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

IC/DoD REST Interface Encoding Specification for CDR Deliver v1.0

12 May 2011

REVISION/HISTORY

Doc Revision	Revised By	Revision Date	Revisions
V0.1	Foster / Coston	10 September 2010	
V0.2	Foster/Coston	17 September 2010	Applied input from 1st draft REST specification comment matrix
V0.3	Foster/Coston	28 September 2010	Revision based on follow up team comments
V0.4	Foster/Coston	8 October 2010	Changed to reflect the 5 October CDR-IPT Design Meeting decision regarding Deliver Service Behavior
V0.5	Foster/Coston	19 October 2010	Changes made to reflect the 14 October 2010 review session
V1.0	Foster/Coston	20 October 2010	Changes made to reflect the 19 October 2010 review session
V1.0	Foster/Coston	25 October 2010	Technical Editing changes
V1.0	Rothrock	29 October 2010	Tech Edit Recommendations
V1.20101223	Coston/Wigglesworth	12 January 2011	Reconciliation with SOAP Specification
V1.20110318	CDR/IPT	18 March 2011	Changed to reflect community comments
V1.20110429	CDR/IPT	29 April 2011	Changed to reflect community comments

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

1.1 Service Overview

This document defines requirements and provides guidelines for the realization of the Content Discovery and Retrieval (CDR) Deliver Component as a web service using the Representational State Transfer REST style binding, hereafter termed a ***Deliver Service*** in this document. It describes a ***Deliver Service's*** interface and other aspects in detail, providing enough information for ***Deliver Service*** providers and implementers to create CDR-compliant Deliver ***Services***.

The ***Deliver Service***, as defined by the Intelligence Community/Department of Defense (IC/DoD) CDR Specification Framework, serves as a "push" mechanism to send information resources. The ***Deliver Service*** relies on mechanisms that are already well established in the service oriented architecture design and development community.

The ***Deliver Service,*** as defined, supports the delivery of a specified resource payload directly to a consumer specified location. In its simplest form, ***Deliver*** will take a consumer-supplied payload and send it to another consumer as specified by the delivery destination and properties. The ***Deliver Service*** may include additional (interim) processing, including but not limited to compression, encryption, or conversion. The specification of interim processing is beyond the scope of this document.

The delivery destination can be:

- A specified location (e.g., ftp folder, shared drive)
- A receiving component implementation
- Another component or service endpoint within the architecture

The implementation method is left to the implementers of the ***Deliver Service.*** The Deliver Specification focuses on a single delivery target, but it does not preclude an implementation having multiple delivery targets. The consumer requesting the delivery may want to obtain the status of the Deliver Function, especially in scenarios where the delivery content is not returned directly to the requestor; in the initial version of this Deliver Specification, we demonstrate status as an output returned to the requestor. However, future versions of this specification may provide other methods for obtaining status.

1.2 Relationship to Other CDR Architecture Elements

The CDR Architecture prescribes an abstract-to-concrete model for the development of architecture elements and guidance for CDR. 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 in Figure 1, discussed in detail within the CDR Reference Architecture (RA), illustrates this model.

