Version 1.1

A 'service oriented' view of the JISC Information Environment

Andy Powell, UKOLN, University of Bath

November 2005

UKOLN is funded by MLA: the Museums, Libraries and Archives Council, the Joint Information Systems Committee (JISC) of the Higher and Further Education Funding Councils, as well as by project funding from the JISC and the European Union. UKOLN also receives support from the University of Bath where it is based.

1 Introduction

The JISC Information Environment (JISC IE)[1] technical architecture specifies a set of standards and protocols that support the development of an integrated set of networked services. The intention is to allow the end-user to more seamlessly discover, access, use and publish digital and physical resources as part of their learning and research activities.

The key standards and protocols specified in the technical architecture are listed in the JISC IE Technical Standards[2].

The kinds of service components[3] made available through the JISC IE are shown in the diagram below. This diagram is not intended to be definitive. However, it is worth noting that at the time of writing, the majority of these components have been instantiated in some form or other as real service components on the network[4].

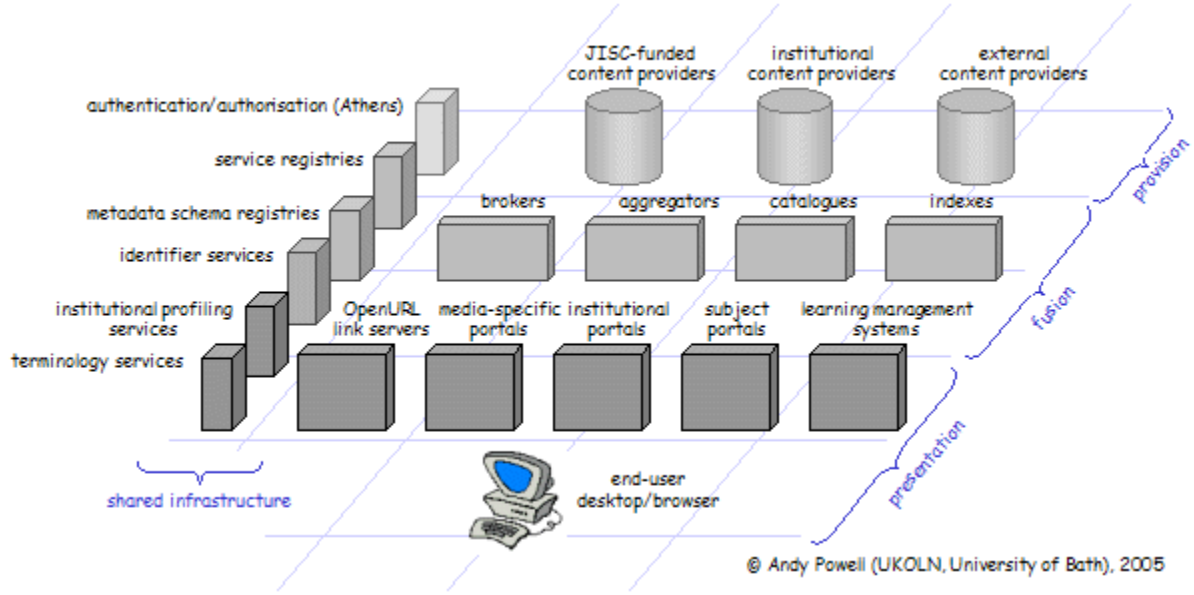

Figure 1 - The JISC IE architecture diagram

Examples of the kind of activities supported by the architecture include:

---

[1] Investing in the Future: Developing an Online Information Environment <http://www.jisc.ac.uk/index.cfm?name=ie_home>

[2] The JISC Information Environment Technical Standards <http://www.ukoln.ac.uk/distributed-systems/jisc-ie/arch/standards/>

[3] In the JISC IE, a 'service component' is a network service, i.e. a service that is provided on-line. Example network services include Web sites, document supply services, abstracting and indexing services, data archives, online catalogues, databases, email archives, format conversion services, printing services, authentication or e-commerce services, etc. Each service component may offer part or all of its functionality through one or more Web services.

