

1 e-Framework Service Usage Model Name

Provide a simple, informative name for the Service Usage Model (SUM).

- Name: OpenURL + Handle Appropriate Copy SUM
- Alternative Names: <Alternative names>

2 Version

Provide the e-Framework Service Usage Model Version for this description.

- <Version number>

2.1 Version History

Include requested information about all versions of this document.

Version	Date	Author	Description	Organization / Project
v.0.1	2007-06-06	NN	Initial Draft	FRED Project
v.0.2	2008-07-23	NN	Editorial	FRED Project

3 Rationale

This SUM documents the development of an appropriate copy selection system by the FRED project, as an orchestration of an appropriate copy service and a multiple resolution service. It has been developed as an instance of a small-scale service usage model developed to address a specific need with specific protocols and interfaces. It also serves as an illustration of a developmental small-scale SUM for the e-Framework community.

4 Classification¹

Type an "X" in the brackets next to the most appropriate choice for each category. Required categories are shown in **Bold**; any non-required categories that are not used may be deleted.

¹ See definitions of the Service Usage Model Classification Scheme categories and their allowable choices at:

<http://www.e-framework.org/Services/ServiceClassificationScheme/ClassificationSchemeForSUMs/tabid/817/Default.aspx>

Copyright © University of Southern Queensland.



This work is licensed under the Creative Commons Attribution-Share Alike 2.5 Australia License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/2.5/au/>

To be provided by the submitter:			
SUM Type	<input checked="" type="checkbox"/> Domain	<input type="checkbox"/> CORE (a commonly recurring SUM; designation requires e-Framework Integrity Group approval)	
Domain(s)	<input checked="" type="checkbox"/> Learning & Teaching	<input type="checkbox"/> Research <input type="checkbox"/> Libraries	<input type="checkbox"/> Administration <input type="checkbox"/> IT Services <input type="checkbox"/> Common
Maturity	<input checked="" type="checkbox"/> Immature	<input type="checkbox"/> Mature	
Purpose(s)	<input type="checkbox"/> Exemplar	<input checked="" type="checkbox"/> Application	<input type="checkbox"/> Modelling <input type="checkbox"/> Toolkit
XOR (exclusive "or")	<input type="checkbox"/> Service Genres	<input checked="" type="checkbox"/> Service Expressions	
Development Status	<input type="checkbox"/> Proposed	<input checked="" type="checkbox"/> Developmental	<input type="checkbox"/> Prototype <input type="checkbox"/> Production
Deployment Scale	<input checked="" type="checkbox"/> Isolated	<input type="checkbox"/> Ubiquitous	
State Behaviour	<input type="checkbox"/> Stateful	<input checked="" type="checkbox"/> Stateless	
Transactional Behaviour	<input checked="" type="checkbox"/> Transactional and ACID	<input type="checkbox"/> Transactional but Non ACID	<input type="checkbox"/> Non-Transactional
Batch Behaviour(s)	<input checked="" type="checkbox"/> Individual	<input type="checkbox"/> Batch	
Time-Constraint Behaviour	<input checked="" type="checkbox"/> Hard Real Time	<input type="checkbox"/> Soft Real Time	<input type="checkbox"/> None
Service End Point	<input checked="" type="checkbox"/> Provider	<input type="checkbox"/> Requestor	<input type="checkbox"/> Transcoder (both requests and provides)
Authentication/ Authorization Dependency	<input type="checkbox"/> Auth-Dependent	<input checked="" type="checkbox"/> Auth-Independent	
Protocol Binding(s) (only applies to service expression-based SUMs)	<input type="checkbox"/> Web Service <input type="checkbox"/> SOAP	<input checked="" type="checkbox"/> REST <input type="checkbox"/> HTTP	<input type="checkbox"/> Other
To be determined by the e-Framework:			
Status	<input type="checkbox"/> Approved	<input type="checkbox"/> Placeholder <input type="checkbox"/> Unapproved	<input type="checkbox"/> Superseded <input type="checkbox"/> Withdrawn
Confidence Level	<input type="checkbox"/> High	<input type="checkbox"/> Medium	<input type="checkbox"/> Low

5 Notation [optional]

List and explain the conventions and terminology used that are essential for understanding the rest of the Service Usage Model description. A complete list of terms should be included in the Glossary (below).

<type text here>

6 Description

This SUM documents the development of an appropriate copy selection system by the FRED project, as an orchestration of an appropriate copy service and a multiple resolution service. The FRED project develops infrastructure to support repository federations for education in Australia.

Appropriate Copy functionality is essential to content delivery from repository federations, in cases where there is content replication (e.g. for reasons of load balancing or different rights regimes). Of the available instances to be delivered, some may be inappropriate for the given context (for example, the requester does not have access privileges for a particular instance); the delivery system needs to forestall attempts to deliver an inappropriate instance. As a result, the existence of multiple instances of the requested object (some of them inappropriate) should be transparent to the requester whenever possible: the requester should not need to make any explicit selection amongst available instances.

An appropriate copy selection system presupposes that there is more than one instance of a requested digital object accessible by a content delivery system, stored on more than one data source. Given contextual information about the request for delivery (and potentially other information), the appropriate copy selection system selects from the available instances the instance most appropriate to the request. It then forwards the locator for that instance to the content delivery system, so that the selected instance is delivered to the user. The criteria the service may use to determine the most appropriate instance include access privileges, and data source location on the network.

The functionality of the appropriate copy selection system can be described as a single service. This functionality has already been documented by the FRED, as the service expression *OpenURL Appropriate Location + HTTP + Inline KEV*. However that service expression only describes the interaction of the appropriate copy service with the end user. A deployment also needs to interact internally with an instances registry, containing entries for all available object instances that the delivery service can access. FRED has modelled this interaction as an exposed service of multiple resolution (a single object identifier resolves to the locators of all available object instances in the instances registry). It implements the appropriate copy selection system as an orchestration of an Appropriate Copy Selection service and a Multiple Resolution service.

