

**Virtual Lifetime Electronic Record (VLER)
Program Management Office (PMO)**

Version 1.022

**VLER Data Access Service (DAS)
Interface Control Document (ICD)
with Consumer and Producer Services**



January 2013

Revision History

Date	Version	Description	Author
4/28/2011	0.01	First Draft	David Alvey
4/29/2011	0.02	Second Draft	David Alvey, Tony Mallia
5/9/2011	0.3	Technical Writer Review	John Heeder
5/12/2011	0.4	Draft 4	David Alvey, Tony Mallia
5/26/2011	0.5	Draft 5 – with feedback from VLER architects, projects, and ISI projects	David Alvey
7/8/2011	1.0 - Candidate	Candidate v1.0 for initial baseline approval. Added content in TBD sections (e.g. caching, communications methods, performance requirements). Deleted hData Document Meta Data concept. Added extensions to Atom:Entry including Preview and a few hData Document Meta Data concepts that relate to security and privacy.	David Alvey Tony Mallia
10/15/11	1.0 –	Approved v1.0 Baseline. <ul style="list-style-type: none"> • Added some clarification text based on review request that did not alter the interface • Added ISI Prototype collections to Collection Hierarchy • Clarified Preview feature within Entries • Clarified links usages within Entries 	David Alvey, Tony Mallia
12/6/2011	1.0	Technical Writer Review	John Heeder
12/17/11	1.01	Minor, but technically important errors changed <ul style="list-style-type: none"> • Appendix A: CareCoordinatorProfiles was made plural • Appendix B: Items within Generator definition changed from subelements to attributes 	David Alvey
3/1/12	1.011	Sec 2.3: Corrected error in DoD OID – one digit was missing. App A: Updated collection hierarchy structure.	David Alvey John Heeder

3/15/12	1.02	Baseline document	John Heeder
01/29/2013	1.021	Add new Table 1 showing ICD versions supported by versions of VRS.	George Ludgate
01/29/2013	1.022	Tech Writer Review	Nancy Burak

DO NOT COPY

Table of Contents

1. Introduction	1
1.1. Purpose.....	1
1.2. Scope	1
1.3. Service Identification	2
1.3.1. VLER DAS	2
1.3.2. Consumer Services.....	2
1.3.3. Producer Services.....	2
1.4. Operational Agreement	2
1.5. Standards and References	3
2. Interface Definition	4
2.1. System Overview	4
2.2. Interface Overview.....	5
2.3. Operations	6
2.3.1. HTTP Operations on the <i>baseURL</i>	7
2.3.2. HTTP Operations on the <i>baseURL/aa/{pid}/root.xml</i>	8
2.3.3. HTTP Operations on the <i>baseURL/aa/{pid}/</i>	8
2.3.4. HTTP Operations on the <i>baseURL/aa/{pid}/collectionpath</i>	8
2.3.5. HTTP Operations on the <i>baseURL/aa/{pid}/collectionpath/section.xml</i> 8	8
GET with Query Parameters.....	8
2.3.6. HTTP Operations on the <i>baseURL/aa/{pid}/collectionpath/documentname</i>	9
2.4. Data Transfer	9
2.4.1. HTTP Headers	10
2.4.2. Caching.....	10
2.5. Transaction Types	10
2.6. Data Exchanges	12
2.7. Precedence and Criticality	12
2.8. Communications Methods	13
2.9. Performance Requirements	14
2.10. Security and Privacy	15
3. Interface Requirements	16

3.1. Consumer Services – VLER DAS Interface.....	16
3.1.1. Interface Processing Time Requirements	16
3.1.2. Message/File Requirements	16
3.1.3. Communication Methods.....	16
3.1.4. Security Requirements.....	16
3.1.5. Physical Requirements	16
3.1.6. Structural Requirements	16
3.1.7. Mechanical Requirements	16
3.1.8. Electrical Requirements.....	17
3.1.9. Thermal Requirements	17
3.1.10. Special Fluid Requirements.....	17
3.1.11. Signal Requirements	17
3.2. VLER DAS – Producer Services Interface.....	17
3.2.1. Interface Processing Time Requirements	17
3.2.2. Message/File Requirements	17
3.2.3. Communication Methods.....	17
3.2.4. Security Requirements.....	17
3.2.5. Physical Requirements	17
3.2.6. Structural Requirements	17
3.2.7. Mechanical Requirements	17
3.2.8. Electrical Requirements.....	18
3.2.9. Thermal Requirements.....	18
3.2.10. Special Fluid Requirements.....	18
3.2.11. Signal Requirements	18
4. Interface Verification.....	19
5. Appendix A – VLER vData Collection Hierarchy.....	20
6. Appendix B – Schemas and Data Elements.....	21
6.1. Feed Document.....	21
6.2. Atom: Entry Element.....	23
6.3. Root.xml.....	31
6.4. vData XML Schemas	32