[4] Powell, Andy. Mapping the JISC IE service landscape. *Ariadne* Issue 36, July 2003. <http://www.ariadne.ac.uk/issue36/powell/>

- Integration of local and remote information resources with a variety of 'discovery' services (for example the RDN subject portals, institutional and commercial portals and personal reference managers) allowing students, lecturers and researchers to find quality assured resources from a wide range of content providers including commercial content providers and those within the higher and further education community and elsewhere.
- Seamless linking from 'discovery' services to appropriate 'delivery' services.
- Integration of information resources and learning object repositories with Virtual Learning Environments (for example, allowing seamless, persistent links from a course reading list or other learning objects to the most appropriate copy of an information resource).
- Open access to e-print archives and other systems for managing the intellectual output of institutions.

Examples of the kinds of content that are available through the JISC IE include scholarly journals, monographs, textbooks, learning objects, abstracts, manuscripts, maps, music scores, Internet resource descriptions, still images, geospatial images and other kinds of vector and numeric data, as well as moving picture and sound collections.

2 A service oriented view of the JISC IE

The JISC IE was developed using a service-oriented approach, in the sense that it encourages machine-oriented services to be exposed on the network in order that others are able to develop applications that call on those services. This approach is also very much in line with the concepts behind Web 2.0[5].

The phrase 'service oriented' was not used in the context of the JISC IE since it had not at that stage come into widespread usage. However, as an Ariadne article in April 2002[6] made clear, it is easy to conceptualise the JISC IE as a service oriented architecture (SOA). Since then some work has been undertaken to more thoroughly enumerate the kinds of abstract services that are made available in the JISC IE[7] and the recent work of the Digital Library Federation Abstract Services Taskforce[8] has

[5] Miller, Paul. Web 2.0: Building the New Library. *Ariadne* Issue 45, October 2005 <http://www.ariadne.ac.uk/issue45/miller/>

[6] Powell, Andy and Lyon, Liz. The JISC Information Environment and Web services. *Ariadne* Issue 31, April 2002 <http://www.ariadne.ac.uk/issue31/information-environments/>

[7] The JISC IE in terms of the E-Learning Framework <http://www.ukoln.ac.uk/distributed-systems/jisc-ie/arch/elfie/>

[8] Dempsey, Lorcan and Lavoie, Brian. DLF Service Framework for Digital Libraries < http://www.diglib.org/architectures/serviceframe/dlfserviceframe1.htm>

made some progress towards describing the JISC IE as a 'discovery to delivery' service oriented reference model[9].

It is worth noting that the original work on the DNER technical architecture[10] and the MODELS Information Architecture that preceded it[11] used a functional analysis technique based on the use of UML use cases. This technique was used to better understand the business requirements and business functions that the JISC IE had to support. The availability of these UML use cases has made it relatively easy to retro-fit the JISC IE into the kind of 'reference model' framework proposed by the DLF and the e-Framework for Education and Research[12].

It is also worth noting that although the JISC IE is service oriented it is not exclusively SOAP-based. In fact, the only SOAP-based protocol listed in the JISC standards document is SRW and current trends would indicate that the uptake of SRW is not as good as the uptake of its REST-based counterpart, SRU. In more general terms, there is relatively little uptake and use of SOAP within the UK academic community at the moment. This is probably because of the added complexity of developing SOAP-based Web services in comparison with lighter-weight REST-based solutions. While there are some advantages in using SOAP rather than REST (for example, additional levels of security), those advantages may not outweigh the implementation costs, particularly in a relatively 'open' environment such as the JISC IE.

3 Service choreography

In a service-oriented architecture, service components that make use of remote Web services are often called service 'consumers'. Components that consume the outputs of more than one Web service are responsible for orchestrating the order in which calls are made to those services, and the way in which any data that is the output of one service is transformed into the input data for another service. Such a process of orchestration is often referred to as 'service choreography'.

Depending on the technical environment within which such choreography is undertaken, the service workflow organisation may be controlled in different ways - from hardwiring a simple linear workflow into a given software application to

[9] The JISC IE Discovery to Delivery (d2d) Reference Model
<http://www.ukoln.ac.uk/distributed-systems/jisc-ie/arch/dlf/>

[10] Powell, Andy and Lyon, Liz. The DNER Technical Architecture: scoping the information environment, May 2001
<http://www.ukoln.ac.uk/distributed-systems/jisc-ie/arch/dner-arch.html>

[11] MOving to Distributed Environments for Library Services
<http://www.ukoln.ac.uk/dlis/models/>

[12] The e-Framework for Education and Research
<http://www.e-framework.org/>

complex rules-based choreography using languages like the Business Process Execution Language for Web Services[13]

In the context of the JISC IE, two primary classes of service components that are responsible for service choreography have been identified (though it should be noted that all the service components listed in the diagram above are potentially responsible for some service choreography) – those in the presentation layer and those in the fusion layer.

