# Intelligence Community and Department of Defense Content Discovery & Retrieval Integrated Project Team (CDR IPT)

## *IC/DoD SOAP Interface Encoding Specification for CDR Brokered Search V1.1*

**12 May 2011**

# REVISION/HISTORY

| Doc Revision | Revised By | Revision Date | Revisions |
|---|---|---|---|
| 0.1 | CDR IPT | 17 March 2010 | Initial draft for subgroup review. |
| 0.2 | CDR IPT | 02 June 2010 | 0.1 Comment Adjudication |
| 0.3 | CDR IPT | 05 July 2010 | 0.2 Comment Adjudication |
| 0.3.1 | CDR IPT | 21 July 2010 | 0.3 Comment Adjudication |
| 0.4 | CDR IPT | 19 August 2010 | 0.3.1 Comment Adjudication |
| 0.4.1 | CDR IPT | 07 September 2010 | 0.4 Comment Adjudication |
| 0.4.2 | CDR IPT | 07 September 2010 | Candidate Release |
| 1.0-20100908 | CDR IPT | 08 September 2010 | Minor updates |
| 1.0-20100908 | Pam Preaseau | 10 September 2010 | Tech Edit |
| 1.0-20101026 | D. Wigglesworth | 26 October 2010 | Community Review |

# TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF TABLES

# 1 Introduction

## *1.1 Component Overview*

The Brokered Search Component, as defined by the IC/DoD Content Discovery and Retrieval (CDR) Specification Framework [SF], serves as the primary mechanism to 1) facilitate the distribution of queries to applicable/relevant Search Components and content collections. These Search Components expose and 2) aggregate the results returned individually into a single uniform results set.

This specification defines requirements and provides guidelines for the realization of the CDR Brokered Search Component in a web service using the SOAP messaging protocol, hereafter termed a ***Brokered Search Component*** in this document. Providing enough information for ***Broker Search Component*** providers and implementers to create CDR-compliant ***Brokered Search Components, the specification*** describes a ***Brokered Search Component's*** behavior, interface, and other aspects in detail.

A ***Brokered Search Component*** uses the basic functionality described by the Search Component for a single search. Additional inputs and outputs are defined as needed to support the four activities that underpin Brokered Search capabilities: brokered search coordination, source identification, search component invocation, and federation results processing. As discussed in CDR Specification Framework, a ***Search*** component's results are resource metadata rather than actual content resources. In the context of Search, resource metadata generally refers to a *subset* of a resource's available metadata, not the entire underlying record[1]. Some of the information contained within each Search result may provide the information necessary for a consumer to retrieve or otherwise use a resource.

Any resource may have associated policies for use. This is especially true for authentication and authorization. These policies may be asserted by both the resource owner and those responsible for governance and management of the enterprise. The implementation of policies related to security considerations SHOULD leverage the specific security components and interactions defined by the Joint IC/DoD Security Reference Architecture (SRA), and MUST be in compliance with requirements and guidance for security outcomes as specified in the SRA and its associated specifications.

### 1.1.1 Relationship to Other CDR Architecture Elements[2]

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

[1] *The Search Component returns metadata about a resource, which may sometimes describe the underlying resource (e.g., an image), while other times representing a sub-set of the data that makes up a resource (e.g., a collection of attributes). In some cases, the metadata returned from an instantiation of the Search function and the Retrieve function, which returns a resource itself, may happen to be the same, though this is considered an edge condition.*

[2] *For a detailed description of each of the layers, please reference the CDR RA Section 1.*

graphic, discussed in detail within the CDR Reference Architecture [RA], illustrates this model.

