



Intelligence Community and Department of Defense Content Discovery & Retrieval Integrated Product Team

REST Interface Encoding Specification for Content Discovery and Retrieval Brokered Search

Version 2.0

6 September 2013

Distribution Notice:

This document has been approved for Public Release and is available for use without restriction.

Table of Contents

Chapter 1 - Introduction	1
1.1 - Purpose	1
1.2 - Scope	1
1.3 - Background	1
1.4 - Enterprise Need	2
1.5 - Audience and Applicability	3
1.6 - Conventions	3
1.7 - Dependencies	4
1.8 - Conformance	4
1.9 - Version Policies	4
1.10 - Namespaces	4
1.11 - License	5
1.12 - Security	5
Chapter 2 - Development Guidance	6
2.1 - Brokered Search	6
2.1.1 - Stateless/Stateful Search	6
2.1.2 - Broker to Broker Search	6
2.1.3 - Heterogeneous Sources Invocation	7
2.1.4 - Results Aggregation and Paging	8
2.2 - Source Identification	9
2.2.1 - Source Description	9
2.2.2 - Source Routing	9
2.2.2.1 - Static routing	9
2.2.2.2 - Smart routing by Query Inspection (Non-Normative)	9
Chapter 3 - Definitions, Interfaces & Constraints	10
3.1 - Brokered Search Function	10
3.1.1 - Preconditions	10
3.1.2 - Input	10
3.1.2.1 - HTTP Method	10
3.1.2.2 - URL Template	10
3.1.2.3 - HTTP Message Header	12
3.1.2.4 - HTTP Message Body	12
3.1.2.5 - Search Request - Message Example	12
3.1.3 - Output	13
3.1.3.1 - HTTP Status Code	13
3.1.3.2 - HTTP Message Header	13
3.1.3.3 - HTTP Message Body	13
3.1.3.4 - Response Elements	13
3.1.3.5 - Output Example	16
3.1.4 - Post-conditions	17
3.1.5 - Fault conditions	17
3.1.5.1 - Fault Message Example	18
3.2 - Subsequent Brokered Search Function	19
3.2.1 - Preconditions	19
3.2.2 - Input	19
3.2.3 - Output	21

3.2.4 - Post-conditions	22
3.2.5 - Fault Conditions	22
3.3 - OSDD based Source Description	22
Chapter 4 - CDR Conformance Validation	24
Appendix A - Feature Summary	25
Appendix B - Change History	26
Appendix C - Mapping to Specification Framework	27
C.1 - Brokered Search Function	27
C.2 - Source Identification Function	27
Appendix D - Changes from Prior Version	29
Appendix E - Acronyms	31
Appendix F - Bibliography	37
Appendix G - Points of Contact	40
Appendix H - IC CIO Approval Memo	41

List of Figures

Figure 1 - CDR Architecture Model	2
---	---

List of Tables

Table 1 - Dependencies	4
Table 2 - Referenced XML Namespaces	4
Table 3 - Initial Request Parameters	11
Table 4 - Response Elements	14
Table 5 - /cdrb:sourceStatus in the Response	15
Table 6 - Faults and HTTP Status	17
Table 7 - Faults in sourceStatus	18
Table 8 - Subsequent Request Parameters	20
Table 9 - cdrb:sourceDescription	22
Table 10 - Revision History	26
Table 11 - Specification Framework Input Variables	27
Table 12 - Specification Framework Output Variables	27
Table 13 - Specification Framework Input Variables	27
Table 14 - Specification Framework Output Variables	28
Table 15 - Changes From V1.1 to V2.0	29
Table 16 - Acronyms	31

Chapter 1 - Introduction

1.1. Purpose

The **Brokered Search Component**, as defined by the IC/DoD Content Discovery and Retrieval (CDR) Specification Framework (CDR-SF),^[6] serves as the primary mechanism to facilitate the distribution of search requests to applicable/relevant sources¹ and aggregate the results returned from different sources. The identification of appropriate sources is an additional capability towards source description and source routing

This specification defines requirements and provides guidelines for the realization of the CDR Brokered Search Component as a RESTful², OpenSearch^[16]³ web service, hereafter termed a broker in this document. The content of this specification describes a broker's behavior, interface and other aspects in detail, providing enough information for Brokered Search providers and consumers to create and use CDR-conformant brokers.

1.2. Scope

This specification defines the RESTful interface between the consumer and the broker to support Brokered Search activities. In particular, the broker's behavior and interfaces are specified to enable stateful/stateless search, synchronous/asynchronous response, broker to broker search, source identification, heterogeneous sources invocation (CDR REST/SOAP conformant sources and non-CDR conformant sources), results aggregation and results paging.

The following capabilities, while related with the broker, are out of the scope of the specification:

- Mediation/translation processes that may be implemented by the broker
- Adding a new source that the broker can use as a federation target
- The process for the consumer to obtain the descriptions of the sources through a source registry
- Routing to unknown sources that are not recognized by the broker
- De-duplicating the same results returned from multiple sources

1.3. Background

This specification is a part of the set of specifications that define the concrete, implementation-specific guidance for the services defined under the auspices of the Content Discovery & Retrieval (CDR) Integrated Project Team (IPT). The CDR Architecture prescribes an abstract-to-concrete model for the development of architecture elements and guidance for content discovery and retrieval. Each layer or tier of the model is intended to provide key aspects of the overall guidance to achieve the goals and objectives for joint DoD/IC content discovery and retrieval. The following graphic, discussed in detail within the CDR Reference Architecture (CDR-RA),^[2] illustrates this model.

¹A source may be a content collection component or another Broker Search component. It acts as the federation target defined in CDR-SF^[6] for Brokered Search Component.

²REST is an architectural style that encapsulates the design principles of the World Wide Web (WWW).

³The OpenSearch specification can be found at <http://www.opensearch.org/Specifications/OpenSearch/1.1/>. Previous drafts can be found by appending Draft_n to the end of the URL, where n is the draft of interest. This specification was originally written using Draft_4 as its basis and is now based on Draft_5.

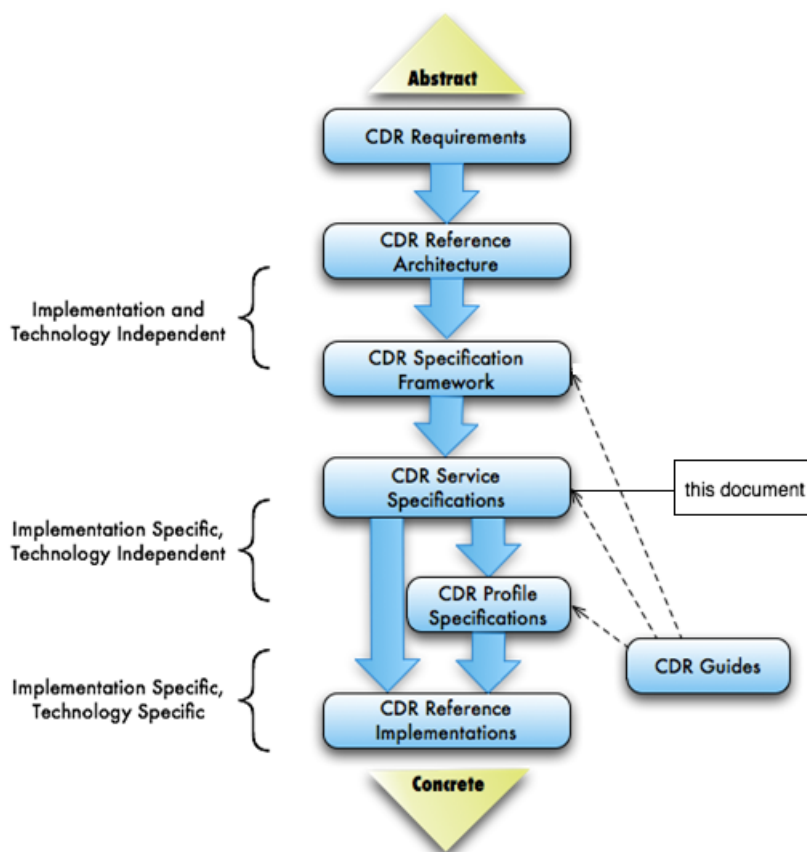


Figure 1 : CDR Architecture Model

As illustrated in Figure 1, the CDR Specification Framework^[6] derives from the CDR Reference Architecture^[2] and describes behavior in terms of the capabilities, components, and usage patterns defined in the CDR-RA.^[2] Multiple CDR Service Specifications are derived from the CDR-SF,^[6] with separate specifications associated with the components of the architecture (e.g., Brokered Search) and, for each service, separate specifications to address Representational State Transfer (REST) and SOAP implementations.