As it turns out, multiple resolution is itself a business requirement for an appropriate copy selection system, when the selection of appropriate instance does need to be made explicitly by the user (because user context cannot be anticipated). So the multiple resolution service is independently exposed to the end user. Moreover, the appropriate copy selection system needs to provide for populating and updating the instances registry, and the business logic driving the selection service.

7 Business Process Modelling

The following business requirements for the system are covered by the service usage model:

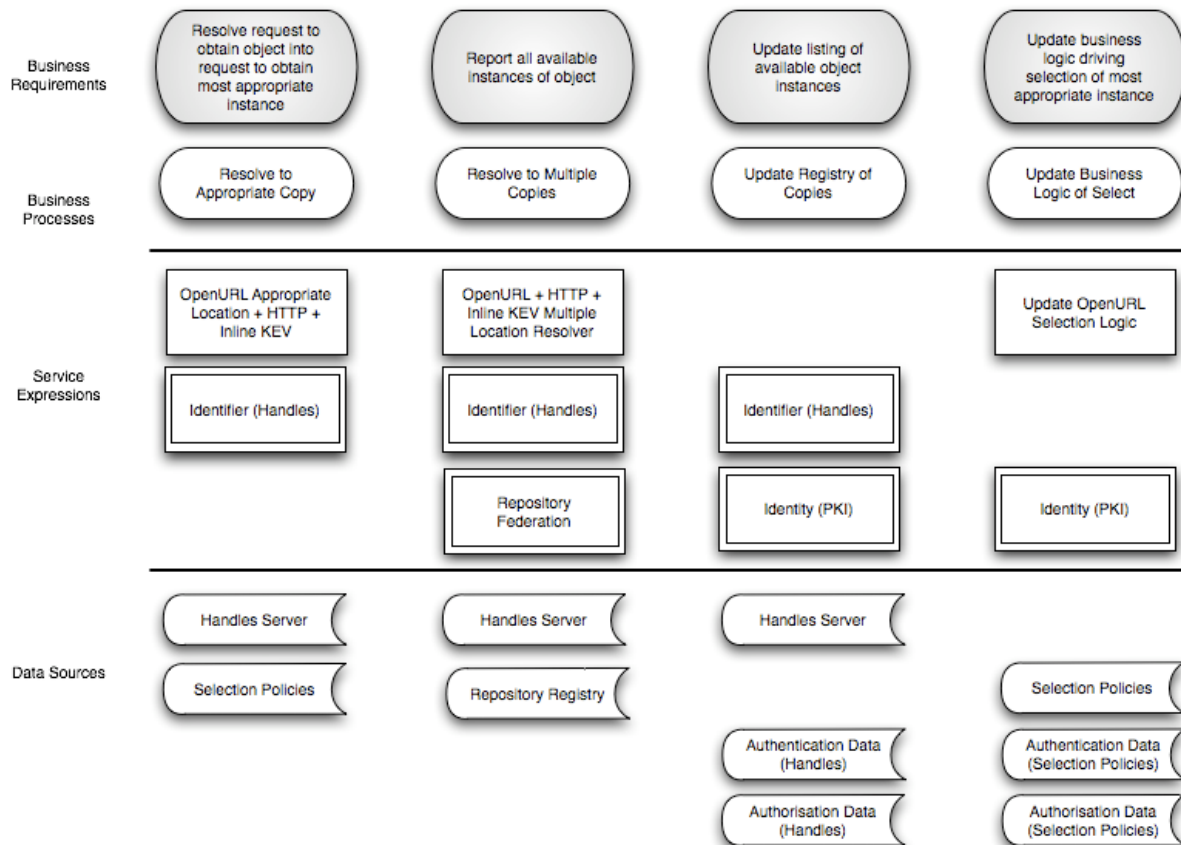
- Resolve a request to obtain an object, into a request to obtain the instance most appropriate for the request.
- Report all available object instances corresponding to a given object.
- Update the listing of all available object instances corresponding to a given object.
- Update the business logic driving the selection of most appropriate instances.

These requirements each map to a single business process:

- Resolve to Appropriate Copy
- Resolve to Multiple Copies
- Update Registry of Copies
- Update Business Logic of Select

The first two processes are end user driven. The last two processes are maintenance-driven. They are triggered either by the selection system administrator, or by the data sources which the system provides resolution for.

8 SUM Diagram



9 Usage Scenarios [optional]

Usage Scenarios for an Appropriate Copy service are available separately: **[AppropriateCopyUsageScenarios]**. They include the following:

- Scenario A: Resolve given only referent identifier: Search (single instance)
- Scenario B: Resolve given only referent identifier: Search (multiple instances)
- Scenario C: Resolve given referent identifier and requester affiliation: Search
- Scenario D: Resolve given referent identifier and requester affiliation: Link from Object
- Scenario E: Resolve given referent identifier and requester affiliation: External Link
- Scenario H: Resolve given referent identifier and requester identity: DRM
- Scenario I: Resolve given referent identifier, requester affiliation, and requester physical location
- Scenario J: Resolve directly to object vs. to metadata

These scenarios cover the two end user driven business processes *Resolve to Appropriate Copy* and *Resolve to Multiple Copies*. The maintainer driven business processes may be illustrated by the following scenarios:

Update Registry of Copies:

- A data source contains objects; the Appropriate Copy selection system enables access to those objects.

