

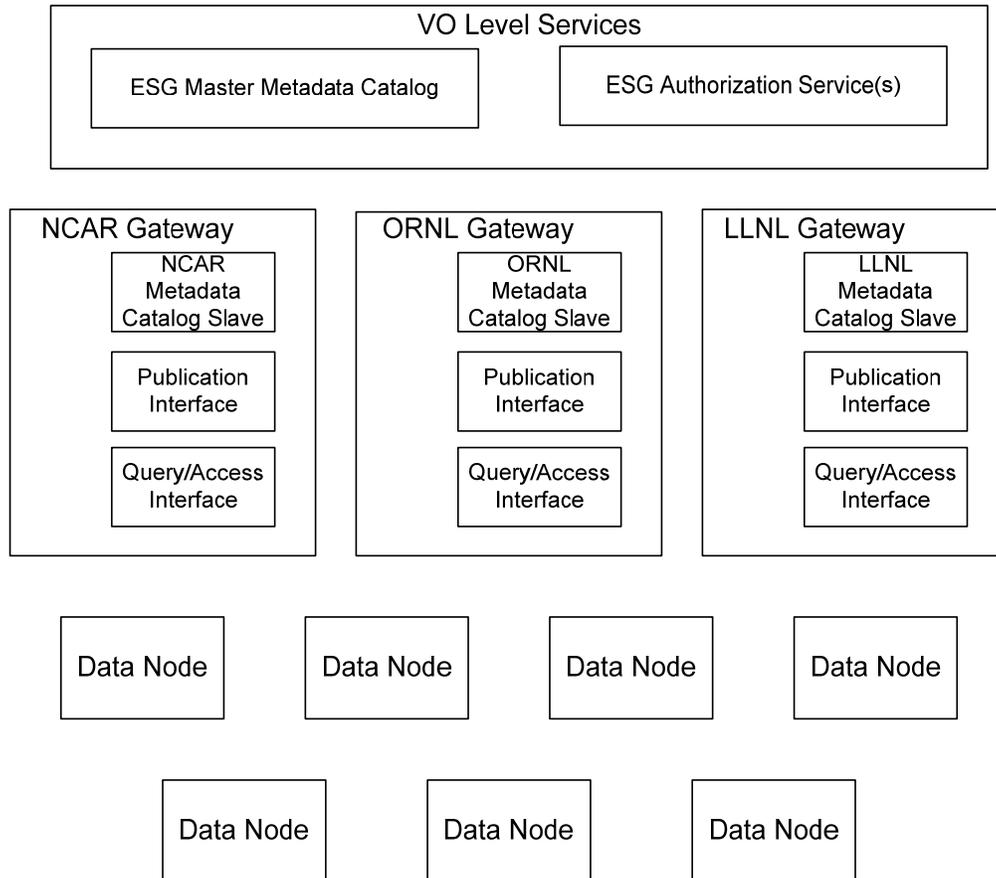
Data Federation Use Cases

Ann Chervenak, Frank Siebenlist

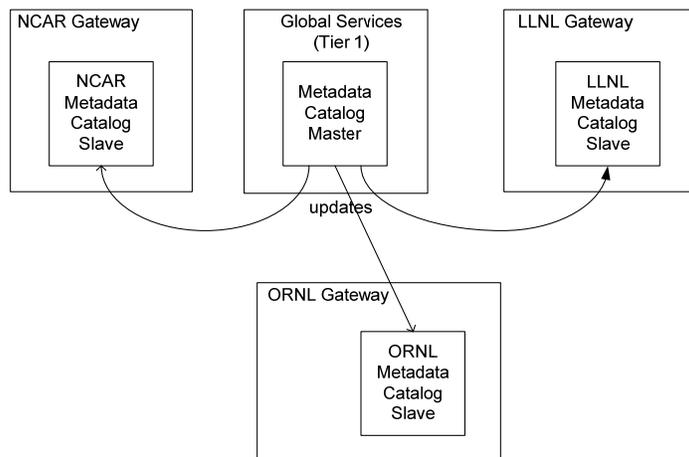
Background:

The federated ESG architecture consists of a set of ESG gateways and data nodes. There will be three Gateway located at NCAR, LLNL and ORNL. Each gateway will provide rich ESG services and sophisticated administrative support. Each gateway will provide access to a distributed and replicated metadata catalog and a query interface through which clients can request any ESG data set by name or by querying for metadata attributes. Gateways need to be fault tolerant, so that they continue to operate even if other gateways in the federated system fail.

The data nodes represent other institutions that contribute data sets to ESG, for example, institutions that publish IPCC. We assume these data nodes typically contain fewer services and need to be run with little administrative support. Data nodes will produce data sets, will be responsible for authoring at least some of the metadata for those data sets, and they may use ESG services to assist with this authoring. Data nodes are considered the authoritative source of the data and the associated metadata. Users at a data node must coordinate with an ESG gateway to publish data sets, as described below.