This document is a specification for implementing the CDR Brokered Search Component using the RESTful OpenSearch^[16] standard. It is intended to parallel the corresponding SOAP specification, the IC/DoD SOAP Interface Encoding Specification for CDR Brokered Search (CDR-SB),^[5] as closely as possible, to minimize the difficulties in interoperating. Additional CDR Guides, Profile Specifications, or Reference Implementations may provide additional guidance on implementing this specification in a particular context.

1.4. Enterprise Need

Enterprise needs and requirements for this specification can be found in the following policies and implementation guidance.

- IC Information Technology Enterprise (IC ITE)

- Intelligence Community Information Technology Enterprise (IC ITE) Increment 1 Implementation Plan^[8]
- 500 Series:
 - Intelligence Community Directive (ICD) 501, Discovery and Dissemination or Retrieval of Information within the IC^[10]
 - Joint IC/DoD Memorandum, IC and DoD Commitment to an Interoperable Service-Based Environment (13 Jul 07)^[15]

1.5. Audience and Applicability

Data Encoding Specifications (DESSs) are primarily intended to be used by those developing tools and services to create, modify, store, exchange, search, display, or further process the type of data being described.

Within the IC, the conditions of use and applicability of this technical specification are defined outside of this technical specification. IC Standard (ICS) 500-20, Intelligence Community Enterprise Standards Compliance, defines the IC Enterprise Standards Baseline (IC ESB) and the applicability of such to an IC element. The IC ESB defines the compliance requirements associated with each version of a technical specification. Each version will be individually registered in the IC ESB. The IC ESB will define, among other things, the location(s) of the relevant artifacts, prescriptive status, and validity period, all of which characterize the version and its utility. Additional applicability and guidance may be defined in separate IC policy guidance.

Within the DoD, the conditions of use and applicability of this technical specifications are defined outside of this technical specification, and can be located within the DoD Information Technology Standards Profile Registry (DISR).

1.6. Conventions

The key words "MUST," "MUST NOT," "REQUIRED," "SHALL," "SHALL NOT," "SHOULD," "SHOULD NOT," "RECOMMENDED," "MAY," and "OPTIONAL" in this specification are to be interpreted as described in the IETF RFC 2119.^[13] When these words are not capitalized, they are meant in their natural-language sense.

When describing concrete XML schemas and example XML documents, this specification uses XPath as the notational convention. Each member of an XML schema is described using an XPath notation (e.g., /x:RootElement/x:ChildElement/@Attribute).

In a URL template, a parameter contained in curly brace, generally represented in the form {name}, is meant to be replaced with an actual value determined at run-time. An optional parameter in a URL template is one whose name is followed by ?, e.g., {name?} and it MAY be replaced by an empty string.

Examples in this text are distinguished by a blue border. These are meant to be illustrative and only one way that the described syntax can be used.

```
<atom:entry>
  <atom:title>This is an example.</atom:title>
</atom:entry>
```

1.7. Dependencies

This technical specification depends on the additional technical specifications or additional documentation listed in the following table. The documents listed below are referenced in this encoding specification, and are normative or informative as indicated in the dependencies table.

Table 1 - Dependencies

Name	Dependency Description
REST Interface Encoding Specification for Content Discovery and Retrieval Search (CDR-RS) ^[4]	CDR-RS ^[4] specifies the CDR Search interface for REST. This version of Brokered Search extends the CDR Rest Search Specification V3.0 (Normative).

1.8. Conformance

For an implementation to conform to this Brokered Search specification, it MUST adhere to all REQUIRED aspects of the specification. A REST broker extends the interface of CDR REST Search Service so it MUST conform to CDR REST Search Specification.^[4]

1.9. Version Policies

The CDR specifications have been developed and are periodically refreshed to incorporate evolving needs and experience. Work is scheduled to meet the most pressing needs for expanding scope and maintaining consistency across the CDR specification family. Whole number versioning, e.g. v1.0 to v2.0, is used for all new versions.

1.10. Namespaces

Namespaces referenced in this document and the prefixes used to represent them are listed in the following table.

Table 2 - Referenced XML Namespaces

Prefix	URI	Description
opensearch	http://a9.com/-/spec/opensearch/1.1/	OpenSearch 1.1
atom	http://www.w3.org/2005/Atom	Atom Syndication format ^[14]
cdrb	urn:cdr:broker:2.0	CDR Brokered Search Specification V2
cdrs	urn:cdr:search:3.0	CDR Search Specification V3.0

1.11. License

This specification is licensed under the Creative Commons Attribution-ShareAlike 2.5 Generic License (<http://creativecommons.org/licenses/by-sa/2.5/>), because it builds on the OpenSearch^[16] standard, which is licensed with the share-alike clause.

1.12. Security

This specification does not directly address security concerns. It will be necessary for any implementation of this specification to address security concerns relevant to the systems with which they interact and the corresponding governance bodies. Several aspects of Brokered Search, to include returning only the results for which the requesting entity is authorized, should be addressed in the detailed security plan of an implementation, but are out of scope for this document.

Chapter 2 - Development Guidance

2.1. Brokered Search

The primary function of a broker is to support federated search by distributing a consumer's search request to one or more sources and returning the aggregated search results from the sources. The broker extends CDR Search Service and defines additional request/response parameters as specified in Section 3. The broker extensions support the capabilities as described in this section.

2.1.1. Stateless/Stateful Search

A broker MAY process all search requests as a new search and no prior state/cache is associated with a search. Such a stateless search MAY provide a simple interface to a consumer. For a stateless search, the broker MUST return the results to the consumer synchronously, i.e. the broker MUST wait until the completion of federation results processing to respond to the consumer.

A broker MAY persist the information associated with an initial search request and provide stateful interaction in order to improve responsiveness and avoid redundant requests to the sources. For a stateful search, the broker response MUST contain the result set identification and the broker MAY support asynchronous response, i.e. the broker MAY return intermediate status information with an empty result to the consumer before the search results are sent. The interim response MAY contain the status for all participating sources to provide diagnostic information to the consumer. The status information could indicate items such as the overall progress for each identified source (e.g. waiting, processing, etc.) or whether results are being added to the set of search results (i.e. number of retrieved results is increasing).

Upon receiving the interim response, a consumer may continue to request the initial search results using the result set identifier. The broker MAY respond with status information or send the results back when it is ready.

Subsequent requests may be directed to the broker to access an identifiable subset of the search results associated with the result set identifier such as a new page or results from a particular source. The broker MAY respond with results as requested or with information on the progress in creating the set of search results.

2.1.2. Broker to Broker Search

A broker MAY act as a consumer and MAY distribute a search request to another broker, i.e. the latter broker may act as a source of the former broker. The second broker may pass the request to a third broker. As search propagation goes on, the search request may reach brokers across various networks. A loop may be created if the same broker receives the same search request again via different search propagation paths. Special consideration must be taken to prevent the potential loop situation; hence, the processing of a broker to broker search request SHOULD include sufficient routing processing in addition to search request processing.

To detect and avoid looping, this specification defines a path parameter to track the identity of brokers through which a search request passes. Each broker MUST include its identifier in the

path parameter, as defined in Section 3. A consumer is considered as broker type if the path parameter is found in its search request. When a consumer does not provide the path, it is assumed to be the search request initiator.

Each time a search request is sent from one broker to another, the path parameter **MUST** be updated by the sending broker to include the broker identifier of the sender. To avoid a loop situation, the receiving broker **MUST** check if its own broker identifier is already included in the path parameter value. If the broker's identifier is already in the path and the broker has available a mechanism to determine whether continuing the broker process is warranted, then the broker **MUST** use and **MUST** follow the results of this mechanism to determine whether to continue to distribute the search request. If the broker does not use this mechanism or the results are ambiguous, then the broker **MUST** terminate the search request as in the circumstance above where no mechanism is available. A broker **SHOULD** have documented the mechanism by which it would decide to continue the processing; however, the mechanism by which a broker would decide whether to proceed or terminate the search request is outside the scope of this specification.

Note that any broker may terminate what it recognizes as a loop, even if the received path information is not complete or accurate.

The broker identifier **MUST** be unique across the brokers. The examples of possible types of identifier include UUID, IC-ID^[9], URI, a fully qualified domain name (FQDN), a distinguished name (DN) as used in Lightweight Directory Access Protocol (LDAP), or names in unknown namespaces.

2.1.3. Heterogeneous Sources Invocation

A broker **MUST** have a specific set of sources it is prepared to invoke. Each source is assigned an identifier and the source identifier **MUST** be unique within the broker. The source identifier is used to link the routing information, the search result, or status information with a source.

