

Distributed Digital Preservation with Samvera

The One-to-Many Grant

Problem Statement

Local repository systems are active: DSpace, Fedora, Hyrax

Distributed Digital Preservation systems (DDPs) are fairly static and often have limited versioning capabilities:

Chronopolis, APTrust, LOCKSS

Information about the data in the DDP (location, audit) is not captured in the local repository space with the rest of the metadata.

Goals

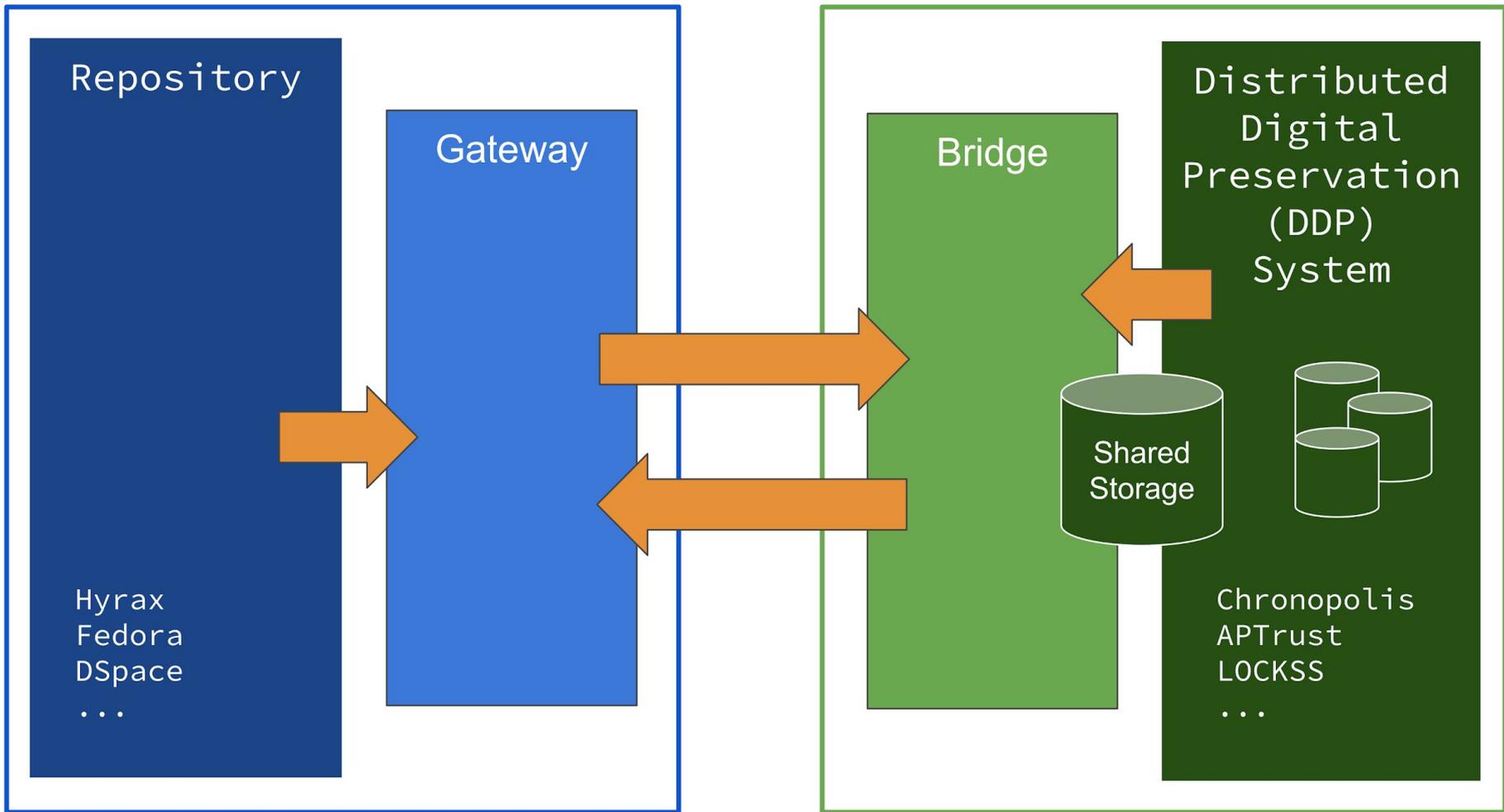
- 1) To define the development work needed to integrate local repositories and DDPs
- 2) To define requirements for an interface for curators to send digital objects from their local repository to a DDP
- 3) To define the requirements for version information and tracking of data sent to a DDP service
- 4) To ensure that the created definitions, specifications, and design documents are applicable to other digital repository software and DDP services