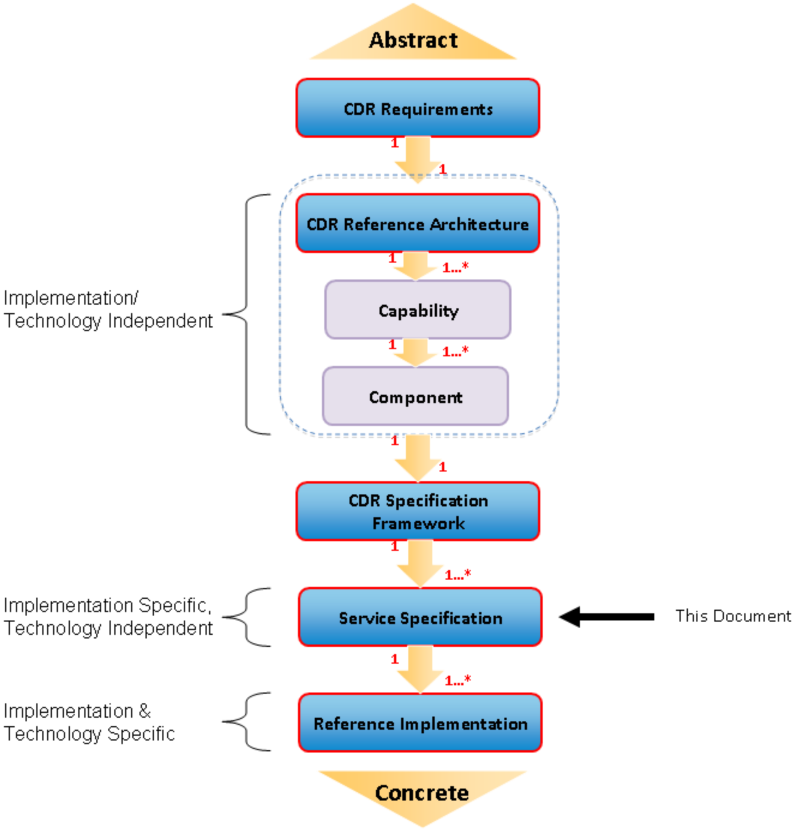

**Figure 1 – CDR Architecture Model**

As illustrated in Figure 1, the Specification Framework derives from the Reference Architecture (RA) and can describe behavior in terms of the capabilities, components, and usage patterns defined in the RA. The Specification Framework then expands on the details of information flows and the information conveyed in those flows providing a consistent basis for multiple Service Specifications which in turn provides consistent interfaces, both in terms of the structure and semantics of the exchanged information. Service Specifications, such as this one, provide implementation-specific guidance.

This specification covers the following aspects of a SOAP-based Brokered Search Component:

- **Service Behavior** maps the Brokered Search interaction patterns defined in the Specification Framework to concrete SOAP constructs.
- **Service Interface** defines the base SOAP constructs to express inputs, outputs, and faults.
- **Implementation** provides additional implementation guidance beyond service behavior and interface guidance.

- **Reference Documentation** provides references to other CDR and community artifacts (i.e., Security Reference Architecture).

## 1.2 Notational Convention

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. 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). The use of {any} indicates the presence of an element wildcard (<xs:any/>). The use of @{any} indicates the presence of an attribute wildcard (<xs:anyAttribute/>).

Examples in this text are distinguished by a black border and items that should be focused on are highlighted in yellow. These are meant to be illustrative and not the only way that the described syntax can be used. Examples may include several lines derived from the IC/DoD Search Specification for SOAP Implementations [S] sub-specifications, e.g. cdrs:Expression, and included solely to complete the example. These lines are presented in green.

## 1.3 Conformance

This specification defines the *IC/DoD Content Discovery & Retrieval Brokered Search Specification for SOAP Implementations*. For an implementation to conform to this ***Brokered Search Specification***, it MUST adhere to all mandatory aspects of this specification in addition to the *IC/DoD Content Discovery & Retrieval Search Specification for SOAP Implementations* to which this specification follows.

## 1.4 Namespaces

Table 1 identifies XML Namespaces that are directly leveraged in this document. Additional namespaces are introduced in the set of specifications that compliment core search functionality, including those used in specific query types (e.g., Keyword, XQuery), response types (e.g., Atom), and data standards (e.g., DDMS, IRM.XML).

**Table 1 – Referenced XML Namespaces**

| Prefix | URI | Description |
|---|---|---|
| soap | http://www.w3.org/2003/05/soap-envelope | W3C SOAP Version 1.2 |
| wsa | http://www.w3.org/2005/08/addressing | WS-Addressing Definition |
| wsaw | http://www.w3.org/2006/05/addressing/wsdl | WS-Addressing – WSDL Binding |

| | | |
|---|---|---|
| cdrs | urn:cdr:1.0:soap:search | CDR v1.0 Search Specification for SOAP Implementations |
| cdrb | urn:cdr:1.0:soap:broker | CDR v1.0 Brokered Search Specification for SOAP Implementations |
| cdrd | urn:cdr:1.0:soap:describe | CDR v1.0 Describe Specification for SOAP Implementations (proposed) |

# 2 Brokered Search Interface

The **Brokered Search Component** makes use of the **Search Component** Search interface as specified by the **Search Component** SOAP Specification.

## 2.1 Brokered Search Function

| Function | Input | Output | Fault |
|---|---|---|---|
| Search | *cdrs:SearchRequest* | {ResultSet}[3] | Defined within CDR Framework |

The ***Brokered Search*** specification is REQUIRED to function as described by the Content Discovery and Retrieval (CDR) Specification Framework with any input, behavior, output, and fault condition extensions listed below. These follow the inputs, outputs, and faults defined in section 2.1 for the Search Component and provide any additional details to adhere to the CDR Specification Framework's description of Brokered Search.