A broker **MAY** distribute the search to a number of different types of sources and brokers. For brevity, the following terms will be used:

- REST source type: a source conforming to CDR-RS^[4]
- SOAP source type: a source conforming to CDR-SS^[7]
- REST broker type: a source conforming to this specification
- SOAP broker type: a source conforming to CDR-SB^[5]
- broker type : any broker whether a REST or SOAP broker type or a broker not conforming to CDR specifications
- non-broker type : any source that is not broker type

A broker receives a search request and uses it to assemble the search requests to send to each identified source:

- For a source of REST type or REST broker type, the broker **SHOULD** distribute a search request that includes the OpenSearch parameters in the original search request, with adjustment when applicable.
- For a source of SOAP type or SOAP broker type, a broker **SHOULD** map the original search request to a CDR SOAP Search request, with adjustment when applicable.

- For a non-CDR conformant source, a broker SHOULD create the search request based on the original request and the syntax supported by the source. This assumes that the broker has specialized knowledge of specific non-CDR proprietary interfaces.
- In addition, for a source of broker type, the broker MUST include its broker identifier in the path and pass it to the targeted broker.

A consumer may specify the expected number of results to view or a timeout value that indicates to the broker how long the requester is willing to wait for slow-to-respond sources.

- The broker MAY determine how to allocate the maximum results number across its sources. For example, it MAY request the same number of results from each source, or it MAY vary the number of results requested from the sources. Not all sources allow a consumer to dictate the maximum number of results, so this parameter MAY simply serve as a suggestion indicating the consumers desired maximum number of results.
- When passing a timeout value to a source, the broker MAY reduce the value to ensure adequate performance, such as accounting for time needed for its internal processing of results.

2.1.4. Results Aggregation and Paging

When the search results become available, the broker MAY wait for responses from all invoked sources, aggregate the entire results set and send the aggregated results to the consumer. This tends to be simpler to implement and has the benefit of providing better sorting capability.

The broker MAY aggregate on the partial result set using consumer specified criteria such as timeout or the number of results to be retrieved to reduce the amount of storage required of the broker for the subsequent search requests. A broker MAY drop the search results received after the requested timeout has expired or after the requested number of results is retrieved. The result set would then remain static for the subsequent requests and the continuity of search results is guaranteed when paging back and forth across subsets of search results.

A broker MAY return results to the consumer as soon as sufficient number of results are retrieved, e.g. the requested number of results in the first page is retrieved, which may provide reduced response time and better scalability across disparate environments.

Each result returned in a result set MUST indicate the source from which it originated. For a source of broker type, the search path MAY be included. This is similar to the path maintained for broker to broker search (Section 2.1.2) but it indicates the return path of results rather than the outbound path of the search request. A broker MAY provide relevance scores for individual search results with respect to the particular search with which it is identified. Relevance scores generated by a given source can be interpreted as relative relevance of results.

When a paging request is received, i.e., a subsequent search request with the result set identifier and *opensearch:startIndex* or *opensearch:startPage* specified, the broker MAY traverse the search results and create next page results based on the information from the original search request and result inspection. Various approaches are possible for the broker to select the results in the results page. For example, the broker MAY distribute the number of selected results equally between all sources or the broker MAY use the relevancy score to select the results.

2.2. Source Identification

2.2.1. Source Description

The broker MAY have an internal list of known sources and MAY provide the consumer with a means to select sources from that list. If the means to select sources is supported, then the broker SHOULD support a means to retrieve descriptions of the known sources.

A broker that implements the means to retrieve source description MUST provide a list of potential sources and SHOULD include links to the description of the source and MAY include elements of that description. As appropriate, the description MAY be embedded in an Opensearch Description Document (OSDD).

2.2.2. Source Routing

Two different strategies are discussed for a broker to identify a specific set of sources to which the search request can be routed.

2.2.2.1. Static routing

The consumer may specify routing information to limit the search request distribution to a subset of the identified sources. A *routeTo* parameter is specified in Section 3 that contains a list of source identifiers, to which the search request should be routed.

The use of *routeTo* parameter by a broker is optional. When supported, the *routeTo* parameter MAY not be provided in the Brokered Search request or its value MAY be empty. In such cases, the broker SHOULD select a default set or select the sources by use of query inspection mechanism as specified below.

2.2.2.2. Smart routing by Query Inspection (Non-Normative)

In addition to static methods already described, the broker may identify the sources by query inspection. The broker MAY select source based on the query submitted by the consumer and the properties of the sources. The properties of the source may be obtained through static configuration or through the dynamic mechanism defined by CDR Describe Specification (CDR-RD).^[3]

Chapter 3 - Definitions, Interfaces & Constraints

This section defines the broker interface and contains the technical descriptions of the functions through which the consumer will interact with the service. Support for input and output parameters for each function as well as the default behaviors is described in the following tables in terms of what is expected of the broker.

In addition, an optional feature of OSDD based source description is described in Section 3.3.

3.1. Brokered Search Function

A broker MUST implement the brokered search function for the consumer to initiate a new search.

3.1.1. Preconditions

The following preconditions MUST be satisfied if the brokered search function is to correctly process inputs and generate results and post-conditions as described.

1. The requester is authenticated and authorized according to applicable policy requirements for this function.
2. The broker has a set of known sources and can use the search interface of the sources.

3.1.2. Input

The broker is the application of an HTTP/HTTPS GET method (request) to a single information resource, as identified by a Uniform Resource Locator (URL).

3.1.2.1. HTTP Method

The Brokered Search Function MUST use the HTTP GET method.

3.1.2.2. URL Template

The URL Template is defined in the OpenSearch specification, and it is a required element in the OpenSearch description document (OSDD). URL Template is further described in the CDR Search REST specification.^[4]

An extension to OpenSearch has been developed in order to support the brokered search function. This extension adds the following request parameters:

- routeTo
- maxResults
- timeout
- path

An example URL template:

```
http://example.com/search?
q={searchTerms}&src={cdrb:routeTo?}&mr={cdrb:maxResults?}&mt={cdrs:
timeout?}&path={cdrb:path?}
```

Table 3 defines the parameters of the initial request. The use of the parameters is illustrated in the examples in Section 3.1.2.5.

Table 3 - Initial Request Parameters

Parameter Name and Description	Support
<p>/cdrb:routeTo</p> <p>A comma-separated list of known source identifiers to which the search request should be routed.</p> <p>The ordering of sources in the list SHOULD NOT be interpreted to indicate preference, i.e. it is NOT REQUIRED that the sources be invoked in the order listed.</p> <p>Restrictions: The value of source identifier MUST be a string of printable ASCII characters, URL encoded and MUST NOT contain commas. A list of source identifiers SHOULD be enumerated using the <i>sourceId</i> attribute of the <i>sourceDescription</i> element in the OpenSearch Description Document (OSDD) (See Section 3.2.3). The use of source identifier is further described in Section 2.1.3.</p> <p>Default: The value is empty by default. Section 2.2.2.1 defines the broker behavior when the value is empty.</p>	<p>MAY be supported by service.</p>
<p>/cdrb:maxResults</p> <p>The maximum number of results expected to be retrieved.</p> <p>The use of this parameter is further described in Section 2.1.3 and Section 2.1.4.</p> <p>Restrictions: The value MUST be a non-negative integer.</p> <p>Default: If the value is not provided by the consumer, the broker SHOULD use its own strategy to limit the maximum number of results (e.g. use of a default number)</p>	<p>MAY be supported by service.</p>

<p>/cdrs:timeout^a (formerly maxTimeout^b)</p> <p>As defined in CDR SOAP Search Specification,^[7] i.e. the maximum wait time, in milliseconds, within which the consumer expects to receive the search results.</p> <p>The use of this parameter is further described in Section 2.1.3 and 2.1.4.</p> <p>Restrictions: The value MUST be a non-negative integer.</p> <p>Default: If the value is not provided by the consumer, the broker SHOULD use its own strategy for timeout (e.g. use of default timeout value).</p>	<p>MAY be supported by service.</p>
<p>/cdrb:path</p> <p>A comma-separated list of broker identifiers that indicates the sequence of brokers through which a search request has been routed.</p> <p>The purpose of the parameter is to facilitate loop avoidance for broker to broker search. A broker MUST add its identifier to the end of the existing list and it MUST initialize the list if it does not exist.</p> <p>Restrictions: The value of the broker identifier MUST be a string of printable characters, URL encoded and MUST NOT contain commas. The broker identifier MUST be unique across the brokers. Section 2.1.2 describes the approaches to generate broker identifiers. When the path parameter is included, the length of search request URL MUST not exceed the maximum URL length that the receiving broker allows.</p> <p>Default: The value is empty by default and the consumer is considered as a search request initiator.</p>	<p>MUST be supported by service for broker to broker search</p>

^acdrs:timeout is not yet defined in CDR-RS^[4] and will be included by the CDR REST Search Specification in the future.