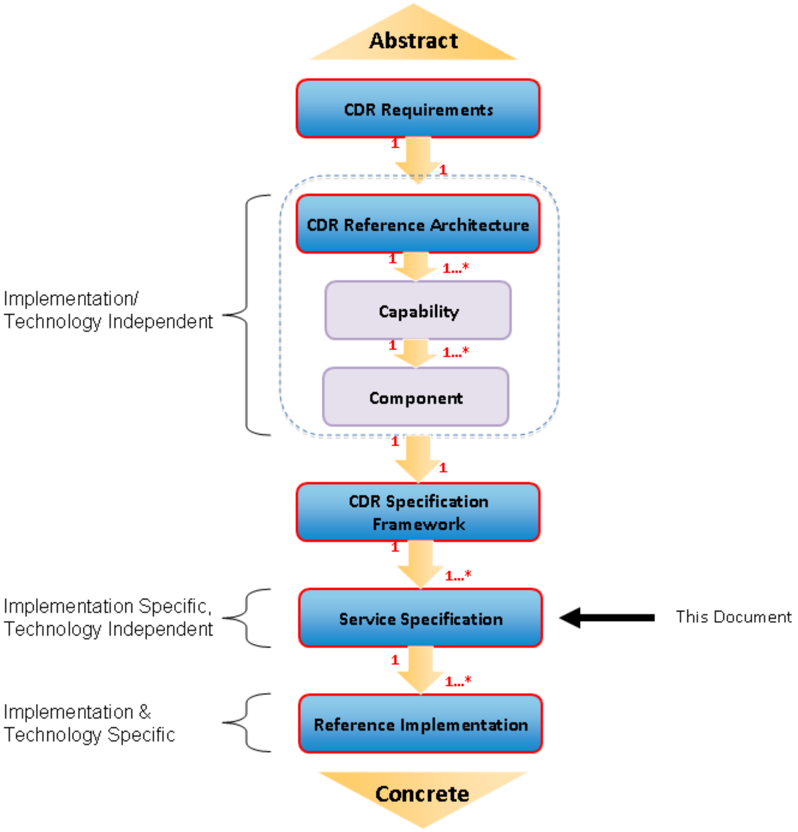

Figure 1 - CDR Architectural Model

As illustrated in Figure 1, the Specification Framework derives from the RA and describes behavior in terms of the capabilities, components, and usage patterns defined in the RA. The Specification Framework expands on the details of information flows and the information conveyed in those flows to provide a consistent basis for multiple Service Specifications to provide consistent interfaces, both in terms of the structure and of the semantics of the exchanged information. Service Specifications, such as this one, provide implementation-specific guidance. More specifically, this Deliver Specification defines the specific guidance for implementing the Deliver ***Service***.

1.3 Scope

As shown in the shaded area in Figure 2, below, this specification is limited to the description of the interaction between the Initiating Consumer and the ***Deliver Service***. The association between these two components is depicted in the diagram as a ***Deliver Service*** invocation with a set of parameters that includes (payload, properties, and destination). Components, Interactions and associations that lie outside the shaded areas clarify the overall design and provide a context for the use of deliver. Interactions/associations that are outside the shaded rectangle are used in this document to clarify the interaction between the Initiating Consumer and the ***Deliver Service***.

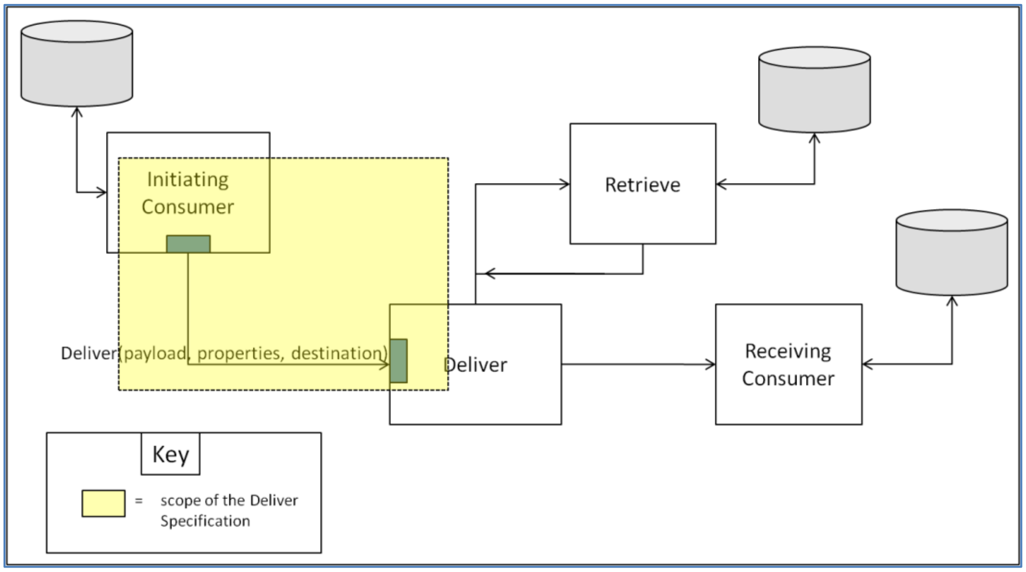
Initiating Consumer
Retrieve
Deliver(payload, properties, destination)
Deliver
Receiving Consumer
Key
= scope of the Deliver Specification

Figure 2 - Scope of the Deliver Service Specification for REST Implementations

This specification covers the following aspects of a REST-based ***Deliver Service***:

- **Service Interface** defines the base REST constructs to expressing inputs, outputs, and faults.
- **Implementation** provides additional implementation guidance beyond the behavior and interface guidance.
- **Reference Documentation** provides references to other CDR and community artifacts (e.g.., Service Security RA).

