

Preservation Flow

Summary

The One To Many (OTM) Specification defines two APIs to support communication between digital content repository systems (Repository) and distributed digital preservation systems (DDP) in order to facilitate depositing, deletion, restoration, and auditing of digital content managed by one or both systems. The APIs defined are the [OTM Repository Gateway API](#) (Gateway) for the Repository and the [OTM Bridge API](#) (Bridge) for the DDP. The Gateway and the Bridge APIs handle intermediary communication between the Repository and DDP and allow each system to operate without any knowledge of the internals of the other system. Each API is designed to facilitate deployment either as part of or extension to the Repository (in the case of the Gateway) or the DDP (in the case of the Bridge) or as a stand-alone application. They each provide an HTTP-based approach for authentication, communication, and data transfer.

The descriptions and diagrams below reference the [OTM Bridge API Specification](#) and [OTM Repository Gateway Specification](#) and are intended to capture the context in which the API calls are expected to be used.

Initialize

Flow

1. An agreement is reached between a repository owner and DDP system that will allow repository content to be deposited into the DDP; appropriate SLA/MOU and other legal documentation is signed and arrangements for billing/invoicing are made
2. The DDP administrator calls the Bridge **Add Account** endpoint to add the repository to the Bridge system and generate the credentials needed for the repository's Gateway to connect to the Bridge
3. The DDP administrator provides the Bridge credentials to the Gateway administrator
4. The Gateway administrator enters the Bridge credentials into the Gateway and the Gateway calls the Bridge **Register** endpoint to provide the Bridge with the details necessary to make calls back to the Gateway

Notes

- Several steps in this flow are necessarily outside the scope of software but are included for completeness

Deposit

Flow

1. The Repository administrator selects a set of objects to be deposited
2. The Repository calls the Gateway **PUT Object** endpoint once for each object to be deposited; this starts the deposit process
3. The Gateway resolves each object into a set of files to be deposited; each file is either copied to the Gateway staging storage area or a link to the file is captured to allow transfer to the Bridge
4. The Gateway calls the Bridge **Deposit Content** endpoint using the object ID as the filegroup identifier and providing an identifier for each file to be deposited
5. The Bridge initiates a deposit action for each filegroup in the deposit request
6. For each file in each filegroup the Bridge calls the Gateway **GET File** endpoint to transfer the file to the Bridge staging storage location
7. As each file transfer into the Bridge staging storage completes, the Bridge compares the checksum of the transferred file to the checksum provided in the deposit request; any mismatches trigger a re-transfer
8. Once all files in a filegroup are in Bridge staging storage and all checksums are validated, the status of the deposit is updated to "STAGED_FOR_DEPOSIT"
9. The DDP calls the Bridge **List Deposits** endpoint on a regular schedule to check for new deposits in the "STAGED_FOR_DEPOSIT" state
10. For each staged deposit in the Bridge the DDP copies the files from Bridge staging storage into the DDP ingest pipeline and performs a deposit (and replication)
11. When the deposit into the DDP is finished, the DDP calls the Bridge **Complete Deposit** endpoint to inform the Bridge that the deposit is complete
12. The Bridge clears the files associated with the completed deposit from Bridge staging storage
13. The Repository administrator checks the object status in the Repository; the Repository requests information about the object from the Gateway to provide information.

Delete

Flow

1. The Repository manager selects an object to be deleted from preservation storage
2. The Repository calls the Gateway **DELETE Object** endpoint for the object to be deleted
3. The Gateway resolves the object into a set of files to be deleted
4. The Gateway calls the Bridge **Delete Content** endpoint with the list of files to be deleted
5. The Bridge initiates a delete action for all files in the delete request
6. The DDP calls the Bridge **List Deletes** endpoint on a regular schedule to check for new delete requests
7. The DDP performs a delete on each requested file; when all deletes are completed, the DDP calls the Bridge **Complete Delete** endpoint to inform the Bridge that the delete is complete

8. The Repository administrator checks the object status in the Repository; the Repository requests information about the object from the Gateway to provide information.

Restore

Flow

1. The Repository manager selects an object to be restored from preservation storage
2. The Repository calls the Gateway **POST Object Restore** endpoint for the object to be restored
3. The Gateway resolves the object into a set of files to be restored
4. The Gateway calls the Bridge **Restore Content** endpoint with the list of files to be restored
5. The Bridge initiates a restore action for all files in the restore request and creates a directory in Bridge staging storage for the restored files
6. The DDP calls the Bridge **List Restores** endpoint on a regular schedule to check for new restore requests
7. The DDP copies each file in the restore request to the specified directory in Bridge staging storage
8. When all files have been copied into Bridge staging storage the DDP calls the Bridge **Complete Restore** endpoint to inform the Bridge that the restored files are available
9. The Bridge validates that all file checksums match the checksums provided in the restore request (when checksums are provided)
10. The Bridge updates the status of the restore action to "STAGED_FOR_RESTORE"
11. The Gateway calls the Bridge **Restore Status** endpoint on a regular basis to determine if the status of the restore is "STAGED_FOR_RESTORE"
12. The Gateway calls the Bridge **Get Restored Content** endpoint for each file in the restore request and stores each file in the Gateway staging storage
13. The Repository calls the Gateway Get Object endpoint and pulls the content into repository storage
14. The Repository sends a notification to the Repository manager that requested the restore

Audit

Flow

1. The Repository manager selects an object and requests a preservation audit history
2. The Repository calls the Gateway **GET Object Audit** endpoint for the object
3. The Gateway calls the Bridge **Get Audit History** endpoint, specifying the object ID as the filegroup identifier
4. The Bridge gathers audit data for the given filegroup and associated files from its internal data store and responds to Gateway with the requested audit history data
5. The Gateway translates the Bridge audit data into a format familiar to the repository and responds to the Repository request
6. The Repository displays the audit data to the Repository manager