

**Virtual Lifetime Electronic Record (VLER)  
Data Access Services (DAS)**

**Gateway Interface Control Document (ICD)**



**Version 6.0**

**March 2015**

## Revision History

| <b>Date</b> | <b>Version</b> | <b>Description</b>                | <b>Author</b> |
|-------------|----------------|-----------------------------------|---------------|
| 05/28/2014  | 0.01           | Initial draft                     | G. Ludgate    |
| 05/28/2014  | 0.02           | Technical Writer Review           | N. Burak      |
| 05/28/2014  | 0.03           | Management Review                 | C Moore       |
| 05/28/2014  | 1.0            | Approved                          |               |
| 11/26/2014  | 1.1            | November Deliverable – no updates | Nancy Burak   |
| 11/26/2014  | 1.2            | Management Review                 | Cam Moore     |
| 11/26/2014  | 2.0            | Approved                          |               |
| 12/30/2014  | 2.1            | December Deliverable – no updates | Nancy Burak   |
| 12/30/2014  | 2.2            | Management Review                 | Cam Moore     |
| 12/30/2014  | 3.0            | Approved                          |               |
| 01/30/2015  | 3.1            | January Deliverable – no updates  | Nancy Burak   |
| 01/30/2015  | 3.2            | Management Review                 | Cam Moore     |
| 01/30/2015  | 4.0            | Approved                          |               |
| 02/27/2015  | 4.1            | February Deliverable – no updates | Nancy Burak   |
| 02/27/2015  | 5.0            | Approved                          |               |
| 03/22/2015  | 5.1            | March Deliverable – no updates    | Nancy Burak   |
| 03/22/2015  | 6.0            | Approved                          |               |

Blue text in the body of the ICD means “to be completed”

# Table of Contents

|   |           |
|---|-----------|
| <b>1. Introduction</b>                                  | <b>1</b>  |
| 1.1. Purpose  | 1         |
| 1.2. Scope  | 1         |
| 1.3. System Identification                              | 2         |
| 1.3.1. External Partner’s Applications                  | 2         |
| 1.3.2. VLER Gateway                                     | 2         |
| <b>2. Interface Definition</b>                          | <b>3</b>  |
| 2.1. System Overview                                    | 3         |
| 2.2. Interface Overview                                 | 5         |
| 2.2.1. Exchange Protocol                                | 5         |
| 2.2.2. Request Web Service                              | 11        |
| 2.2.3. Response Web Service                             | 17        |
| 2.3. Operations   | 21        |
| 2.4. Security   | 21        |
| 2.4.1. Certificates                                     | 21        |
| 2.4.2. On Boarding                                      | 21        |
| <b>Appendix A – Request WSDL Files</b>                  | <b>22</b> |
| <b>Appendix B – Response WSDL Files</b>                 | <b>23</b> |
| <b>Appendix C – Harris-Integration Specific Details</b> | <b>24</b> |
| C.1 DBQ Automation Solution                             | 24        |
| C.2 Web Services  | 25        |
| C.3 Exchange Protocol                                   | 25        |
| C.4 Request Web Service                                 | 25        |
| <b>Appendix D – D2D-Integration Specific Details</b>    | <b>27</b> |
| D.1 D2D Automation Solution                             | 27        |
| D.2 Web Services  | 27        |
| D.3 Exchange Protocol                                   | 32        |
| <b>Appendix E – VES-Integration Specific Details</b>    | <b>33</b> |
| E.1 VDBQ Portal   | 33        |
| E.2 Web Services  | 33        |
| E.3 Exchange Protocol                                   | 39        |

## List of Tables

|  |           |
|--|-----------|
| <i>Table 1-1 – Partner’s Identification Information .....</i>                            | <i>2</i>  |
| <i>Table 1-2 - VLER Gateway Identifiers .....</i>  | <i>2</i>  |
| <i>Table 2-1 – Transfer Protocol Parameters.....</i>                                     | <i>10</i> |
| <i>Table 2-2 – Request Envelope variables and example-values .....</i>                   | <i>13</i> |
| <i>Table 2-3 – Location of web Service operation details .....</i>                       | <i>21</i> |
| <i>Table 0-1 - DBQ Automation Solution Identifiers .....</i>                             | <i>24</i> |
| <i>Table 0-2 – Transfer Protocol Parameters.....</i>                                     | <i>25</i> |
| <i>Table 0-1 – Inbound Services used by D2D.....</i>                                     | <i>28</i> |
| <i>Table 0-2 – Outbound Services implemented by External Partner’s Applications.....</i> | <i>30</i> |
| <i>Table 0-3 – D2DTransfer Protocol Parameters.....</i>                                  | <i>32</i> |
| <i>Table 0-4 – D2DTransfer Protocol Parameters (cont) .....</i>                          | <i>32</i> |
| <i>Table 0-1 – Inbound Services used by VDBQ Portal.....</i>                             | <i>34</i> |
| <i>Table 0-2 – Outbound Services implemented by VDBQ Portal.....</i>                     | <i>36</i> |
| <i>Table 0-3 – Outbound Services VDBQ Portal Endpoint .....</i>                          | <i>36</i> |
| <i>Table 0-4 – VDBQ Portal Transfer Protocol Parameters .....</i>                        | <i>39</i> |

## List of Figures

|   |    |
|---|----|
| <i>Figure 2-1 – VLER Gateway Context Diagram</i>  | 3  |
| <i>Figure 2-2 – Application-architecture for servicing external requests.</i>                       | 4  |
| <i>Figure 2-3 – Interface Extended-Data-Flow-Diagram showing implementation mechanisms.</i>         | 5  |
| <i>Figure 2-4 – Web Services flows.</i>   | 6  |
| <i>Figure 2-5 – State Transition Dgm of possible statuses of one instance of a Request Payload.</i> | 6  |
| <i>Figure 2-6 – Request web service concrete WSDL</i>   | 11 |
| <i>Figure 2-7 – Request web service abstract WSDL</i>   | 12 |
| <i>Figure 2-8 – Example request which should succeed</i>  | 15 |
| <i>Figure 2-9 – Example web service success return value (acknowledgement) for a request</i>        | 16 |
| <i>Figure 2-10 – Example web service failure return value (acknowledgement) for a request</i>       | 16 |
| <i>Figure 2-11 – Response web service concrete WSDL</i>   | 17 |
| <i>Figure 2-12 – Response web service abstract WSDL</i>   | 18 |
| <i>Figure 2-13 - Example “success” response (no returned response data)</i>                         | 19 |
| <i>Figure 2-14 - Example “failure” response (no returned response data)</i>                         | 20 |
| <i>Figure 2-15 - Example web service success return value (acknowledgement) for a response</i>      | 20 |
| <i>Figure 2-16 - Example web service failure return value (acknowledgement) for a response</i>      | 20 |
| <i>Figure 0-1 – Example D2D request which should succeed</i>  | 29 |
| <i>Figure 0-2 – Example D2D success response containing a response-data-document</i>                | 31 |
| <i>Figure 0-1 – Example VDBQ Portal request (which should succeed)</i>                              | 35 |
| <i>Figure 0-2 – Example VES “success” response</i>  | 37 |
| <i>Figure 0-3 – Example VES “failure” response</i>  | 38 |

# 1. Introduction

This document is the Interface Control Document (ICD) for the interface between External Partner's applications and the internal Virtual Lifetime Electronic Record (VLER) Gateway application.

In this ICD "external" means "external to the Department of Veterans Affairs (VA) network" while "internal" means "on the VA network". In addition "inbound" means toward the VA and "outbound" toward the External Partners.

External Partner's applications are those applications running external to the VA but interfacing to the VLER Gateway to exchange data with VA internal systems.

## 1.1. Purpose

This ICD serves as a specification of the interface between External Partner's applications and the internal VLER Gateway application. It will be used by:

1. Developers of the internal VLER Gateway application.
2. Developers of the External Partner's applications .

This interface is designed to allow the External Partner's applications to send requests for services to the VLER Gateway and to receive back a response, for each request, specifying the success or failure of the request.

## 1.2. Scope

This ICD focuses on the software interface between External Partner's applications and the internal VLER Gateway application. It describes the operations (or transaction types), data transfers and communication methods of the service interface supported by the VLER Gateway.

Upon formal approval by each External Partner this ICD shall be incorporated into the requirements baseline for the VLER Gateway and all External Partner's applications .

This document does not specify the software design of the VLER Gateway nor of the applications which interact with the VLER Gateway.

## 1.3. System Identification

### 1.3.1. External Partner's Applications

The details of External Partner's applications currently interfaced to the VLER Gateway are provided in appendices of this ICD as follows:

**Table 1-1 – Partner's Identification Information**

| <i>Project Short Name</i> | <i>Location</i>                                  |
|---------------------------|--|
| "DBQ"                     | Appendix C – Harris-Integration Specific Details |
| "D2D"                     | Appendix D – D2D-Integration Specific Details    |
| "VES"                     | Appendix E – VES-Integration Specific Details    |

### 1.3.2. VLER Gateway

The VLER Gateway serves as a gateway and a processes execution service between External Partner's applications and all internal VLER applications.

Table 1-2 identifies the VLER Gateway application for which this ICD defines the communications interface.

**Table 1-2 - VLER Gateway Identifiers**

| <i>System</i>         | <i>Details</i>                                       |
|-----------------------|--|
| Identification number | VLER 17  |
| Title                 | Virtual Lifetime Electronic Record Gateway           |
| Abbreviation          | Gateway  |
| Version number        | 7.0  |
| Release number        | 7.0  |
| Point of Contact      | VA Government Program Manager or Contracting Officer |

## 2. Interface Definition

This ICD specifies the request and response interfaces between External Partner’s applications and the VLER Gateway, as shown in Figure 2-1. The interface is designed to guarantee the processing of every request.