## 2.2 Input

The input to this function is REQUIRED to be compliant with the input defined by the CDR Search SOAP Specification. Table 2 shows each input variable defined in this specification, and maps each to the Brokered Search variables as defined in the IC/DoD Content Discovery and Retrieval Specification Framework [CDR-SF] (see Table 9 in the CDR-SF).

**Table 2 – Specification Framework Input Variables**

| Specification Framework Variables | SOAP Specification Variable | Required/Optional |
|---|---|---|
| Search Activity Inputs | [Search Component Inputs] | Per SOAP Specification |
| Brokered Search Properties | N/A | Optional |
| Identified Content Collections and access | routeTo | Optional |
| | wsa:EndpointReference | Optional |

[3] *{ResultSet} represents an unspecified results set. Its use is discussed in section 3.2.*

The following example illustrates a search request message to a ***Brokered Search Component***, using a notional keyword query (for more information on Query Types, please reference the IC/DoD Search Specification for SOAP Implementations [S]):

```
<soap:Envelope>
  <soap:Header>
    ...
    <wsa:Action>
      urn:cdr:1.0:soap:action:search
    </wsa:Action>
    ...
  </soap:Header>
  <soap:Body>
    <cdrs:SearchRequest
         startIndex=”1” resultsPerPage=”50” timeout=”20000”
         cdrb:routeTo=”exampleSource1, exampleSource2”>
      <wsa:EndpointReference cdrb:sourceId=”exampleSource1”>…</wsa:EndpointReference>
      <wsa:EndpointReference cdrb:sourceId=”exampleSource2”>…</wsa:EndpointReference>
      <cdrs:Query queryTypeURI=”cdr:1.0:soap:query:keyword”>
        <cdrs:Expression>
            UNMANNED AERIAL VEHICLE
        </cdrs:Expression>
      </Query>
    </cdrs:SearchRequest>
  </soap:Body>
</soap:Envelope>
```

Description of significant elements:

***/cdrs:SearchRequest***

This REQUIRED element, located directly inside the `env:Body`, encapsulates the search request, as shown in the example.

***/cdrs:SearchRequest/*@startIndex**[4]

This OPTIONAL element describes the desired start index of the search execution. Its value, if provided, MUST be greater than or equal to 1. The default value is 1.

***/cdrs:SearchRequest/*@resultsPerPage**[4]

This OPTIONAL element describes the desired number of search results per page. Its value, if provided, MUST be greater than or equal to 1. The default value is 10.

***/cdrs:SearchRequest/*@timeout**[4]

This OPTIONAL element describes the desired timeout period (in milliseconds) of a brokered search request. If present, a ***Brokered Search Component*** must return results (even if partial) by the end of the given timeout period. If no results are available at the end of the timeout period and the search has not yet completed, a fault SHOULD be returned. If partial results are returned due to a timeout, a service provider SHOULD return an appropriate indication in the "Results Metadata" of the Result Set (see 3.3.3 for more information).

***/cdrs:SearchRequest/*@cdrb:routeTo**

[4] *startIndex, resultsPerPage and timeout are specified as part of the Search Component interface*

This OPTIONAL attribute provides a comma-separated list of the source identifiers that identify the ***Search Component*** sources to which the Query should be distributed

***/cdrs:SearchRequest/wsa:EndpointReference***

This OPTIONAL element provides the endpoint of the ***Search Component*** source to which the Query should be distributed. A separate EndpointReference is necessary for each source identified in the routeTo attribute that is not implicitly understood by the Brokered Search Component.

***/cdrs:SearchRequest/wsa:EndpointReference/*@sourceId**

This OPTIONAL attribute associates the source identifier with the wsa:EndpointReference. This sourceId should match a sourceId in the routeTo attribute.

***/cdrs:SearchRequest/Query***

This REQUIRED element identifies the query being provided in the search request.

***/cdrs:SearchRequest/Query/*@queryTypeURI**

This REQUIRED element identifies the type of query being provided in the search request.

