

ECL_Features

Introduction

- Website
[ECL: eduSource Communication Layer](#)
- Version
0.5
- Brief description
ECL is an implementation of the IMS DRI specification. The software distribution provides an ECL connector, used for connecting a repository to the eduSource network. It also includes the ECL Gateway, which is a middleware framework for connecting to repositories and networks which implement different protocols or specifications.
The IMS DRI specification addresses interoperability at a very high level, leaving many details of repository management up to the specific implementations. Functions which are in scope of the specification are search/expose, gather/expose, submit/store, request/deliver, and alert /expose.
- Related Documents
[Secure Communication Layer for Scalable Network of Learning Object Repositories](#) Marek Hatala, Griff Richards, Timmy Eap, Ashok Shah
[Interoperability Frameworks for Learning Object Repositories](#), Griff Richards and Marek Hatala
[IMS Digital Repositories Interoperability - Core Functions Information Model](#)
[The EduSource Communication Language: Implementing Open Network for Learning Repositories and Services](#) Marek Hatala, Griff Richards, Timmy Eap and Jordan Willms
[Federated Security: Lightweight Security Infrastructure for Object Repositories and Web Services](#) Marek Hatala, Timmy Eap, Ashok Shah
- Sites running the software
ECL connectors have been used for a variety of protocols and networks, including OAI, SRW/SRU (RDN network in UK), EdNA (Australia), SMETE (USA), SQI (Ariadne), LionShare (gnutella based P2P network).

Features

Storage

Add data

Clients make requests to a *Submit* SOAP service on a server. To create or modify an object, the "save" action of the *Submit* function is used. If asynchronous messaging is used, a *Store* SOAP service on the client handles the results of the *Submit* call. Asynchronous messaging is not supported in the version under evaluation. Persistent identification appears to be left up to the implementation. Because objects should be using LOM, the catalog and entry metadata elements could combine to form a globally unique identifier. Alternatively, a handle system could be used.

Access data

Clients make a request to a *Request* SOAP service on a server. If asynchronous messaging is used, a *Deliver* SOAP service on the client handles the results of the *Request* call. Asynchronous messaging is not supported in the version under evaluation.

Remove data

Objects can be removed using the "delete" action of the *Submit* function.

Manage metadata

Metadata (LOM) is submitted along with the object via the *Submit* function.

Aggregation

There is no specific support in the interface for managing aggregates.

Management

Bulk ingest

This feature is left to implementations.

Bulk export

This feature is left to implementations.

Security

Authentication

Authentication support is provided by the ECL security infrastructure layer.

Access control

The interface supports getting and setting object access permissions. As the form of these will vary, formats are specified in the ECL Registry.

User management

This feature is left to implementations.

Policy management

This feature is left to implementations.

Locking

This feature is left to implementations.

Virtual object representation

This feature is left to implementations.

Transactions

This feature is left to implementations.

Versioning

There is some support for versioning through the use of modification timestamps.

Searching

Clients make a request to a *Search* SOAP service on a server. If asynchronous messaging is used, an *Expose* SOAP service on the client handles the results of the *Search* call.

Asynchronous messaging is not supported in the version under evaluation. ECL uses XQuery, and provides a set of templates which repositories may use to register which XQuery capabilities they support.