In this ICD the term “request” refers to a **business-request** sent from the External Partner to the VLER Gateway while the term “response” refers to a **business-response** sent to the External Partner from the VLER Gateway.

### 2.1. System Overview

The VLER Gateway serves as a gateway between External Partner’s applications and internal VLER applications and Internal Service Applications as shown in Figure 2-1 (a data flow diagram). The External Partner’s applications currently initiate all requests for service to the VA Gateway and may receive back a response, for each request, specifying the success or failure of the request.

In this version of the ICD, all requests from the External Partner’s applications must be requests for asynchronous processing of requests. Synchronous processing of requests is not yet supported.

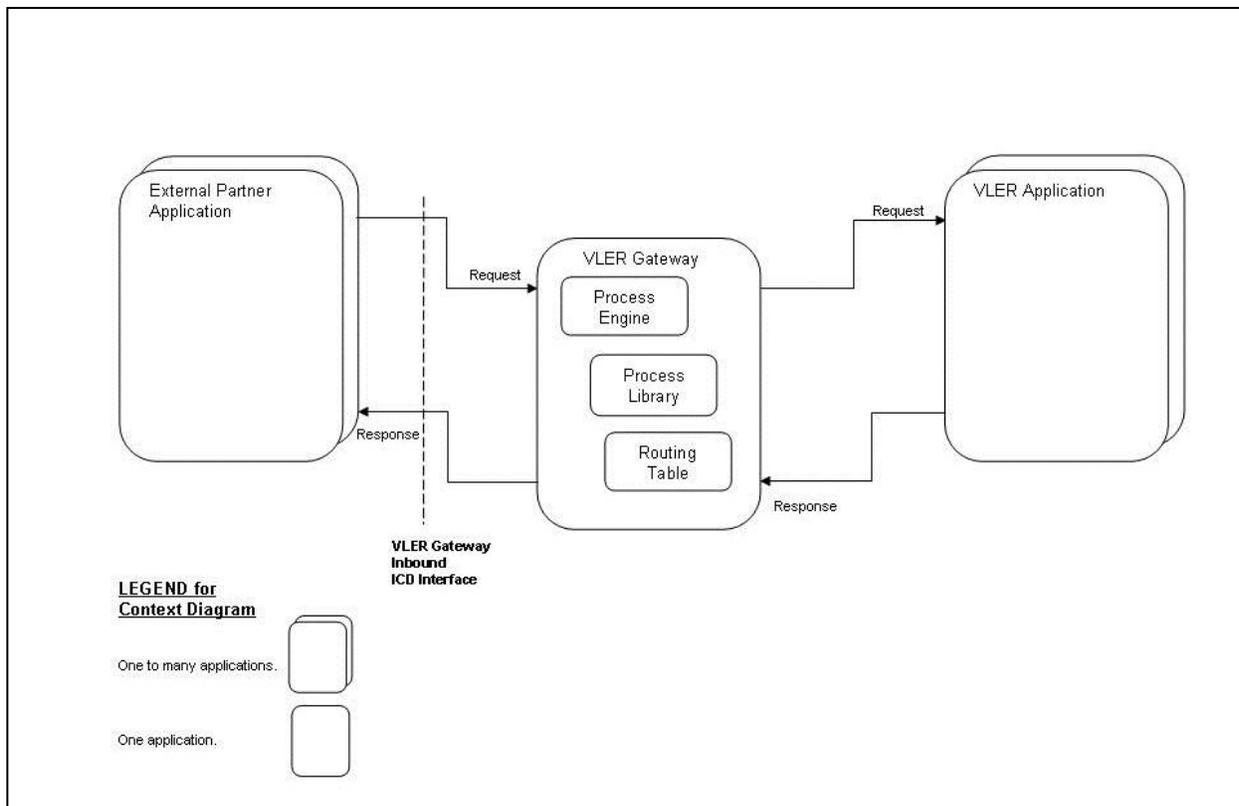
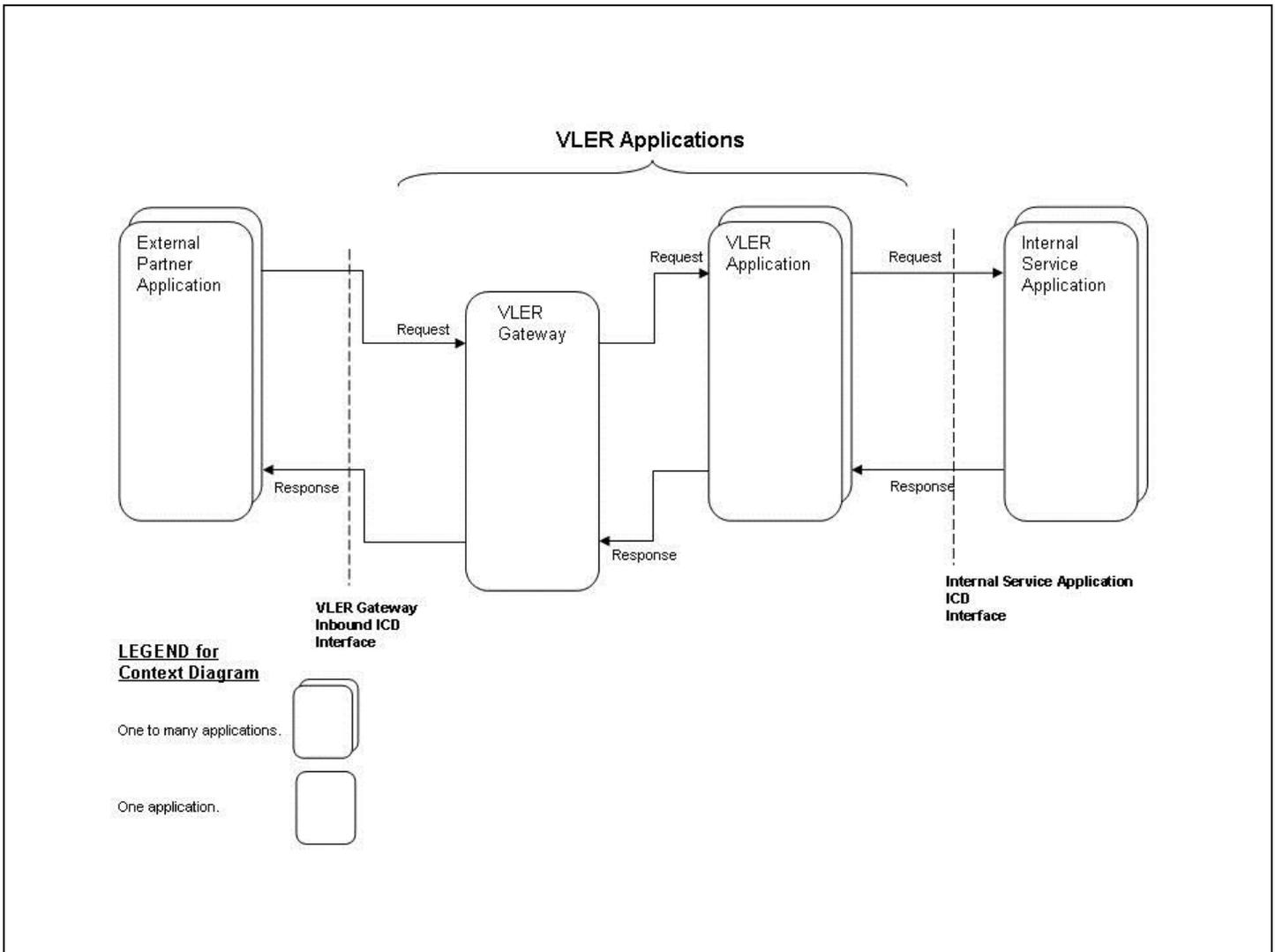


Figure 2-1 – VLER Gateway Context Diagram

Requests for service are transported by VLER applications (see Figure 2-2) to Internal Service Applications which satisfy the requests and return appropriate responses. VLER applications return the responses to the original External Partner's application which sent in the request. Figure 2-2 shows the application architecture for servicing external requests. The decision as to whether a response is provided is made by the Internal Service Application.

A detailed description of the Internal Service Applications and their interfaces is outside the scope of this ICD. This data-flow diagram (Figure 2-2) of the application architecture is included to provide an end-to-end understanding of the flow of requests and responses within VA.



**Figure 2-2 – Application-architecture for servicing external requests.**

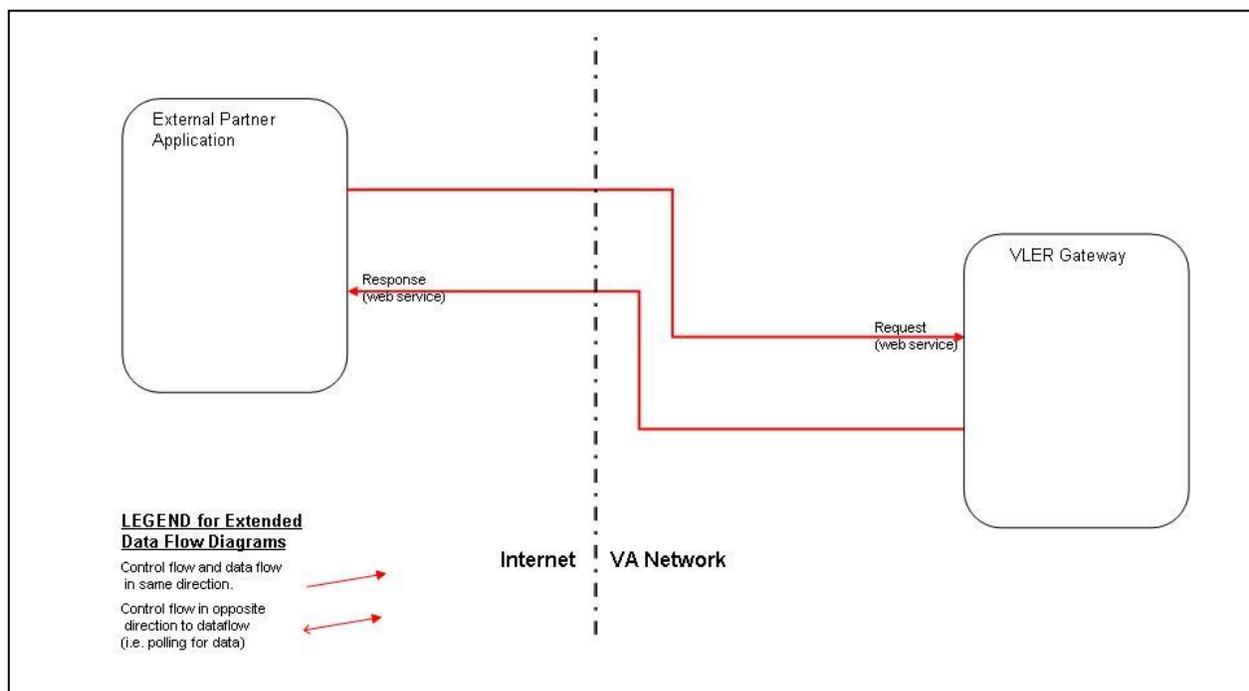
Request response times, the maximum possible incoming request rate and the maximum size of an incoming request may be determined by VLER Applications or by the Internal Service Applications. The values for specific External Partner's Applications are listed in the appropriate appendix defined in Table 1-1.