^bmaxTimeout was used in IC/DOD CDR REST interface for Brokered Search specification v1.1.

3.1.2.3. HTTP Message Header

There are no additional entries in the request message header beyond those defined in the HTTP specification.

3.1.2.4. HTTP Message Body

There is no request message body for this function.

3.1.2.5. Search Request - Message Example

The following examples should be interpreted in the context of the example URL template shown in Section 3.1.2.2.

Example initial search request from a search request initiator to broker:

```
HTTP message:
http://example.com/search?q=watson+ibm&src=abc,def&mt=3000&mr=1000

Results in:
GET /search?q=watson+ibm&src=abc,def&mt=3000&mr=1000 HTTP/1.1
Host: example.com
```

Example initial search request from broker "example.com" to broker "example2.com":

```
HTTP message:
http://example.com/search?q=watson
+ibm&src=xyz&mt=3000&mr=1000&path=example.com

Results in:
GET /search?q=watson+ibm&src=xyz&mt=3000&mr=1000&path=example.com HTTP/1.1
Host: example2.com
```

3.1.3. Output

The output of the broker is a set of results that describe resources matching the query provided in the search request. In addition to Brokered Search requirement specified in this section, the output MUST conform to the output defined by the CDR REST Search Specification.^[4]

3.1.3.1. HTTP Status Code

If the GET request is successful, the service will respond with a 200 OK Status Code and the content resource.

3.1.3.2. HTTP Message Header

The response header SHOULD include the content type (i.e., the content-type header field with a MIME type as the assigned value) and content encoding of the content resource so that the consumer may anticipate how it should be processed and to inform any user agents (outside the scope of CDR) that may be available in the future to assist in mediating the referenced information resource to a preferred format.

3.1.3.3. HTTP Message Body

The body of the HTTP message MUST consist of the result set consisting of zero or more results.

3.1.3.4. Response Elements

Brokered Search extension defines three primary response elements: *resultSource*, *resultSetID*, and *sourceStatus*, as specified in Table 4. The *sourceStatus* element has a number of child elements associated with it as specified in Table 5. The use of the parameters is illustrated in the examples in Section 3.1.3.5.

Table 4 - Response Elements

Attribute/Element Name and Description	Support
<p>/cdrs:resultSetID</p> <p>As defined in CDR SOAP Search Specification,^[7] i.e., a unique identifier to identify a result set corresponding to a search request.</p>	MAY be supported by Service.
<p>/cdrb:sourceStatus</p> <p>XML element whose child elements convey status information about the source identified by <i>cdrb:sourceStatus/@sourceId</i>.</p> <p>When <i>/cdrb:sourceStatus</i> is included, there MUST be one <i>sourceStatus</i> element for each source to which the search request was sent by the broker. The content of <i>sourceStatus</i> element MAY be extended with elements from another XML namespace to provide additional information from the broker.</p> <p>The input parameter <i>cdrb:includeStatus</i> is related with this element (See Table 8).</p>	MAY be supported by Service.
<p>/cdrb:resultSource</p> <p>XML element that MUST be included with each result in order to identify the source from which the result was found. The XML element MUST have the <i>sourceId</i> attribute (defined below) and MUST have a value of type string that contains the short name of the source, where the short name is defined in Table 9 and corresponds to <i>/cdrb:resultSource/@sourceId</i>.</p>	MUST be supported by Service.
<p>/cdrb:resultSource/@sourceId</p> <p>XML attribute whose value is a string conveying the source identifier, as defined for the <i>/cdrb:routeTo</i> parameter in Table 3.</p>	MUST be supported by Service.
<p>/cdrb:path</p> <p>A comma-separated list of broker identifiers that indicates the sequence of brokers through which a search result has been returned.</p> <p>The XML element is intended as an indication of the brokers involved for broker to broker search. A broker MAY add its identifier to the beginning of the existing list and it MAY initialize the list if does not exist.</p> <p>Use of the path parameter for activities such as auditing is outside the scope of this specification.</p>	MAY be supported by Service.

When */cdrb:sourceStatus* is provided in the broker's response to the initial search request, the status information provided MUST conform to that shown in Table 5. Additional status information MAY also be provided.

Table 5 - /cdrb:sourceStatus in the Response

Attribute/Element Name and Description	Support in the context of providing sourceStatus
/cdrb:sourceStatus/@sourceId XML attribute whose value is a string conveying the source identifier as defined for /cdrb:routeTo parameter in Table 3.	MUST be supported by Service.
/cdrb:sourceStatus/shortName XML element whose value is a string conveying the short name of the source, where the short name is defined in Table 9 and corresponds to /cdrb:sourceStatus/@sourceId.	MUST be supported by Service.
/cdrb:sourceStatus/Status The current status of a single source. It MUST contain one of the following values: <ul style="list-style-type: none"> • <i>excluded</i>: The source was excluded by the broker. There may be a number of reasons for excluding a source, for example, if a maximum number of sources is exceeded, or if the source does not support query parameters in the request. • <i>waiting</i>: The search request has been sent to the source, and the broker is waiting for a complete response from the source. • <i>error</i>: The source returned an error response. • <i>timeout</i>: The source failed to respond within the configured timeout period. • <i>processing</i>: The broker received a complete response from the source, but is processing the result set (e.g., converting format, merging with other results, re-ranking). • <i>complete</i>: A response was successfully received and the result set from this source has been processed. 	MUST be supported by Service.
/cdrb:sourceStatus/resultsRetrieved The number of search results that the broker retrieved from the source.	MUST be supported by Service.
/cdrb:sourceStatus/totalResults The number of total results matching the query, as reported by the source.	MAY be supported by Service.
/cdrb:sourceStatus/elapsedTime The elapsed time, in milliseconds, for the source response.	MAY be supported by Service.
/cdrb:sourceStatus/fault As defined in Section 3.1.5.	MAY be supported by Service.

3.1.3.5. Output Example

The results presentation SHOULD follow a documented Result Set specification.

The CDR Specification set includes a single predefined Result Set formatting definition that IC/DoD organizations can leverage in their search service implementations, the IC/DoD ATOM Data Encoding Specification for Content Discovery and Retrieval Result Sets^[1]. Implementers SHOULD consult appropriate policy and implementation guidance to determine requirements or recommendations concerning the use of particular Result Set formatting.

The following represents a sample output using the Atom format for the result set augmented with OpenSearch response elements as identified in CDR-IPT REST Search Specification.^[4] Search service implementations SHOULD include these OpenSearch response elements, although the exact mechanism and syntax for including these elements may vary depending on the Result Set specification being used. Consult the Atom specification^[14] and CDR ATOM Specification^[1] for specifics on using the Atom format.

The example below shows the output with two sources: the source "abc" is a source of non-broker type and source "def" is of broker type. The namespaces referenced in the example are listed in Table 2.

```
HTTP/1.1 200 OK
Content-Length: 300
Content-Type: application/atom+xml
<atom:feed>
  <atom:id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</atom:id>
  <atom:title>Query Results for watson ibm</atom:title>
  <atom:updated>2003-12-13T18:30:02Z</atom:updated>
  <atom:author><atom:name>Enterprise Catalog</atom:name></atom:author>
  <opensearch:totalResults>492420</opensearch:totalResults>
  <opensearch:startIndex>1</opensearch:startIndex>
  <opensearch:itemsPerPage>10</opensearch:itemsPerPage>
  <atom:link rel="self" href="http://example.com/search?q=watson
+ibm&src=abc,def&mt=3000&mr=1000" type="application/atom+xml"/>
  <cdrs:resultSetID>1234</cdrs:resultSetID>
  <cdrb:sourceStatus cdrb:sourceId="abc">
    <cdrb:shortName>non-broker</cdrb:shortName>
    <cdrb:status>waiting</cdrb:status>
    <cdrb:resultsRetrieved>100</cdrb:resultsRetrieved>
    <cdrb:totalResults>222222</cdrb:totalResults>
    <cdrb:elapsedTime>2000</cdrb:elapsedTime>
  </cdrb:sourceStatus>
  <cdrb:sourceStatus cdrb:sourceId="def">
    <cdrb:shortName>broker</cdrb:shortName>
    <cdrb:status>waiting</cdrb:status>
    <cdrb:resultsRetrieved>50</cdrb:resultsRetrieved>
    <cdrb:totalResults>12345</cdrb:totalResults>
    <cdrb:elapsedTime>1500</cdrb:elapsedTime>
  </cdrb:sourceStatus>
  <atom:entry>
    <atom:id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af7</atom:id>
    <atom:title>IBM Watson</atom:title>
```

```
<atom:updated>2011-02-21T00:00:00Z</atom:updated>
<atom:link rel="alternate" href="http://www-03.ibm.com/innovation/us/
watson/index.html"/>
<relevance:score>0.97</relevance:score>
<cdrb:resultSource cdrb:sourceId="abc">non-broker</cdrb:resultSource>
</atom:entry>
<atom:entry>
<atom:id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af8</atom:id>
<atom:title>IBM Watson</atom:title>
<atom:updated>2011-02-21T00:00:00Z</atom:updated>
<atom:link rel="alternate" href="http://www-02.ibm.com/innovation/us/
watson/index.html"/>
<relevance:score>0.95</relevance:score>
<cdrb:resultSource cdrb:sourceId="def">broker</cdrb:resultSource>
<cdrb:path>example.com,example2.com</cdrb:path>
</atom:entry>
...
</atom:feed>
```

3.1.4. Post-conditions

The following conditions MUST be met upon completion of a search.