Presentation layer service components are primarily responsible for interacting with end-users and for transforming the actions of the end-user into calls against the available services. The language of the JISC IE tended to refer to these services as 'portals'. However, in many ways this use of terminology was unhelpful, since a much wider range of services exist in the presentation layer than would be typically thought of as 'portals', some Web-based, others desktop applications and Web browser modules.

Furthermore, as described in Scott Wilson's paper about Web service workflows in the context of eLearning[14], there is also a special class of 'composite applications' that are both service consumers and service producers. In the JISC IE, the fusion layer is made up of these kinds of services – brokers, aggregators, catalogues and indexes. Each of the services in this layer is responsible for choreographing its own workflow against a set of other services and for offering the results of that process as a service output for others to build on.

Composite applications may combine several remote services in order to offer the same kind of service as a result, or they may combine remote service of one kind into a service of a different kind. For example, a broker combines several remote 'search' interfaces into a single 'search' interface. An aggregator may combine several 'harvest' interfaces into a single 'search' interface, a single 'harvest' interface, or both.

4 Shared services

One of the key service components in any service-oriented architecture is the service registry – the service (or set of services) that maintains and exposes information about the services that are available on the network. Without such a shared service, it is difficult for service consumers to achieve any level of automated service choreography.

In the JISC IE, the service registry is delivered as a centralised shared service component in the form of the IE Service Registry (IESR)[15] though there are ongoing

[13] Business Process Execution Language for Web Services version 1.1
<http://www-128.ibm.com/developerworks/library/specification/ws-bpel/>

[14] Wilson, Scott. Workflow and web services
<http://www.e-framework.org/resources/SOAandWorkflow2.pdf>

[15] JISC IE Service Registry
<http://www.iesr.ac.uk/>

discussions about whether this service should be delivered in a distributed form, in collaboration with similar initiatives elsewhere, and if so how.

As noted above, the JISC IE is a hybrid service-oriented architecture, in the sense that the available services are a mix of SOAP Web services, REST services and others (for example, Z39.50). This is also reflected in the technical implementation of the IESR because it needs to support the description of a wide range of different services and to make those descriptions available for harvesting and searching in multiple ways. So, although in a pure SOAP-based architecture, UDDI might be an obvious choice for the underlying technology of the service registry, this is not so clear-cut in the case of the IESR.

Finally it is perhaps worth noting that, with the exception of the authentication and authorisation shared services, the shared services listed in the JISC IE diagram above are not particularly mature. Most are still defining the exact nature of the abstract services that they will offer, and making appropriate technology implementation choices.

5 JISC IE business requirement, functions and processes

As mentioned above, the JISC e-Framework for Education and Research and the DLF Abstract Services Taskforce are both trying to develop mechanisms for defining 'reference models' that describe the set of services used in a particular application area. Whilst it is to be hoped that these two activities can work together to develop a shared way of doing this, it is currently the case that our thinking in this area is not sufficiently refined to make this task particularly straightforward. Attempts so far at describing the JISC IE discovery to delivery reference model (see above) need significantly more work to make a compelling and useful description of the space occupied by the JISC IE.

Part of the problem lies in finding the best way of 'picturing' the functional aspects of a reference model. UML use-case diagrams provide one way forward but there is a real danger that they will prove to be overly complex for parts of the intended audience. As an alternative, it might be possible to use 'mind maps' but more work is required to see how well this approach works.

A second problem lies in untangling 'abstract services' from 'service components'. For example, it is likely that the 'content provider', 'service registry' and 'licence registry' service components each offer essentially the same set of 'abstract services' (e.g. a search interface, a harvest interface, a deposit interface, a news channel and a delivery service). The difference between these service components lies not in the services offered, but in the kind of content that is passed across the service interfaces. We need to find ways of documenting these abstract services and their bindings in the form of 'reference models' without replicating a large amount of the information.