1.4 Notational Convention

The key words "MUST," "MUST NOT," "REQUIRED," "SHALL," "SHALL NOT," "SHOULD," "SHOULD NOT," "RECOMMENDED," "MAY," and "OPTIONAL" in this document 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.

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

1.5 Conformance

This specification defines an interface to a ***Deliver Service*** to which an implementation MUST conform. For an implementation to conform to the requirements contained in this specification, it MUST adhere to all mandatory aspects of the specification.

2 Deliver Service Behavior

This section uses basic message exchange patterns to clarify the behavior of the Deliver Service Component in the context of the CDR architecture components.

2.1 Component Interactions

The ***Deliver Service*** supports two fundamental message exchange patterns. The first pattern, shown in Figure 3, reflects a case where the Initiating Consumer supplies the information content via the payload parameter to be delivered to the Receiving Consumer, as specified via the destination parameter. The properties included as part of the ***Deliver Service*** request may be used to provide delivery specific information, including but not limited to interim processing, routing, and security. The second exchange pattern, shown in Figure 4, reflects a case where the Initiating Consumer requests that the content associated with the resource payload be *retrieved* so that it can be used in the subsequent deliver activity. In this case, the information necessary for the ***Deliver Service*** to utilize the CDR Retrieve Service MUST be specified via the Deliver Properties.

In the event that a particular implementation of the ***Deliver Service*** makes use of "default" values for message retrieval and/or delivery, the service implementer is responsible for publishing this information using an agreed upon mechanism. Cases where information is not supplied as part of the Deliver Service Request and service defaults are not available will result in a fault condition.