Use Case

UC San Diego sends about **45 TB** of data from its local repository to Chronopolis quarterly.

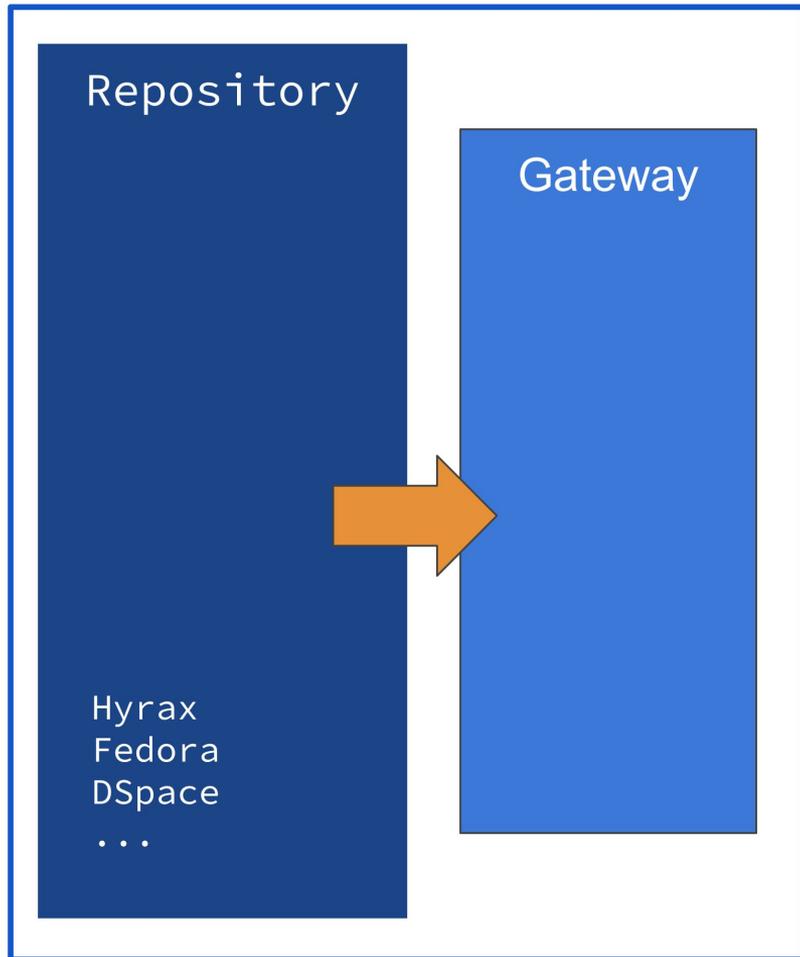
This currently takes about **30 days**, almost two thirds of which is for data packaging (tarring, bagging).

Actual changes to the entire data set tend to be minor - metadata changes or new data added. Usually only a max of 6 TB difference.