The federated metadata catalog architecture will include a single master metadata catalog that is hosted at the Tier1 or Global services layer of the ESG architecture. All updates to metadata must be performed on this master catalog. The contents of the master metadata catalog will be replicated periodically to each of the ESG gateway nodes. This allows users to issue metadata queries at any gateway. There will be some delay between when updates are made at the master metadata catalog and when these updates are propagated to the metadata catalog replicas at each gateway.



Use Cases

1. A user at a data node wants to publish a dataset into ESG.

A data publisher at a data node wants to publish an IPCC data set into the ESG collaboration. The initial data production and metadata generation happen at the data node. When the data set and metadata are prepared, the user authenticates himself with one of the ESG gateway nodes. The authorization system determines whether the user is allowed to publish data to ESG. If so, then the user interacts with publication services on the gateway node, where additional metadata may be produced or extracted. The data set is registered with the master ESG metadata catalog and with ESG replica catalogs, etc. After some period of time, information about the new data set propagates to all metadata catalogs in the federated system, and these new data sets can be discovered by clients at any ESG gateway.

Note that this relates directly to Bob Drach's publication use case:

A data provider stages a set of model runs. He runs a process that extracts the metadata from the data, then runs validation and quality control on the metadata to ensure it meets:

- ESG standards
 - project-specific standards
- for the project that generated the datasets.

The data provider authenticates with a gateway, giving him sufficient privilege to publish to the gateway metadata DB.

The data provider publishes to the gateway. Additional metadata not extracted from the base data is added manually. After a certain period of time, the data is visible and accessible from all gateways.

2. User accesses data sets via an ESG gateway node

A user authenticates to an ESG gateway, which is able to determine the user's group information and access permissions. The user makes a query to the identify data sets with particular metadata characteristics, and the gateway metadata catalog returns a list of the query results that the user is authorized to see. Next, the user can request access to some of those data sets. There are three alternatives for providing this access..

a) Data access through the gateway node

This is the scenario that Arie proposed, in which the gateway acts on behalf of the user, contacts the relevant data nodes to access the data sets that reside on each node, caches these data sets on the gateway, and returns the results to the user. This is the preferred trust model where the "ESG-Gateway-Identity" is the single authority that is trusted by all the nodes. This scenario requires that each ESG gateway knows about and has access permissions on each of the data nodes.

b) Data access at each data node with authorization credentials

An alternative scenario is that after receiving a list of data sets that match the query and their locations, the user is responsible for contacting each data node and requesting data access from that node directly. This scenario would probably use a Community Authorization Service (CAS) model, where the ESG Gateway issues authorization assertions for each user. When a user makes a data request at a gateway, the user presents its CAS credential that specifies the user's access permissions. The data node should then grant authority to the data sets according to the permissions specified in this credential.

c) Data access at data node with call-outs

An alternative but trust-equivalent scenario is where the data nodes call-out to the ESG Authorization Service to ask for access control information for users making requests at the data node.

From a trust perspective, all three access models are equivalent, and the deployment choice should be based on use-case requirements. We may have to support all three models.

Note this relates directly to the scenario that Arie identified earlier:

A user comes to a gateway, and wishes to login using his login name and password. The gateway checks with the ESG login service that the user is an ESG user and the group he belongs to. This can be done either locally or accessing login service remotely – a question yet to be decided. Accordingly, the user is allowed to browse and explore the metadata (stored locally) he/she is allowed to see. The user makes a selection of one or more files. Now, the user cannot get the files directly from the corresponding nodes because he has no security credentials that nodes will honor. Thus, the gateway will get the files into its managed cache (similar to what is done now) using the gateways credential (but adding to it the identity of the user). In this scenario, we assume that each gateway is able to contact all nodes and that a single request can be from files from any nodes (an open issue). The user then pulls the files from the gateway. One advantage of this scenario is that files that end up at the gateway cache can be accessed directly from the gateway by subsequent users who need them if they are still in the gateway cache. This can help reduce the load on nodes that are not rich in resources.

3. User accesses via a downloaded ESG client tool

Use case from Arie Shoshani: We assume here that the user downloaded an ESG client tool. For the sake of discussion, we assume it is an enhanced version of DML. Now, the user logs into an ESG gateway, browses and/or searches the metadata, and makes selections of some files as in Use Case 2. However, in this case the gateway prepares a proxy, as well as a DML-file of the desired files to be downloaded, and gives the user a reference token. The user invokes DML with the reference token. The DML contacts the gateway, gets the proxy and the DML-file, and proceeds to get the files directly from the nodes using the gateways proxy. The DML can also get some or all the files from the gateway if the gateway still has

them. This is particularly useful for remote files being brought into a local gateway, and accessed by multiple users local to the gateway.

