending that data to the Federation Repository. In addition, the module ETL must be structured properly to allow the Federation Master to loop over the ingestion actions for all Clients that have sent data and ingest their data from the Federated Repository.

For example, a Client will have its own set of resource ids that are unique to the Client but not the federation. Prior to sending data to the Master Repository, the Client must be able to map from a local resource id to a global resource id. Another example is an organizational hierarchy. Clients will likely have their own local hierarchies while the Master will provide a normalized view of this data so the clients must map between their own Local Value Space to the federation's Global Value Space.
It should be noted that initially, only data stored in the data warehouse will be available for federation. This includes data from the Jobs, Allocations, and Accounts realms as well as *summarized* data from the SUPReMM realm, **but not job-level information from SUPReMM that is stored in MongoDb** since this data is not stored in the data warehouse. A future task will enable pass-through of job-level requests to individual Clients.

Data Federation Modules

Two optional modules will be provided to allow a local XDMoD instance to participate in a Federation: a Federation Client and a Federation Master module. These modules will facilitate data transfer and communication between Clients and Master. *These modules are mutually exclusive in that they cannot both be installed into the same XDMoD instance.*

Federation Client Module

The Federation Client module is installed into an existing XDMoD instance to allow data to be extracted from the local XDMoD data warehouse and shipped to the Federation Repository. It provides the following functionality

- Generate a request to the Federation Master for inclusion in the Federation. This request should include information such as the name of the Federation Client site, administrator email address, IP address of the Client for configuring access rules, the local resources to be included in the federation, and any other information deemed necessary for proper configuration.
- If a Client supports multiple resources, select which of the resources will participate in the federation.
- Ship data from the local Client data warehouse to the Federated Repository, performing mapping to a common format where required.
- Provide a mechanism for mapping or normalizing, where necessary, local data values to global values supported by the federation prior to sending it to the Repository.
- Based on configuration options, send only new or changed data to the Federated Repository.

- Send data to the Federated Repository that was added between a provided start and end date.
- Provide a status report of data that has been sent to the Federation Master

Federation Master Module

The Federation Master module is installed into an XDMoD instance to allow it to read SP data from the Federation Repository and import it into the Federation Master's data warehouse to provide metrics across the federation. Data will be presented to the user from the Federation Master's data warehouse. It provides the following functionality:

- Process a Client request and generate configuration information to enable proper configuration of the Client. Configuration information will include any normalization information such as resource or organization ids unique to the federation to use when mapping data prior to sending to the Repository.
- Create database schemas, accounts, and access control in the Federation Repository for each approved Client.
- Provide information to the Clients that will allow them to properly map data between local values and values supported by the federation. For example, each Client may have their own organizational hierarchy and will need to be able to map these values to the global hierarchy supported by the federation as a whole.
- Retrieve data from the Federated Repository and add it to the Master's local data warehouse, performing normalization as required.
- Provide a status report of data that has been received from each Federation Client.

Data Mapping and Normalization

Individual XDMoD Clients will configure their instances to collect and organize data relevant to their local needs. For example, resources will be locally named and an organizational hierarchy would be created based on the hierarchy used at the local institution. When displaying data for the federation as a whole, the Master will provide a global, unified view of the data. This will likely require some mapping of local values provided by the Client into the unified view presented by the Master. For example, the federation will decide on a global organizational hierarchy to be presented by the Master but Clients may have established their own hierarchies. A mechanism must be provided for the client to map data from their local values to the federation-supported values prior to uploading to the Master Repository.

XDMoD will provide a mechanism for the federation, through the Federation Master, to define a global view to be used when displaying values across the federation. This view will be made available to the Clients by the Federation Master module and the Federation Client module will contain a mechanism to perform the mapping prior to sending the information to the Repository. Multiple mappings will be supported since a Client can report to multiple federations.

Federated Repository

The Federated Repository is the central data repository that will be used to receive and collect data from the Federation Clients (SPs) and serve as the primary data source for the Federation

Master. The repository will be a single database instance, with each client depositing data into a separate schema to compartmentalize information. Schemas will be named appropriately to ensure there are no namespace collisions (i.e., federated_nics) with names controlled by the Federation Master. Rather than provide a REST API (and need to maintain that software stack), the repository will be accessed directly by both the Client and the Master XDMoD modules. While a separate database for the Repository is recommended, it is not required as long as there are no namespace conflicts.