- The data source ingests an instance of a digital object.
- The digital object is identified through a Handle. The instance in the data source is identified through a locator (URL).
- The data source requests an update to the instances registry, to include the new locator as an instance of the digital object, and included in multiple resolution of the digital object Handle.
- The instances registry record is updated.
- An end user requests appropriate copy resolution for the digital object.
- The new object instance is now a candidate for appropriate copy resolution.

Update Business Logic of Select:

- One of the rules by which the selection system decides which instance to direct a resolution request to is updated. For example:
 - A new data source is added to those accessed by the system. For some contexts the new data source is the most appropriate location for delivery.
 - A data source has been removed from those accessed by the system. Any rules selecting that data source as an appropriate location need to select another location.
 - There is a change in the access regime of a data source. As a result, there are contexts in which the data source used to be appropriate and now is not, or vice versa.
 - There is a change in the network accessibility of a data source. As a request, there are contexts in which the data source used to be appropriate and now is less appropriate, or vice versa.
 - There is a change in the model of request contexts that is used in selection requests. The information available about the context of the request may be augmented, decreased, or refactored.
- The system administrator requests an update to the data source for selection policies, to reflect the new rule formulation.
- The selection policy data source is updated.

10 Applicability [optional]

The applicability constraints of the service expression *OpenURL Appropriate Location + HTTP + Inline KEV* also apply when that service expression is used in this service usage model. In particular, the service usage model, like the service expression, assumes an information model distinguishing between objects and object instances. The differentiation applies to the instances registry, where the locators of instances are keyed to identifiers of objects. However, this service usage model allows requests for objects to be resolved both to requests for the most appropriate instances (*OpenURL Appropriate Location + HTTP + Inline KEV*), and to all available instances (*OpenURL + HTTP + Inline KEV Multiple Location Resolver*).

The system assumes that the data source managers have given consent to end users accessing their content via the selection system. The selection system provides a single point of access to content on a range of repositories; this means that the data sources have entered into a federation, even if that federation is only instantiated through the selection system. The service usage model is not applicable if the data source managers refuse to participate in a federation and deliver their content through a common access point.

The system assumes that information about data source holdings of instances is freely available to the system, and is updated promptly. The update can be either through pull or push; in this service usage model, it is assumed to be through push. This means that the data source managers enter into an agreement, and commit to updating the instances registry promptly when they ingest a new instance. This agreement occurs out of band, and is therefore outside the scope of this service usage model.

The Handles server is primarily used by the system as an instances registry; as a resolver, it serves as a multiple resolution service, and not an authoritative or last-resort resolution service. End user access to object instances is through the OpenURL interface, and not directly through the Handle resolution interface. A Handle must be created for an object for the purposes of the instances registry, even if the object already has a Handle under a different naming authority: the Handles generated for the this SUM are for use specifically in the delivery system, and do not supplant any existing Handles.

There is no authentication of Handle resolution: all the locators resolved to by Handle are openly accessible without authorisation. Authenticated resolution may be imposed on the service usage model as an extension.

11 Functionality

The business processes outlined above are realized in terms of system functionality as follows:

- *Resolve to Appropriate Copy*: **Select Appropriate Copy**; Multi-Resolve Identifier
- *Resolve to Multiple Copies*: Multi-Resolve Identifier; **Present Identifier Resolutions**
- *Update Registry of Copies*: Create Identifier; Update Identifier
- *Update Business Logic of Select*: **Create Selection Rule Object**; **Update Selection Rule Object**; **Delete Selection Rule Object**

Functionality within the scope of this service usage model is in boldface. Functionality properly covered by other service usage models, which this service model leverages, is in plain style.

These functionalities may be described as follows:

Select Appropriate Copy:

Given a list of available locators, context information about the request, and selection policies, select the locator most appropriate to the request.

Basic workflow steps (without error handling) are:

- Retrieve the list of available locators to select amongst. (This functionality is covered by *Multi-Resolve Identifier*)
- Retrieve the selection policy objects (rules) applicable to the given context information.
- Apply the selection policy objects to the context information and the locators, to select a single locator
- Return the selected locator.

Business Process: *Resolve to Appropriate Copy*

Multi-Resolve Identifier:

Given an identifier for an object, return a list of locators of all object instances corresponding to that object, and recorded in the instances registry. The endpoint for this

functionality is in the Identifier (Handle) SUM, though the service expression OpenURL + HTTP + Inline KEV Multiple Location Resolver serves as a “wrapper” to that endpoint consistent with the OpenURL interface of *Select Appropriate Copy*.

Basic workflow steps (without error handling) are:

- Pass the object identifier to the instances registry
- Retrieve the list of locators corresponding to the identifier
- Return the list of locators as an XML object

Business Process: *Resolve to Appropriate Copy, Resolve to Multiple Copies*

Present Identifier Resolutions:

Present a list of locators to an end user as an HTML page allowing the user to select a locator manually. Enhance the presentation with metadata about data sources to will guide the user's selection.

Basic workflow steps (without error handling) are:

- Retrieve the list of available locators to select amongst. (This functionality is covered by *Multi-Resolve Identifier*)
- For each locator, extract the locator (or identifier) of the repository (data source) containing it.
- Retrieve the description of each repository from a repository registry.
- Render the list of locators as an HTML page, enhanced appropriately with metadata drawn from the repository descriptions.

Business Process: *Resolve to Multiple Copies*

Create Identifier:

Create an identifier for a new object and register it with the instances registry. This functionality does not include the assignment of a locator to the identifier, but it is not intended to be a mere “create name” function: the identifier should still incorporate sufficient discriminant metadata to allow the object to be identified at the object rather than instance level. The endpoint for this functionality is in the Identifier (Handle) SUM, and full description of the functionality is outside the scope of this service usage model. This functionality is only invoked if the object does not already have an identifier in the Handle Server's namespace. This functionality must be invoked if the object has a Handle in a different server's namespace, but the selection system does not have write privileges to that namespace.

