Use Cases Summary - API Extension Architecture

This page documents the patterns and other commonalities that have been extracted from the API Extension Architecture use cases.

Guidelines

The common patterns should be identified and described on this page, and then related (e.g. a matrix of use cases and pattern names) to the use cases. Any clarifying questions (and answers) regarding a use case should be posed in the comments section for the individual use case.

I think we decided that the authors of individual use cases would take the first attempt to extract patterns from their own use cases.

We also decided to get this wiki page into a stable form by **Friday September 4th**, so that we can use the week of September 6th to think about how the patterns fit with the design proposal. We would be prepared to discuss the patterns as they relate to the design on our next call, Friday September 11.

Patterns

Name	Mediated Ingest
Description	This pattern exposes an object-based resource endpoint for complex ingest requests
Use cases	The Content Validation and Content Modeling use cases would seem to fit this pattern as well as the Transaction Support pattern below

Name	Mediated Retrieval
Description	This pattern exposes an object-based resource endpoint for complex retrieval requests
Use cases	OpenSearch API support - would use mediated retrieval to populate indices as well as to actually retrieve complex objects Content Modeling requires transformation patterns that filter the Fedora output of GET requests. Extra-repository Access Control filters results of a triplestore query based on user privileges.

Name	Workflow Monitoring
Description	This pattern exposes a service for monitoring the state/progress of API-Extension Architecture workflows
Use cases	

Name	Deployment/Configuration
Description	This pattern involves the mechanics of extension deployment, configuration, and sharing
Use cases	

Name	Transaction support
Description	A pattern that guarantees the state of the repository (Fedora + associated indexes) after a mutating operation completes
Use cases	- In the Recover from failed package deposit use case, one option for handling failed deposits is to roll back the deposit as if it never occurred. - In the Content Modeling use case, roll back the deposit of an object if it cannot be successfully transformed upon POST - Rolling back a deposit in the Content Validation use case if an incoming object cannot be successfully validated

Name	Selection and Filtering of Fedora resources or requests
Description	A pattern that allows for selecting Fedora resources and/or requests by some criteria
Use cases	- In the signposting use-case, selecting which responses are to be filtered - In the content and structural validation use case, selecting which objects are subject to validation. - in the Earth Sciences OpenSearch use case, knowing which collections are amenable to Earth Science OpenSearch queries.

Name	Stability of representations
Description	The stability of the representation of resources is a concern of multiple use cases. Ideally the representation of a resource can remain stable (identical or equivalent to previously retrieved representations), even if the underlying model used to represent the resource changes.
Use cases	SI - Stable Representations, SI - Assembly of representation from graph discuss retrieving stable representations. The JHU Retrieve Domain Objects, and AIC Content Model use cases explicitly want object representations to conform to a domain model. The AIC Content and Structural Validation discusses mechanisms to ensure conformity to a content model.