## 2.2. Interface Overview

### 2.2.1. Exchange Protocol

The communications mechanism selected to operate between the External Partner's applications and the internal VLER Gateway is SOAP based web services over HTTPS, as shown in the extended data flow diagram Figure 2-3 and the web services diagram Figure 2-4.

The External Partner's applications must use one or more web services to send requests for service to the VLER Gateway, over both the Internet and the VA Network. Similarly the internal VLER Gateway must use a web service to return responses to the requesting External Partner's application over the same networks.

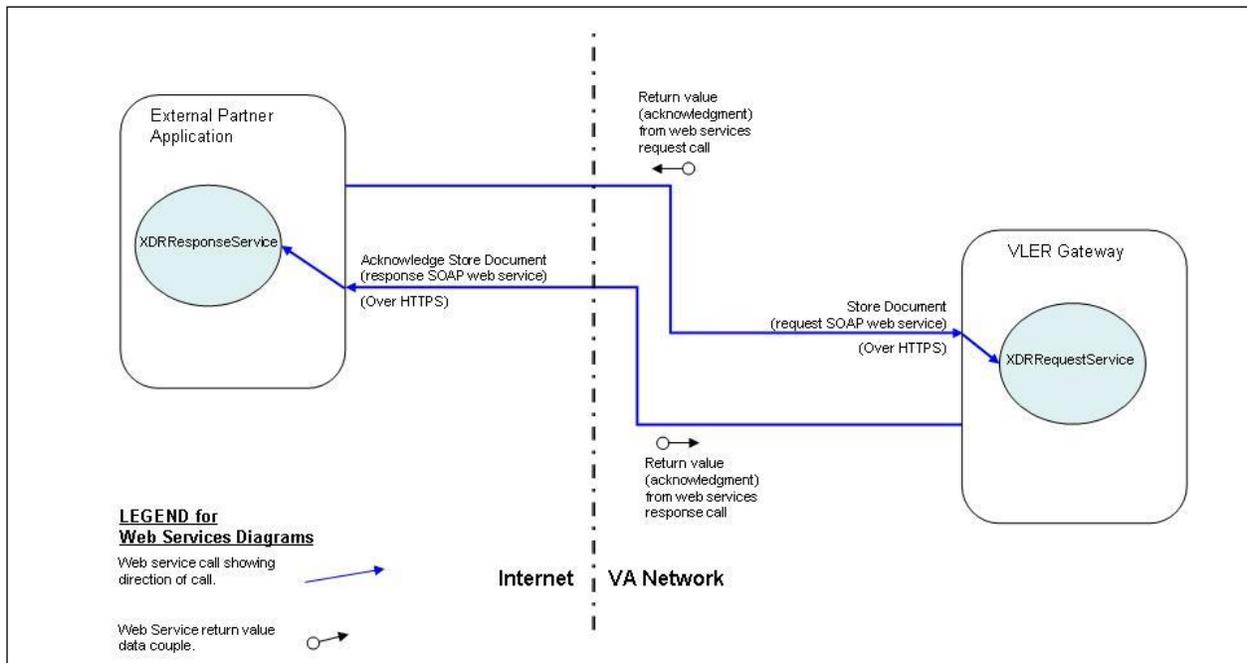


**Figure 2-3 – Interface Extended-Data-Flow-Diagram showing implementation mechanisms.**

Inbound web services allow External Partner's applications to submit requests for service to the VLER Gateway, one request per web service call, for later, asynchronous, processing by VLER applications.

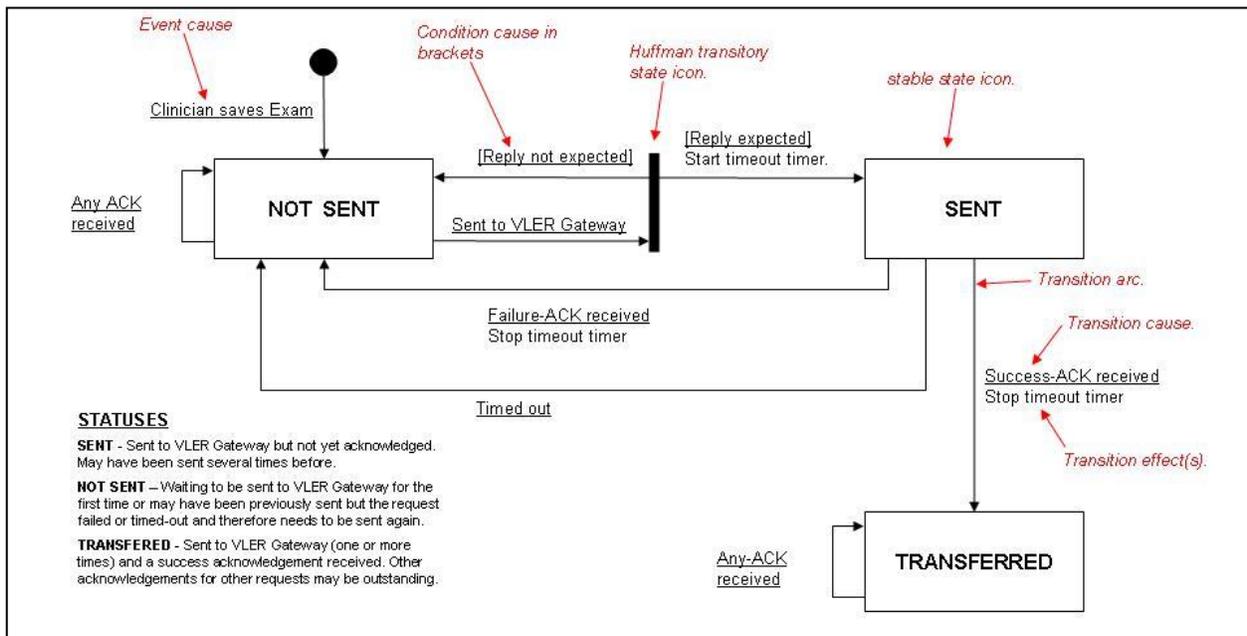
The outbound web service allows the VLER Gateway to return a particular response to the particular instance of an External Partner's application which earlier sent in a specific request for service.

The request and response web service calls between one requester (an External Partner's application) and the responder (the VLER Gateway) are illustrated in Figure 2-4.



**Figure 2-4 – Web Services flows.**

Figure 2-5 shows possible statuses which could be used to track the processing of one asynchronous request for service sent to VA. The statuses are shown as the states of a State Transition Diagram.



**Figure 2-5 – State Transition Dgm of possible statuses of one instance of a Request Payload.**

The protocol for sending one request for service to the VLER Gateway is described by the process below, comprising three threads of individual steps. Only the “happy day” path is described i.e. the one in which all actions are assumed to be carried out perfectly and as expected. This protocol assumes that individual requests are idempotent<sup>1</sup>. This process can only take place when all systems involved in the request operation are operating in a normal production mode. Planned downtimes by internal VLER applications must be allowed for in the External Partner’s production schedules. See Table 2-1 for details of how to obtain the internal systems planned downtimes (e.g. for VLER Gateway and VLER applications).

The “happy day” sequence of protocol steps (the sending thread) to transfer one instance of a new request to the VLER Gateway is as follows:

1. The External Partner’s application selects one request for service to send to the VLER Gateway. This request may be “flagged” as “NOT SENT” after its creation (see Figure 2-5). The External Partner’s application may keep performance statistics on the number of times a request has been made, the number of times a failure acknowledgement has been received back for a request, the number of times the transmission has timed out for a request , etc. If the number of transfer attempts for this request exceeds the maximum number specified in Table 2-1 then VLER Operations must be contacted to notify them of the production problem. It is suggested that this entire requesting process be ended and transfers stopped until the situation is remedied. See Table 2-1 for details of how to contact VLER Operations for this eventuality.
2. The External Partner’s application must Base64 encode the request XML and insert it into the “Document” element of the request-envelope XML as described in Section 2.2.2 and Table 2-2.
3. The External Partner’s application must fill in values for the remaining elements of the request-envelope XML as described in Section 2.2.2 and Table 2-2.
4. The External Partner’s application must transfer the request-envelope XML via the request web service call to the VLER Gateway. This web service call constitutes a request to VA to perform one service for that specific External Partner’s application . See Section 2.2.2 for details of the service. There is no guarantee that (1) requests will be processed, even if accepted by VLER Gateway, (2) requests will be processed in the order they are received by the VLER Gateway and (3) a response will be returned for every request.
5. The return-value of the request web service call (returned synchronously to the sender as shown in Figure 2-4) will indicate whether the request was received successfully by the VLER Gateway, for later asynchronous processing by VLER applications.
  - a. If the return-value indicates the request was received successfully then
    - i. If a response is expected for this request then the External Partner’s application must start a timeout timer for this request operation and the request may be “flagged” as “SENT”.
6. This thread of the process can continue with Step 1 if the External Partner’s application determines all conditions are acceptable, e.g.:
  - a. The VLER Gateway will accept requests up to a maximum rate as defined in Table 2-1.