Basic workflow steps (without error handling) are:

- Authenticate a request to create a Handle
- Authorise a request to create a Handle
- Create a Handle record
- Populate the Handle record with adequate discriminant metadata for the object it is to refer to

Business Process: *Update Registry of Copies*

Update Identifier:

Update the data contained in the identifier record for an object. In particular, update the listing of locators associated with the identifier, adding, deleting, or changing a locator value. Note that while the locators are *resolutions* of the object identifier, they are not the referents of the identifier: the identifier refers to an object, not to object instances. The endpoint for this functionality is in the Identifier (Handle) SUM, and full description of the functionality is outside the scope of this service usage model.

Basic workflow steps (without error handling) are:

- Authenticate a request to update a Handle
- Authorise a request to update a Handle
- Update the given Handle record to contain the desired values.

Business Process: *Update Registry of Copies*

Create Selection Rule Object]

Create a rule to be used in the selection of appropriate copies, and register it in the Selection Policies data store.

Basic workflow steps (without error handling) are:

- Formulate a selection rule object following the format specific to the Selection Policies data source. This rule will be parameterised on information contained in the OpenURL Context Object, and in the first instance on repository identifiers
- Authenticate a request to add to the Selection Policies data source
- Authorise a request to add to the Selection Policies data source
- Add the formulated rule object to the Selection Policies data source

Business Process: *Update Business Logic of Select*

Update Selection Rule Object]

Update a rule to be used in the selection of appropriate copies, and change it in the Selection Policies data store.

Basic workflow steps (without error handling) are:

- Identify a selection rule object in the Selection Policies data source.
- Reformulate the selection rule object to account for the change in selection policy.
- Authenticate a request to update the Selection Policies data source
- Authorise a request to update the Selection Policies data source
- Change the selected rule object to the Selection Policies data source

Business Process: *Update Business Logic of Select*

Delete Selection Rule Object]

Delete a rule to be used in the selection of appropriate copies, and change it in the Selection Policies data store.

Basic workflow steps (without error handling) are:

- Identify a selection rule object in the Selection Policies data source.
- Authenticate a request to update the Selection Policies data source

- Authorise a request to update the Selection Policies data source
- Delete the selected rule object from the Selection Policies data source

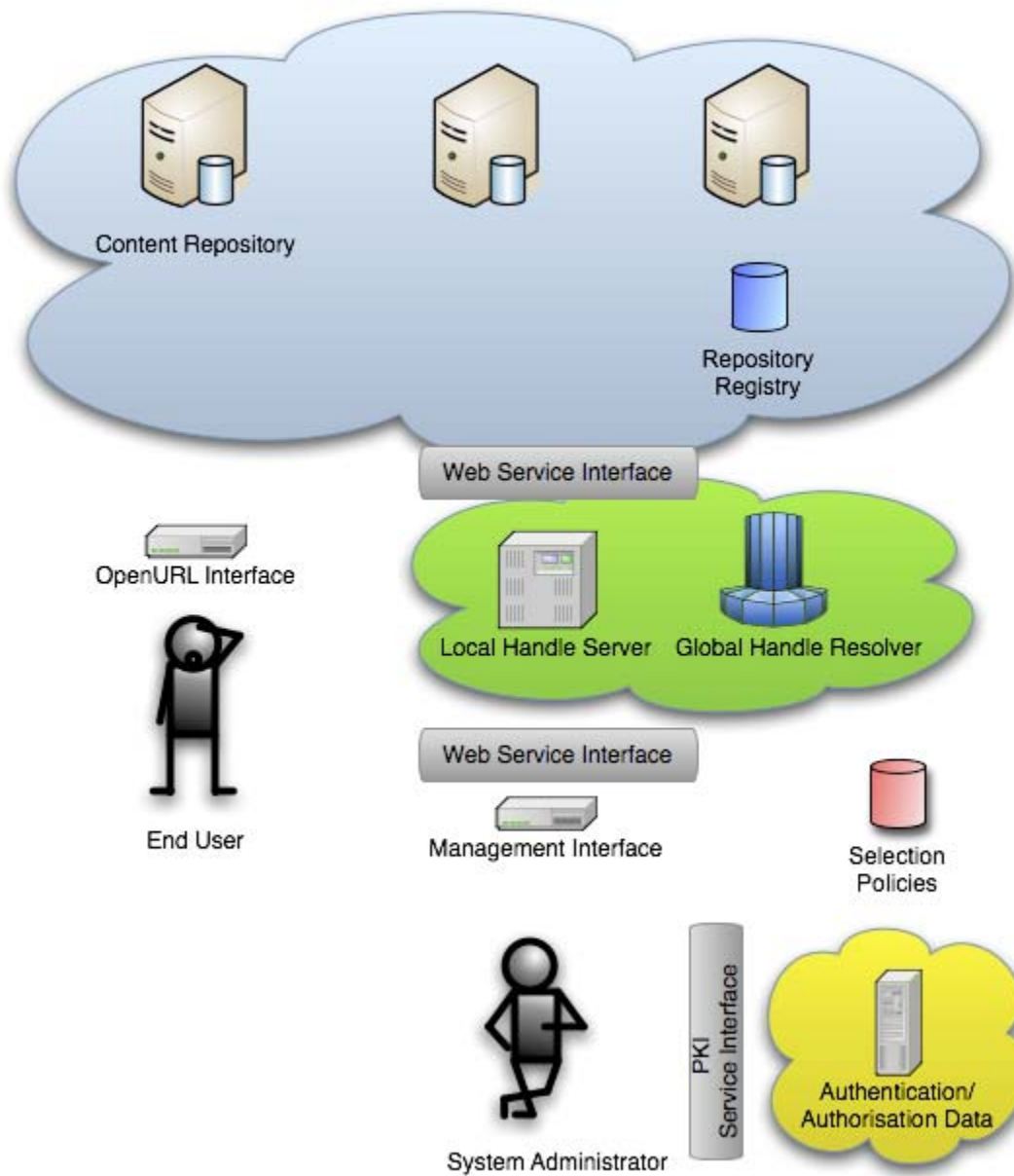
Business Process: *Update Business Logic of Select*