A final problem lies in trying to conceptualise the highly complex and varied way in which end-users use our services. We are no longer trying to build monolithic services (Web-based or otherwise) that meet all the needs of our end-users. Instead,

our users are using their Web browsers and other desktop tools to navigate their way between a wide range of human-oriented services, each of which may have several Web services sitting behind it. (To complicate matters even further, some of those desktop tools may be embedded into the end-user's browser in the form of a browser module, while others will be true desktop applications.) For example, an end-user looking for an academic journal article may move fairly seamlessly along the following path – Google (initial search) -> Desktop RSS channel viewer (viewing the RSS channel associated with the discovered resource) -> Connotea (saving bookmarks for later use) -> Local OpenURL resolver (finding a local copy) -> Abstracting and indexing service (checking citations) -> Local OpenURL resolver (following-up related resources) -> Local library OPAC (checking local library holdings) -> IngentaJournals or publisher Web site (obtaining electronic copy) -> Desktop reference manager (storing citations for future use in papers). All of this happens largely by navigating a set of hypertext links using a Web browser (i.e. that is the end-users primary experience of what is happening) but there may well be a number of Web services being called in the background (which are, of course, invisible to the end-user). A view of the world in which the user visits a single monolithic application (a 'portal') to undertake all these tasks is no longer relevant.

Given this complexity and the lack of maturity in declaring reference models, this document does not attempt to present a reference model for the JISC IE. Instead, the remaining sections by-pass any attempt to describe the functions and processes being undertaken by the end-user and instead consider the kinds of abstract services likely to be offered by each of the service components currently identified in the JISC IE.

6 JISC IE abstract services and bindings

The following sections list the abstract services that are likely to be offered by the service components shown in the JISC IE diagram above, describe each of those abstract services, and finally list the specific 'service bindings' associated with those abstract services (corresponding to the standards and protocols listed in the JISC IE Standards Framework). In this context, an abstract service is:

> *An identifiable portion of a business process. An abstract service includes a description of its functional scope, and an abstract model of its behavior and data.*

And a service binding is:

> *A specific instantiation of an abstract service. A service binding elaborates on an abstract service by providing all of the following which are applicable: 1) a specific data representation; 2) an Application Programming Interface (API) specification; and 3) a Web service specification.*

6.1 Mapping JISC IE service components to abstract services

This section lists the abstract services that are likely to be offered by each of the service components shown on the JISC IE diagram above (plus one or two others).

Content provider / repository

Stores, manages and makes available content, metadata and/or news channels.

- Search interface
- Harvest interface
- News channel
- Deposit interface
- Delete interface
- Obtain interface

Broker

Passes search requests on to one or more remote search interfaces (typically those offered by content providers / repositories) in parallel, combines the results and returns them to the requester.

- Search interface

Aggregator

Gathers metadata records from one or more remote harvest interfaces or news channels (typically those offered by content providers / repositories) and offers them for searching or harvesting by others or as an aggregated news channel.

- Harvest interface
- Search interface
- News channel

Catalogue

Stores, manages and makes available metadata records (typically human-generated) about content held elsewhere.

- Harvest interface
- Search interface
- News channel
- Metadata deposit interface
- Delete interface

Index

Uses Web robots or other techniques to gather content held elsewhere and indexes it for searching by others.

- Search interface

OpenURL link server

The target of an OpenURL link. (Also known as an OpenURL resolver.) Offers links to content and other information based on the metadata carried on the OpenURL.

- OpenURL interface

Portal

Provides a personalised, single point of access to a range of heterogeneous network services, local and remote, structured and unstructured. Portal functionality often includes resource discovery, email access and online discussion fora. Portals are intended for (human) end-users using common Web 'standards' such as HTTP, HTML, Java and JavaScript.

- None

VLE

A Virtual Learning Environment. (Also known as a Learning Management System.) Offers a range of functionality that supports teachers, lecturers, tutors and learnings in their teaching and learning activities.

- None (though VLE with integrated learning object repository may offer the content provider services above)

Authentication/authorisation (Athens)

An authentication service component determines whether the real-world individual who has the rights to use it is using the digital ID being presented to a network service. This is often achieved through the use of a username/password combination or a digital certificate, depending on the degree of assurance required. An authorisation service component indicates whether a particular real-world individual has the necessary access-rights to access a particular resource.

- Authentication service
- Authorisation service

This functionality is combined in the Athens service, but this is not the case in other systems such as Shibboleth.

Collection registry

Stores, manages and makes available descriptions of collections (i.e. the collections of content being made available by content providers / repositories).

- Collection description search interface
- Collection description harvest interface
- News channel
- Collection description deposit interface
- Delete interface

Service registry

Stores, manages and makes available descriptions of the services being offered on the network by other service components.

