

1

2



3

4

5

6

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

9

***IC/DoD Content Discovery & Retrieval
Brokered Search Specification for SOAP
Implementations***

10

11

12

13

14

15

16

V1.0-20101026

17

18

19

26 October 2010

20

21

22

REVISION/HISTORY

23

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

24

25

26

27

TABLE OF CONTENTS

28			
29			
30	1	Introduction.....	5
31	1.1	Component Overview	5
32	1.1.1	Relationship to Other CDR Architecture Elements	5
33	1.2	Notational Convention	7
34	1.3	Conformance.....	7
35	1.4	Namespaces.....	7
36	2	Brokered Search Interface.....	8
37	2.1	Brokered Search Function.....	8
38	2.2	Input	8
39	2.3	Output	10
40	2.4	Fault Conditions.....	11
41	3	Brokered Search Behavior.....	12
42	3.1	Brokered Search Coordination.....	12
43	3.2	Source Identification Activity.....	12
44	3.2.1	Sources Identified from a static, internal list (<code>cdrb:routeTo</code>).....	12
45	3.2.2	Sources Identified by Consumer (<code>cdrb:routeTo</code> ,	
46		<code>wsa:EndpointReference</code>).....	13
47	3.2.3	Sources Identified by Consumer specified Criteria/Query Introspection (<i>Non-</i>	
48		<i>Normative</i>)	14
49	3.2.4	Including the Identified Source in the {ResultSet}	
50		(<code>cdrb:sourceStatus</code>).....	14
51	3.3	Search Component Invocation Activity.....	15
52	3.3.1	Search Component Query (<code>cdrs:Query</code>)	15
53	3.3.2	Search Component Paging (<code>cdrs:resultsPerPage</code>)	15
54	3.3.3	Search Component Invocation Status (<code>cdrb:sourceStatus</code>).....	15
55	3.4	Federation Results Processing Activity	17
56	3.4.1	Source Identification (<code>cdrb:sourceId</code>).....	17
57	3.4.2	Rank/Relevance	17
58	3.4.3	Paging of Search Results (<code>cdrs:resultsPerPage</code>).....	18
59	3.4.4	Start Index Out of Range (<code>cdrs:startIndex</code>).....	18
60	3.4.5	Cached Query Results (<code>cdrs:queryId</code>)	18
61	3.4.6	Streaming Results to Consumer.....	18
62	3.5	Additional References.....	18
63	4	Discovery and Publishing.....	19
64	4.1	SOAP Interface	19
65	4.2	Policy	19
66	4.3	Query Types.....	19
67	4.4	Result Sets.....	19
68	4.5	Delivery.....	19
69	4.6	Security Considerations	19
70	5	Reference Documents	20
71	5.1	Specifications.....	20
72	5.1.1	Content Discovery and Retrieval Specifications	20

73 5.1.2 Other Specifications 21
74 5.2 Policy and Guidance 21
75 5.2.1 Content Discovery and Retrieval Policy and Guidance..... 21

76
77

LIST OF TABLES

78
79 Figure 1 – CDR Architecture Model 6

80
81

LIST OF TABLES

82 Table 1 – Referenced XML Namespaces 7
83 Table 2 – Specification Framework Input Variables 8
84 Table 3 – Specification Framework Output Variables 10

85

86 1 Introduction

87 1.1 Component Overview

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

93
94 This specification defines requirements and provides guidelines for the realization of the
95 CDR Brokered Search Component in a web service using the SOAP messaging protocol,
96 hereafter termed a *Brokered Search Component* in this document. Providing enough
97 information for *Broker Search Component* providers and implementers to create CDR-
98 compliant *Brokered Search Components*, the *specification* describes a *Brokered Search*
99 *Component's* behavior, interface, and other aspects in detail.

100

101 A *Brokered Search Component* uses the basic functionality described by the Search
102 Component for a single search. Additional inputs and outputs are defined as needed to
103 support the four activities that underpin Brokered Search capabilities: brokered search
104 coordination, source identification, search component invocation, and federation results
105 processing. As discussed in CDR Specification Framework, a *Search* component's
106 results are resource metadata rather than actual content resources. In the context of
107 Search, resource metadata generally refers to a *subset* of a resource's available metadata,
108 not the entire underlying record¹. Some of the information contained within each Search
109 result may provide the information necessary for a consumer to retrieve or otherwise use
110 a resource.