4. A new data node joins the ESG federation

When a new site (e.g., an IPCC institution) wants to join the ESG federation as a data node, a process will be required of mapping the metadata ontologies, data models and authorization permissions used at the data node to those of the ESG federation so that the new data node can interoperate with the federation. In particular, the user groups and authorization policies used on the data node must map to those used in ESG. The data node will need to delegate its authorization capabilities to the ESG Authorization Service or CAS. Finally, the data node will want to make sure that its access policies are reflected in the access permissions that are supported by the ESG Authorization Service.

The new data node must register its users with the ESG VO services via the user registration and management use cases described in a separate document. If all ESG gateways need to know about all data nodes (see scenario 2a above), then the ESG VO services need to register the new data node with ESG gateway nodes. To publish into the ESG collaboration, the new data node needs access to the ESG Gateway data publication services. It needs to publish its data sets with appropriate metadata and set up access permissions for these data sets with the ESG Authorization Service. For data access, the new data node needs to allow users who are authorized by the ESG federation to access data sets.

5. A data node leaves the ESG federation

When an existing site leaves the ESG federation, should all the data sets that were published by that data set also be removed? This would require updating the master metadata catalog and the appropriate replica catalog. Eventually, these updates would propagate to all metadata and replica catalogs. It would require removing any cached copies of the data sets.

6. A new ESG gateway joins the ESG federation

A new gateway node will need to run a minimal set of ESG Gateway services, including hosting a replica of the ESG metadata catalog and providing interfaces for data publication, discovery and access, and user authentication/authorization via the ESG Authorization Service. Clients may also run additional specialized services for their clients. Under scenario 2a above, the gateway needs to be informed by the ESG Virtual Organization about all existing data nodes in the federation and get authorization rights on each of those data nodes to access their data sets.

7. An ESG gateway leaves the ESG federation

If a gateway leaves the federation, its access permissions on data nodes should be removed by each data node. All metadata catalogs, replica catalogs, and cached data on the ESG gateway node should be removed.

8. User modifies data sets already available in ESG

When a user at a data node modifies or replaces a data set that has already been registered in the federated ESG system, several actions must occur. The user making the modification needs to authenticate to a gateway, as usual. Any copies of the data set being modified that are cached on gateway nodes must be invalidated so that old data will not be mistakenly accessed. Master metadata catalog and replica catalogs must be updated to reflect the data modifications, and these changes eventually propagate to all gateway metadata catalogs and to replica catalog indexes. One open question is what happens to the old data files. Are data files immutable once written, so that modifications result in new versions of data sets? This is easier to implement than a version in which data files actually change, and pointers to old data sets might result in accessing the wrong data.

This use case relates to scenarios from Bob Drach:

Data Replace: The data provider replaces an existing file, set of files, or aggregation. In some cases, the filenames do not match the original, however the metadata in the files is sufficient to identify the data to be replaced. In the case of spatio-temporal data, this information may include time ranges. In particular, the time ranges may not match the original data. The replace operation is reversible by default, and may be made permanent by the data provider.

9. User modifies the metadata associated with a data set already available in ESG

When a user (typically at a data node, where authoritative copies of data and metadata reside) modifies the metadata associated with a data set, but not the data set itself, the master metadata catalog must be updated to reflect these changes. The user authenticates to an ESG gateway as usual and must have permission to publish/modify metadata. The ESG gateway allows the user to update the master metadata catalog. When updates are complete, the changes need to be propagated to the replicated metadata catalogs at each ESG gateway node as soon as possible so that the old metadata associated with the data sets can be invalidated to avoid getting stale query results based on this obsolete metadata.

This use case relates to a scenario from Bob Drach:

Metadata Update: Data provider modifies the metadata for a file, set of files, or aggregation, but does not modify the underlying data itself. He publishes the modifications to the gateway.

10. User deletes data sets that are available in ESG

When a user that has previously published data sets (e.g., an authorized publisher from a data node) deletes an existing data set, several actions are required. The master metadata catalog and appropriate replica catalogs must be update to delete or invalidate all references to the data sets. These updates are eventually propagated to all the replicated metadata catalogs and to the replica catalog index services. Also, if there are cached copies of the data sets on any of the ESG gateway nodes, these copies must be invalidated or deleted.

This use case relates to a scenario from Bob Drach:

Data Delete: Data provider removes a file, set of files, or aggregation. The operation is recoverable by default, although the data provider may choose to make it permanent. He publishes the deletion to the gateway. After a period of time, the changes are visible on all gateways.