---

<sup>1</sup> Requests can have the property of "idempotence" in that (aside from error or expiration issues) the side-effects of N > 0 identical requests is the same as that for a single request. However, it is possible that a sequence of several requests is non-idempotent, even if all of the requests executed in that sequence are idempotent. (A sequence is idempotent if a single execution of the entire sequence always yields a result that is not changed by a re-execution of all, or part, of that sequence.) For example, a sequence is non-idempotent if its result depends on a value that is later modified in the same sequence. A sequence that never has side effects is idempotent, by definition (provided that no concurrent operations are being executed on the same set of resources).

The “happy day” sequence of protocol steps (the acknowledgement thread) to receive back an acknowledgement (the response) for a request is as follows:

7. The External Partner’s application receives a web service call from the VLER Gateway containing the response to an earlier request to the VLER Gateway.
8. The External Partner’s application examines the response to determine for which request the response was sent. (See Section 2.2.3 for a description of how to correlate requests with responses).
9. If a response for a given request arrives before the request operation timeout period expires (timeout was set in Step 5(a) ), then the External Partner’s application :
  - a. Stops the timeout timer for this request operation .
  - b. If the response indicates the request operation was successful then:
    - i. The original request may be “flagged” as “TRANSFERRED” (see Figure 2-5) .
    - ii. End this thread.
  - c. If the response indicates the request operation was not successful then:
    - i. *(The overall process will retry sending this request again when Step 1 is performed in the sending thread.)*
    - ii. The original request may be “flagged” as “NOT SENT” (see Figure 2-5).
    - iii. End this thread.
10. If a response for a given request arrives after the request operation timeout period expires (i.e. the original request may have been “flagged” as “TRANSFERRED” or “NOT SENT” (see Figure 2-5) ) then the External Partner’s application :
  - a. Ignores the response.
  - b. *(This situation may arise if the request has been sent several times without receiving a response, each time encountering a timeout, because VLER applications are running slowly. The requests may be processed a long time after they are sent and responses generated. The responses that are not lost as they pass back through VLER applications are returned later to the External Partner’s application but may arrive in any order).*
  - c. Ends this thread.

Some requests may not expect to receive back a response. This choice is at the discretion of the External Partner’s application architect.

In the sending thread above, a timeout timer may be set for each request operation. It is expected that VLER Gateway will receive and process a request within this timeout period. However, if the timeout timer for a request expires, the following sequence of protocol steps (the timeout thread) must be followed:

11. The External Partner's application examines the timeout to determine for which request the timeout was set.
12. If the request in this transfer is "flagged" as "SENT" (see Figure 2-5) i.e. having been sent but not yet successfully transferred then:
  - a. The request may be "flagged" as "NOT SENT" (see Figure 2-5)
  - b. *(The overall process will retry sending this request again when Step 1 is performed in the sending thread.)*
13. End this thread.

Some requests may not expect to receive back a response. This choice is at the discretion of the External Partner's application architect.

In addition to contacting VLER Operations for those situations which arise in the above process, VLER Operations must be contacted to notify them of a production problem if:

1. The External Partner's applications are unable to connect to the VLER Gateway web service for network-related reasons.
2. The VLER Gateway returns error codes for transfers and does not complete inbound or outbound transfers as expected.

See Table 2-1 for details of how to contact VLER Operations.

The VLER Gateway and its supporting applications are designed to accept multiple instances of the same request for service as identified by the "id" attribute of the "Document" element in Table 2-2. Only one instance of a given request for service should be executed successfully regardless of the number of times that specific request is transferred to VA (see Footnote 1). If, therefore, one or more of the External Partner's computers should fail or a disastrous event occur then the recovered computers should resend any instances of requests for which an uncertainty exists as to whether they were executed by VA or not. Requests for service are assumed in this ICD to be idempotent. It is the responsibility of the system executing, and thereby satisfying, a request to ensure this property (i.e. the Internal Service Application). Such a system may be designed to return a variety of success messages, for example indicating whether the request has been processed before.

The following is assumed for all requests for service:

1. The "id" attribute of the "Document" element (see Table 2-2) serves as a unique identifier of a given request for service. If the given request is updated by the External Partner's application, for any reason, a new document ID value must be assigned.
2. The VLER Gateway will accept multiple instances of the same request for service, identified by a single document ID value. If the request for service is successful then a success response will eventually be returned to the originating External Partner's application.

Table 2-1 list parameters of the request/response transfer protocol. These parameters may change in future versions of this ICD.

**Table 2-1 – Transfer Protocol Parameters**

| Protocol Parameter  | Value  |
|---|--|
| VLER Gateway Server URL available on the Internet.                              | VLER.VA.GOV – for production.<br>TESTVLER.VA.GOV – for SQA testing. <b>This URL will be deprecated in future. Please use the GOLD URL below.</b><br>GOLDVLER.VA.GOV – for SQA testing.<br>SILVERVLER.VA.GOV – for integration testing.                           |
| Request for service operation-timeout period.                                   | See appendices for External Partner’s application specific values. This value depends on the VA system satisfying the request.   |
| VLER Operations contact.  | Contact the VA Government Program Manager or Contracting Officer for details of how to contact staff knowledgeable about VLER operations .   |
| Internal systems planned downtimes.   | Contact the VA Government Program Manager or Contracting Officer for details of how to obtain scheduled downtime periods.  |
| Maximum number of transfer retries for a given request for service, any reason. | 3  |
| Maximum request send rate.  | See appendices for External Partner’s application specific values. This value depends on the VA system satisfying the request.   |
| Maximum size of request   | See appendices for External Partner’s application specific values. This value may depend on: <ol style="list-style-type: none"> <li>1. The VA system satisfying the request.</li> <li>2. The value of the “operationName” slot described in Table 2-2</li> </ol> |

## 2.2.2. Request Web Service

The request web service is implemented by the VLER Gateway as shown in Figure 2-4.

The concrete Web Services Description/Definition Language (WSDL) for the VLER Gateway web service is shown below in Figure 2-6 and the abstract WSDL in Figure 2-7. The files defining the request web service WSDL are attached in Appendix A – Request WSDL Files.

```
<?xml version="1.0" encoding="utf-8"?>
<definitions
  xmlns:tns="http://_2007.request.async.xdr.iti.ihe/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
  name="XDRRequestService" targetNamespace="http://_2007.request.async.xdr.iti.ihe/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
  <import
    namespace="urn:ihe:iti:xdr:async:request:2007"
    location="http://SERVER:PORT/XDRRequest_PortTypeWS/XDRRequestService?wsdl=1" />
  <types />
  <binding
    xmlns:ns1="urn:ihe:iti:xdr:async:request:2007"
    name="XDRRequest_PortTypeBinding"
    type="ns1:XDRRequest_PortType">
    <wsaw:UsingAddressing />
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="ProvideAndRegisterDocumentSet-bRequest">
      <soap:operation
        soapAction="tns:ProvideAndRegisterDocumentSet-bRequest" />
      <input>
        <soap:body use="literal" />
      </input>
      <output>
        <soap:body use="literal" />
      </output>
    </operation>
  </binding>
  <service name="XDRRequestService">
    <port name="XDRRequest_PortType"
      binding="tns:XDRRequest_PortTypeBinding">
      <soap:address
        location = "http://SERVER:PORT/XDRRequest_PortTypeWS/XDRRequestService" />
    </port>
  </service>
</definitions>
```

Figure 2-6 – Request web service concrete WSDL

The value of the **yellow highlighted** “location” attributes in the WSDL of Figure 2-6 must be specified by VLER Operations in concert with the VLER Gateway team with:

1. URL scheme:
  - a. “HTTP” when the communications path is not protected by SSL and
  - b. “HTTPS” when the communications path is protected by SSL.
2. The fully qualified SERVER host-name as shown in Table 2-1
3. The PORT selected for communications with VLER Gateway.
  - a. 80 for HTTP
  - b. 443 for HTTPS

There will be differences between the values for test and production. Contact VLER Operations to confirm the above details (see Table 2-1 for contacting VLER Operations).

The abstract WSDL is shown below in Figure 2-7.

```
<?xml version="1.0" encoding="utf-8"?>
<definitions
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="urn:ihe:iti:xdr:async:request:2007"
  targetNamespace="urn:ihe:iti:xdr:async:request:2007"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
  <types>
    ... types omitted to save space.
  </types>
  <message name="ProvideAndRegisterDocumentSet-bRequest">
    <part xmlns:ns1="urn:ihe:iti:xds-b:2007"
      name="body" element="ns1:ProvideAndRegisterDocumentSetRequest" />
  </message>
  <message name="ProvideAndRegisterDocumentSet-bRequestResponse">
    <part xmlns:ns2="urn:ihe:iti:xdr:2007"
      name="body" element="ns2:Acknowledgement" />
  </message>
  <portType name="XDRRequest_PortType">
    <operation name="ProvideAndRegisterDocumentSet-bRequest">
      <input message="tns:ProvideAndRegisterDocumentSet-bRequest" />
      <output message="tns:ProvideAndRegisterDocumentSet-bRequestResponse" />
    </operation>
  </portType>
</definitions>
```

**Figure 2-7 – Request web service abstract WSDL**

Table 2-2 lists those elements, attributes and value text fields of the request envelope document which may vary from request to request. Rows are paired (color coded rows) in the table with the first row of the pair identifying a variable name and the second row describing the variable's allowed values. This set of variables define the request to store a document in VA. Figure 2-8 is an example completed request.