Each Client will utilize a separate database schema to deposit their data and will authenticate using a user created by the Federation Administrator through tools provided by the Federation Master Module specifically for this purpose. The schema and username will be generated as part of the process of adding a new Client (see Configuration) and will be named based on the short name assigned to the Client. For example, a Client named “NICS” would deposit their data into the schema “federated_nics” using the user “xdmod_nics”. Data deposited into this schema must be in a known structure as defined by the Client Module.

Each Client will transfer data collected locally into their Federated Repository schema using a process defined by the Client Module to ensure that the data is in the correct format. Mapping may need to be performed by the Client in cases where local data does not match the organization or format supported by the Master (e.g., a different organizational hierarchy). The client will specify this mapping.

- People, local usernames (system accounts)
- Organizational hierarchy, Field of Science hierarchy
- Allocations/projects, PIs/project managers
- Resource definitions (types, specs, code, etc.)
- Job records (Master Records and Tasks)
- SUPReMM and Application Kernel data

The Repository, through the Federation Master, will define the following items. Federation Clients will be able to query this information to synchronize their own data structures, if desired, or create/update mappings from their own local data to the format supported by the Repository. The Client is responsible for providing a mapping between their own local data and the definition provided by the Master in the ETL action that transfers data from the Client data warehouse to the Repository.

- Master Field of Science definition
- Master Organizational Hierarchy definition
- Location information (state, country)
- Bucket/histogram information?
- Master list of organizations
- Master (disambiguated) person list
- Master list of types (resource type, account type, user type, etc.)
- List of service providers (clients)

- Resource metadata (types, specs, code, etc.)

Optional Modules

Optional XDMoD modules may introduce new realms or dimensions into the data warehouse. In order to support federation of optional data, the Master and at least one Client must have the module installed and the module must support data federation. Supporting federation of data generated by optional modules will be supported at a later date.

Federation ETL

Federation Clients

Clients will collect data based on the existing XDMoD ETL process. This typically includes the shredding of resource manager log files, querying cloud resource tools, and importing of CSV files. This data will be sent to the Federation Repository where it will be processed by the Master. Sending data to the Federation Master should be performed by a distinct ETL action that can be run independently of local ETL processes. Normalization and/or mapping of the data will be performed by this Client ETL action as needed. Data to be mapped includes:

- Organizational Hierarchy: The hierarchy is based on the organizational unit that a user running a job has been assigned to. **We do not currently support a user belonging to multiple organizational units, although this is a requested feature and may be handled via tagging to avoid double-counting.** The Master will support a single organizational hierarchy for the federation. A method will be developed that will allow Clients to map local hierarchies to the master. Since we assume a tightly coupled federation, we assume that SPs will work together to do this.
- Field of Science (FoS): The field of science for a job is not associated with a person, but is assigned through a project or allocation that the job was run under. A person may run jobs under multiple FoS. Mapping Client FoS data prior to sending to the Federation Repository will be handled similarly to the organizational hierarchy.

Federation Master

The Master will perform ETL on Client data in the Federation Repository to ingest and aggregate data across all Clients, adding this data to its own data warehouse and performing normalization of the data as needed. The Master will generate all metrics from the data in its own local data warehouse. Data to be normalized includes:

- Person disambiguation: The same person may have accounts at multiple sites in the federation (possibly with different usernames). A method must exist to disambiguate them.

The Master will perform ETL using the Federation Repository as its primary data source. When performing ETL from the Repository, the master will execute a loop over the ingestion ETL actions for each client and store the data in its own data warehouse. Following ingestion, a standard aggregation ETL can occur to populate the aggregate tables. This ensures that the Master is a fully functional instance of XDMoD containing all necessary data in the event that access to a fact table is needed. The Master will need to maintain enough information to properly execute ETL appropriately based on the installed modules and the data that each Client has sent to the Repository.