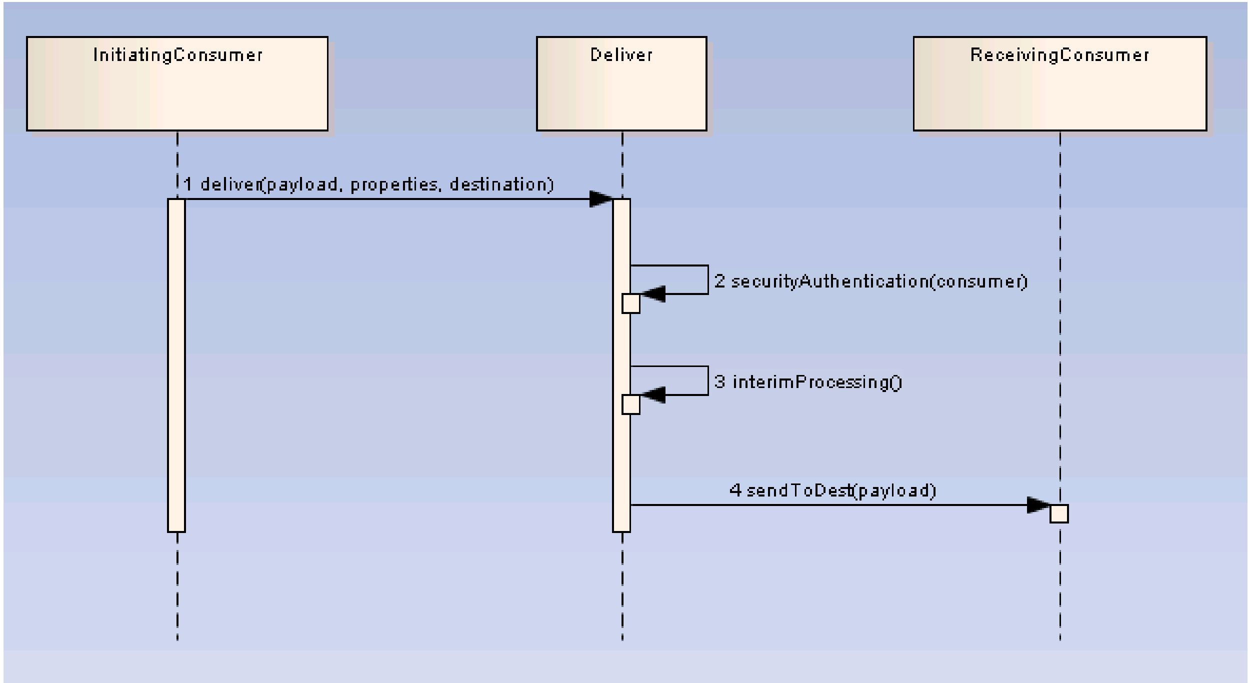

Figure 3 - Deliver: Payload Provided by the Initiator

Step 1 – Initiating Consumer sends a Deliver Service Request to the ***Deliver Service***.

Step 2 – The ***Deliver Service*** leverages a set of security components to verify that the Initiating Consumer is authenticated and authorized to send a specific information resource (payload) to the Receiving Consumer; and the Receiving Consumer is authenticated and authorized to receive the specified information payload. The Joint IC/DoD Security Reference Architecture defines the specific security components and interactions needed to perform this verification.

Step 3 – The optional interim processing may represent internal capabilities of the Deliver implementation or may be external capabilities for with the Deliver implementation acts as a consumer. The ***Deliver Service*** implementation is NOT required to include any interim processing (e.g., applying compression algorithms or translating the payload to a different format).

Step 4 – Payload is delivered to Receiving Consumer.

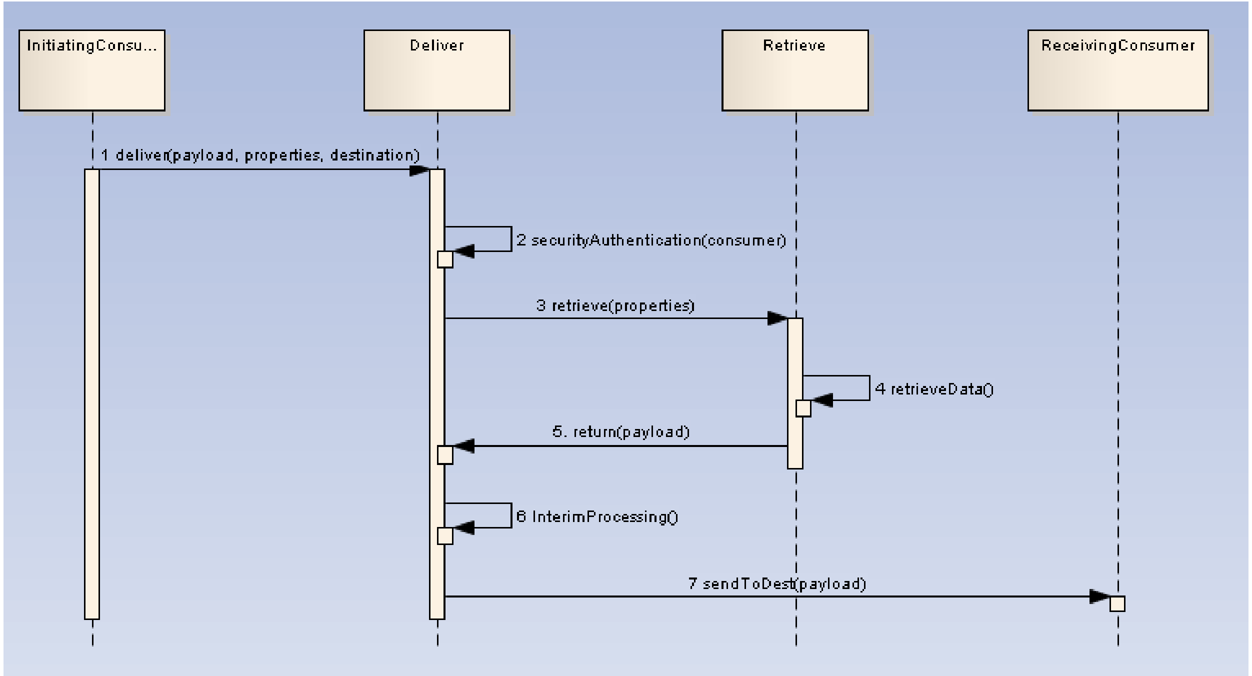

Figure 4 - Deliver: Payload Retrieved by the Deliver Service

Step 1 – Initiating Consumer sends Deliver Properties and Retrieve Properties to the ***Deliver Service***.