Any data (i.e. a document to be stored) associated with the request must be Base64 encoded and included as the value of the <Document> element in the request envelope. This data must satisfy the following:

1. Be valid XML conforming to an XSD provided to VA before the first transfer of such XML in a production environment.
2. Have a character encoding of UTF-8 before being Base64 encoded into the <Document> element of the envelope document.

**Table 2-2 – Request Envelope variables and example-values**

| Variable/<br>Value   | Description of Variable/<br>Value   |
|--|---|
| <code>&lt;urn1:SubmitObjectsRequest id="a498fb06-43b0-452e-94d7-1776d42ce71e"&gt;</code> | Required<br>The “id” attribute is the unique identifier of this request message. The asynchronous response message for this request will have a matching value in the “requestID” attribute of the <RegistryResponse> element (See Figure 2-13 or Figure 2-14). |
| <code>&lt;urn1:SubmitObjectsRequest id="a498fb06-43b0-452e-94d7-1776d42ce71e"&gt;</code> | Required<br>Set the value of this attribute to be the unique identifier of this message.<br><br>Type: UUID or unique identifier   |
| <code>&lt;urn3:Slot name="operationName" &gt;</code>                                     | Required<br>This slot identifies the need for requestors to specify an operation name.  |
| <code>&lt;urn3:Value&gt;StoreExaminationResults&lt;/urn3:Value&gt;</code>                | Required.<br>Set the value of this slot to be the name of the operation to be performed by the VLER Gateway for this request.<br><br>Type: Text   |
| <code>&lt;urn3:Slot name="originatingOrganizationName" &gt;</code>                       | Required<br>This slot identifies the need for requestors to specify the name of the originating organization making the request.  |

| Variable/<br>Value                               | Description of Variable/<br>Value  |
|--|--|
| <urn3:Value>Harris Corporation</urn3:Value>      | Required<br>Set the value of this slot to be the name of the originating organization making the request.<br>Type: Text                                |
| <urn3:Slot name="originatingApplicationName" >   | Required<br>This slot identifies the need for requestors to specify the name of the originating External Partner's application making the request.     |
| <urn3:Value>Forms Service</urn3:Value>           | Required<br>Set the value of this slot to be the name of the originating External Partner's application making the request.<br>Type: Text              |
| <urn:Document id="0001">                         | Required<br>The document "id" attribute is the unique identifier of the instance of the request payload.   |
| <urn:Document id="0001">                         | Required<br>Set the value of this attribute to be a unique identifier of the instance e.g. a UUID.<br>Type: UUID or unique identifier.                 |
| <urn:Document > ... </urn:Document>              | Required<br>The document element contains the Base64 encoded data associated with this request for service.  |
| <urn:Document >PGdvd... lbWlhPg==</urn:Document> | Required<br>Set the document element value to be the Base64 encoded value of the data associated with this request for service.<br>Type: Text (Base64) |

Figure 2-8 shows an example request envelope document completed according to the guidelines for elements listed in Table 2-2. The Base64 encoded instance of a request for service document (highlighted in green) has been truncated to fit the space available on this page. Values for the other variable have been highlighted in yellow.

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:urn="urn:ihe:iti:xds-b:2007"
  xmlns:urn1="urn:oasis:names:tc:ebxml-regrep:xsd:lcm:3.0"
  xmlns:urn2="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0"
  xmlns:urn3="urn:oasis:names:tc:ebxml-regrep:xsd:rsm:3.0">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:ProvideAndRegisterDocumentSetRequest>
      <urn1:SubmitObjectsRequest id="a498fb06-43b0-452e-94d7-1776d42ce71e">
        <urn2:RequestSlotList>
          <!--Zero or more repetitions:-->
          <urn3:Slot name="operationName" >
            <urn3:ValueList>
              <!--Zero or more repetitions:-->
              <urn3:Value>StoreExaminationResults</urn3:Value>
            </urn3:ValueList>
          </urn3:Slot>
          <urn3:Slot name="originatingOrganizationName" >
            <urn3:ValueList>
              <!--Zero or more repetitions:-->
              <urn3:Value>Harris Corporation</urn3:Value>
            </urn3:ValueList>
          </urn3:Slot>
          <urn3:Slot name="originatingApplicationName" >
            <urn3:ValueList>
              <!--Zero or more repetitions:-->
              <urn3:Value>Forms Service</urn3:Value>
            </urn3:ValueList>
          </urn3:Slot>
        </urn2:RequestSlotList>
        <urn3:RegistryObjectList/>
      </urn1:SubmitObjectsRequest>
      <!--Zero or more repetitions:-->
      <urn:Document id="0001">
        PGdvdI52YS5hZ2VudG9yYW5nZS5mYXN0dHJhY2subW9kZWwuZGJxLmxiay5iYWlyeUNlbgxMZXRr
        ... ..
        aWNhdGlvbnM+PC9hbnlPdGhlcjNvbXBsaWNhdGlvbnM+DQoJCTwvZ292LnZhLmFnZW50b3Jhbmdl
        LmZhc3R0cmFjay5tb2R1bC5kYnEubHVrLkhhaXJ5Q2VsbE*ldWt1bWlhPg==
      </urn:Document>
    </urn:ProvideAndRegisterDocumentSetRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Figure 2-8 – Example request which should succeed

The request web service call from a External Partner’s application to the VLER Gateway will receive back a return value (acknowledgement) from the VLER Gateway indicating the success or failure of the submission of a request. If the call is successful then the request will be processed at some future time by VLER applications and a response returned, if needed. If the call fails then the request must be resubmitted.

Figure 2-9 is an example successful return value (acknowledgement). The “message” element contains the return value (highlighted in yellow) and will always start with “SUCCESS”.

```
<?xml version="1.0" encoding="utf-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns9:Acknowledgement
      xmlns:ns9="urn:ihe:iti:xdr:2007"
      xmlns:ns8="urn:ihe:iti:xds-b:2007"
      xmlns:ns7="urn:gov:hhs:pha:nhinc:gateway:samltokendata"
      xmlns:ns6="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
      xmlns:ns5="urn:oasis:names:tc:ebxml-regrep:xsd:lcm:3.0"
      xmlns:ns4="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0"
      xmlns:ns3="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0">
      <message>SUCCESS: Request Received</message>
    </ns9:Acknowledgement>
  </S:Body>
</S:Envelope>
```

**Figure 2-9 – Example web service success return value (acknowledgement) for a request**

Figure 2-10 is an example failure return value (acknowledgement). The “message” element contains the return value (highlighted in yellow) and will always start with “FAILURE”. Additional text will indicate the nature of the failure.

```
<?xml version="1.0" encoding="utf-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns9:Acknowledgement
      xmlns:ns9="urn:ihe:iti:xdr:2007"
      xmlns:ns8="urn:ihe:iti:xds-b:2007"
      xmlns:ns7="urn:gov:hhs:pha:nhinc:gateway:samltokendata"
      xmlns:ns6="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
      xmlns:ns5="urn:oasis:names:tc:ebxml-regrep:xsd:lcm:3.0"
      xmlns:ns4="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0"
      xmlns:ns3="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0">
      <message>FAILURE:
gov.va.integration.core.component.ComponentProcessorException:
gov.va.integration.core.router.RouterException:
gov.va.integration.core.validator.ValidatorException: Subject Objects Request/Id
cannot be null</message>
    </ns9:Acknowledgement>
  </S:Body>
</S:Envelope>
```

**Figure 2-10 – Example web service failure return value (acknowledgement) for a request**

### 2.2.3. Response Web Service

A response web service must be implemented by each External Partner's application as shown in Figure 2-4.

The concrete WSDL for the service is shown below in Figure 2-11 and the abstract WSDL in Figure 2-12. The files defining the request web service WSDL are attached in Appendix B – Response WSDL Files

```
<?xml version="1.0" encoding="utf-8"?>
<definitions
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://_2007.response.async.xdr.iti.ihe/"
  name="XDRResponseService"
  targetNamespace="http://_2007.response.async.xdr.iti.ihe/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
  <import
    namespace="urn:ihe:iti:xdr:async:response:2007"
    location="XDRResponseService0.wsdl" />
  <types />
  <binding
    xmlns:ns1="urn:ihe:iti:xdr:async:response:2007"
    name="XDRResponse_PortTypeBinding"
    type="ns1:XDRResponse_PortType">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="ProvideAndRegisterDocumentSet-bResponse">
      <soap:operation
        soapAction="tns:ProvideAndRegisterDocumentSet-bResponse" />
      <input>
        <soap:body use="literal" />
      </input>
      <output>
        <soap:body use="literal" />
      </output>
    </operation>
  </binding>
  <service name="XDRResponseService">
    <port name="XDRResponse_PortType"
      binding="tns:XDRResponse_PortTypeBinding">
      <soap:address
        location="http://SERVER:PORT/gateway/XDRResponseService" />
    </port>
  </service>
</definitions>
```

Figure 2-11 – Response web service concrete WSDL

The value of the **yellow highlighted** “location” attribute of the <soap:address> element in the WSDL of Figure 2-11 must be specified by the External Partner’s application team with:

1. URL scheme:
  - a. “HTTP” when the communications path is not protected by SSL and
  - b. “HTTPS” when the communications path is protected by SSL.
2. The fully qualified External Partner’s SERVER host-name.
3. The PORT selected for communications with External Partner’s application .
  - a. 80 for HTTP
  - b. 443 for HTTPS
4. The fully qualified External Partner’s URL path.

There will be differences between the values for test and production.