111

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

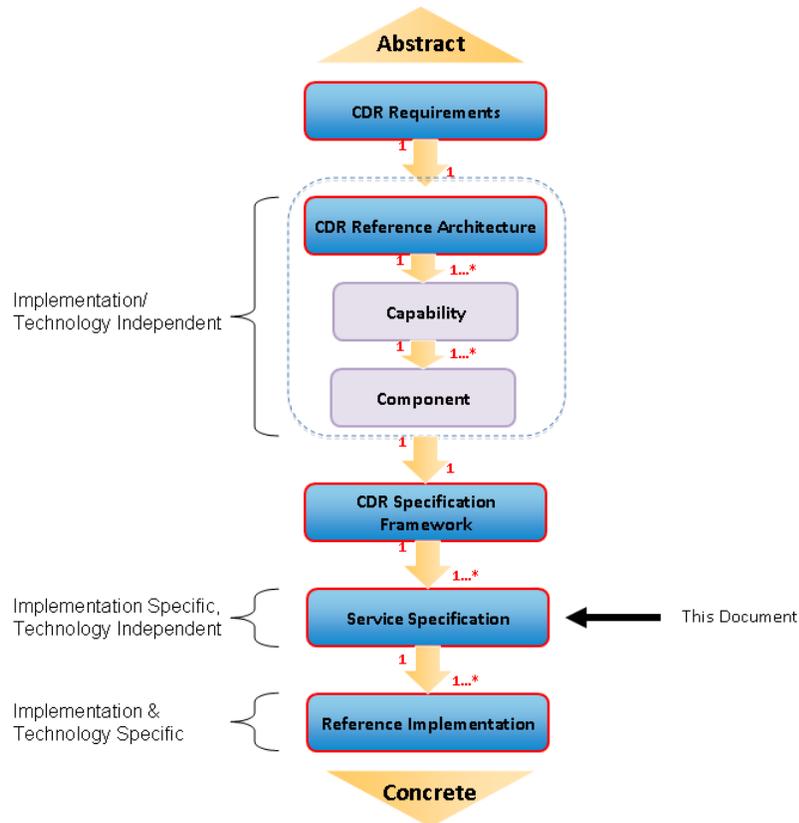
119 1.1.1 Relationship to Other CDR Architecture Elements²

120 The CDR Architecture prescribes an abstract-to-concrete model for the development of
121 architecture elements and guidance for content discovery and retrieval. Each layer or tier
122 of the model is intended to provide key aspects of the overall guidance to achieve the
123 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.

124 graphic, discussed in detail within the CDR Reference Architecture [RA], illustrates this
 125 model.
 126



127

128

Figure 1 – CDR Architecture Model

129

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

137

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

140

- 141 • **Service Behavior** maps the Brokered Search interaction patterns defined in the
 Specification Framework to concrete SOAP constructs.
- 142 • **Service Interface** defines the base SOAP constructs to express inputs, outputs,
 143 and faults.
- 144 • **Implementation** provides additional implementation guidance beyond service
 145 behavior and interface guidance.
- 146 • **Reference Documentation** provides references to other CDR and community
 147 artifacts (i.e., Security Reference Architecture).

148 1.2 Notational Convention

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

153
154 When describing concrete XML schemas and example XML documents, this
155 specification uses XPath as the notational convention. Each member of an XML schema
156 is described using an XPath notation (e.g., /x:RootElement/x:ChildElement/@Attribute).
157 The use of {any} indicates the presence of an element wildcard (<xs:any/>). The use of
158 @{any} indicates the presence of an attribute wildcard (<xs:anyAttribute/>).

159
160 Examples in this text are distinguished by a black border and items that should be
161 focused on are highlighted in yellow. These are meant to be illustrative and not the only
162 way that the described syntax can be used. Examples may include several lines derived
163 from the IC/DoD Search Specification for SOAP Implementations [S] sub-specifications,
164 e.g. cdrs:Expression, and included solely to complete the example. These lines are
165 presented in green.

166 1.3 Conformance

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

172 1.4 Namespaces

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

177

178

179

180 **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)

181

182 2 Brokered Search Interface

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

185 2.1 Brokered Search Function

Function	Input	Output	Fault
Search	<i>cdrs:SearchRequest</i>	{ResultSet} ³	Defined within CDR Framework

186

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

193 2.2 Input

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

199

200 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

201

202 The following example illustrates a search request message to a **Brokered Search**
203 **Component**, using a notional keyword query (for more information on Query Types,
204 please reference the IC/DoD Search Specification for SOAP Implementations [S]):
205