Step 2 – The ***Deliver Service*** leverages a set of security components to verify that the Initiating Consumer is authenticated and authorized to send a specific information resource (payload) to the Receiving Consumer; and the Receiving Consumer is authenticated and authorized to receive the specified information payload. The Joint IC/DoD Security Reference Architecture defines the specific security components and interactions needed to perform this verification.

Step 3 – The optional interim processing may represent internal capabilities of the Deliver implementation or may be external capabilities for with the Deliver implementation acts as a consumer. The ***Deliver Service*** implementation is NOT required to include any interim processing (e.g., applying compression algorithms, or translating the payload to a different format)

Step 4 - The ***Deliver Service*** acts as a Consumer in initiating the Retrieve Interaction as specified in the *Retrieve Service Specification for REST Implementations*, which can be obtained via the unclassified Intelink web site [4].

Step 5 – The Retrieve Service returns the resource requested in step 4.

Step 6 – The ***Deliver Service*** may supply requested interim processing.

Step 7 –The ***Deliver Service*** delivers resource payload to Receiving Consumer.

2.2 REST Specific Behavior Information

The ***Deliver Service*** is the application of an HTTP/HTTPS[1] POST[2] method (request) to send an information resource to a Receiving Consumer via the ***Deliver Service***, which MUST be identified by a Uniform Resource Locator (URL).

2.3 Functional Behavior

The ***Deliver Service*** is REQUIRED to function as described by the CDR Specification Framework with any input, behavior, output, and fault condition extensions listed below.

Function	**Input**	**Output**	**Fault**
Deliver	*DeliverTo, {Deliver Properties}, {Resource Payload}*		HTTP Status Code[3]

3 Deliver Service Interface

3.1 Input

The following table shows each input variable defined in the Deliver Service's Deliver function, and maps each to the Deliver Service variables as defined in the IC/DoD CDR Specification Framework (see Section 5.1 in this document).

Table 1 - Deliver Specification Input Variables

Input Variable	Required/Optional
DeliverTo	Optional[1]
Deliver Properties	Optional
Resource Payload	Required

The following examples illustrate the mechanisms for providing input in the Deliver request message. The examples assume that this service is hosted at https://example.com/DeliverComponent/deliverTo. The HTTP POST request is used to send the resource payload to the destination that is specified after the base DeliverTo element of the request Uniform Resource Identifier (URI) parameter.

Example 1: Deliver Request Message with a {Resource Payload} described as an XML document sent to a targetURL via the deliverTo service. {Deliver Properties} are included as URL parameters

```
POST /DeliverComponent?deliverTo="targetURL1” &DeliverProperty1=”DeliverProperty1Value” HTTP/1.1
Content-Type: application/xml
            {Resource Payload}
```

Example 2: Deliver Request Message with a {Resource Payload} sent to multiple targetURL(s) via the deliverTo service.

```
POST /DeliverComponent?deliverTo ="targetURL1, targetURL2, targetURL3, targetURLn” HTTP/1.1
Content-Type: application/xml
         {Resource Payload}
```

The following is a description of significant elements:

3.1.1 Deliver To

This element provides implementers with a parameter that refers to the Receiving Consumer. The ***Deliver Service*** endpoint MUST be identified within the Initiating Consumer's request.

[1] *If a DeliverTo Element is not specified, the component may deliver the {Resource Payload} to a default recipient*

3.1.2 Deliver Properties

The Deliver Properties provide a flexible mechanism to describe the characteristics of the Deliver request. As explained in the *CDR Specification Framework (see section 5.1)*, Properties MAY be expressed via a set of property identifiers, property values, and an optional value description.

3.1.3 Resource Payload

The Resource Payload refers to the information resource to be delivered. If the Resource Payload is not supplied, the information required by the ***Deliver Service*** to retrieve the information resource MUST be provided via the Deliver Properties. If a payload is included in the Deliver Request: the payload MUST be contained in HTTP/HTTPS POST Message Body. Additionally, if the Content-Type and Content-Length are known, the Deliver Service Request SHOULD populate this information into the corresponding HTTP Header.