The abstract WSDL is shown below in Figure 2-12.

```
<?xml version="1.0" encoding="utf-8"?>
<definitions xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="urn:ihe:iti:xdr:async:response:2007"
  targetNamespace="urn:ihe:iti:xdr:async:response:2007"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
  <types>
    ... types omitted to save space.
  </types>
  <message name="ProvideAndRegisterDocumentSet-bResponse">
    <part xmlns:ns1="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0"
      name="body" element="ns1:RegistryResponse" />
  </message>
  <message name="ProvideAndRegisterDocumentSet-bResponseResponse">
    <part xmlns:ns2="urn:ihe:iti:xdr:2007"
      name="body" element="ns2:Acknowledgement" />
  </message>
  <portType name="XDRResponse_PortType">
    <operation name="ProvideAndRegisterDocumentSet-bResponse">
      <input message="tns:ProvideAndRegisterDocumentSet-bResponse" />
      <output message="tns:ProvideAndRegisterDocumentSet-bResponseResponse" />
    </operation>
  </portType>
</definitions>
```

**Figure 2-12 – Response web service abstract WSDL**

The response document (e.g. Figure 2-13), which may be returned to a External Partner’s application by the VLER Gateway for a request, defines:

1. The original request for which this is the response. A response is the response for a particular request if the value of the “requestID” attribute of the “RegistryResponse” element in the response matches the value of the “id” attribute of the “SubmitObjectsRequest ” element in the associated original request document.
2. The success or failure of the request in the “status” attribute of the “RegistryResponse” element.

Figure 2-13 shows an example response indicating that the request was successfully satisfied by VA.

```
<?xml version="1.0" encoding="utf-8"?><env:Envelope
xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header/>
  <env:Body>
    <ns2:RegistryResponse
      xmlns:ns2="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0"
      xmlns="urn:oasis:names:tc:ebxml-regrep:xsd:rsm:3.0"
      xmlns:ns4="urn:oasis:names:tc:ebxml-regrep:xsd:lcm:3.0"
      xmlns:ns3="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
      requestId="a498fb06-43b0-452e-94d7-1776d42ce71e"
      status="urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success"/>
  </env:Body>
</env:Envelope>
```

**Figure 2-13 - Example “success” response (no returned response data)**

Figure 2-14 shows an example failure response indicating that the request was not successfully satisfied by VA. The “RegistryErrorList” element provides additional information in the response about the error which caused the failure of the request.

```
<?xml version="1.0" encoding="utf-8"?><env:Envelope
xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header/>
  <env:Body>
    <ns2:RegistryResponse
      xmlns:ns2="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0"
      xmlns="urn:oasis:names:tc:ebxml-regrep:xsd:rsm:3.0"
      xmlns:ns4="urn:oasis:names:tc:ebxml-regrep:xsd:lcm:3.0"
      xmlns:ns3="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
      requestId="a498fb06-43b0-452e-94d7-1776d42ce71e"
      status="urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Failure">
      <ns2:RegistryErrorList>
        <ns2:RegistryError
          codeContext="Something went wrong with the application"
          errorCode="XDSRegistryMetadataError"
          severity="urn:oasis:names:tc:ebxml-
            regrep:ErrorSeverityType:Error"/>
      </ns2:RegistryErrorList>
    </ns2:RegistryResponse>
  </env:Body>
</env:Envelope>
```

### Figure 2-14 - Example “failure” response (no returned response data)

The response web service call from the VLER Gateway to an External Partner’s application must receive back a return value (acknowledgement) from that External Partner’s application indicating the success or failure of the receipt of the response by the External Partner’s application .

The processing carried out by an External Partner’s application on receipt of the response from the VLER Gateway is outside the scope of this ICD.

Figure 2-15 is an example successful return value. The “message” element contains the return value and will always start with “SUCCESS”.

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:urn="urn:ihe:iti:xdr:2007">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:Acknowledgement>
      <message>SUCCESS</message>
    </urn:Acknowledgement>
  </soapenv:Body>
</soapenv:Envelope>
```

### Figure 2-15 - Example web service success return value (acknowledgement) for a response

Figure 2-16 is an example failure return value (acknowledgement). The “message” element contains the return value and will always start with “FAILURE”. Additional text will indicate the nature of the failure.

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:urn="urn:ihe:iti:xdr:2007">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:Acknowledgement>
      <!--Optional:-->
      <message>FAILURE: Something went wrong</message>
    </urn:Acknowledgement>
  </soapenv:Body>
</soapenv:Envelope>
```

### Figure 2-16 - Example web service failure return value (acknowledgement) for a response

## 2.3. Operations

Currently this interface supports the operations listed in the appendices as shown in Table 2-3.

**Table 2-3 – Location of web Service operation details**

| <i>Project Short Name</i> | <i>Location</i>                                  |
|---------------------------|--|
| “DBQ”                     | Appendix C – Harris-Integration Specific Details |
| “D2D”                     | Appendix D – D2D-Integration Specific Details    |
| “VES”                     | Appendix E – VES-Integration Specific Details    |
|                           |  |

## 2.4. Security

### 2.4.1. Certificates

Transfers between External Partner’s applications and the VLER Gateway must be protected by two-way SSL within the HTTPS application layer protocol. This protocol requires certificates which identify the host computers running:

1. The External Partner’s applications and
2. The VLER Gateway application.

Certificates must be public CA (Certificate Authority) signed certificates for this interface. External Partners are responsible for obtaining and keeping their certificates up to date based on the lifetime of each certificate.

The certificates for External Partner’s host computers and the VLER Gateway computers must be securely exchanged. See Table 2-1 for contact information. The VLER Gateway will need a copy of each External Partner’s certificate and any other certificates needed to establish a chain of trust to the signing CA. Similarly The each External Partner’s host computer will need a copy of the VLER Gateway certificate and other certificates needed to establish a chain of trust to the signing CA.

### 2.4.2. On Boarding

Each external host computer wishing to interface to the VLER Gateway (test or production environments) may have to have each connection approved by the VA ESCCB (Enterprise Security Change Control Board). This approval process takes 10 days minimum after all needed information is made available to the ESCCB. Contact VLER Operations for information on how to start this procedure. See Table 2-1 for contact information.

Once approval has been obtained, a change must be made to network equipment (an F5 router) to allow incoming connections from the host, identified by its IP address and certificate. Appropriate certificates must be made available at or before this time.

# Appendix A – Request WSDL Files

The WSDL files for the store-document request service implemented by the VLER Gateway are included below in this ICD.

| Embedded File  | Embedded File  | Description              |
|--|--|--------------------------|
| <br>C:\XDR\<br>XDRRequestService\   |  | Concrete-WSDL file.      |
| <br>C:\XDR\<br>XDRRequestService\   |  | Abstract-WSDL file.      |
| <br>C:\XDR\<br>XDRRequestService\   | <br>C:\XDR\<br>XDRRequestService\   | XSD of needed XML types. |
| <br>C:\XDR\<br>XDRRequestService\  | <br>C:\XDR\<br>XDRRequestService\  | XSD of needed XML types. |
| <br>C:\XDR\<br>XDRRequestService\ | <br>C:\XDR\<br>XDRRequestService\ | XSD of needed XML types. |
| <br>C:\XDR\<br>XDRRequestService\ |  | XSD of needed XML types. |
| <br>C:\XDR\<br>XDRRequestService\ |  | XSD of needed XML types. |
| <br>C:\XDR\<br>XDRRequestService\ |  | XSD of needed XML types. |

# Appendix B – Response WSDL Files

The WSDL files for the store-document response service to be implemented by External Partner’s applications are included below in this ICD.

| Embedded File  | Embedded File  | Description              |
|--|--|--------------------------|
| <br>C:\XDR\<br>XDRResponseService   |  | Concrete-WSDL file.      |
| <br>C:\XDR\<br>XDRResponseService   |  | Abstract-WSDL file.      |
| <br>C:\XDR\<br>XDRResponseService   | <br>C:\XDR\<br>XDRResponseService   | XSD of needed XML types. |
| <br>C:\XDR\<br>XDRResponseService  | <br>C:\XDR\<br>XDRResponseService  | XSD of needed XML types. |
| <br>C:\XDR\<br>XDRResponseService | <br>C:\XDR\<br>XDRResponseService | XSD of needed XML types. |
| <br>C:\XDR\<br>XDRResponseService |  | XSD of needed XML types. |
| <br>C:\XDR\<br>XDRResponseService |  | XSD of needed XML types. |
| <br>C:\XDR\<br>XDRResponseService |  | XSD of needed XML types. |

# Appendix C – Harris-Integration Specific Details

## C.1 DBQ Automation Solution

The DBQ Automation Solution provides functionality as specified in the following document:

PERFORMANCE WORK STATEMENT (PWS)

DEPARTMENT OF VETERANS AFFAIRS  
Office of Information & Technology  
Enterprise Program Management Office (EPMO)

Compensation Services Disability Benefits Questionnaire (DBQ) Automation Solution

Date: April 27, 2012  
TAC-12-04156

The DBQ Automation Solution provides a Forms Service external to VA which Clinicians can use to complete Disability-Benefits Questionnaire (DBQ) forms for Veterans. Completed DBQs can then be sent to VA for storage and used in adjudicating Veteran’s claims.

The information listed in Table 0-1 identifies the DBQ Automation Solution.

**Table 0-1 - DBQ Automation Solution Identifiers**

| <i>System</i>         | <i>Details</i>          |
|-----------------------|-------------------------|
| Identification number |                         |
| Title                 | DBQ Automation Solution |
| Abbreviation          |                         |
| Version number        | 1.0                     |
| Release number        | 1.0                     |
| Point of Contact      |                         |

## C.2 Web Services

The inbound web service allows the DBQ Automation Solution to submit Examination Results to VA, one at a time, for later, asynchronous, processing by VLER applications. An Examination Results comprises a DBQ and possibly several attached digital documents supporting the claim.