12 Structure & Organisation

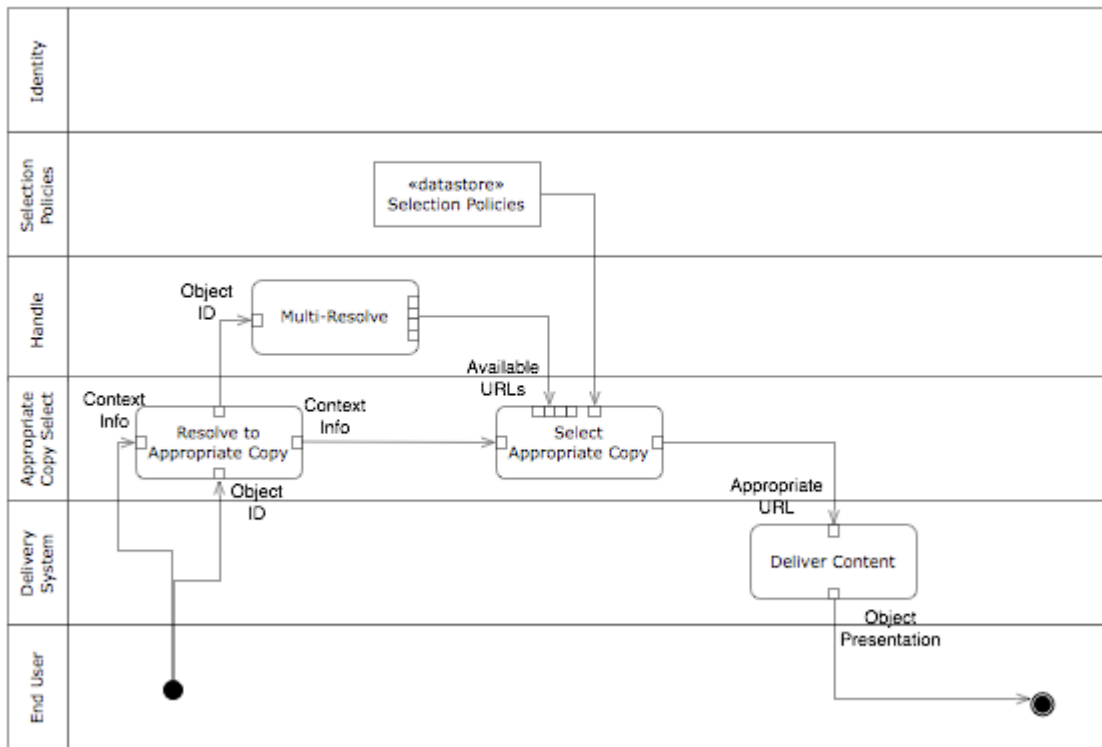
The structure of data sources involved in the service usage model is given below. Note that only the Selection Policies data source is described primarily by this service usage model, as a domain data source. The content repositories are accessed through services in the Repository Federation SUM; the Handle Server through the Identity (Handle) SUM; and the authentication/authorization data sources through Identity (PKI). There are service interfaces used for management functions to interact with the data sources, and triggered by content repositories and system managers: creating and updating Handles and selection policy rulesets, and authenticating and authorising creation and update requests.

End users, on the other hand, interact with the system through the OpenURL protocol, via a web browser. Once their request is resolved through the selection system, content delivery is done by the content repository delivery system; this is outside the scope of this service usage model. End user access to appropriate copy and multiple copy resolution is not authenticated; authentication is done only when there is an attempt to access the content object itself.

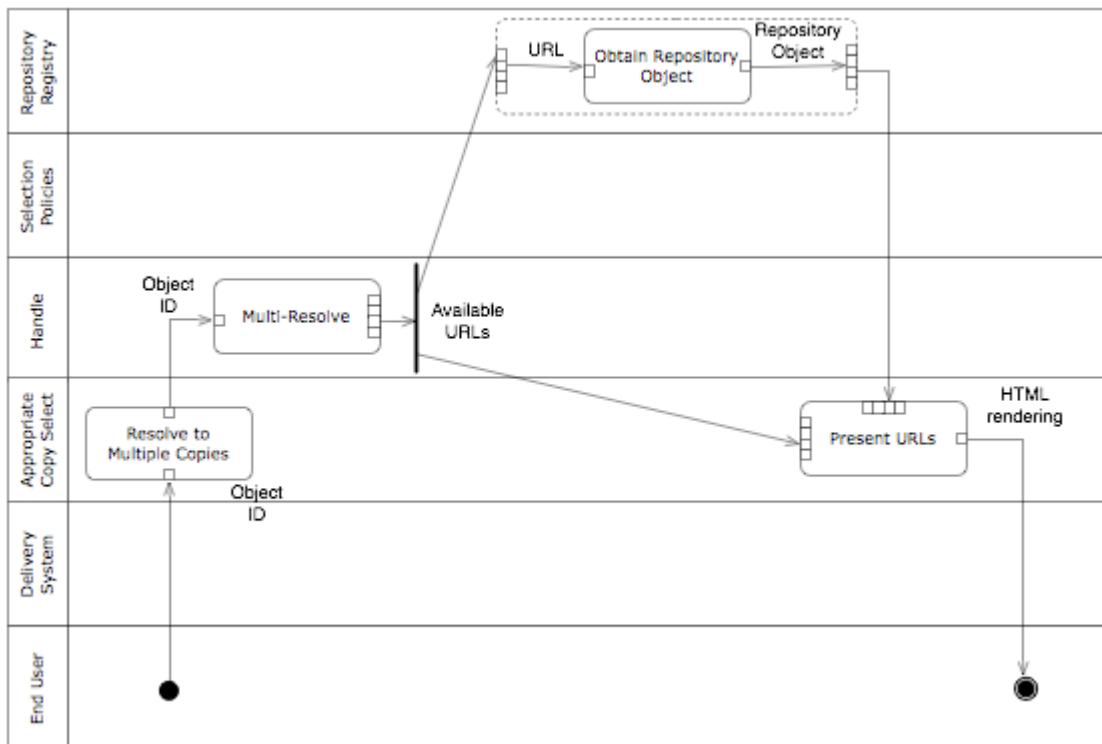
There is no defined overall application. This service usage model is assumed to be used within a set of applications that are used to provide end user access to the content repositories (typically but not always as a formal federation), provide management and provide operational support. Operations may require transactional- and session-based controls.



