DSpace 2.0 Modelling Services

Hack This Document!

Introduction:

Corrections, Comments, Opinions and below are welcome. The goal of this document is to begin a dialog outlining the service base of DSpace+2.0 and our expectations on those services. These services were outlined in the June 25th Funded work meeting. The list below is a tentative set of service boundaries to take into concern.

What are Services?

The concept of "services" used here refers to the "Service Locator" Design Pattern. Something that is used in many Enterprise Java Applications and a prevalent pattern in the Sakai Framework.

References

1. "CodeProject: A Basic Introduction On Service Locator Pattern. Free source code and programming help," http://www.codeproject.com/KB/architecture/Service_Locator_Pattern.aspx.

2. "Core J2EE Patterns - Service Locator," http://java.sun.com/blueprints/corej2eepatterns/Patterns/ServiceLocator.html.

3. "Design Patterns: Service Locator," http://java.sun.com/blueprints/patterns/ServiceLocator.html.

4. "Introduction+to+Sakai+and+Sakai+Services.ppt," http://bugs.sakaiproject.org/confluence/download/attachments/32712/Introduction+to+Sakai+and+Sakai+Services.ppt?version=1.

5. "Inversion of Control Containers and the Dependency Injection pattern," http://martinfowler.com/articles/injection.html.

6. "SakaiDevPractice.ppt," http://bugs.sakaiproject.org/confluence/download/attachments/27777/SakaiDevPractice.ppt?version=1.

7. "ServiceLocator Pattern: Does EJB 3 Really Kill It Off?," http://www.javalobby.org/articles/service-locator/.

8. "Session Facade, meet Business Delegate and his friend Service Locator... « ranting without the raving...," http://etweb.wordpress.com/2007/09/27/session-facade-meet-business-delegate-and-his-friend-service-locator/.

Outlined Service Domains:

Content Storage (Assetstore, Content/Read, Content/Write):

Includes aspects of storing and retrieving content files from a DSpace repository.

Criticisms about 1.5 (MRD):

- Current DSpace Asset-storage limits the ability to select the storage solution appropriate for a specific use case within the system (User and Collection centric content, Versions of Content)
- Assetstore is not a Plugable service
- Metadata cannot be attached or associated directly to Assets

Proposed Solution in 2.0 (MRD):

- Support Stackable/Plugable Assetstore Providers that can be selected/configured on a case by case basis (Local User non-persistent store (workspace), local store for resources related to presentation (logos, css theming etc), local store for caching derivative transformations (MediaFilter, text extraction, etc).
- Utilize ObjectIdentifiers (UUID's) to reference Content in Assetstore and Resource representation in Metadata storage layers, utilize separate Metadata Contexts to allow access control and mutability of metadata properties, checksums, files sizes, filenames/descriptions, relations)

Design Ideas (MRD):

- DSpace Content Service:
  - Content Service Aspect: Encapsulates the functionality required to store Assets delivered as InputStreams or resolvable URL's.
  - Metadata Service Aspect: Responsible for storing the metadata related to the technical aspects of these bitstreams.
    - Signatures
    - Size
    - Creation Timestamp
  - The above values are not mutable after the resource has been created.
  - All Assets have a Universally Unique Identifer (UUID) generated for resolution within DSpace services.

Metadata Management (Search/Browse/Indexing, Metadata store, Metadata schema, Versioning, External Identifiers, ObjectIdentifiers)

All aspects of managing and navigating the Metadata space. This includes supporting metadata properties attached to any DSpace Resource Identifier may be represented by one or more services.

Design Ideas (MRD):

- DSpace Technical Metadata Service: Holds metadata that is mutable on a Resource:
    - Preferred Filename
    - Preferred Mime type
    - Asset Format

- DSpace Administrative Metadata Service: Holds Metadata describing DSpaces Managerial layer over the above Assets.
    - Resource Relationships
        - Bitstream 2 Bundles, Bundles 2 Items,Items 2 Collections, Collections 2 Communities, Collections 2 Workflows, Workflows 2 Stages
    - Resource Descriptive Meta
        - DC/TERMS, MODS, BIBO

Policy Management (Roles, Policies, Workflows)

Design Ideas (MRD):

- DSpace Policy Management Service: Holds query-able Policy related metadata
    - Roles
        - Owner/Submitter
        - Administrator
        - Manager
        - User
    - Policies
        - CRUD

User Management (Users/Groups)

Design Ideas (MRD):

- DSpace User Management Service: Holds and Relates User/Group metadata
    - Users
    - Groups

Monitoring and Event Processing (Logging / Statistics, Events, Monitoring and Integrity)

A very core aspect of DSpace, Allows event notifications to be attached to higher level services that may report Access and Mutation of DSpace Content and its associated Properties.

Design Ideas (MRD):

- DSpace Event Notification Service

Authentication and Authorization (AuthN/AuthZ)

Criticisms in 1.5

Needs greater Abstraction away from HTTP Servlet API and Separation of Authentication substrate, Method and Realms.

Design Ideas (MRD):

- Should be javax Principal based and possibly XACML or some other standard protocol for which backends can be reused/repurposed.

Content Dissemination (Export / disseminate)

Content Transformation (MediaFilter, Data extraction, Transformation, etc)

Criticisms in 1.5

Conflated with Preservation. (Text extraction for Search turned into Preserving Content in a open accessable format)? Text Extraction, Thumbnail Creation all should have configurable levels of preservation (cached derivative vs preserved derivative)

Content Ingestion (Import / ingest)

Criticisms in 1.5

Not granular enough, cannot add Content to Items. Not CRUD enough, no implementation of replace or delete on both Content and Metadata.

I'm leaving these for Aaron or others to flesh out.

- Configuration
- Describe (introspection)
- Structure
- Service locator

Standard Service Protocols / Support

Some thoughts on Content Services:

- Would be nice to align with JSR-170/Fedora/Sakai (Nodes/Properties)
- Beneficial to consider Resourceful/Restful
- Content API should probably expose a basic search capability (Fedora/JSR-170: SQL subset

Thoughts on Services in relation to Metadata / Resource Modeling:

- Metadata for any Resource (Content or not) Fits nicely into one or more RDF Descriptions for that Resource.
- Different Services May hold different Statements about the Resource.
  - Policy Service holds Statements about Policies Attached to the Resource
  - Content Service holds Statements about Size, Signature and Creation for a Content Backed Resource).
  - Administrative Metadata Service holds Statements about how the Resources are related to each other "terms:isPartOf/hasPart". (This is similar to Fedora RELS-EXT).

DSpace Service API

DSpace Content Hosting Sevice

Modeled on common design aspects of JCR, Sakai, Fedora. It provides the following set of functionality:

- Resolution of Identifiers (Strings/URI) to an Object graph that can be navigated, manipulated and persisted.
- Attachment of properties (metadata) as well and content to the objects within the Graph.
- Provides Domain independent API for interacting with the underlying Store.

Important mappings/aspects

Fedora

- RELS-EXT expresses relationships between Fedora Digital Objects (Nodes)
- Datastreams are various representations of the Object (Metadata, Content, Derivatives)
- Fedora Content Model may be used to "Type" objects to adhere to specific constraints.

JCR

Sakai

http://bugs.sakaiproject.org/confluence/display/CHS/Roadmap+for+JCR+Migration

For an excellent overview of

External Resources

The Digital Library Technology Jester...

http://dltj.org/article/fedora-plus-sakai/trackback/
http://dltj.org/article/fedora-plus-sakai-3/trackback/
http://dltj.org/article/sakai-gets-jsr170/trackback/