The processing of each inbound request includes:

1. Adding the Veteran's Internal Control Number (ICN) to the Examination Results, if possible.
2. Storing the updated instance of Examination Results in a database for later validation and adjudication of a Veteran's claim, if possible.
3. Returning an acknowledgement to the DBQ Automation Solution, indicating the success or failure of storing the Examination Results, if possible.

One instance of Examination Results is assumed to contain:

1. One completed Disability Benefits Questionnaire for a Veteran (in XML format)
2. Optionally one or more attachments; other digital examination results for the same Veteran, included by the examining clinician, e.g., photographs.

## C.3 Exchange Protocol

Table 0-2 lists parameters of the request/response transfer protocol. These parameters may change in future versions of this ICD.

**Table 0-2 – Transfer Protocol Parameters**

| Protocol Parameter                | Value   |
|-----------------------------------|---|
| Storage-operation timeout period. | 4 hrs (Should be configurable by DBQ Automation Solution) |
| Maximum request send rate.        | 25 Hz – determined by the HDR                             |
| Maximum size of request           | 25 MB – determined by the HDR                             |

## C.4 Request Web Service

An instance of Examination Results sent (Base64 encoded) in the <urn:Document> element of a request must satisfy the following:

1. Be valid XML conforming to an XSD provided to VA before the first transfer of such XML in a production environment.
2. Have a character encoding of UTF-8.
3. Contain XML elements for:
  - a. Document ID – an Integer uniquely identifying the instance and state of the Examination Results XML document

- b. DBQ Document type – the type of DBQ, from a standardized list of DBQ Document types, contained in the Examination Results XML document.
- c. DBQ status – set to “complete”
- d. Examination Results creation date – in XSD:DateTime format
- e. Facility ID – Physician’s associated facility ID.
- f. Facility Station Number
- g. Facility Name
- h. Patient SSN – without dashes or spaces
- i. Patient First Name
- j. Patient Last Name
- k. Patient Date of Birth – in XSD:DateTime format
- l. Patient Gender – from HL7 gender codes
- m. Patient ID – a unique identifier of the patient assigned by the Assigning Authority - can be blank
- n. Patient ID Assigning-Authority – an identifier of the authority assigning IDs to patients - can be blank
- o. Patient ID Assigning-Facility – can be blank
- p. Physician Id
- q. Physician Id Assigning Facility or Assigning Authority – an identifier of the authority assigning IDs to Physicians
- r. Physician First Name
- s. Physician Middle Initial
- t. Physician Last Name

# Appendix D – D2D-Integration Specific Details

## D.1 D2D Automation Solution

The D2D Automation Solution provides a Forms Service, at VSO offices external to VA, which can be used to complete a variety of VA-forms for Veterans. Completed forms and associated attachments can then be sent to VA for storage and used in adjudicating Veteran's claims.

## D.2 Web Services

The inbound web service, provided by VLER Gateway, allows the D2D Automation Solution to submit:

1. Forms to VA, one at a time, for later asynchronous processing by VA applications. The forms first go into a holding system within VA awaiting the completion of an entire claim submission.
2. Attachments to VA, one at a time, for later asynchronous processing by VA applications when associated to the appropriate form as part of a complete claim submission. The attachments first go into a holding system within VA awaiting the completion of an entire claim submission.
3. Requests to check the status of a claim submission in the holding system, optimally comprising a form and possibly several related attachments.
4. Confirmation of Submission which sends an entire submission from the holding system into the VBMS for claims processing.

One instance of a submission is assumed to contain:

1. One completed VA form (in XML format)
2. Optionally one or more attachments supporting the claim.

The inbound service, implemented in VLER Gateway and used by External Partner's applications to submit requests to VA, is described in

Table 0-1 and an example shown in Figure 0-1.

A single web service operation is provided by VLER Gateway to receive requests to store documents in VA. The different business functions to be requested are distinguished by the value of the **<Slot name="operationName">** element in the request as defined in

Table 0-1 and in the example shown in Figure 0-1.

**Table 0-1 – Inbound Services used by D2D**

| <b>Business Function<br/>(&lt;Slot name="operationName"&gt;<br/>Value)</b> | <b>Effect</b>  |
|--|--|
| VSO.submitForm   | Request to store a form in the holding system.                     |
| VSO.submitAttachmentForm   | Request to store an attachment in the holding system.              |
| VSO.checkStatus  | Request to check the status of a submission in the holding system. |
| VSO.confirmSubmission  | Request to send a submission forward for claims processing.        |

The value of the <Slot name="originatingOrganizationName"> element should be constant for a given VSO organization making requests to store forms and attachments. This value will be used by VLER Gateway to identify the host to which the response for such a given request will be returned. Only VSO organizations approved by the D2D project will be permitted to send requests to VLER Gateway. VLER gateway will reject request from unknown VSO organizations.

Figure 0-1 shows an example request envelope document completed according to the guidelines for elements listed in Table 2-2. The Base64 encoded instance of a request for service (highlighted in green) has been truncated to fit the space available on this page. Values for the other variable have been highlighted in yellow.

```

<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:urn="urn:ihe:iti:xds-b:2007"
  xmlns:urn1="urn:oasis:names:tc:ebxml-regrep:xsd:lcm:3.0"
  xmlns:urn2="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0"
  xmlns:urn3="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0">
  <env:Header/>
  <env:Body>
    <urn:ProvideAndRegisterDocumentSetRequest>
      <urn1:SubmitObjectsRequest
        id="a498fb06-43b0-452e-94d7-1776d42ce71e">
        <urn2:RequestSlotList>
          <urn3:Slot name="operationName" >
            <urn3:ValueList>
              <urn3:Value>VSO.submitForm</urn3:Value>
            </urn3:ValueList>
          </urn3:Slot>
          <urn3:Slot name="originatingOrganizationName" >
            <urn3:ValueList>
              <urn3:Value>VSO Blaine, WA</urn3:Value>
            </urn3:ValueList>
          </urn3:Slot>
          <urn3:Slot name="originatingApplicationName" >
            <urn3:ValueList>
              <urn3:Value>ACME Forms Service</urn3:Value>
            </urn3:ValueList>
          </urn3:Slot>
        </urn2:RequestSlotList>
        <urn3:RegistryObjectList/>
      </urn1:SubmitObjectsRequest>
      <urn:Document id="0001">
        PGdvdI52YS5hZ2VudG9yYW5nZS5mYXN0dHJhY2subW9kZWwuZGJxLmxiay5IYWlyeUNlbGxMZXRr
        .. : : :
        aWNhdGlvbnM+PC9hbnRpZGhlcjNvbXBsaWNhdGlvbnM+DQoJCTwvZ292LnZhLmFnZW50b3Jhbmdl
        LmZhc3R0cmFjay5tb2R1bC5kYnEubHVrLkhhaXJ5Q2VsbExldWt1bWlhPg==
      </urn:Document>
    </urn:ProvideAndRegisterDocumentSetRequest>
  </env:Body>
</env:Envelope>

```

**Figure 0-1 – Example D2D request which should succeed**

The outbound service, implemented in External Partner’s applications and used by VLER Gateway to return responses, is described in Table 0-2 and an example shown in Figure 0-2.

A single web service operation must be provided by External Partner’s applications to receive the responses to earlier requests to store documents in VA. The responses for the different business functions, specified in the original request, are distinguished by the value of the **<Slot name=“operationName”>** element in the response as defined in Table 0-2 and in the example shown in Figure 0-2. If:

1. The original request specified **<Slot name=“operationName”>** value of “VSO.submitForm”

- Then in the response the **<Slot name="operationName">** will be "VSO.submitForm-response".

**Table 0-2 – Outbound Services implemented by External Partner’s Applications**

| <b>Business Function<br/>(&lt;Slot name="operationName"&gt;<br/>Value)</b> | <b>Effect</b>  |
|--|--|
| VSO.submitForm-response  | Returns the response to a request to store a form in the holding system.                     |
| VSO.submitAttachmentForm-response  | Returns the response to a request to store an attachment in the holding system.              |
| VSO.checkStatus-response   | Returns the response to a request to check the status of a submission in the holding system. |
| VSO.confirmSubmission-response   | Returns the response to a request to send a submission forward for claims processing.        |

The value of the **<Slot name="originatingOrganizationName">** element in the request will be used by VLER Gateway to identify the host to which the response to the request will be returned. The **<Slot name="originatingOrganizationName">** in the response will be set to match that of the same field in the original request.

The value of the **<Slot name="responseDocument">** element in the response will be used by VLER Gateway to return response-data to the requesting VSO. Table 0-4 lists those business-functions which are expected to normally return data.

```

<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header/>
  <env:Body>
    <ns2:RegistryResponse
      xmlns="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"
      xmlns:ns2="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0"
      requestId="a498fb06-43b0-452e-94d7-1776d42ce71e"
      status="urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success">
      <ns2:ResponseSlotList>
        <Slot name="operationName">
          <ValueList>
            <Value>VSO.submitForm</Value>
          </ValueList>
        </Slot>
        <Slot name="originatingOrganizationName">
          <ValueList>
            <Value>VSO Blaine, WA</Value>
          </ValueList>
        </Slot>
        <Slot name="originatingApplicationName">
          <ValueList>
            <Value>ACME Forms Service</Value>
          </ValueList>
        </Slot>
        <Slot name="responseDocument">
          <ValueList>
            <Value>PGdvdi52YS5hZ2VudG9yYW5nZS5mYX .. ..
              .. .. ..
              LmZhc3R0cmdWtlbWlhPg==
            </Value>
          </ValueList>
        </Slot>
      </ns2:ResponseSlotList>
    </ns2:RegistryResponse>
  </env:Body>
</env:Envelope>

```

Figure 0-2 – Example D2D success response containing a response-data-document

## D.3 Exchange Protocol

Table 0-3 lists parameters of the request/response transfer protocol. These parameters may change in future versions of this ICD.