## 2.3 Output

The output is REQUIRED to be compliant to the requirements imposed by the CDR Search SOAP Specification. In addition, Table 3describes further output constraints on the ***Brokered Search Component's*** Search function:

**Table 3 – Specification Framework Output Variables**

| Specification Framework Variables | SOAP Specification Variable | Required/Optional |
|---|---|---|
| Search Activity outputs | [Search Component Outputs] | Per SOAP Specification |
| Identified Content Collections and access | wsa:EndpointAddress | Optional |
| | sourceStatus | Optional |

The following example illustrates the high level components of a response message (containing a Result Set of unspecified type) from a ***Brokered Search Component***:

```
<soap:Envelope>
  <soap:Header>

  ...
  <wsa:Action>
    urn:cdr:1.0:soap:action:response
  </wsa:Action>
  ...
  </soap:Header>
  <soap:Body>
    <{ResultSet}>
      <wsa:EndpointReference sourceId=”exampleSource1”>…</wsa:EndpointReference>
      <cdrb:sourceStatus sourceId=”exampleSource1”>
        <cdrb:status>…</cdrb:status>
        <cdrb:resultsRetrieved>…</cdrb:resultsRetrieved>
        <cdrb:totalResults>…</cdrb:totalResults>
      </cdrb:sourceStatus>
      <wsa:EndpointReference sourceId=”exampleSource2”>…</wsa:EndpointReference>
      <cdrb:sourceStatus sourceId=”exampleSource2”>
        <cdrb:status>…</cdrb:status>
        <cdrb:resultsRetrieved>…</cdrb:resultsRetrieved>
        <cdrb:totalResults>…</cdrb:totalResults>
      </cdrb:sourceStatus>
      <{Result} sourceId=”exampleSource1”>…</{result}>
      <{Result} sourceId=”exampleSource1”>…</{result}>
      <{Result} sourceId=”exampleSource2”>…</{result}>
    </{ResultSet}>
  </soap:Body>
</soap:Envelope>
```

Description of significant elements:

**/soap:Envelope/soap:Body/<{ResultSet}>**

This is a placeholder for the result set that holds the resource metadata corresponding to the individual results for each identified content collections

**/soap:Envelope/soap:Body/<{ResultSet}>/wsa:EndpointReference**

This OPTIONAL element provides the endpoint of the ***Search Component*** source to which the Query was distributed.

**/soap:Envelope/soap:Body/<{ResultSet}>/wsa:EndpointReference/@sourceId**

This OPTIONAL attribute associates the wsa:EndpointReference element with the source identifier. The sourceId should correspond to the sourceId specified in the routeTo attribute of the SearchRequest.**/soap:Envelope/soap:Body/<{ResultSet}>/cdrb:sourceStatus**

This OPTIONAL element indicates the individual status of the search requests issued to the collaborating ***Search Component*** implementations

**/soap:Envelope/soap:Body/<{ResultSet}>/cdrb:sourceStatus/@sourceId**

This OPTIONAL attribute associates the cdrb:sourceStatus element with the source identifier. The sourceId should correspond to the sourceId specified in the routeTo attribute of the SearchRequest.

**/soap:Envelope/soap:Body/<{Result}>**

This is a placeholder for the individual result containing the resource metadata

## *2.4 Fault Conditions*

An implementation of the ***Brokered Search Component*** MUST allow for the Fault Conditions defined in the CDR Specification Framework. This includes those specified in the *Search Component SOAP Specification*, in addition to those listed below:

- **Source Identification Fault** – A fault used if the Brokered Search implementation cannot identify Search Components to invoke

- **Search Component Invocation Fault** – A fault used if the Brokered Search implementation cannot invoke the identified Search Components. This fault MUST clearly identify which Search Component invocation caused the problem.

- **Federated Results Processing Fault** – A fault used when the Brokered Search implementation cannot process the results set of an individual Search Component Invocation. This may indicate an error in the returned results set or an inconsistency in interpreting the results set specified format.

# 3 Brokered Search Behavior

An implementation of the ***Brokered Search Component*** MUST follow the behavior defined in the CDR Specification Framework with the following extensions detailed in the subsequent sections.

## *3.1 Brokered Search Coordination*

As specified in the IC/DoD Content Discovery and Retrieval Specification Framework [CDR-SF], the Brokered Search Coordination activity is the primary entry point to the Brokered Search function and provides coordination of the other activities that identify, invoke, and process results from the federation targets. In addition to managing internal communications among the activities, the Brokered Search Coordination activity MUST manage individual federation target invocations and respond to information exchanges with the federation targets. It may also be the point of invoking mediation to enable a larger number of targets to participate.

## *3.2 Source Identification Activity*

As specified in the Content Discovery and Retrieval (CDR) Specification Framework, there are multiple strategies in the identification of **Search Component** sources, and, as noted in Section 2, a broker may implement a number of these. However, the most direct strategy is for the consumer to explicitly provide the sources.

### 3.2.1 Sources Identified from a static, internal list (`cdrb:routeTo`)

A Brokered Search Component implementation can identify a specific set of Search Components implementations it is prepared to invoke, and the service consumer may use the cdrb:routeTo attribute to limit the query to a subset of the identified Search Components. The cdrb:routeTo attribute contains a comma-separated list of source identifiers, to which the search query should be routed. The same identifier is also referenced by the sourceId attribute in response elements described in sections 3.2.4 and 3.4.1.