- Service description search interface
- Service description harvest interface
- News channel
- Service description deposit interface
- Delete interface

Note that the JISC IESR shared 'service registry' service component currently combines Collection Registry and Service Registry functionality into a single service component known as the IESR.

Metadata schema registry

Stores, manages and makes available descriptions of the metadata schemas in use by other service components. Typically some combination of metadata semantics and syntax will be described.

- Metadata schema search interface
- Metadata schema harvest interface
- News channel
- Metadata schema deposit interface
- Delete interface

Identifier service component

Stores, manages and resolves identifiers.

- Identifier resolver
- Identifier search interface
- Identifier harvest interface
- Identifier deposit interface
- Delete interface

Institutional profiling service component

Stores, manages and makes available institutional profiles (descriptions institutional preferences such as their preferred e-journal suppliers, licensing agreements in place, etc.).

- Institutional profile search interface
- Institutional profile harvest interface
- News channel
- Institutional profile deposit interface
- Delete interface

Terminology service component

Stores, manages and makes available vocabularies (ontologies, classification schemes, thesauri and/or other controlled vocabularies) and provides terminology-related services, such as mapping a term from one controlled vocabulary to another or expanding terms within a thesaurus.

- Vocabulary search interface
- Vocabulary harvest interface
- News channel
- Vocabulary deposit interface
- Delete interface
- Terminology service

Ratings and annotation service component

Stores, manages and makes available user-supplied ratings and annotations about content held elsewhere.

- Ratings search interface
- Ratings harvest interface
- Ratings deposit interface
- Ratings delete interface
- Annotations search interface
- Annotations harvest interface
- Annotations deposit interface
- Delete interface

User preferences service component

Stores, manages and makes available user preferences (typically for the purposes of personalising other service components).

- User preferences search interface
- User preferences harvest interface
- User preferences deposit interface
- Delete interface

Note that for data protection and other operational reasons, a user preferences service component is unlikely to be a suitable candidate for a 'shared service' and is more likely to be offered within the confines of a single institution.

Terms and conditions service component

Provides information about the terms and conditions associated with a resource.

- Licence search interface
- Licence harvest interface
- Licence deposit interface
- Delete interface

As noted above, this list of services is not exhaustive. There will be a wide range of other service developed and offered on the network, either globally, as national shared services or within specific institutions. Examples include automatic document classification and metadata generation services, packaging and unpackaging services, format conversion services, validation services, reading list services, bookmarking services, etc., etc.

6.2 Abstract services

This section briefly describes each of the abstract services listed above. Each abstract service is described in terms of its overall function, the intelligence it requires (i.e. what business entities does it need to know about) and its inputs and outputs.

Search interface

Accepts a structured query and issues a result set (a set of metadata records) in response.

Intelligence: Content, result set schema
Data in: Structured query (keywords, boolean operators, etc.)
Data out: Result set

Harvest interface

Makes metadata records available for harvesting (regular gathering). Typically, this service will be invoked in order to harvest metadata records into a local database so that an end-user search or browse interface can be offered or so that the harvested records can be re-exposed for harvesting or searching by other services.

Intelligence: Content, response schema
Data in: Harvesting request (record ID(s), set ID(s), time period, etc.)
Data out: Harvesting response

News channel

Provides a feed of information about new or updated content (or other resources).

Intelligence: Content, channel feed schema
Data in: Channel request (channel identifier, etc.)
Data out: Channel feed

Note that a 'news channel is really an obtain interface (see below) for a channel feed.

Deposit interface

Provides an interface through which content and metadata (possibly in the form of a 'complex object', i.e. a package) can be deposited and initiates ingest process for local storage.

Intelligence: Data format, packaging standard
Data in: Deposit request (content, metadata, etc.)
Data out: Deposit status (success, failure, pending, etc.) and content identifier

Note that the subsequent ingest process may include both automated and manual procedures including format checking, editorial control, quality assurance mechanisms, etc.

Delete interface

Provides an interface through which content can be deleted.

Intelligence:
Data in: Delete request (content identifier, etc.)
Data out: Delete status (success, failure, pending, etc.)

Obtain interface

Provides an interface through which content (or appropriate representation of content – possibly in the form of a 'complex object', i.e. a package) can be requested and initiates delivery process.

Intelligence: Data format, packaging standard
Data in: Content request (content identifier, preferred representation format, etc.)
Data out: Content (or appropriate representation of content)

Metadata deposit interface

Deposit interface (see above) carrying metadata.