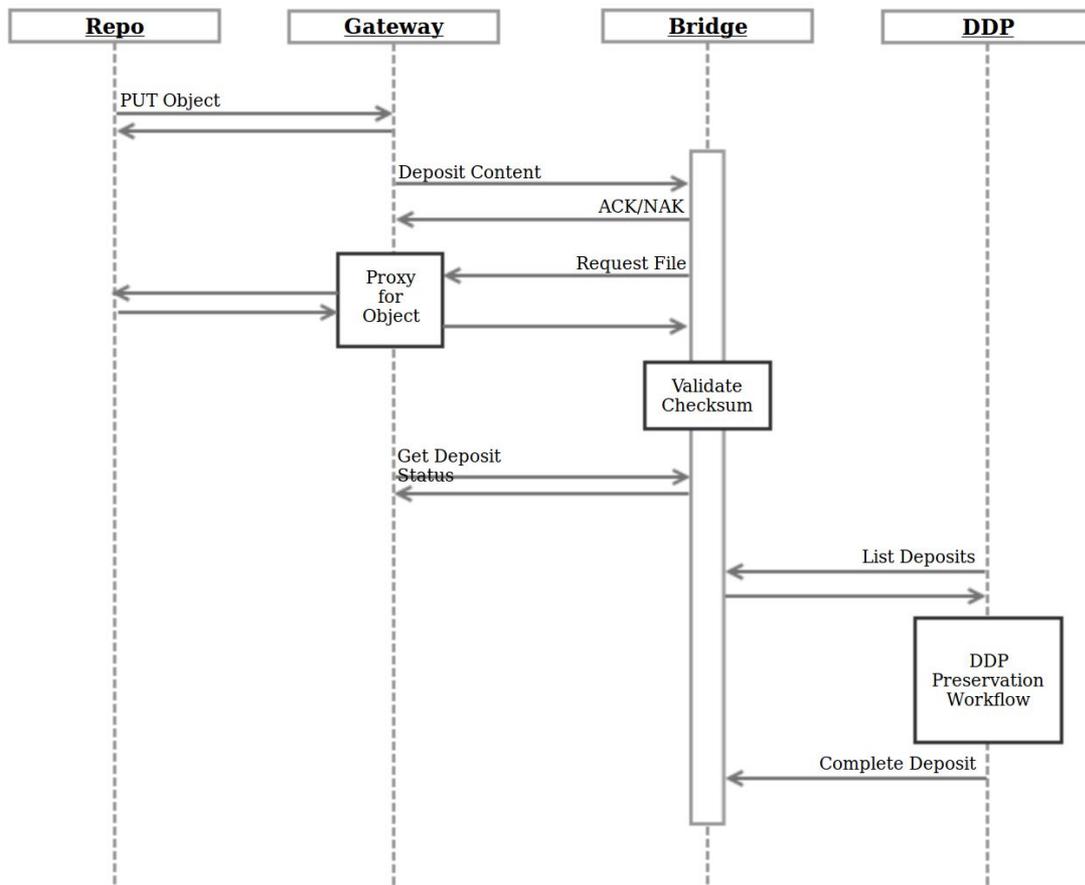


Gateway

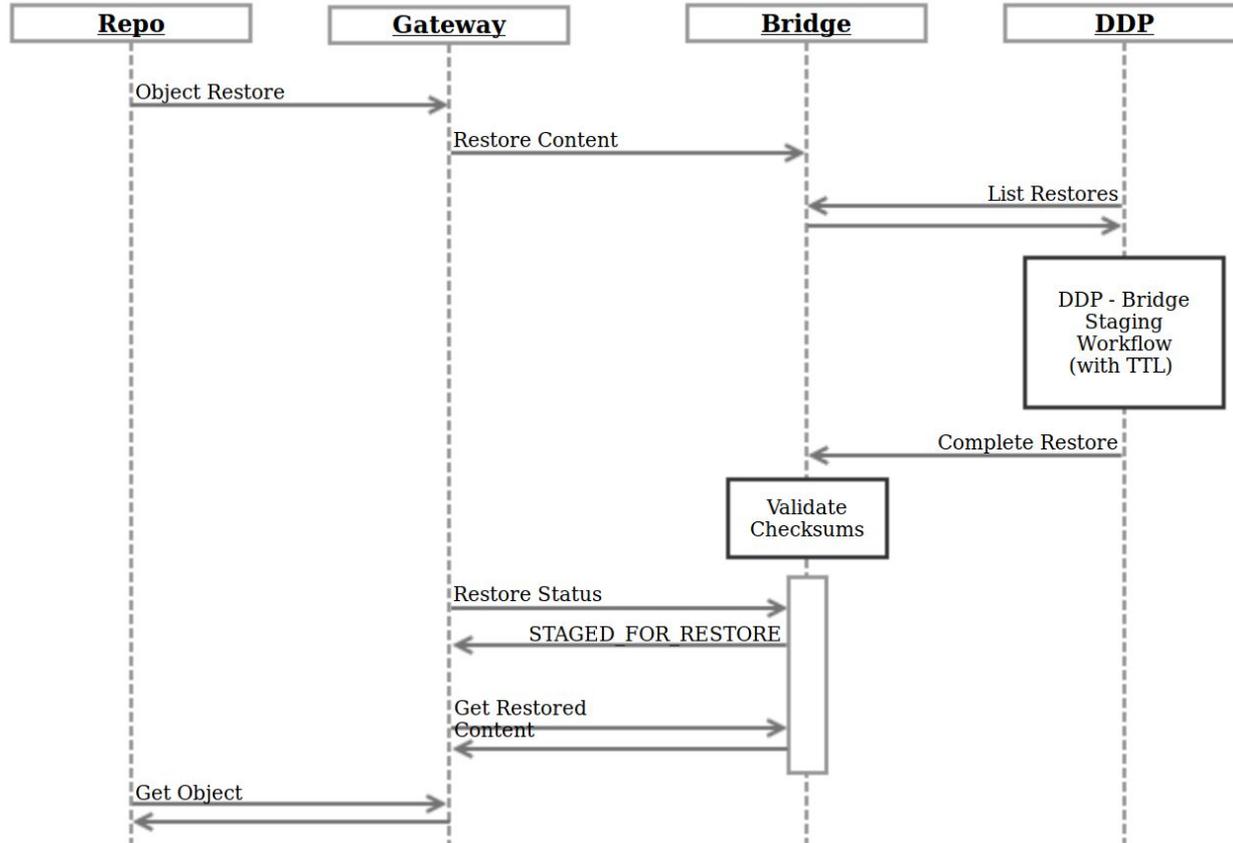
- Synchronous API for asynchronous actions.
- Translate repository semantics (“Objects”, “versions”) into storage concepts
 - Accepts BagIt bags
 - Sends manifest/exposes *files* to Bridge/DDP
- Deploy close to Repository
 - Minimize data transfer latency/cost
 - Repository owns deployment/data/logs
- S3 superset API on Repository side
 - Repository implementations can interact with object stores (and Glacier/Deep Archive)



Deposit Content



Restore Content



Repository Stories

Capture these features/functions:

- Send all your data to a DDP or multiple DDPs
- Send partial amounts of data to a DDP or multiple DDPs
- Configure workflows for when data should be sent
- Send versions of data
- Send updated metadata
- Remove particular objects or files from the DDP
- Retrieve/restore data from the DDP
- Tracking audit info from the DDP - if fixity checks were OK, etc.

Hyrax Workflows

Use Hyrax/Sipity workflow engine for configurable preservation actions, approval, notifications.

Progress & Next Steps

- Progress:
