RESEARCH DATA ALLIANCE

PID Collections: Perseus Use Case

Bridget Almas, Frederik Baumgardt
Perseus Digital Library, Tufts University

research data sharing without barriers
rd-alliance.org

Perseus Use Case

- Data Management of dynamic collections of texts, linguistic/scholarly annotations, commentaries, assertions, images, artifacts, bibliographic records …
  - Primary data object types include URIs, XML, Image Binaries, RDF Triples, Named Graphs, Plain Text, HOCR,
- Primary requirements
  - PID Assignment
  - Technology independent identifier strategy (support for URNs as well as URLs)
  - PID resolution
  - Throughout the lifecycle of a data object (pre-publication, published, updated, replaced, retired)
  - Grouping of items into semantically meaningful collections by data type *and* metadata characteristicsi
  - Ability to define citation *capabilities* by collection
    - Status
    - Fragment
    - Query
  - CRUD + L operations on objects within a collection
  - Ability to associate and update provenance metadata with collection items
  - Ability to template collections and collection items for patterns of reuse
  - Ability to validate adherence of collection metadata to a template or schema
  - Support for controlled vocabularies
  - Ability to perform batch operations on items within a collection
  - AAI support for item consumers and producers
  - Ability to search, filter and aggregate items within and across collections

Perseus Use Case – Fragments?

- We would like to be able to manage and cite these data objects not only as a whole objects, but also at the level of a fragment of a data object.
- Operations on fragments of data objects necessarily differ depending upon the data type
  - we want to cite a ROI on an image, an XPath or Xpointer into an XML file, a specific triple or set of triples in an RDF graph, etc.
  - and within a given types, there are often multiple accepted “standard” approaches to citing a fragment
  - some objects will support read operations on fragments but not write
- Should a collections API support generalizable approach to managing CRUDL operations on fragments ?

What do we want to get out of this WG?

- An abstract definition of an API for data management operations on Collections and Collection Items that meets the previously identified requirements
- A robust, open-source RESTful implementation of this API that is used and sustained by a multi-disciplinary community of users
- The ability easily create, populate and manage new collections, regardless of data type, in a standard, repeatable fashion
- The ability to more easily share collection items with other projects and communities and invite contributions to these datasets