DRS2 Public Search & Object Delivery APIs

Spencer McEwen, Library Technology Services

Goal

• Create a unified entry point into DRS2. DRS2 introduced an architecture whose capabilities are not being fully taken advantage of. A much larger amount of metadata is generated, stored, indexed and made searchable. Current DRS1 delivery applications have been modified to work with DRS2, but they are silos, specific to the file formats that they understand. This project will create a single application and interface utilizing modern application frameworks for searching and exploring DRS2 content, leveraging the service oriented architecture, as well as other features designed into DRS2. A public API for access to the search index and metadata and content will also be created.

Create

- Robust search that leverages the existing DRS2 Solr index.
    - The DRS2 search index is extensive. This project will create a public search interface, exposing the 150+ searchable fields for discovering objects and files. This will include searching of metadata as well as the full text contents of known formats. It will replicate the current FTS (Full Text Search service) API, making it a drop in replacement for FTS.
- Basic delivery of objects, content files, and metadata utilizing the internal service API provided by DRS2.
    - The major shift in DRS2 was introduction of complex managed objects - structures that aggregate multiple individual files into a single, more easily managed structure. This project will provide an interface for exploring DRS2 content from the perspective of the object. Navigation of an object, its metadata, files, and relationships will be possible.
    - Simple content viewers for specific file formats:
        - Images
        - Audio
        - Video (maybe?)
        - Text
        - Documents
        - Generic file view for formats not displayable in the browser
- Public API so that others can more easily build applications
    - This project will expose the numerous internal REST APIs provided by the core DRS2 Services layer and make them publicly accessible. Metadata, files, and entire objects will be accessible, allowing content owners, researchers, and the public to build applications utilizing the data.

Other Benefits

- This project would provide a centralized location for making content delivery mobile friendly.
- It would create a centralized point for logging access to DRS content and enforcing access controls. Logging features could be designed into the system for more detailed information about which objects belonging to which owners are accessed.
- Access controls:
    - AMS and pin will restrict access to non public content. DRS2 Premis rights restrictions will be enforced. For example embargos, max image delivery sizes, and stream only flags will be used. Secured content would not be searchable or viewable as well.
- There would be a feature to get an embeddable/permalink link or URN for displaying content in other sites
- Views for technical (MIX, TextMD, DocumentMD, AES57), administrative, and descriptive metadata (MODS) - possibly reuse or create new stylesheets for reuse in other systems.
- If images have geospatial coordinates they can be used.
- Examples of future objects:
    - Complex aggregate objects comprised of multiple sub objects of different types can be prototyped
- If fully adopted this application could replace several aging DRS delivery applications and become a key component of the Harvard Library infrastructure, while also reducing application maintenance and administrative overhead:
    - PDS
    - IDS
    - FDS
    - FTS
    - SDS

Resources

I admit that this is a large and ambitious project. Each of the 3 major components (search, delivery, public API) are almost complete projects on their own. I don't have enough insight at this time to estimate the number of resources required or duration of the project. With the end of the core implementation of DRS2 almost in sight I'm unsure how much time I would be able to contribute to development. I'm also not a front end GUI designer. I have general ideas of how content delivery could look, but a programmer with solid web design skills would benefit the project greatly.