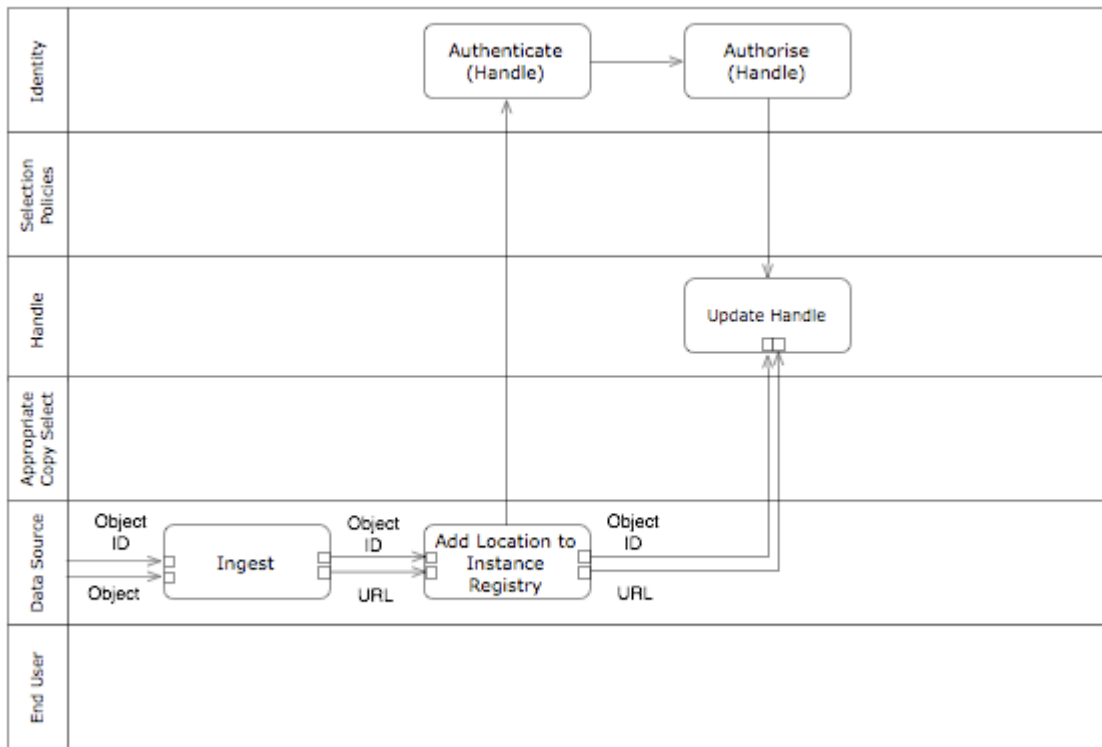
The following UML activity diagrams give the choreography of services to realize the four business processes of the service usage model.



