Bamboo phase two eighteenth-century use case - BHB
DRAFT

Summary
Allow users to compare and correct texts from ECCO and the HathiTrust Digital Library, share their corrections, and use analytical tools on the resulting editions.

Notes
Identifying the "best" digital surrogate of a volume from across several collections that have individually scanned and transcribed the work is a common task for scholars working with digital texts from the eighteenth and nineteenth centuries. In this example TextShop adds value over the current TypeWright deployment in three areas:

1. The ability to pull texts from multiple repositories, compare them, and choose the cleanest one (or possibly to choose between the sources at the level of the page).
2. The ability to apply analytical tools to the editions without a lot of manual munging.
3. Tracking of metadata about provenance and operations performed on the texts.

Whether users would have the ability to compose a "Frankenstein" edition by selecting the best transcriptions at the page level from among the surrogates depends on e.g. our discussions with the TypeWright team.

Workflow Steps
1. Create a set of references (using Zotero bookmarks) to multiple digital surrogates of a particular volume in ECCO and the HathiTrust Digital Library.
2. Authenticate to a Bamboo RE.
3. Retrieve the objects from the external repository into the CI HUB cache and inspect them.
4. Create a location in the local object store in which to place the objects.
5. Copy the object from the CI HUB to the local object store.
   a. Provenance metadata are created and the stored in datastreams on the object.
6. Estimate the transcription error rates for the surrogates using a service developed by TypeWright and deployed on the SSP.
7. Select the best text (or compose a new edition by selecting the best version of each page).
8. Perform any transformations on the resulting object prior to editing, in effect, creating derivative objects.
9. Open the text in TypeWright (the degree to which this is an export process depends on how closely TypeWright is integrated with the local object store).
10. Collaboratively correct the text in TypeWright with other users of the same RE.
11. Save the corrected text in (or re-import into) the local object store.
12. Perform morphological annotation and named entity recognition on the text using MorphAdorner and OpenNLP services running on the SSP.
13. Inspect the results.
14. Open the text in one of the VEP tools for analysis and visualization. This may include associating commentary, visualizations, or other annotations with the object.
15. Create a content package that includes the original version retrieved from the repository, the revised version, change logs, annotations, visualizations, and provenance metadata, all of which are recorded in a manifest.
16. Export the package to 18thConnect for permanent hosting (which includes metadata indexing by the 18thConnect COLLEX instance).

Bamboo phase two capabilities supporting the use case

Integration with TypeWright
This includes both services (error estimation) and application (collaborative interface for transcription correction).

Annotation
Annotations are stored and exposed for querying.

Modeling
- content package: This is a bundle of resources with a manifest for transmitting a curated object to a source repository for (re)deposit.
- provenance: Retains references to objects and tracks provenance between revisions.
- annotation bodies: Structure bodies to support mining.
- curation logs: Structure logs to support mining.

CI Adapters and Hub
One-way adapters for ECCO and HTDL. Some kind of facility for exporting the content package metadata to Collex (which probably would not be through the CI Hub).

N-tier IAM
The use of texts from the ECCO repository must be restricted to users with the appropriate institutional affiliation. Users should also be able to restrict access to texts they are working on in the local object store to specific groups of collaborators.

**Discussion**

In what senses is this a narrow use case? And how does being narrow in these senses reduce scope?

1. **Integration with a relatively mature tool.** TypeWright is known to work well with ECCO texts.

**Building on the use case**