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

```

206 <soap:Envelope>
207   <soap:Header>
208     ...
209     <wsa:Action>
210       urn:cdr:1.0:soap:action:search
211     </wsa:Action>
212     ...
213   </soap:Header>
214   <soap:Body>
215     <cdrs:SearchRequest
216       startIndex="1" resultsPerPage="50" timeout="20000"
217       cdrb:routeTo="exampleSource1, exampleSource2">
218       <wsa:EndpointReference cdrb:sourceId="exampleSource1">...</wsa:EndpointReference>
219       <wsa:EndpointReference cdrb:sourceId="exampleSource2">...</wsa:EndpointReference>
220       <cdrs:Query queryTypeURI="cdr:1.0:soap:query:keyword">
221         <cdrs:Expression>
222           UNMANNED AERIAL VEHICLE
223         </cdrs:Expression>
224       </Query>
225     </cdrs:SearchRequest>
226   </soap:Body>
227 </soap:Envelope>

```

228
229 Description of significant elements:

230 */cdrs:SearchRequest*

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

233 */cdrs:SearchRequest/@startIndex*⁴

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

237 */cdrs:SearchRequest/@resultsPerPage*⁴

238 This OPTIONAL element describes the desired number of search results per page.
239 Its value, if provided, MUST be greater than or equal to 1. The default value is
240 10.

241 */cdrs:SearchRequest/@timeout*⁴

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

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

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

253 */cdrs:SearchRequest/wsa:EndpointReference*

254 This OPTIONAL element provides the endpoint of the *Search Component* source
255 to which the Query should be distributed. A separate `EndpointReference` is
256

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

257 necessary for each source identified in the routeTo attribute that is not implicitly
258 understood by the Brokered Search Component.

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

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

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

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

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

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

269 **2.3 Output**

270 The output is REQUIRED to be compliant to the requirements imposed by the CDR
271 Search SOAP Specification. In addition, Table 3 describes further output constraints on
272 the *Brokered Search Component's* Search function:
273

274 **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

275

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

```

278 <soap:Envelope>
279   <soap:Header>
280   ...
281   <wsa:Action>
282     urn:cdr:1.0:soap:action:response
283   </wsa:Action>
284   ...
285   </soap:Header>
286   <soap:Body>
287     <{ResultSet}>
288       <wsa:EndpointReference sourceId="exampleSource1">...</wsa:EndpointReference>
289       <cdrb:sourceStatus sourceId="exampleSource1">
290         <cdrb:status>...</cdrb:status>
291         <cdrb:resultsRetrieved>...</cdrb:resultsRetrieved>
292         <cdrb:totalResults>...</cdrb:totalResults>
293       </cdrb:sourceStatus>
294       <wsa:EndpointReference sourceId="exampleSource2">...</wsa:EndpointReference>
295       <cdrb:sourceStatus sourceId="exampleSource2">
296         <cdrb:status>...</cdrb:status>
297         <cdrb:resultsRetrieved>...</cdrb:resultsRetrieved>
298         <cdrb:totalResults>...</cdrb:totalResults>
299       </cdrb:sourceStatus>
300       <{Result} sourceId="exampleSource1">...</{result}>
301       <{Result} sourceId="exampleSource1">...</{result}>
302       <{Result} sourceId="exampleSource2">...</{result}>
303     </{ResultSet}>
304   </soap:Body>
305 </soap:Envelope>

```

307 Description of significant elements:

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

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

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

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

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

315 This OPTIONAL attribute associates the wsa:EndpointReference element with the
316 source identifier. The sourceId should correspond to the sourceId specified in the
317 routeTo attribute of the

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

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

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

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

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

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

327 **2.4 Fault Conditions**

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

- 331 • **Source Identification Fault** – A fault used if the Brokered Search implementation
332 cannot identify Search Components to invoke
333
- 334 • **Search Component Invocation Fault** – A fault used if the Brokered Search
335 implementation cannot invoke the identified Search Components. This fault **MUST**
336 clearly identify which Search Component invocation caused the problem.
337
- 338 • **Federated Results Processing Fault** – A fault used when the Brokered Search
339 implementation cannot process the results set of an individual Search Component
340 Invocation. This may indicate an error in the returned results set or an inconsistency
341 in interpreting the results set specified format.
342

343 **3 Brokered Search Behavior**

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

347 **3.1 Brokered Search Coordination**

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

356 **3.2 Source Identification Activity**

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

361 **3.2.1 Sources Identified from a static, internal list (`cdrb:routeTo`)**

362 A Brokered Search Component implementation can identify a specific set of Search
363 Components implementations it is prepared to invoke, and the service consumer may use
364 the `cdrb:routeTo` attribute to limit the query to a subset of the identified Search
365 Components. The `cdrb:routeTo` attribute contains a comma-separated list of source
366 identifiers, to which the search query should be routed. The same identifier is also
367 referenced by the `sourceId` attribute in response elements described in sections 3.2.4 and
368 3.4.1.
369