3.2 Output

Implementations of the Deliver Service MUST return the appropriate HTTP status code (based on values from the HTTP Status Code Registry maintained by IANA). The Deliver Service MUST adhere to approved information assurance guidelines and constraints regarding distribution of information.

3.3 Fault Conditions

An implementation of the Deliver Service MUST allow for the Fault Conditions defined in the CDR Specification Framework.

4 External Dependencies

4.1 Service Security

The Security focus area provides a set of security-focused services to the IC and DoD for protecting access to services, data, and their interactions within the IC/DoD Enterprise. Integration of Security capabilities is advocated, both from the service discovery and the service access standpoint, to protect content providers and consumers from attack from any unknown entities. Security capabilities are responsible for authenticating and authorizing of consumers and consumer agents, binding IA metadata to the information that it describes (query, search result, or retrieved content), controlling access to content resources, and enabling cross-domain search and retrieval. Furthermore, security capabilities provide integrity, confidentiality, and audit services that CDR providers can leverage. CDR providers together with their security engineers should reference the Joint IC/DoD Security Reference Architecture (SRA) [5] for guidance on integrating and using the security services within and between CDR components[2]. It is expected that the SRA and derived specifications will provide guidance for implementers of the CDR components which identifies interface points for requesting security services. As appropriate, this guidance will be documented within the CDR Architecture Model to achieve secure CDR services.

4.1.1 Service Security Concerns

The following security relevant considerations are consolidated in this section to more clearly define points of intersection and dependency upon the Joint IC/DOD Security Reference Architecture (SRA)[5] that may be of importance in realizing the CDR Compliant Services:

- Identification and Authentication: The operations defined here require the Consumer Component to provide an authenticated identity to the CDR Component it is calling. The authentication requirement extends to authenticating CDR Components acting on behalf of a consumer (chained authentication).
- Activity Authorization: The CDR Component must determine if the authenticated consumer is authorized to perform the requested activity. In addition, it must determine if the intended recipients of delivered metadata or resource content are authorized to receive it.
- Access Control: The CDR Component must abide by the access control policies for search results and retrieved content based on their IA Metadata, and on Consumer and CDR Component security attributes. Access control is applied to both the Content Collection and individual Content Resources within the Collection.
- Classification: General rules and specifications referring to the classification of saved resources also apply to CDR Components, but are not described in this framework.
- Auditing and Logging: General rules and specifications referring to the auditing and logging of data apply to CDR Components, but are not described in this framework.
- Protecting confidentiality, integrity, availability and non-repudiation: General rules and specifications referring to these security concerns apply to CDR components, but are not described in this specification. This includes message level and transport level security.

4.1.2 Security Fault Conditions

The following potential security fault conditions are common to most of the CDR capabilities:

- Action Not Authorized: The Consumer does not have permission to perform the requested function on the requested resource.

[2] *This guidance could also cover the security aspects of CDR components interacting with non-CDR components.*

- Identity Not Authenticated: The Consumer could not be authenticated. (The claimed identity could not be confirmed.)

5 Reference

This section includes additional references that may be used to provide further insight into the overall design concepts that serve to guide the CDR-IPT engineering efforts.

5.1 Content Discovery and Retrieval Specifications

The CDR Reference Architecture and Specification Framework provide essential background and context to service designers. This document was based on the following CDR Reference Architecture and Specification Framework document versions:

- “IC/DoD Content Discovery and Retrieval Reference Architecture Version 1.0”, 19 December 2009
- “IC/DoD Content Discovery and Retrieval Specification Framework Version 0.9” 6 June 2010

The most recent version of the documents can be found at the unclassified Intelink web site [4].

5.2 Additional References

[1] HTTP/HTTPS – http://www.w3.org/Protocols/

[2] HTTP Method Definitions - http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html

[3] Hypertext Transfer Protocol (HTTP) Status Code Registry – http://www.iana.org/assignments/http-status-codes

[4] Unclassified Intelink web site – https://www.intelink.gov/site/odni/cio/i2e/focus/iads/cdript/default.aspx

[5] Intelligence Community and Department of Defense Service Security Working Group, Joint IC/DoD Security Reference Architecture, available via Intelink