Federated vs. Local Authentication

The XDMoD Federation Client running at each SP may be a Single Sign On Service Provider (SSO-SP) using a supported institutional Identity Provider (IdP) as well as local (non-SSO) XDMoD accounts. The Federation Master can provide authentication via each supported Federation Client's configured and will also support XDMoD accounts local to the Federation Master mapping accounts to their disambiguated identity. Authenticating to the Federation Master using accounts local to a Federation Client is not supported.

Configuration

Ideally, adding a new Client to the federation should be as simple as creating a new Client request (a request and approval is necessary to avoid namespace conflicts) via a command-line tool and sending that request to the Master for approval. Upon approval the Master will generate a configuration template to be installed on the Client that will enable the federation. The configuration might include an approved name and metadata for the Client as well as information for connecting to the a destination host, username, and credentials. Note that the abbreviation assigned to a Client may not be the same as the one requested in order to keep abbreviations and other identifying information unique.

Use Cases

Use Case: Aristotle Cloud Federation

The Aristotle Cloud Federation (https://www.cac.cornell.edu/about/news/aristotle.aspx) is a project to build a federation of Eucalyptus cloud installations at Cornell, the University at Buffalo, and UC Santa Barbara. One goal of the project is to build an allocations and accounting model that will allow institutional administrators to track utilization across federated sites as well as allowing users to utilize resources at any of the three sites and migrate instances between sites. Each site will install a local copy of XDMoD along with the Client Module and the Federation Master will be hosted in the cloud. The Master will provide utilization metrics for the federated cloud as a whole as well as each individual cloud installations. It is expected that Aristotle will use inCommon to authenticate users at each site. Note that, as shown in Figure 1, Buffalo will have multiple resources included in its local XDMoD installation and only the DIBBS cloud data will be configured to be a part of this federation.

Use Case: NSF-funded Projects (Non-XSEDE)

NSF is interested in viewing utilization metrics for multiple NSF-funded HPC resources in a single instance of XDMoD. Currently, metrics for all XSEDE resources are available via a single XDMoD instance but there are other NSF-funded resources such as Blue Waters, Open Science Grid, and LIGO that are not part of XSEDE. Rather than requiring program officers to access separate XDMoD instances local to each resource, a single federated instance where data can be viewed across all of the resources will be made available.

Use Case: CloudyCluster

Clemson is working to develop CloudyCluster (http://www.cloudycluster.com/) an infrastructure allowing users to easily deploy an HPC compute cluster an AWS. It deploys complete with an included scheduler, compute infrastructure, and storage based on OrangeFS, EFS and S3, as well as a variety of popular Open HPC software. Initial discussion between XMS and Clemson generated some interest for building XDMoD into a CloudyCluster deployment, as well as providing a federated XDMoD where each CloudyCluster deployment would provide data to a central repository for Clemson to view and use to provide user support.

Future Features

1. Allow the Master to request that the client re-send data for a particular date range. This grants the Federation Administer the ability to possibly rectify issues with data transfer without needing to involve staff at Client sites.
2. Pass-through of SUPReMM job viewer requests from the Master to the Clients. Initially, we will not plan to support federation of non-summarized job data.
3. Allow the SP to act as an IdP for the Federation Master, allowing SP local accounts to be used to login to the Federation Master.

Implementation Details

Implementation details recorded from discussions among XMS team members.

Client Identities

When sending data to the Repository, can we use a multi-column key using the federation organization id for the client along with client-specific keys. For example, can a client resource_id be combined into a key (federation_client_id, resource_id) for unique identification?

Mapping Local to Global Values

Local values may need to be mapped to globally determined values. For example, the federation will decide on a global organizational hierarchy to be presented by the Master but Clients may have established their own hierarchies. A mechanism must be provided for the client to map data from their local values to the federation-supported values prior to uploading to the Master Repository.

ETL

The current plan is for Clients to send data (appropriately mapped) to the Federated Repository and for the Master to use the Repository as its primary data source. The Master will perform ETL ingestion actions to bring Client data into its own data warehouse where it can then perform standard aggregation actions. This will ensure that the Master is still a fully functional XDMoD instance in its own right, while incurring the storage overhead of the redundant data. If we were to attempt to aggregate directly on data in the Repository (spread across multiple Clients) we would need to join across the multiple schemas and possibly impose restrictions on the ETL actions.