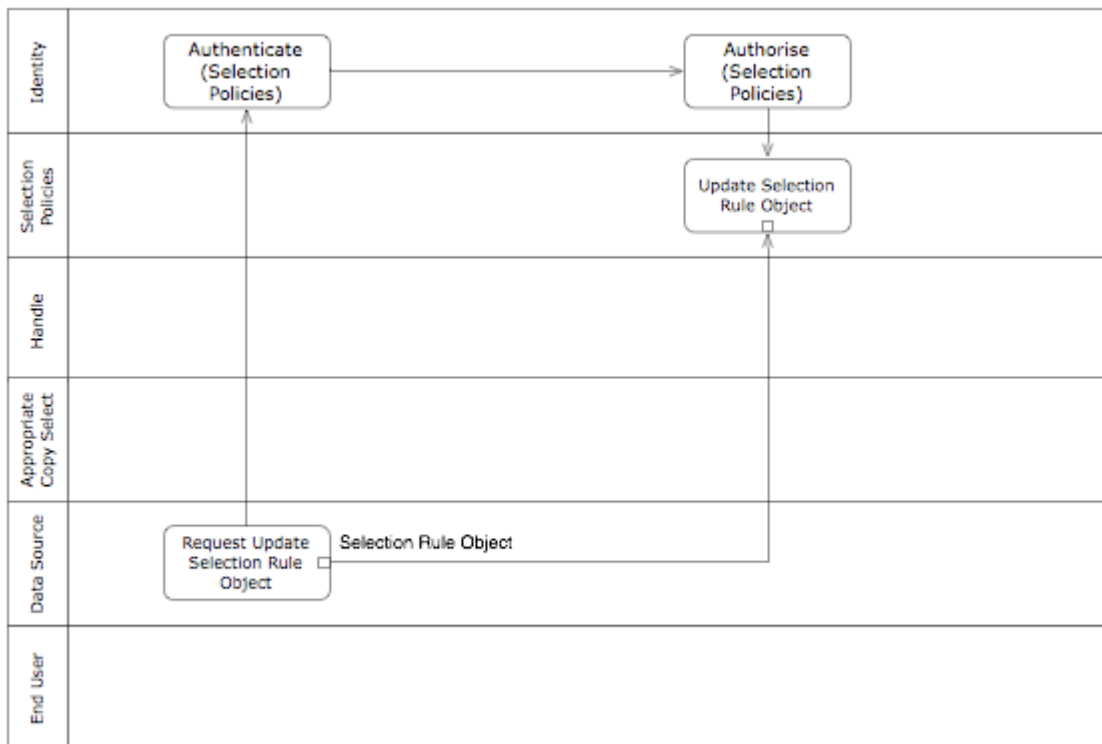
Resolve to Appropriate Copy



Resolve to Multiple Copies



Update Registry of Copies



Update Business Logic of Select

The functionality described above is realized through services as follows:

Select Appropriate Copy:

End Point: OpenURL Appropriate Location + HTTP + Inline KEV

Supporting Service Genre(s):

Supporting Service Usage Model(s):

Primary Data Source: Selection Policies

Secondary Data Source(s): Handles Server

Object(s): OpenURL ContextObject, Locator Listing (XML), Selection Policy Rule Object, Locator

Multi-Resolve Identifier:

End Point: Identifier (Handle) :: Handle Multiple Resolve

Supporting Service Genre(s): OpenURL + HTTP + Inline KEV Multiple Location Resolver (as "wrapper" service)

Supporting Service Usage Model(s): Identifier (Handle)

Primary Data Source: Handles Server

Secondary Data Source(s):

Object(s): OpenURL ContextObject (for "wrapper" service), Handle, Locator Listing (XML)

Present Identifier Resolutions:

End Point: Render Transform?

Supporting Service Genre(s): Obtain {Repository Object}

Supporting Service Usage Model(s): Repository Federation

Primary Data Source:

Secondary Data Source(s): Repository Registry

Object(s): Locator Listing (XML), Repository Object, HTML Page

Create Identifier:

End Point: Identifier (Handle) :: Create Identifier

Supporting Service Genre(s): Authenticate (PKI); Authorise (PKI)

Supporting Service Usage Model(s): Identity (PKI)

Primary Data Source: Handle Server

Secondary Data Source(s): Authentication Data (Handles); Authorisation Data (Handles)

Object(s): Handle; Object Metadata

Update Identifier:

End Point: Identifier (Handle) :: Update Identifier

Supporting Service Genre(s): Authenticate (PKI); Authorise (PKI)

Supporting Service Usage Model(s): Identity (PKI)

Primary Data Source: Handle Server

Secondary Data Source(s): Authentication Data (Handles); Authorisation Data (Handles)

Object(s): Handle; Instance URL

Create Selection Rule Object]

End Point: Update OpenURL Selection Logic

Supporting Service Genre(s): authenticate (PKI); authorise (PKI)

Supporting Service Usage Model(s): identity (PKI)

Primary Resource: Selection Policies

Secondary Resource(s): Authorisation Data (Selection Policies); Authentication Data (Selection Policies)

Object(s): Selection Rule Object

Update Selection Rule Object]

End Point: Update OpenURL Selection Logic

Supporting Service Genre(s): authenticate (PKI); authorise (PKI)

Supporting Service Usage Model(s): identity (PKI)

Primary Resource: Selection Policies

Secondary Resource(s): Authorisation Data (Selection Policies); Authentication Data (Selection Policies)

Object(s): Selection Rule Object

Delete Selection Rule Object]

End Point: Update OpenURL Selection Logic

Supporting Service Genre(s): authenticate (PKI); authorise (PKI)

Supporting Service Usage Model(s): identity (PKI)

Primary Resource: Selection Policies

Secondary Resource(s): Authorisation Data (Selection Policies); Authentication Data (Selection Policies)

Object(s): Selection Rule Object

13 Applicable Standards [recommended]

- American National Standards Institute 2005. *The OpenURL Framework for Context-Sensitive Services*. Bethesda (Md.): NISO Press. ANSI/NISO Z39.88-2004. ISSN: 1041-5653. http://www.niso.org/standards/standard_detail.cfm?std_id=783 . JISC Standards Catalogue: <http://standards-catalogue.ukoln.ac.uk/index/Z39.88-2004>

The OpenURL service expressions profile the service expressions to the requirements of the FRED community, and have created formal specifications which need to be included in service expression calls. See the service expressions for more details.

- Sun., S., Lannom, L. & Boesch, B. 2003. *Handle System Overview*. RFC 3650. The Internet Society. <http://www.ietf.org/rfc/rfc3650.txt> . JISC Standards Catalogue: <http://standards-catalogue.ukoln.ac.uk/index/Handle>

14 Design Decisions & Tradeoffs [optional]

Only the *Resolve to Appropriate Copy* and *Update Registry of Copies* functions of the service usage model are essential. Presenting repository metadata through *Resolve to Multiple Copies* is not essential. Therefore the service usage model does not have to invoke services from the Repository Federation service usage model, if such functionality is left out.

The functionality of the service model has been broken up into reusable components (particularly Multi-Resolve), and a particular implementation of the Instances Registry, rather than leaving the specification as the monolithic service expression *OpenURL Appropriate Location + HTTP + Inline KEV*. So there is no overt connection between this service usage model and the service expression *OpenURL + HTTP + Inline KEV Multiple Location Resolver*.

This service usage model has not been specialized to the needs of a particular community. Such specialization should occur in extensions of the service expressions, as well as refinements of the overall architecture.