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

373

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

377

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

380

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

383

```
384 <cdrs:SearchRequest cdrb:routeTo="exampleSource1, exampleSource2">
385 ...
386 </cdrs:SearchRequest>
```

387 Description of significant elements:

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

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

392 3.2.2 Sources Identified by Consumer (`cdrb:routeTo`, 393 `wsa:EndpointReference`)

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

398

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

401

```
402 <cdrs:SearchRequest routeTo="exampleSource1, exampleSource2">
403   <wsa:EndpointReference sourceId="exampleSource1">
404     <wsa:Address>http://example.com/search/example1</wsa:Address>
405     <wsa:Metadata>
406       <wsaw:InterfaceName>cdrs:Search_PortType</wsaw:InterfaceName>
407       <wsaw:ServiceName>example:SearchService1</wsaw:ServiceName>
408     </wsa:Metadata>
409   </wsa:EndpointReference>
410   <wsa:EndpointReference sourceId="exampleSource2">
411     <wsa:Address>http://example.com/search/example2</wsa:Address>
412     <wsa:Metadata>
413       <wsaw:InterfaceName>cdrs:Search_PortType</wsaw:InterfaceName>
414       <wsaw:ServiceName>example:SearchService2</wsaw:ServiceName>
415     </wsa:Metadata>
416   </wsa:EndpointReference>
417   ...
418   ...
419 </cdrs:SearchRequest>
```

420 Description of significant elements:

- 421 **/cdrs:SearchRequest/@cdrb:routeTo**
 422 This OPTIONAL attribute provides a comma-separated list of the source
 423 identifiers that identify the *Search Component* sources for which the Query
 424 should be distributed
- 425 **/cdrs:SearchRequest/wsa:EndpointReference**
 426 This OPTIONAL element provides the endpoint of the *Search Component* source
 427 in which the Query should be distributed
- 428 **/cdrs:SearchRequest/wsa:EndpointReference/@sourceId**
 429 This OPTIONAL attribute associates the source identifier with the
 430 wsa:EndpointReference

431 3.2.3 Sources Identified by Consumer specified Criteria/Query 432 Introspection (*Non-Normative*)

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

437
 438 To help facilitate the use of query introspection, a consumer may append additional
 439 description information to the query element. If this description information is included
 440 in the `cdrs:Query`, a *Brokered Search* can be used in conjunction with the Service
 441 Discovery Capability to determine the available *Search* implementations that are
 442 categorized under the requested description value. The *Brokered Search* implementation
 443 MUST then distribute requests to the selected *Search Components*.

444
 445 This introspection capability is not explicitly supported in the present specification.

446 3.2.4 Including the Identified Source in the {ResultSet} 447 (`cdrb:sourceStatus`)

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

452
 453 Example source XML:

```
454 <{ResultSet}>
455   ...
456   <cdrb:sourceStatus sourceId="Intelink-Open">
457     ...
458     <cdrb:status>complete</cdrb:status>
459     <cdrb:resultsRetrieved>100</cdrb:resultsRetrieved>
460     <cdrb:totalResults>222222</cdrb:totalResults>
461   </cdrb:source>
462   <cdrb:sourceStatus sourceId="AKO-DKO">
463     ...
464     <cdrb:status>waiting</cdrb:status>
465   </cdrb:source>
466   ...
467 </{ResultSet}>
```

468

469 **3.3 Search Component Invocation Activity**

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

476
477 The search MAY be propagated concurrently or consecutively to each identified *Search*
478 *Component* implementation. Implementers SHOULD use a concurrent approach to
479 provide shorter overall response times, but MAY choose a consecutive approach to
480 reduce implementation time.

481 **3.3.1 Search Component Query (`cdrs:Query`)**

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

486 **3.3.2 Search Component Paging (`cdrs:resultsPerPage`)**

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

493 **3.3.3 Search Component Invocation Status (`cdrb:sourceStatus`)**

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

504
505 Example source XML:

```

506 <{ResultSet}>
507 ...
508 <cdrb:sourceStatus sourceId="Intelink-Open">
509   <cdrb:status>complete</cdrb:status>
510   <cdrb:resultsRetrieved>100</cdrb:resultsRetrieved>
511   <cdrb:totalResults>222222</cdrb:totalResults>
512 </cdrb:sourceStatus>
513 <cdrb:sourceStatus sourceId="AKO-DKO">
514   ...
515   <cdrb:status>waiting</cdrb:status>
516   <cdrb:comment>Query sent successfully. Awaiting a response.</cdrb:comment>
517 </cdrb:sourceStatus>
518 <cdrb:sourceStatus sourceId="Example3">
519   <cdrb:status>error</cdrb:status>
520   <soap:Fault>
521     <soap:faultcode>wsa:DestinationUnreachable</soap:faultcode>
522     <soap:faultstring>
523       No route can be determined to reach http://example.com/search/case1
524     </soap:faultstring>
525   </soap:Fault>
526 </cdrb:sourceStatus>
527 ...
528 </{ResultSet}>