1. The results available to be returned to the requester are relevant to the input query.
2. The response will consist of a list of results or an appropriate fault.
3. The results are in the correct format.
4. The authenticated requester has been authorized to receive each result in the response.
5. The use of this function has been audited according to applicable policy.

3.1.5. Fault conditions

Besides common fault conditions that a Search Service implementation SHOULD handle, there are Brokered Search specific errors. Table 6 and Table 7 below indicate the HTTP status codes that SHOULD be referenced, and fault types of *sourceStatus*, respectively.

Table 6 - Faults and HTTP Status

Fault	HTTP Status	HTTP Status Description
Brokered Search Properties Fault	400	Bad Request
Duplicated Request	400	Bad Request
Unknown Source Fault	400	Bad Request
Size of the Path Parameter Over the limit	414	URI too long
Merge Fault	500	Server Error

Broker timeout	504	Gateway Timeout
----------------	-----	-----------------

If errors occur when invoking a source or processing the results from the source, the faults are returned in the "sourceStatus" element of the corresponding source. The search request has been accepted for processing and the HTTP response code 200 is returned.

Table 7 - Faults in sourceStatus

/cdrb:sourceStatus/fault	Fault Descriptions
SourceInvocation	A fault used if the broker cannot invoke the identified source.
ResultsProcessing	<p>A fault used when the broker cannot process the results set of an individual source .</p> <p>This may indicate an error in the returned results set or an inconsistency in interpreting the results set specified format.</p>

3.1.5.1. Fault Message Example

The following shows a fault message of type SourceInvocation. The namespaces referenced in the example are listed in Table 2.

```
HTTP/1.1 200 OK
Content-Length: 300
Content-Type: application/atom+xml
<atom:feed>
  <atom:id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</atom:id>
  <atom:title>Query Results for watson ibm</atom:title>
  <atom:updated>2003-12-13T18:30:02Z</atom:updated>
  <atom:author><atom:name>Enterprise Catalog</atom:name></atom:author>
  <opensearch:totalResults>492420</opensearch:totalResults>
  <opensearch:startIndex>1</opensearch:startIndex>
  <opensearch:itemsPerPage>10</opensearch:itemsPerPage>
  <atom:link rel="self" href="http://example.com/search?q=watson
+ibm&src=abc,def&mt=3000&mr=1000" type="application/atom+xml"/>
  <cdrs:resultSetID>1234</cdrs:resultSetID>
  <cdrb:sourceStatus cdrb:sourceId="abc">
    <cdrb:shortName>My Source</cdrb:shortName>
    <cdrb:status>waiting</cdrb:status>
    <cdrb:resultsRetrieved>100</cdrb:resultsRetrieved>
    <cdrb:totalResults>222222</cdrb:totalResults>
    <cdrb:elapsedTime>2000</cdrb:elapsedTime>
  </cdrb:sourceStatus>
  <cdrb:sourceStatus cdrb:sourceId="def">
    <cdrb:shortName>Other Source</cdrb:shortName>
    <cdrb:status>error</cdrb:status>
    <cdrb:resultsRetrieved>0</cdrb:resultsRetrieved>
    <cdrb:fault>SourceInvocation</cdrb:fault>
  </cdrb:sourceStatus>
```

```
<atom:entry>
  <atom:id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af7</atom:id>
  <atom:title>IBM  Watson</atom:title>
  <atom:updated>2011-02-21T00:00:00Z</atom:updated>
  <atom:link rel="alternate" href="http://www-03.ibm.com/innovation/us/
watson/index.html"/>
  <relevance:score>0.97</relevance:score>
  <cdrb:resultSource cdrb:sourceId="abc">My Source</cdrb:resultSource>
</atom:entry>
...
</atom:feed>
```

3.2. Subsequent Brokered Search Function

A broker MAY implement subsequent brokered search function to support stateful searches.

A broker MAY support result paging with cached results using the subsequent brokered search function. Note, the OpenSearch specification defines paging parameters and behavior but does not specify explicit support for accessing cached results. Please refer to IC/DoD REST Search Specification^[4] for interface definitions for a stateless search without results caching.

3.2.1. Preconditions

In addition to the preconditions of Brokered Search as specified in Section 3.1.1, the subsequent search function **REQUIRES** that:

1. An initial search request has been made and the broker has established the result set associated with this request.
2. The broker has returned a resultSetID in its initial response to the Search Request.
3. The result set identified by the resultSetID as defined in Table 4 must still be accessible through reference to the resultSetID.

3.2.2. Input

The subsequent search function is realized by the extension to OpenSearch. The following request parameters are added:

- resultSetID
- sourceFilter
- includeStatus

An example URL template:

```
http://example.com/subsequentSearch?
id={cdrs:resultSetID}&filter={cdrb:sourceFilter?}&
status={cdrb:includeStatus?}&start={startIndex?}&count={count?}
```

The results paging functionality can be realized by using the URL template for subsequent search request and varying the values of */opensearch:startIndex* or */opensearch:startPage* (The use of */opensearch:startIndex* and */opensearch:startPage* is described in CDR-RS.[\[4\]](#))

Table 8 defines the parameters of the subsequent request.

Table 8 - Subsequent Request Parameters

Parameter Name and Description	Support in the context of Subsequent Search Request
<p><i>/cdrs:resultSetID</i>^a (formerly <i>queryId</i>^b)</p> <p>As defined in Table 3.</p> <p>This parameter MUST be provided in a subsequent search request.</p> <p>Restrictions: The value MUST be a string and MUST have been provided in the output of a previous search request.</p> <p>Default: The value is empty by default. A valid value MUST be provided to avoid error.</p>	<p>MUST be supported by service</p>
<p><i>/cdrb:sourceFilter</i></p> <p>A source identifier as defined for <i>/cdrb:routeTo</i> parameter in Table 3, that is used in a subsequent request to constrain a result set to contain only results from the indicated source.</p> <p>Restrictions: The value MUST be the URL encoded identifier for one source.</p> <p>Default: The value is empty by default, i.e. without source filtering.</p>	<p>MAY be supported by service.</p>

<p>/cdrb:includeStatus</p> <p>A Boolean value that indicates to the broker whether or not the consumer wishes to receive status information on the sources with the results.</p> <p>If the broker supports this parameter, then a value of 1 MUST result in the status information being returned with the results. A value of 0 MUST result in the status information not being included in the response.</p> <p>Note that for the initial search request, it is up to the broker to decide whether or not to include source status in the response and the consumer cannot specify the includeStatus option.</p> <p>Restrictions: The value MUST be 0, 1.</p> <p>Default: The value is 0 by default.</p>	<p>MAY be supported by Service.</p>
---	-------------------------------------

^acdrs:resultSetID is used only for results paging request in CDR-SS.^[7] It is used in the subsequent requests for Brokered Search to support stateful interaction. It is not yet defined in CDR-RS^[4] and will be included by the CDR REST Search Specification in the future.

^bqueryId was used in IC/DOD CDR REST interface for Brokered Search specification v1.1.

Example subsequent search request:

```
HTTP message:
http://example.com/subsequentSearch?
id=1234&status=1&filter=abc&start=100&count=10

Results in
GET /subsequentSearch?id=1234&status=1&filter=abc&start=100&count=10 HTTP/1.1
Host: example.com
```

3.2.3. Output

The output of subsequent brokered search function is the same as the output of brokered search function defined in Section 3.1.3. The namespaces referenced in the example are listed in Table 2.