A Brokered Search Component MAY treat the routeTo attribute as optional. That is, if the routeTo attributer is missing, the broker may route the query to a default set of sources or route to a set of sources based on attributes of the query.

There is no significance to the order in which the sources are listed (i.e., it should not be assumed that the sources will be queried in the order they are listed in this attribute). The source identifiers MUST NOT contain commas.

If a source in the routeTo list is not recognized by the broker, it MUST return an Unknown Source Fault.

An example of the `cdrb:routeTo` attribute within a `cdrs:SearchRequest` is shown below:

```
<cdrs:SearchRequest cdrb:routeTo="exampleSource1, exampleSource2">
  …
</cdrs:SearchRequest>
```

Description of significant elements:

**/cdrs:SearchRequest/@cdrb:routeTo**
**Optional.** This attribute provides a comma-separated list of the source identifiers that identify the ***Search Component*** sources for which the Query should be distributed

### 3.2.2 Sources Identified by Consumer (`cdrb:routeTo, wsa:EndpointReference`)

An identified **Search Component** source may be explicitly specified by the service consumer in the SearchRequest using the `wsa:EndpointReference` element. If the `wsa:EndpointReference` is available, it MUST be resolved and the appropriate cdrs:SearchRequest MUST be distributed to that location.

An example of the `cdrb:routeTo` attribute being used with a `wsa:EndpointReference` within a `cdrs:SearchRequest` is shown below:

```
<cdrs:SearchRequest routeTo="exampleSource1, exampleSource2">
  <wsa:EndpointReference sourceId="exampleSource1">
    <wsa:Address>http://example.com/search/example1</wsa:Address>
    <wsa:Metadata>
      <wsaw:InterfaceName>cdrs:Search_PortType</wsaw:InterfaceName>
      <wsaw:ServiceName>example:SearchService1</wsaw:ServiceName>
    </wsa:Metadata>
  </wsa:EndpointReference>
  <wsa:EndpointReference sourceId="exampleSource2">
    <wsa:Address>http://example.com/search/example2</wsa:Address>
    <wsa:Metadata>
      <wsaw:InterfaceName>cdrs:Search_PortType</wsaw:InterfaceName>
      <wsaw:ServiceName>example:SearchService2</wsaw:ServiceName>
    </wsa:Metadata>
  </wsa:EndpointReference>

  …
</cdrs:SearchRequest>
```

Description of significant elements:

**/cdrs:SearchRequest/@cdrb:routeTo**

This OPTIONAL attribute provides a comma-separated list of the source identifiers that identify the ***Search Component*** sources for which the Query should be distributed

**/cdrs:SearchRequest/wsa:EndpointReference**

This OPTIONAL element provides the endpoint of the ***Search Component*** source in which the Query should be distributed

**/cdrs:SearchRequest/wsa:EndpointReference/@sourceId**

This OPTIONAL attribute associates the source identifier with the wsa:EndpointReference

### 3.2.3 Sources Identified by Consumer specified Criteria/Query Introspection *(Non-Normative)*

As specified in the Content Discovery and Retrieval (CDR) Specification Framework and in section 3.3.2, there are multiple strategies in the identification of **Search Component** sources. In addition to *Static* methods already described, the **Brokered Search Component** can identify the sources by use of query introspection.

To help facilitate the use of query introspection, a consumer may append additional description information to the query element. If this description information is included in the `cdrs:Query`, a ***Brokered Search*** can be used in conjunction with the Service Discovery Capability to determine the available ***Search*** implementations that are categorized under the requested description value. The ***Brokered Search*** implementation MUST then distribute requests to the selected ***Search Components***.

This introspection capability is not explicitly supported in the present specification.

### 3.2.4 Including the Identified Source in the {ResultSet} (`cdrb:sourceStatus`)

In order to record the ***Search*** component implementations involved in the collaboration, the Brokered Search Component MUST include one `cdrb:sourceStatus` element per identified ***Search Component*** in the {*ResultSet}*. Each `cdrb:sourceStatus` should articulate the identified source by including the sourceId attribute.