```

529 Description of significant elements:

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

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

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

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

- 536 • *excluded* – The source was excluded by the broker. There may be a number of reasons for
537 excluding a source, for example, if a maximum number of sources is exceeded, or if the
538 source doesn't support query parameters in the request.
- 539 • *waiting* – The search request has been sent to the source, and the broker is waiting for a
540 complete response from the source.
- 541 • *error* – The source returned an error response.
- 542 • *timeout* – The source failed to respond within the configured timeout period.
- 543 • *processing* – The broker received a complete response from the source, but is processing
544 the result set (e.g., converting format, merging with other results, re-ranking).
- 545 • *complete* – A response was successfully received and the result set from this source has
546 been processed.

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

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

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

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

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

559 **Optional.** This element reports the number of total results matching the query, as
560 reported by the source.
561

562 **3.4 Federation Results Processing Activity**

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

566
567 Regardless of the distribution method, concurrent or consecutive, results **SHOULD** be
568 aggregated based on order of response time. Under this aggregation ordering scheme,
569 results that are received first **MUST** be returned first in the response feed. A result is
570 considered received when the *Search Component* has returned a complete response and
571 the *Broker Search* implementation has processed the response per input to the Results
572 Presentation and Results Paging activities defined for the Search Component.
573

574 **3.4.1 Source Identification (cdrb:sourceId)**

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

577

578 Source Identification example:

```
579   ...
580   </{ResultSet}>
581   ...
582   </{Result} sourceId = "Intelink-Open">
583    <title>This is an Example Page</title>
584    <link href="http://example.com/foo/index.html" type="alternate"/>
585    <id>http://example.com/foo/index.html</id>
586    <date-created>2010-05-05</date-created>
587    <summary>... As the US Army transitions to a force for the 21st Century, so does the
588      Army&#39;s only independent operational test organization - the US Army Operational ...
589    </summary>
590   </{Result}>
591   ...
```

592 Description of significant elements:

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

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

596 **3.4.2 Rank/Relevance**

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

599 **3.4.3 Paging of Search Results (`cdrs:resultsPerPage`)**

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

609 **3.4.4 Start Index Out of Range (`cdrs:startIndex`)**

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

612 **3.4.5 Cached Query Results (`cdrs:queryId`)**

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

616
617 In the case where the `cdrs:queryId` is present in the search request, but is unknown to
618 the implementation or is not present, then the implementation MUST throw fault as
619 specified in the CDR Search Specification.

620 **3.4.6 Streaming Results to Consumer**

621 Although aggregating all the results tends to be simpler to implement and has the benefit
622 of providing better sorting capability, the *Brokered Search* component can also
623 OPTIONALLY return results to the service consumer as soon as a search request is
624 completed which would provide better scalability across disparate environments.

625 A search request is completed when any one of the following conditions are satisfied:

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

629

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

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

633

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

637 **3.5 Additional References**

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

640 4 Discovery and Publishing

641 4.1 SOAP Interface

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

646 4.2 Policy

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

654 4.3 Query Types

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

659 4.4 Result Sets

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

666 4.5 Delivery

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

672 4.6 Security Considerations

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

677 5 Reference Documents

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

684

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

687 5.1 Specifications

688 5.1.1 Content Discovery and Retrieval Specifications

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

694

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

695

696 5.1.1.1 Result Set Specifications

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

700

701 The following documents define the expected format and content of a particular type of
702 collection returned from a CDR Search function --beyond that specified by the
703 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

704 **5.1.2 Other Specifications**705 **5.1.2.1 Security Specifications**

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

706 **5.1.2.2 Service Discovery Specifications**

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

707 **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

708

709 **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

710 **5.2 Policy and Guidance**711 **5.2.1 Content Discovery and Retrieval Policy and Guidance**

712 Since this specification inherits the interface and all behaviors of the CDR Search
713 Component. The following documents provide additional requirements and expectations
714 set by policy:
715

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

716

717