Example output:

```
HTTP/1.1 200 OK
Content-Length: 300
Content-Type: application/atom+xml
<atom:feed>
  <atom:id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</atom:id>
  <atom:title>Query Results for watson ibm</atom:title>
  <atom:updated>2003-12-13T18:30:02Z</atom:updated>
  <atom:author><atom:name>Enterprise Catalog</atom:name></atom:author>
  <opensearch:totalResults>492420</opensearch:totalResults>
  <opensearch:startIndex>100</opensearch:startIndex>
```

```
<opensearch:itemsPerPage>10</opensearch:itemsPerPage>
<atom:link rel="self" href="http://example.com/subsequentSearch?
id=1234&status=1&filter=abc&start=100&count=10" type="application/atom+xml"/>
<cdrs:resultSetID>1234</cdrs:resultSetID>
...
</atom:feed>
```

3.2.4. Post-conditions

The post-conditions of subsequent brokered search function are identical to the post-conditions of brokered search function as specified in Section 3.1.4.

3.2.5. Fault Conditions

The fault conditions of subsequent brokered search function are the same as the fault condition of brokered search function defined in Section 3.1.5.

3.3. OSDD based Source Description

A broker MAY embed the descriptions of its sources in the OSDD file. The sourceDescription element is an extension to the core OpenSearch elements.¹ The use of OSDD based source description is OPTIONAL. When supported, a *sourceDescription* element MUST be provided for each source known to the broker.

Table 9 - cdrb:sourceDescription

Attribute/Element Name and Description	Support in the context of using OSDD Based Source Description
/cdrb:sourceDescription XML element whose child elements convey description information of a source.	MUST be supported by service.
/cdrb:sourceDescription/@sourceId XML attribute whose value is a source identifier as defined for / <i>cdrb:routeTo</i> parameter in Table 3.	MUST be supported by service.
/cdrb:sourceDescription/shortName XML element whose value is a brief human-readable title that identifies this source. It MUST be composed of 16 or fewer characters of plain text and MUST NOT contain HTML and other markup.	MUST be supported by service.

¹ <http://www.opensearch.org/Specifications/OpenSearch/1.1#Extensibility>

<p>/cdrb:sourceDescription/longName</p> <p>XML element whose value MUST be a human-readable, extended name identifying the source. It MUST be composed of 48 or fewer characters of plain text and MUST NOT contain HTML and other markup.</p>	<p>MAY be supported by service.</p>
<p>/cdrb:sourceDescription/description</p> <p>XML element whose value MUST be a human-readable description of the source. It MUST be composed of 1024 or fewer characters of plain text and MUST NOT contain HTML and other markup.</p>	<p>MAY be supported by service.</p>
<p>/cdrb:sourceDescription/link</p> <p>An empty XML element that MUST convey the attributes as defined below:</p> <ul style="list-style-type: none"> • <i>href</i>: REQUIRED. Contains a URL for the linked resource. • <i>rel</i>: REQUIRED. Contains a string representing the relationship of the linked resource. Its value MAY be self: The linked resource provides additional information about the source and the content collection(s) it exposes. The list of allowed rel values is defined in the OpenSearch specification.^[16] • <i>type</i>: REQUIRED. Contains an Internet Assigned Numbers Authority (IANA) content type for the linked resource. 	<p>MAY be supported by service.</p>

Example source XML as part of the OSDD:

```
<cdrb:sourceDescription cdrb:sourceId="abc">
  <cdrb:shortName>My Source</cdrb:shortName>
  <cdrb:longName>My Example Source</cdrb:longName>
  <cdrb:description>An OpenSearch service for foo data.</cdrb:description>
  <cdrb:link rel="self" type="text/xml" href="http://example.com/
description.xml"/>
</cdrb:sourceDescription>
<cdrb:sourceDescription cdrb:sourceId="xyz">
  <cdrb:shortName>My Source 2</cdrb:shortName>
  <cdrb:longName>My Example Source 2</cdrb:longName>
  <cdrb:description>An OpenSearch service for bar data.</cdrb:description>
  <cdrb:link rel="self" type="text/xml" href="http://example2.com/
description.xml"/>
</cdrb:sourceDescription>
```

Chapter 4 - CDR Conformance Validation

Conformance is defined herein as “adherence to all relevant organizationally-mandated standards within a related family of specifications such that all implementations of web services based on these standards will achieve measurable levels of increased interoperability with others implementing the same standards.” Conformance is not necessarily compliance. Conformance addresses specific technical interoperability while compliance is associated with standards policy adherence as a whole. While not sufficient by itself, technical conformance is a necessary condition that helps in validating compliance with specifications.

Conformance of an implemented service to this specification will be validated by using certified tests or other means that are approved by the relevant authority by which this standard is mandated. In the absence of a certified test to validate the technical conformance, documentation that enables unambiguous traceability from each specification requirement to satisfactory execution of that requirement in the implementation must be provided. This is often in the form of a mapping matrix called a Conformance Test Matrix (CTM).

Note: The ability to validate conformance is greatly enhanced by the use of the OASIS Test Assertion Guidelines, both for articulating individual specification requirements during specification authoring, and for developing the associated implementation tests.

Appendix A Feature Summary

This document specifies the Brokered Search interface and behaviors and provides the implementation guidance for the following activities as defined in the Specification Framework:^[6]

- The brokered search coordination activity is the primary entry point to the brokered search function and provides coordination of the other activities.
- The source identification activity uses criteria provided by or derived from the consumer (which could be another broker) to identify sources to which a search request should be directed.
- The search component invocation activity sends a search request to each identified source and receives search results or fault returned by each source. The search component invocation activity does not interact directly with the consumer therefore there is no interface associated with it.
- The federation results processing activity performs the required processing necessary to combine the outputs of the individual search component invocation activities into a single uniform result set per any processing instructions provided in the brokered search properties. The federation results processing activity does not provide standalone functionality to warrant its own interface.

An explicit mapping between this specification to the corresponding sections in CDR-SF^[6] is found in Appendix C.

Appendix B Change History

The following table summarizes the version identifier history for this DES.

Table 10 - Revision History

Doc Revision	Revised By	Revision Date	Revisions
0.1		May 5 2010	Initial draft for subgroup review.
0.2	Dave Lemen		Revised based on comments.
0.3	Dave Lemen	June 19 2010	Revised based on comments, harmonized with SOAP spec.
0.4	Pam Preaseau	August 30 2010	Technical Edits
1.1	Dave Lemen	October 25, 2010	Revisions based on community comments
2.0 draft v1	Lisa Xie	Feb 25, 2013	Revised towards content re-writing, re-formatting, use of consistent examples
2.0 draft v2	Lisa Xie	March 11, 2013	Revised based on the subgroup comment for draft v1 and to add broker to broker support
2.0-draft v3	Lisa Xie	March 25, 2013	Revised based on the subgroup comment for draft v2
2.0-draft v4	Lisa Xie	Apr 10, 2013	Revised based on the subgroup comment for draft v3
2.0-draft v5	Lisa Xie	Apr 24, 2013	Revised based on the subgroup comment for draft v4
2-0-draft v6	Lisa Xie	May 9th, 2013	Revised based on the subgroup comment for draft v5

Appendix C Mapping to Specification Framework

This section explicitly ties the items in this specification to the requirements of the CDR-SF.^[6] The CDR-SF^[6] identifies the requirements for creating specifications, while implementation details are outlined in this document.

C.1. Brokered Search Function

Input

Table 11 maps the Brokered Search Function inputs in the CDR-SF^[6] to the elements and attributes defined in this specification.

Table 11 - Specification Framework Input Variables

Specification Framework Variable	REST Brokered Search Specification
Search Function Inputs	[Search Component Inputs]
Brokered Search Properties	cdrb:routeTo, cdrb:maxResults, cdrb:path

Output

Table 12 maps the Brokered Search Function outputs in the CDR-SF^[6] to the elements and attributes defined in this specification.

Table 12 - Specification Framework Output Variables

Specification Framework Variable	REST Brokered Search Specification
Merged Results Set and associated output	[Search Component Outputs]
Total Results by Source	cdrb:totalResults, cdrb:resultsRetrieved

C.2. Source Identification Function

Input

Table 13 maps the Brokered Search Function inputs in the CDR-SF^[6] to the elements and attributes defined in this specification.

Table 13 - Specification Framework Input Variables

Specification Framework Variable	REST Brokered Search Specification
Search Function Inputs	[Search Component Inputs]

Output

Table 14 maps the Brokered Search Function outputs in the CDR-SF^[6] to the elements and attributes defined in this specification.

Table 14 - Specification Framework Output Variables

Specification Framework Variable	REST Brokered Search Specification
Identified Content Collections and access (as Results Set of Search Function Outputs)	cdrb:sourceStatus

Appendix D Changes from Prior Version

The following table outlines the significant changes that were made from **IC/DoD Content Discovery & Retrieval Brokered Search Service Specification for OpenSearch Implementations V1.1** to the current version. These changes, shown in Table 15, were made to accommodate changes in the CDR-RA^[2] and CDR-SF^[6], to harmonize the content of this specification with that contained in the other specifications produced by this group, to incorporate feedback on the specification from pilot implementations, to resolve inconsistency across REST and SOAP specification for CDR Brokered Search and to introduce new capabilities.