Example source XML:

```
<{ResultSet}>
  …
  <cdrb:sourceStatus sourceId="Intelink-Open">
     …
     <cdrb:status>complete</cdrb:status>
     <cdrb:resultsRetrieved>100</cdrb:resultsRetrieved>
     <cdrb:totalResults>222222</cdrb:totalResults>
  </cdrb:source>
  <cdrb:sourceStatus sourceId="AKO-DKO">
     …
     <cdrb:status>waiting</cdrb:status>
  </cdrb:source>
…
</{ResultSet}>
```

## 3.3 Search Component Invocation Activity

Once the participating ***Search Component*** implementations have been identified, a `cdrs:SearchRequest` must be created and it MUST be forwarded to each participating SOAP based ***Search Component*** implementation. The invoking consumer's identity MAY be propagated with the distributed query using the brokered trust mechanisms (including all user authentication and authorization information) as discussed in the Security Reference Architecture [SRA] and its associated sub-documents.

The search MAY be propagated concurrently or consecutively to each identified ***Search Component*** implementation. Implementers SHOULD use a concurrent approach to provide shorter overall response times, but MAY choose a consecutive approach to reduce implementation time.

### 3.3.1 Search Component Query (`cdrs:Query`)

A `cdrs:SearchRequest` SHOULD include the `cdrs:Query` element (in its entirety) from the original Brokered Search Component Search request. It may not apply to cases when the `cdrs:Query` may have to be modified prior to distributing the query to a particular source, e.g. if mediation needs to be applied.

### 3.3.2 Search Component Paging (`cdrs:resultsPerPage`)

In the process of distributing search requests to individual ***Search Component*** providers, an implementation SHOULD consider how many results it should request from any one ***Search Component*** implementation. To facilitate this process, it is RECOMMENDED that the implementation leverage the `cdrs:resultsPerPage` Search attributes to determine a target number of total results that it needs to receive from the collaborating ***Search Components*** to which it distributes the search request.

### 3.3.3 Search Component Invocation Status (`cdrb:sourceStatus`)

In order to record the status of those ***Search*** component implementations involved in the collaboration, the Brokered Search component MUST include one `cdrb:sourceStatus` element per identified source. The `cdrb:sourceStatus` should articulate the reported search invocation status, including the total number of results expected by that Search Component implementation or the fault that was thrown. This element may also be included in an interm broker response to give the consumer diagnostic information on the overall progress of a search request, for each source that was included in the request. The sourceStatus element may be extended with elements or attributes from another XML namespace to provide additional information if required by the Search Broker Component implementation.

Example source XML:

```
<{ResultSet}>
  …
  <cdrb:sourceStatus sourceId=”Intelink-Open”>
    <cdrb:status>complete</cdrb:status>
    <cdrb:resultsRetrieved>100</cdrb:resultsRetrieved>
    <cdrb:totalResults>222222</cdrb:totalResults>
  </cdrb:sourceStatus>
  <cdrb:sourceStatus sourceId=”AKO-DKO”>
    …
    <cdrb:status>waiting</cdrb:status>
    <cdrb:comment>Query sent successfully. Awaiting a response.</cdrb:comment>
  </cdrb:sourceStatus>
  <cdrb:sourceStatus sourceId=”Example3”>
   <cdrb:status>error</cdrb:status>
   <soap:Fault>
     <soap:faultcode>wsa:DestinationUnreachable</soap:faultcode>
     <soap:faultstring>
       No route can be determined to reach http://example.com/search/case1
     </soap:faultstring>
   </soap:Fault>
  </cdrb:sourceStatus>
  …
</{ResultSet}>
```

Description of significant elements:

**/<{ResultSet}>/cdrb:sourceStatus/@sourceId**

**Required.** This attribute provides an identifier to which the ***Search Component*** source can be identified..

**/<{ResultSet}>/cdrb:sourceStatus/cdrb:status**

**Required**. This element reports the current status of a single source. It may contain one of the following values:

- *excluded* – 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 doesn’t support query parameters in the request.
- *waiting* – The search request has been sent to the source, and the broker is waiting for a complete response from the source.
- *error* – The source returned an error response.
- *timeout* – The source failed to respond within the configured timeout period.
- *processing* – 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).
- *complete* – A response was successfully received and the result set from this source has been processed.

**/<{ResultSet}>/cdrb:sourceStatus/soap:Fault**

This OPTIONAL element describes any `soap:Fault` element that was thrown by the referenced ***Search Component***. If the given `wsa:Address` for the ***Search Component*** was unreachable, the `soap:Fault` should describe a `wsa:DestinationUnreachable` fault. If the ***Search Component*** is disabled, the `soap:Fault` should describe a `wsa:EndpointUnavailable` fault. If the ***Search Component*** throws a known ***Search Component*** fault, the `soap:Fault` should reflect that information.

**/<{ResultSet}>/cdrb:sourceStatus/cdrb:resultsRetrieved**

**Required**. This element reports the number of search results that were returned by the source.

**/<{ResultSet}>/cdrb:sourceStatus/cdrb:totalResults**

**Optional**. This element reports the number of total results matching the query, as reported by the source.

## 3.4 Federation Results Processing Activity

As the ***Brokered Search Component*** does not produce any search results itself, but rather acts as a broker of search results from one or more ***Search Component*** implementations, special care must be taken when crafting an aggregated result set.