OpenURL interface

Resolves an OpenURL into a set of links to delivery services and/or other information.

> Intelligence: Content provider holdings, terms and conditions, institutional preferences, user preferences
> Data in: OpenURL
> Data out: Links to delivery services and other information

Authentication service

Determines that the real-world individual who has the right to use it is using the user-ID being presented to a network service.

> Intelligence: Users
> Data in: IP address, username/password, digital certificate
> Data out: Result indicator (authenticated/not authenticated)

Authorisation service

Indicates whether an authenticated user ID has the necessary rights to access a particular resource.

> Intelligence: Access control lists, User Profiles, Licenses/T&C
> Data in: IP address, username, digital certificate, session token, resource ID
> Data out: Binary result indicator (authorized/not authorized)

Collection description search interface

Search interface (see above) to registry of collection descriptions.

Collection description harvest interface

Harvest interface (see above) offering collection descriptions.

Collection description deposit interface

Deposit interface (see above) carrying collection descriptions.

Service description search interface

Search interface (see above) to registry of service descriptions.

Service description harvest interface

Harvest interface (see above) offering service descriptions.

Service description deposit interface

Deposit interface (see above) carrying service descriptions.

Metadata schema search interface

Search interface (see above) to registry of metadata schemas.

Metadata schema harvest interface

Harvest interface (see above) offering metadata schemas.

Metadata schema deposit interface

Deposit interface (see above) carrying metadata schemas.

Identifier resolver

Resolves an identifier into a location.

> Intelligence: Identifier/location mappings
> Data in: Identifier
> Data out: Location (e.g. a URL)

Identifier search interface

Search interface (see above) to registry of identifiers and associated metadata.

Identifier harvest interface

Harvest interface (see above) offering identifiers and associated metadata.

Identifier deposit interface

Deposit interface (see above) carrying identifiers and associated metadata.

Institutional profile search interface

Search interface (see above) to registry of institutional profiles.

Institutional profile harvest interface

Harvest interface (see above) offering institutional profiles.

Institutional profile deposit interface

Deposit interface (see above) carrying institutional profiles.

Vocabulary search interface

Search interface (see above) to registry of vocabularies.

Vocabulary harvest interface

Harvest interface (see above) offering vocabularies.

Vocabulary deposit interface

Deposit interface (see above) carrying vocabularies.

Terminology service

Provides terminology-related services, such as mapping a term from one controlled vocabulary to another or expanding terms within a thesaurus.

> Intelligence: Ontologies, classification schemes, thesauri or other controlled vocabularies
> Data in: Terminology request (term, scheme, operation)
> Data out: Terminology response (term(s))

Ratings search interface

Search interface (see above) to database of resource ratings.

Ratings harvest interface

Harvest interface (see above) offering resource ratings.

Ratings deposit interface

Deposit interface (see above) carrying ratings.

Annotations search interface

Search interface (see above) to database of resource annotations.

Annotations harvest interface

Harvest interface (see above) offering resource annotations.

Annotations deposit interface

Deposit interface (see above) carrying annotations.

User preferences search interface

Search interface (see above) to registry of user preferences.

User preferences harvest interface

Harvest interface (see above) offering user preferences.

User preferences deposit interface

Deposit interface (see above) carrying user preferences.

Licence search interface

Search interface (see above) to registry of licensing information.

Licence harvest interface

Harvest interface (see above) offering licensing information.

Licence deposit interface

Deposit interface (see above) carrying licensing information.

6.3 Service bindings

This section provides a very minimal list of candidate protocols and standards for each of the abstract services listed above.

More information about most of these standards is available from the JISC IE Standards document.

Search interface

- Z39.50
- SRW/SRU
- A9 Opensearch
- Google API
- DC
- IEEE LOM
- METS
- IMS Content Packaging Specification
- MPEG-21 DID

Harvest interface

- OAI-PMH
- DC
- IEEE LOM
- METS
- IMS Content Packaging Specification
- MPEG-21 DID

News channel

- RSS
- Atom
- HTTP

Deposit interface

- HTTP POST
- SRW/SRU Update
- Fedora API (?)
- WebDAV
- Atom Publishing Protocol
- DC
- IEEE LOM
- METS
- IMS Content Packaging Specification
- MPEG-21 DID

Delete interface

- Atom Publishing Protocol
- WebDAV
- Fedora API (?)