15 Implementation Guidance & Dependencies [optional]

The service usage model is keyed to specific protocols and technologies through its choice of service expressions. In particular:

- Although Object delivery is out of scope for this service usage model, any object delivery system must be interoperable with the outputs of the two resolution service expressions, which are URLs. So delivery is across HTTP, and the objects delivered must be able to be processed by a web browser.
- Since the instances registry is expressed though Handles, the deployment must have write access to a Handles server: it must add new Handle records to the server, and new locator (URL) values to that server.

As noted, the deployment of this service usage model creates an ad hoc repository federation of the data sources it redirects to, but does not require the federation to be formalized with explicit policy infrastructure. It does require data sources to commit to pushing new locations of instances into the instance registry.

The Multiple Resolution functionality operates on batches of data. Details of batch process (including flow control) and transaction processing are deferred to the service expressions involved.

This service usage model does not specify the infrastructure or components required to implement an application. Such a system instance may require distributed or replicated servers, and multiple service instances and resource sets, for load balancing and performance.

Management functions may be performed asynchronously with end user requests. The implementation should ensure consistency of behaviour and results.

16 Known Uses [optional]

- *Repository Federation*. This service usage model may be used in conjunction with a Repository Federation, in order to provide a single access point to its content. The appropriate copy selection service would need to be integrated with any presentation of the federation's content delivery functionality; so calls to appropriate copy selection would supplant calls to object delivery, and content would be presented to end users at an object rather than object instance level, the object instance selection being hidden from the end user.
- *TLF Open URL*. This service usage model was used as a basis for a design service usage model, describing the appropriate copy system to be developed by The

Le@rning Federation. That service usage model was used in turn as a starting point for the eventual design and implementation of the appropriate copy system ultimately developed by The Le@rning Federation (the *Identifier and Access Services* component of the *Sharing Exchange Services*:
http://www.thelearningfederation.edu.au/for_developers/learn_about_our_technology/technical_projects/sharing_exchange_services.html).

17 Data Sources Used

- **Handles Server.** The Handles server is used by this service usage model as its Instances Registry. It contains a mapping of object identifiers to object instance locators. The locators can be retrieved from the instances registry through the Handle protocol, and selected amongst by the selection service.
- **Selection Policies.** The business logic through which appropriate copies are selected is represented as a ruleset stored in a selection policy data store. Individual rules of the ruleset can be retrieved, consulted, and updated.
- **Authentication Data (Handles).** This data source contains data through which access to the Handles Server is authenticated. Such access includes updating Handle records.
- **Authorisation Data (Handles).** This data source contains data through which administrative access to the Handles Server is authorised. Such access includes updating Handle records.
- **Authentication Data (Selection Policies).** This data source contains data through which access to the Selection Policies data source is authenticated.
- **Authorisation Data (Selection Policies).** This data source contains data through which administrative access to the Selection Policies data source is authorised. Such access includes updating Selection Rule objects.
- **Repository Registry:** Managed storage for repository objects describing the repositories accessible through the selection system.

18 Related SUMs [optional]

- *Identifier (Handle).* The Identifier (Handle) service usage model is an integrated set of service expressions that provides the entire identifier infrastructure used by this service usage model. The Identifier (Handle) service usage model is an expression of the Identifier service usage model through the Handle protocol. Functionality includes provisioning and management of the identifier system, and the creation, registering, publishing, managing and resolving of individual identifiers.
- *Identity (PKI).* The Identity (PKI) service usage model is an integrated set of service expressions that provides the entire user identity infrastructure used by this service usage model and by the Identifier (Handle) service usage model. The Identity (PKI) service usage model is an expression of the Identity service usage model through Public Key infrastructure. Functionality includes provisioning and management of the identity system, the creation of user right and roles, user authentication and user authorization. Note that authentication and authorization are combined within a single service usage model.
- *Repository Federation.* The Repository Federation service usage model is an integrated set of service genres that provides the infrastructure for managing and exposing a federation of repositories. As a single delivery point, the selection system defines a federation, but does not require a full implementation of the Repository Federation service usage model. However, the *Present Identifier Resolutions* function

does make use of a repository registry, such as is described in that service usage model.

19 Services Used

- OpenURL Appropriate Location + HTTP + Inline KEV
- OpenURL + HTTP + Inline KEV Multiple Location Resolver
- Update OpenURL Selection Logic
- Handles Multiple-Resolve. This service expression is included in the Identifier (Handle) SUM.
- Handles Update. This service expression is included in the Identifier (Handle) SUM.
- Authenticate (PKI). This service expression is included in the Identity (PKI) SUM.
- Authorise (PKI). This service expression is included in the Identity (PKI) SUM.
- Obtain Repository Object?. This service genre is included in the Repository Federation SUM.

20 CORE SUMs Used [recommended]

List the names and versions of all Commonly Recurring (CORE) SUMs that this Service Usage Model is based on.

<type text here>

21 References

Caplan, P. & Flecker, D. 1999. *Choosing the Appropriate Copy*. NISO.
<http://www.niso.org/news/reports/DLFarch.html>

Rehak, D. 2005. *The Appropriate Version Problem: Separating Learning Designs and Course Structures from Learning Object Versions, Variants and Copies*. CORDRA.
hdl:2000.01/D6E3BF9462684182AC293D64D3DDE192 .
<http://hdl.handle.net/2000.01/D6E3BF9462684182AC293D64D3DDE192>

22 Glossary [optional]

List and define the domain-specific terms used in documenting the Service Usage Model.

<type text here>



This SUM is licensed under:

Creative Commons Attribution-NonCommercial-ShareAlike 2.5 licence

<http://creativecommons.org/licenses/by-nc-sa/2.5/au/>