Regardless of the distribution method, concurrent or consecutive, results SHOULD be aggregated based on order of response time. Under this aggregation ordering scheme, results that are received first MUST be returned first in the response feed. A result is considered received when the ***Search Component*** has returned a complete response and the ***Broker Search*** implementation has processed the response per input to the Results Presentation and Results Paging activities defined for the Search Component.

### 3.4.1 Source Identification (`cdrb:sourceId`)

Each result returned in the search results MUST include the cdrb:sourceId attribute which indicates the identified source(s) that returned it.

Source Identification example:

```
…
<{ResultSet}>
  …
  <{Result} sourceId =”Intelink-Open“>
    <title>This is an Example Page</title>
    <link href="http://example.com/foo/index.html" type="alternate"/>
    <id>http://example.com/foo/index.html</id>
    <date-created>2010-05-05</date-created>
    <summary>… As the US Army transitions to a force for the 21st Century, so does the
      Army&amp;#39;s only independent operational test organization - the US Army Operational …
    </summary>
  </{Result}>
…
```

Description of significant elements:

**/<{Result}>/@cdrb:sourceId**

This OPTIONAL attribute references the individual **Search Component** source specified in the {ResultSet}.

### 3.4.2 Rank/Relevance

A ***Brokered Search*** implementation MAY provide relevance scores for individual search results with respect to the particular search with which it is identified.

### 3.4.3 Paging of Search Results (`cdrs:resultsPerPage`)

The ***Brokered Search Component*** paging is that of the **Search Component** and will follow those principles and behaviors outlined in the ***Search Component*** specification. Search result pages may be traversed using the information from the original ***Brokered Search*** request combined with the endpoint information provided by the wsa:EndpointReference describing the ***Brokered Search Component*** from which the current result set was generated. The ***Brokered Search Component*** wsa:EndpointReference allows a service consumer to issue a search request for the next "page" of data. However, to avoid the repeated execution of potentially costly queries, an indexing or caching mechanism SHOULD be implemented by service providers.

### 3.4.4 Start Index Out of Range (`cdrs:startIndex`)

If a requested `cdrs:startIndex` is out-of-range, then the implementation should return an empty {ResultSet}.

### 3.4.5 Cached Query Results (`cdrs:queryId`)

When the `cdrs:queryId` is present in the search request and is known to the implementation, then the implementation SHOULD return elements from the cache mechanism it has associated with the `cdrs:queryId`.

In the case where the `cdrs:queryId` is present in the search request, but is unknown to the implementation or is not present, then the implementation MUST throw fault as specified in the CDR Search Specification.

### 3.4.6 Streaming Results to Consumer

Although aggregating all the results tends to be simpler to implement and has the benefit of providing better sorting capability, the ***Brokered Search*** component can also OPTIONALLY return results to the service consumer as soon as a search request is completed which would provide better scalability across disparate environments.
A search request is completed when any one of the following conditions are satisfied:

- The requested results per page (`cdrs:resultsPerPage`)
- The requested timeout has expired (`cdrs:timeout`)
- All identified Search Components have been invoked

The `cdrs:SearchRequest` MUST include the following from the Search request:

- `cdrs:Query` element (in its entirety)
- all Search Properties (`cdrs:startIndex`, `cdrs:resultsPerPage`, etc)

Incremental ***Brokered Search*** results MAY be traversed using the next page information (in the {ResultSet}) created from original ***Brokered Search Component*** request using the `cdrs:queryId` and `cdrs:startIndex`.

### 3.5 Additional References

XML Schemas (XSD) are provided as supplements to this service specification to provide the necessary data structures for the ***Brokered Search Component***.

# 4 Discovery and Publishing

### 4.1 SOAP Interface

The SOAP interface is defined through a WSDL document and MUST be published through the workflow described by the Service Discovery Reference Architecture and Specifications. The SOAP interface MAY be discovered through any interface defined in Service Discovery.

### 4.2 Policy

This specification defines the technical requirements and guidelines for implementing a ***Brokered Search Component***. Policy for ***Brokered Search Component*** implementations is described in auxiliary documents. See the Reference Documents section for a listing of relevant policy documents. Implementers MUST follow the guidance in those policy documents, as appropriate for their organization. For instance, DoD consumers should follow DoD policy guidance when sharing data across organizational boundaries, whereas IC consumers would follow IC policy, if differences between the two exist.