Obtain interface

- HTTP
- Inter-Library Loan (ILL)
- OAI-PMH (carrying complex objects)
- Atom Publishing Protocol

Metadata deposit interface

- HTTP POST
- SRW/SRU Update
- Fedora API (?)
- WebDAV
- Atom Publishing Protocol
- DC
- IEEE LOM
- METS
- IMS Content Packaging Specification
- MPEG-21 DID

OpenURL interface

- OpenURL
- COinS

Authentication service and authorisation service

- Athens
- Shibboleth

Collection description search interface

- SRW/SRU
- DC Collection Description Application Profile

Collection description harvest interface

- OAI-PMH
- DC Collection Description Application Profile

Collection description deposit interface

- HTTP POST
- SRW/SRU Update
- Fedora API (?)
- WebDAV
- Atom Publishing Protocol
- DC Collection Description Application Profile

Service description search interface

- SRW/SRU
- IESR Application Profile
- WSRP
- UDDI

Service description harvest interface

- OAI-PMH
- DC
- IESR Application Profile
- WSRP

Service description deposit interface

- HTTP POST
- SRW/SRU Update
- Fedora API (?)
- WebDAV
- Atom Publishing Protocol
- IESR Application Profile
- WSRP

Metadata schema search interface

- SRW/SRU
- DC
- IEEE LOM
- RDF/RDFS
- OWL
- SKOS Core

Metadata schema harvest interface

- OAI-PMH
- DC
- IEEE LOM
- RDF/RDFS
- OWL
- SKOS Core

Metadata schema deposit interface

- HTTP POST
- SRW/SRU Update
- Fedora API (?)
- WebDAV
- Atom Publishing Protocol
- DC
- IEEE LOM
- RDF/RDFS
- OWL
- SKOS Core

Identifier resolver

- HTTP
- URI
- DOI
- Handle
- PURL

Identifier search interface

- SRW/SRU
- URI
- DOI
- Handle
- PURL

Identifier harvest interface

- OAI-PMH
- DC
- URI
- DOI
- Handle
- PURL

Identifier deposit interface

- HTTP POST
- SRW/SRU Update
- Fedora API (?)
- WebDAV
- Atom Publishing Protocol
- URI
- DOI
- Handle
- PURL

Institutional profile search interface

- SRW/SRU

Institutional profile harvest interface

- OAI-PMH
- DC

Institutional profile deposit interface

- HTTP POST
- SRW/SRU Update
- Fedora API (?)
- WebDAV
- Atom Publishing Protocol

Vocabulary search interface

- SRW/SRU
- Zthes
- RDF/RDFS
- OWL
- SKOS Core
- IMS VDEX

Vocabulary harvest interface

- OAI-PMH
- DC
- Zthes
- RDF/RDFS
- OWL
- SKOS Core
- IMS VDEX

Vocabulary deposit interface

- HTTP POST
- SRW/SRU Update
- Fedora API (?)
- WebDAV
- Atom Publishing Protocol
- Zthes
- RDF/RDFS
- OWL
- SKOS Core
- IMS VDEX

Terminology service

- SOAP
- HTTP
- Zthes
- RDF/RDFS
- OWL
- SKOS Core
- IMS VDEX

Ratings search interface

- SRW/SRU
- DC

Ratings harvest interface

- OAI-PMH
- DC

Ratings deposit interface

- HTTP POST
- SRW/SRU Update
- Fedora API (?)
- WebDAV
- Atom Publishing Protocol
- DC

Annotations search interface

- SRW/SRU
- DC

Annotations harvest interface

- OAI-PMH
- DC

Annotations deposit interface

- HTTP POST
- SRW/SRU Update
- Fedora API (?)
- WebDAV
- Atom Publishing Protocol
- DC

User preferences search interface

- LDAP
- SRW/SRU
- eduPerson

User preferences harvest interface

- OAI-PMH
- DC
- EduPerson

User preferences deposit interface

- HTTP POST
- SRW/SRU Update
- Fedora API (?)
- WebDAV
- Atom Publishing Protocol
- DC
- EduPerson

Licence search interface

- SRW/SRU
- ODRL
- XrML
- Creative Commons

Licence harvest interface

- OAI-PMH
- DC
- ODRL
- XrML
- Creative Commons

Licence deposit interface

- HTTP POST
- SRW/SRU Update
- Fedora API (?)
- WebDAV
- Atom Publishing Protocol
- ODRL
- XrML
- Creative Commons