 - User stories finalized
<https://wiki.duraspace.org/display/OTM/User+Stories>
 - Overall architecture determined
 - UI wireframes drafted
 - Specifications in draft
- Next Steps:
 - Complete Specifications
 - Finalize UI wireframes
 - Planning for development phase (follow-on grant)
- Grant wrap-up Q1 2020

Teams

Core Team

- Sibyl Schaefer (UC San Diego)
- Jessica Hilt (UC San Diego)
- Mike Ritter (University of Maryland)
- David Trujillo (UC San Diego)
- Andrew Woods (Lyrasis)
- Bill Branan (Lyrasis)
- Tom Johnson (UC Santa Barbara)
- Rosalyn Metz (Emory University)

Advisory Team

- Tim Marconi (UC San Diego)
- Erin Glass (UC San Diego)
- David Minor (UC San Diego)
- Andrew Diamond (APTrust)
- Collin Brittle (Emory University)
- Brendan Quinn (Northwestern University)
- Tom Wrobel (Oxford University)

Questions

Project goals, user stories, and draft specifications available at: <https://wiki.duraspace.org/display/OTM>

Questions/Comments? Please contact:

Sibyl: sschaefer@ucsd.edu

Or Tom: tomjohnson@ucsb.edu

This grant project was made possible by funding from the ***Andrew W. Mellon Foundation.***