### 4.3 Query Types

A ***Brokered Search Component*** implementation does not use the service consumer's query. It may, however, introspect this query when identifying the participating ***Search Components***. *A* ***Brokered Search Component*** may determine which Query Types it can or cannot accept and return the appropriate fault.

### 4.4 Result Sets

The CDR Specification set includes at least one Result Set definition that IC/DoD organizations can leverage in their ***Brokered Search Component*** implementations. If the **Brokered Search** implementation cannot consolidate the result set of a participating ***Search Component***, a *Federated Results Processing Fault* should be returned in the Broker status. Consult the policy documents to determine requirements or recommendations concerning the use of particular Result Sets.

### 4.5 Delivery

The results of the Brokered Search Component can be rerouted by use of WS-Addressing. WS-Addressing allows SOAP-based services to route search results to a specified interface, where that specified interface must accept the output format of the Brokered Search Component. For more information on using WS-Addressing to re-route search results, please refer to the Web Services Addressing 1.0 Core Specification [I2].

## *4.6 Security Considerations*

The “Joint IC/DoD Security Reference Architecture” [S] and its associated specifications define the specific security components and interactions needed to perform authorization and authentication. ***Brokered Search*** implementations MUST follow the guidance in those documents.

# 5 Reference Documents

The documents in this section provide the foundation for, define extensions to, or include implementation guidance for the ***Brokered Search Component***. They define additional specifications, including those provided as part of the greater CDR specification set, and guidance documents that communicate current policy or implementation details. Each document is assigned a reference identifier, which is cited when the document is referenced within this Search Component Specification.

In some cases, documents have been referenced with a version and date of “Future” in order to track the iterative development of some of these extensions.

## *5.1 Specifications*

### 5.1.1 Content Discovery and Retrieval Specifications

The following documents provide a foundation and guidance for the development of this Brokered Search Specification document. Brokered Search Component implementers should have a thorough understanding of the concepts and guidance in these documents. This Brokered Search Specification represents a realization of the Brokered Search Component defined therein.

| Ref. | Title | Version | Date |
|---|---|---|---|
| SF | IC/DoD Content Discovery and Retrieval Specification Framework | DRAFT 0.6.2 | 29 Jan 2010 |
| RA | IC/DoD Content Discovery and Retrieval Reference Architecture | DRAFT 0.4 | 16 Dec 2009 |
| S | IC/DoD Search Specification for SOAP Implementations | DRAFT Milestone 1 | 09 March 2010 |

#### 5.1.1.1 Result Set Specifications

This Brokered Search Component Specification response can contain any Result Set (or other object). See Community Data Specifications for guidance on how to combine data standards (e.g., DDMS) into Result Sets.

The following documents define the expected format and content of a particular type of collection returned from a CDR Search function --beyond that specified by the underlying Result Set type itself, if the Result Set is based on an industry standard:

| Ref. | Title | Version | Date |
|---|---|---|---|
| RS | IC/DoD Content Discovery and Retrieval Atom 1.0 Result Set Specification | DRAFT 1.0-Milestone 1 | 09 Mar 2010 |

## 5.1.2 Other Specifications

### 5.1.2.1 Security Specifications

| Ref. | Title | Version | Date |
|---|---|---|---|
| SRA | Joint IC/DoD Security Reference Architecture | 1.0 | 25 Jul 2008 |

### 5.1.2.2 Service Discovery Specifications

| Ref. | Title | Version | Date |
|---|---|---|---|
| SDRA | Joint IC/DoD Service Discovery Architecture | DRAFT 1.2 | 28 Sep 2007 |

### 5.1.2.3 Community Data Specifications

| Ref. | Title | Version | Date |
|---|---|---|---|
| C1 | DDMS Data Query Type and Result Type Guidance | 1.0-Milestone 1 | 09 Mar 2010 |
| C2 | IRM.XML Data Query Type and Result Type Guidance | Future | Future |
| C3 | UCore Data Query Type and Result Type Guidance | Future | Future |

### 5.1.2.4 Industry Specifications

| Ref. | Title | Version | Date |
|---|---|---|---|
| I1 | The Atom Syndication Format | 1.0 | Dec 2005 |
| I2 | Web Services Addressing 1.0 - Core | W3C Recommendation 9 May 2006 | 09 May 2006 |

# *5.2 Policy and Guidance*

## 5.2.1 Content Discovery and Retrieval Policy and Guidance

Since this specification inherits the interface and all behaviors of the CDR Search Component. The following documents provide additional requirements and expectations set by policy:

| Ref. | Title | Version | Date |
|---|---|---|---|
| P1 | IC/DoD Content Discovery and Retrieval Search Component Policy for SOAP Implementations | DRAFT 1.0-Milestone 1 | 09 Mar 2010 |