**Table 0-3 – D2DTransfer Protocol Parameters**

| Protocol Parameter                | Value |
|-----------------------------------|-------|
| Storage-operation timeout period. | TBD   |
| Maximum request send rate.        | TBD   |
| Maximum size of request           | 10 MB |

Table 0-4 lists additional parameters of the request/response transfer protocol. The estimated sizes of the request and response types of messages are shown to an order of magnitude. Not every type of incoming request will get a response. The last column indicates whether a response is normally expected or not.

To ensure forms are processed ahead of attachments (which can be large) a priority is applied to processing D2D requests. Requests to save forms will be processed with a higher priority than any of the three other request types. All responses will be treated with equal priority.

**Table 0-4 – D2DTransfer Protocol Parameters (cont)**

| Business Function  | Request Est Rate | Request Est size | Priority | Response Est size | Response required   |
|--------------------|------------------|------------------|----------|-------------------|---|
| Save Form          | TBD              | kB               | High     | B                 | On error only.  |
| Save Attachment    | TBD              | MB               | Low      | B                 | On error only.  |
| Check Status       | TBD              | B                | High     | kB                | Yes.<br>Contains a Base64 encoded response-document.                                      |
| Confirm Submission | TBD              | B                | High     | kB                | Yes, possibly after a long response time.<br>Contains a Base64 encoded response-document. |

# Appendix E – VES-Integration Specific Details

## E.1 VDBQ Portal

VDBQ Portal (an External Partner) provides a forms service external to VA, which can be used to Examination Results for Veterans. Completed Examination Results and associated attachments can then be sent to VA for storage and used in adjudicating Veteran's claims.

## E.2 Web Services

This appendix describes the interface to be used for storing an exam result (aka DBQ) from Veterans Evaluation Services (VES – [www.vesservices.com](http://www.vesservices.com)). VES has a long history of working with VHA and VBA, generating DBQs in PDF form and sending them into VA. This story describes the evolution of VES capabilities to send DBQs to VA in XML format (similar to the system recently implemented wherein the Harris Forms System sends DBQs to VA over the Internet for storage in CDS/HDR).

This same interface can also be used to send PDF files (or other artifacts) to VA by appropriately specifying the value of the XML `<urn3:Slot name="operationName" >` element in the Soap request.

VES DBQs are sent into VA from the VES DBQ Portal (VDBQ Portal). The interface between VES and VA is an asynchronous, web services interface provided by the Virtual Lifetime Electronic Record (VLER) Gateway application. The interface accepts requests to store DBQ XML documents into the VLER SOR database and, later, returns a response indicating the success or failure of the store operation to the VDBQ Portal. The response also indicates whether the store operation detected an attempt to store a duplicate of a DBQ already stored.

The inbound web service allows VDBQ Portal to submit Examination Results to VA, one at a time, for later, asynchronous, processing by VLER applications.

The processing of each inbound request from VDBQ Portal includes:

1. Adding the Veteran's Internal Control Number (ICN) to the Examination Results, if possible.
2. Storing the updated instance of Examination Results in a database for later validation and adjudication of a Veteran's claim, if possible.
3. Returning an acknowledgement to VDBQ Portal, indicating the success or failure of storing the Examination Results, if possible.

One instance of Examination Results is assumed to contain:

1. One completed Disability Benefits Questionnaire for a Veteran (in XML format)
2. Optionally one or more attachments; other digital examination results for the same Veteran, included by the examining clinician, e.g., photographs.

The inbound web service is used by VDBQ Portal to submit requests to store Examination Results in VA. It is described in

Table 0-1 and an example request is shown in Figure 0-1. A single web service operation is provided by VLER Gateway to receive requests to store documents in VA. The different business functions which can be requested are distinguished by the value of the **<Slot name="operationName">** element in the request as defined in

Table 0-1 and in the example shown in Figure 0-1.

**Table 0-1 – Inbound Services used by VDBQ Portal**

| <b>Business Function</b><br>( <b>&lt;Slot name="operationName"&gt;</b><br><b>Value)</b> | <b>Effect</b>  | <b>Contents of &lt;Document&gt;</b><br><b>Element</b> |
|---|--|---|
| StoreExaminationResults   | Request to store one Examination Results into the VLER SOR database.               | Base64 encoded XML DBQ                                |
| StoreExaminationResultsPDF  | Request to store one Examination Results in PDF format into the VLER SOR database. | Base64 encoded PDF document                           |
|   |  |   |

The value of the **<Slot name="originatingOrganizationName">** element in the request should be constant for all VDBQ Portal requests. This value is used by VLER Gateway to identify the host to which the response for such a given request will be returned. VLER Gateway will reject requests from unknown organizations. The value is shown below:

```

<urn3:Slot name="originatingOrganizationName" >
  <urn3:ValueList>
    <urn3:Value>Veterans Evaluation Services</urn3:Value>
  </urn3:ValueList>
</urn3:Slot>

```

The value of the **<Slot name="originatingApplicationName">** element in the request should also be constant for all VDBQ Portal requests. The value is shown below:

```

<urn3:Slot name="originatingApplicationName" >
  <urn3:ValueList>
    <urn3:Value>VDBQ Portal</urn3:Value>
  </urn3:ValueList>
</urn3:Slot>

```

Figure 0-1 shows an example request envelope document completed according to the guidelines for elements listed in Table 2-2. The Base64 encoded instance of one Examination Results (highlighted in green) has been truncated to fit the space available on this page. Values for the other variable have been highlighted in yellow. An instance of Examination Results sent (Base64 encoded) in the **<urn:Document>** element of a request must satisfy the following:

1. Be valid XML conforming to an XSD provided by VA before the first transfer of such XML in a production environment.



One outbound service must be implemented by VDBQ Portal. It is used by VLER Gateway to return responses to a request for service. It is described in Table 0-2 and an example “success” response shown in Figure 0-2. An example “failure” response shown in Figure 0-3.

**A single web service operation must be provided by VDBQ Portal to receive responses from VLER Gateway to earlier requests to store Examination Results in VA.**

The response business function is identified by the value of the <Slot name=“operationName”> element in the response as defined in Table 0-2 and in the example shown in Figure 0-2.

**Table 0-2 – Outbound Services implemented by VDBQ Portal**

| <b>Business Function</b><br>(<Slot name=“operationName”><br>Value) | <b>Effect</b>  |
|--|--|
| StoreExaminationResults-response                                   | Returns the response to an earlier request to store one Examination Results in XML format. |
| StoreExaminationResultsPDF-response                                | Returns the response to an earlier request to store one Examination Results in PDF format. |
|  |  |

The value of the <Slot name="originatingOrganizationName"> element in the request will be used by VLER Gateway to identify the host to which the response to the request will be returned. The <Slot name="originatingOrganizationName"> in the response will be set to match that of the same field in the original request.

The <Slot name="originatingApplicationName"> in the response will be set to match that of the same field in the original request.

VLER Gateway maintains a table of endpoints to which responses may be sent. Table 0-3 shows the entries in the production and test versions of that table for the VDBQ Portal.

**Table 0-3 – Outbound Services VDBQ Portal Endpoint**

| <b>Originating Organization Name</b> | <b>Environment</b> | <b>URL</b>              |
|--------------------------------------|--------------------|-------------------------|
| Veterans Evaluation Services         | Production         | ICG.VESSERVICES.COM     |
| Veterans Evaluation Services         | Test               | TESTICG.VESSERVICES.COM |

```

<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header/>
  <env:Body>
    <ns2:RegistryResponse
      xmlns="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"
      xmlns:ns2="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0"
      requestId="a498fb06-43b0-452e-94d7-1776d42ce71e"
      status="urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success">
      <ns2:ResponseSlotList>
        <Slot name="operationName">
          <ValueList>
            <Value>StoreExaminationResults-response</Value>
          </ValueList>
        </Slot>
        <Slot name="originatingOrganizationName">
          <ValueList>
            <Value>Veterans Evaluation Services</Value>
          </ValueList>
        </Slot>
        <Slot name="originatingApplicationName">
          <ValueList>
            <Value>VDBQ Portal</Value>
          </ValueList>
        </Slot>
      </ns2:ResponseSlotList>
    </ns2:RegistryResponse>
  </env:Body>
</env:Envelope>

```

**Figure 0-2 – Example VES “success” response**

```

<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header/>
  <env:Body>
    <ns2:RegistryResponse
      xmlns="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"
      xmlns:ns2="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0"
      requestId="a498fb06-43b0-452e-94d7-1776d42ce71e"
      status="urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success">
      <ns2:RegistryErrorList>
        <ns2:RegistryError
          codeContext="Something went wrong with the application"
          errorCode="XDSRegistryMetadataError"
          severity="urn:oasis:names:tc:ebxml-
            regrep:ErrorSeverityType:Error"/>
      </ns2:RegistryErrorList>
      <ns2:ResponseSlotList>
        <Slot name="operationName">
          <ValueList>
            <Value>StoreExaminationResults-response</Value>
          </ValueList>
        </Slot>
        <Slot name="originatingOrganizationName">
          <ValueList>
            <Value>Veterans Evaluation Services</Value>
          </ValueList>
        </Slot>
        <Slot name="originatingApplicationName">
          <ValueList>
            <Value>VDBQ Portal</Value>
          </ValueList>
        </Slot>
      </ns2:ResponseSlotList>
    </ns2:RegistryResponse>
  </env:Body>
</env:Envelope>

```

Figure 0-3 – Example VES “failure” response

## E.3 Exchange Protocol

Table 0-3 lists parameters of the request/response transfer protocol. These parameters may change in future versions of this ICD.

**Table 0-4 – VDBQ Portal Transfer Protocol Parameters**

| <b>Protocol Parameter</b>         | <b>Value</b> |
|-----------------------------------|--------------|
| Storage-operation timeout period. | TBD          |
| Maximum request send rate.        | TBD          |
| Maximum size of request           | 10 MB        |