Table 15 - Changes From V1.1 to V2.0

V1.1	V2.0	Rationale
First service specification developed.	Structure of document revised to reflect experience gained in writing other service specs and to use the latest EI&A template. Format of interface presentation revised to reflect experience gained in writing other service specs.	Leverage consistent structure, consistent format to make specs easier to read
	broker to broker search capability defined	New capability
Namespace fs	Namespace changed to cdrb	Resolve inconsistency between REST and SOAP specification
Section 2 Brokered Search Behavior	Section 2 Development Guidance redone	Provide more explicit guidelines for broker and resolve inconsistency across REST and SOAP specification for CDR Brokered Search
Section 3: Brokered Search Interface	Section 3 "Definitions, Interfaces and Constraint" Redone Brokered Search function input, output and fault examples updated accordingly.	Provide consistent structure and examples across REST and SOAP specification for CDR Brokered Search
Section 3 resultsRetrieved requirement	Section 3.1.3 resultsRetrieved requirement changed from OPTIONAL to REQUIRED	Resolve inconsistency between REST and SOAP specification

Section 3 maxTimeout and queryId	Section 3.1.2 maxTimeout replaced by timeout and queryId replaced by resultSetId	Synchronize with Search Specification 3.0
Section 3.4: Fault Conditions	Section 3.1.5 updated with sourceStatus faults.	Match with Specification Framework V2.0 Requirement Match with Search Specification 3.0 format
	Section 3.2 Subsequent Brokered Search Function Function added	Provide consistency and clarity
Section 3.2 OpenSearch Description Document Elements	Moved to Section 3.3 OSDD Based source description	Include supported interface under section 3
Section 4 Service Implementation	Section deleted and contents were merged with other sections where appropriate	Leverage consistent structure to make specs easier to read
Section 4.1 Policy	Merged with contents of section 1.3	Leverage consistent structure to make specs easier to read
Section 4.2 Query Extension Handling	Merged with contents of section 2	Leverage consistent structure to make specs easier to read
Section 4.3 Result Types	Merged with contents of section 3.1.3	Leverage consistent structure to make specs easier to read
Section 4.4 Security Consideration	Security statement in section 1.11	Leverage standard security statement for all service specifications.
Reference	Reference updated with the latest version. Missing references added Reference names modified for consistency.	Align with the latest CDR IPT specifications.
	Appendix added: mapping to specification framework	Align with the latest CDR IPT specifications.

Appendix E Acronyms

This appendix lists all the acronyms referenced in this encoding specification and lists other acronyms that may have been used in other encoding specifications. This appendix is a shared resource across multiple documents so in any given encoding specification there are likely acronyms that are not referenced in that particular encoding specification.

Table 16 - Acronyms

Name	Definition
A&A	Assessment and Authorization
AAS	Authoritative Attribute Sources
ABAC	Attribute Based Access Control
ABNF	Augmented Backus-Naur Form
ACSS	Allied Collaborative Shared Services
ADD	Abstract Data Definition
AICP	Authorized IC Person
AOI	Area of Interest
AOR	Area of Responsibility
API	Applications Programming Interface
APS	Attribute Practice Statement
ARH	Access Rights and Handling
AS	Attribute Service
ATO	Authority To Operate
BBOX	Bounding Box
BNF	Backus-Naur Form
CA	Certification Authority
CAPCO	Controlled Access Program Coordination Office
CAT	Catalog Services Interface Standard
CDR	Content Discovery and Retrieval
CF-NetCDF	Climate and Forecast - Network Common Data Format
CIA	Central Intelligence Agency
CIO	Chief Information Officer
CMS	Cryptographic Message Syntax
CNWDI	Critical Nuclear Weapons Design Information
COMET	Completely Open Mapping Environment
CONOPS	Concept of Operations
CORBA	Common Object Request Broker Architecture

Name	Definition
CQL	Common Catalog Query Language (CQL)
CRL	Certificate Revocation List
CSW	Catalog Service for Web
CTM	Conformance Test Matrix
CUI	Controlled Unclassified Information
CVE	Controlled Vocabulary Enumeration
D & R	Discovery and Retrieval
DAA	Designated Approval Agent
DC MES	Dublin Core Metadata Element Set
DCMI	Dublin Core Metadata Initiative
DDMS	Department of Defense Discovery Metadata Specification
DES	Data Encoding Specification
DI	Digital Identifier
DIA	Defense Intelligence Agency
DISR	DoD Information Technology Standards Registry
DN	Distinguished Name
DNI	Director of National Intelligence
DNS	Domain Name System
DOD	Department of Defense
DOE	Department of Energy
DOI	Digital Object Identifier
DOMEX	Document and Media Exploitation
EA	Enterprise Architecture
EI&A	Enterprise Integration and Architecture
E.O.	Executive Order
EBNF	Extended Backus-Naur Form
EDH	Enterprise Data Header
EPR	Endpoint Reference
ES&IS	Enterprise Search & Integration Services
ESB	Enterprise Standards Baseline
FD&R	Foreign Disclosure & Release
FOUO	For Official Use Only
FSD	Full Service Directory
FTP	File Transfer Protocol
FY	Fiscal Year

Name	Definition
GENC	Geopolitical Entities, Names, and Codes
GeoRSS	Geographic Really Simple Syndication
GeoTIFF	Geographic Tagged Image File Format
GIF	Graphics Interchange Format
GIS	Geospatial Information System
GML	Geography Markup Language
GNS	Geographic Names Server
GUIDE	Globally Unique Identifiers for Everything
GVS	GEOINT Visualization Services
HDF-EOS	Hierarchical Data Format - Earth Observing System
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
I2	Information Integration
IC	Intelligence Community
IC.ADD	Intelligence Community Abstract Data Definition
IC CIO	Intelligence Community Chief Information Officer
IC EA	IC Enterprise Architecture
IC ESB	Intelligence Community Enterprise Standards Baseline
IC ITE	IC Information Technology Enterprise
ICD	Intelligence Community Directive
ICEA	Intelligence Community Enterprise Architecture
ICPG	Intelligence Community Program Guidance
ICS	Intelligence Community Standard
ICSR	Intelligence Community Standards Registry
ICTS	Intelligence Community Technical Specification
IdAM	Identity and Access Management
IDM	Interface Data Model
IDMView	Interface Data Model View
IETF	Internet Engineering Task Force
IOC	Initial Operating Capability
IP	Internet Protocol
IPT	Integrated Project Team
IRM	Information Resource Metadata
ISBN	International Standard Book Number
ISM	Information Security Marking

Name	Definition
ISO	International Organization for Standardization
ISOO	Information Security Oversight Office
ITE	Information Technology Enterprise
JPEG	Joint Photographic Experts Group
JPIP	JPEG 2000 Interactive Protocol
JSON	JavaScript Object Notation
JWE	JSON Web Encryption
JWICS	Joint Worldwide Intelligence Communications System
JWT	JSON Web Token
KA	Knowledge Assertion
KML	Keyhole Markup Language
KOS	Knowledge Organization System
KVP	Key Value Pair
LDAP	Lightweight Directory Access Protocol
LIMDIS	Limited Distribution
LNI	Library of National Intelligence
MAC	Multi Audience Collection
MC&GIL	Mapping, Charting, and Geodesy Information Library
MC&GView	Mapping, Charting, and Geodesy View
MIME	Multipurpose Internet Mail Extensions
MTOM	Message Transmission Optimization Mechanism
NARA	National Archives and Records Administration
NATO	North Atlantic Treaty Organization
NCES	Net-Centric Enterprise Services
NGA	National Geospatial Intelligence Agency
NGDS	Net-Centric GEOINT Discovery Services
NGIC	National Ground Intelligence Center
NGT	Next Generation Trident
NIPRNet	Non-Classified Internet Protocol Router Network
NIEM	National Information Exchange Model
NIST	National Institute of Standards and Technology
NITF	National Imagery Transmission Format
NPE	Non-Person Entity
NMEC	National Media Exploitation Center
NRO	National Reconnaissance Office

Name	Definition
NSA	National Security Agency
NSG	National System for Geospatial Intelligence
NSI	National Security Information
NTK	Need-To-Know Metadata
OCIO	Office of the Intelligence Community Chief Information Officer
OCSP	Online Certificate Status Protocol
ODNI	Office of the Director of National Intelligence
OGC	Open Geospatial Consortium
OGCA	Open Geospatial Consortium Australia
OGCE	Open Geospatial Consortium Europe
ONEMail	Optimized Network Email
OPM	Office of Personnel Management
OWS	OGC Web Services
PAP	Policy Administration Point
PAYL	Payload
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PK	Private Key
PKI	Public Key Infrastructure
PNG	Portable Network Graphics
PUBS	Intelligence Publications
PURL	Persistent Uniform Resource Locator
RA	Reference Architecture
RDBMS	Relational Database Management System
REST	REpresentational State Transfer
RFC	Request for Comments
RR-ID	REST Security Encoding Specification for End-to-End Identity Propagation
SAML	Security Assertion Markup Language
SAP	Special Access Program
SCI	Sensitive Compartmented Information
SIPRNet	Secret Internet Protocol Router Network
SLA	Service Level Agreement
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SSD	Special Security Directorate

Name	Definition
SSL	Secure Sockets Layer
STIL	St Louis Information Library
TCP/IP	Transmission Control Protocol/Internet Protocol
TDC	Trusted Data Collection
TDF	Trusted Data Format
TDO	Trusted Data Object
TGN	Thesaurus of Geographic Names
TIFF	Tagged Image File Format
TIN	Triangulated Irregular Network
TLS	Transport Layer Security
TS	Top Secret
UAAS	Unified Authorization and Attribute Services
UIAS	Unified Identity Attribute Set
UDDI	Universal Description, Discovery and Integration
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
US	United States
UUID	Universal Unique Identifier
VIRT	Virtual Coverage
W3CDTF	World Wide Web Consortium Date Time Format
WARP	Web Based Access and Retrieval Portal
WCS	Web Coverage Service
WFS	Web Feature Service
WMS	Web Map Service
WSDL	Web Service Definition Language
XACML	eXtensible Access Control Markup Language
XML	Extensible Markup Language
XPath	XML Path Language
XPointer	XML Pointer Language
Xquery	XML Query
XSLT	XML Stylesheet Language for Transformations

Appendix F Bibliography

Bibliography

[1] ATOM.XML

Intelligence Community / Department of Defense Content Discovery & Retrieval Integrated Project Team. *Atom Data Encoding Specification for Content Discovery and Retrieval Result Sets*.

Available online Intelink-U at: <http://purl.org/IC/Standards/ATOM>

Available online at: <http://purl.org/IC/Standards/public>

[2] CDR-RA

Intelligence Community/Department of Defense Content Discovery & Retrieval Integrated Project Team. *Content Discovery and Retrieval Integrated Project Team Reference Architecture (CDR-RA)*.

Available online Intelink-U at: <http://purl.org/IC/Standards/CDR-RA>

Available online at: <http://purl.org/IC/Standards/public>

[3] CDR-RD

Intelligence Community/Department of Defense Content Discovery & Retrieval Integrated Project Team. *REST Interface Specification for Content Discovery and Retrieval: Describe (CDR-RD)*.

Available online Intelink-U at: <http://purl.org/IC/Standards/CDR-RD>

Available online at: <http://purl.org/IC/Standards/public>

[4] CDR-RS

Intelligence Community/Department of Defense Content Discovery & Retrieval Integrated Project Team. *REST Interface Encoding Specification for Content Discovery and Retrieval: Search (CDR-RS)*.

Available online Intelink-U at: <http://purl.org/IC/Standards/CDR-RS>

Available online at: <http://purl.org/IC/Standards/public>

[5] CDR-SB

Intelligence Community/Department of Defense Content Discovery & Retrieval Integrated Project Team. *SOAP Data Encoding Specification for Content Discovery and Retrieval: Brokered Search (CDR-SB)*.

Available online Intelink-U at: <http://purl.org/IC/Standards/CDR-SB>

Available online at: <http://purl.org/IC/Standards/public>

[6] CDR-SF

Intelligence Community/Department of Defense Content Discovery & Retrieval Integrated Project Team. *Data Encoding Specification for Content Discovery and Retrieval: Specification Framework (CDR-SF)*.

Available online Intelink-U at: <http://purl.org/IC/Standards/CDR-SF>

Available online at: <http://purl.org/IC/Standards/public>

[7] CDR-SS

-
- Intelligence Community/Department of Defense Content Discovery & Retrieval Integrated Project Team. *SOAP Interface Encoding Specification for Content Discovery and Retrieval: Search (CDR-SS)*.
Available online Intelink-U at: <http://purl.org/IC/Standards/CDR-SS>
Available online at: <http://purl.org/IC/Standards/public>
- [8] IC ITE INC1 IMPL
Office of the Director of National Intelligence. *Intelligence Community Information Technology Enterprise (IC ITE) Increment 1 Implementation Plan*. July 2012.
Available online Intelink-TS at: <http://go.ic.gov/HvBHBmY>
- [9] IC-ID.XML
Office of the Director of National Intelligence. *Text and XML Data Encoding Specification for Intelligence Community Identifier (IC-ID.XML)*.
Available online Intelink-U at: <http://purl.org/IC/Standards/ICID>
Available online at: <http://purl.org/IC/Standards/public>
- [10] ICD 501
Office of the Director of National Intelligence. *Discovery and Dissemination or Retrieval of Information within the Intelligence Community*. Intelligence Community Directive 501. 21 January 2009.
Available online Intelink-TS at: <http://go.ic.gov/GG61roi>
Available online at: http://www.dni.gov/files/documents/ICD/ICD_501.pdf
- [11] ICPG 710.1
Assistant Director of National Intelligence for . *Application of Dissemination Controls: Originator Control*. Intelligence Community Policy Guidance 710.1. 25 July 2012.
Available online Intelink-TS at: <http://go.ic.gov/yAqVQ0H>
- [12] ICS 500-20
Director of National Intelligence Chief Information Officer. *Intelligence Community Enterprise Standards Compliance*. Intelligence Community Standard 500-20. 16 December 2010.
Available online Intelink-TS at: <http://go.ic.gov/QUDIJKZ>
Available online Intelink-U at: https://intelshare.intelink.gov/sites/odni/cio/ea/library/Data%20Specifications/500-21/500_20_signed_16DEC2010.pdf
- [13] IETF-RFC 2119
Internet Engineering Task Force. *Key words for use in RFCs to Indicate Requirement Levels*. March 1997.
Available online at: <http://tools.ietf.org/html/rfc2119>
- [14] IETF-RFC 4287
M. Nottingham, R. Sayre. *The Atom Syndication Format*. December 2005.
Available online at: <http://www.ietf.org/rfc/rfc4287.txt>
- [15] Joint IC/DoD Memorandum
Intelligence Community Chief Information Officer, and Department of Defence Chief Information Officer. *IC and DoD Commitment to an Interoperable Service-Based Environment*. 13 July 2007.

Available online at: <http://www.docstoc.com/docs/797594/Department-of-Defense-DoD-and-Intelligence-Community-IC-Commitment-to-an-Interoperable-Services-Based-Environment---Enterprise-Services>

[16] OS

OpenSearch. *OpenSearch Specification*. Version 1.1.

Available online at: <http://www.opensearch.org>

Appendix G Points of Contact

The Intelligence Community Chief Information Officer (IC CIO) facilitates one or more collaboration and coordination forums charged with the adoption, modification, development, and governance of IC technical specifications of common concern. This technical specification was produced by the IC CIO and coordinated with these forums, approved by the IC CIO or a designated representative, and made available at DNI-sponsored web sites. Direct all inquiries about this IC technical specification to the IC CIO, an IC technical specification collaboration and coordination forum, or IC element representatives involved in those forums.

Public Website: <http://purl.org/ic/standards/public>

E-mail: <ic-standards-support@intelink.gov> .

Appendix H IC CIO Approval Memo

An Office of the Intelligence Community Chief Information Officer (OCIO) Approval Memo should accompany this enterprise technical data specification bearing the signature of the Intelligence Community Chief Information Officer (IC CIO) or an IC CIO-designated official(s). If an OCIO Approval Memo is not accompanying this specification's version release package, then refer back to the authoritative web location(s) for this specification to see if a more complete package or a specification update is available.

Specification artifacts display a date representing the last time a version's artifacts as a whole were modified. This date most often represents the conclusion of the IC Element collaboration and coordination process. Once the IC Element coordination process is complete, the specification goes through an internal OCIO staffing and coordination process leading to signature of the OCIO Approval Memo. The signature date of the OCIO Approval Memo will be later than the last modified date shown on the specification artifacts by an indeterminable time period.

Upon signature of the OCIO Approval Memo, IC Elements may begin to use this specification version in order to address mission and business objectives. However, it is critical for IC Elements, prior to disseminating information encoded with this new specification version, to ensure that key enterprise services and consumers are prepared to accept this information. IC Elements should work with enterprise service providers and consumers to orchestrate an orderly implementation transition to this specification version in concert with mandatory and retirement usage decisions captured in the IC Enterprise Standards Baseline as defined in Intelligence Community Standard (ICS) 500-20.^[12]