1. Introduction

This document is the Interface Control Document (ICD) for interfaces between Virtual Lifetime Electronic Record (VLER) Data Access Service (DAS) and consumer and producer systems, services, or applications. Throughout the rest of this ICD, “services” will be used in referring to consumers and producers whether they are systems, services, or applications. The ICD defines two related interfaces: one for consumer services to access any of the information components of a VLER and the second for the VLER DAS to in turn access VLER-related information from producer services.

Throughout this ICD, VLER is the tangible collection of documents and records associated with any given service member or veteran. VLER is also an initiative and anytime VLER the initiative is referred to in this ICD the phrase “VLER initiative” is used. This spec also includes a description of how information in a VLER is organized.

This approach draws extensively on both the Atom 1.0 web feed standards and the hData concepts created by Gerald Beuchelt, et al of MITRE Corporation. References for these are cited in the Standards and References section.

1.1. Purpose

The ICD serves as a specification for developers of the VLER DAS as well as consumer and producer services which interact with the VLER DAS.

1.2. Scope

The ICD focuses on information flows at VLER DAS service ports. In DoDAF v2 as well as UML v2, a port is a distinct interaction point through which a service interacts with its environment. This isolates dependencies between services to particular interaction points rather than to the service as a whole. The interfaces provided over a port are denoted directly at the port. As many interfaces as one likes can be defined at a port.

It describes the operations, data transfers, transaction types, data exchanges, communication methods, and performance requirements of consumer and producer service interfaces supported by the VLER DAS. This document does not specify the internal design of the VLER DAS nor services that interact with the VLER DAS.

Table 1 shows the version of the VLER Read Service (VRS) deployed in each release of VLER DAS. In addition it shows the major version of this ICD supported by each released VRS.

Table 1: ICD versions used in VLER DAS Releases

VLER DAS Release #	VLER Read Service version #	Major ICD version #
1.0	1	1
2.0	2	1
3.0	2	1

1.3. Service Identification

The services that this ICD applies to are:

- VLER DAS
- Consumer Services
- Producer Services

1.3.1. VLER DAS

The VLER DAS serves as a proxy between consumers and producers of VLER data. Consumers initiate all data transactions. In responding to consumers, the VLER DAS aggregates data from multiple producers.

The VLER DAS interacts with Identity Management and Veteran Authorization and Policy (VAP) to ensure that all such data transactions maintain the integrity of personal data associated with individuals and enforce all related security and privacy policies. The VLER DAS interfaces with Identity Management and VAP are not in within the scope of this ICD since this ICD focuses on the Consumer and Producer interfaces. Any references to Identity Management and VAP are to illustrate identity management and access control context within transactions that span Consumers, VLER DAS, and Producers.

Table 2 provides identifier specifics for VLER DAS.

Table 2: VLER DAS Identifiers

System	Details
Identification number	S19 VLER Data Access Service
Title	Virtual Lifetime Electronic Record Data Access Service
Abbreviation	VLER DAS
Version number	1.0
Release number	
Point of Contact	VLER PMO
Vendor [optional]	

1.3.2. Consumer Services

Consumers are services that have a need to engage with producer services to obtain VLER data. VLER DAS acts as a proxy to the producer services providing a common port for consumers.

1.3.3. Producer Services

Producer services are any system, application, or service that offers VLER data persistence in support of consumer services. VLER DAS acts as a proxy between such servers and consumer services thereby providing a consistent interface to consumer services.

1.4. Operational Agreement

This ICD is maintained by the VLER PMO. Requested changes to the interfaces described in this document, or to just this document itself, are to be submitted to the VLER PMO for consideration. Changes that materially affect the implementation of systems, applications, or services maintained by separate VA or DoD program offices will be escalated to the appropriate governing board.

All consumer and producer services interacting with the VLER DAS are required to either use and/or support these ports.

1.5. Standards and References

This ICD makes use of the following standards and approaches:

- a. The Atom Syndication Format standard, RFC 4287: <http://tools.ietf.org/html/rfc4287>
- b. The Atom Publishing Protocol, RFC 5023: <http://tools.ietf.org/html/rfc5023>
- c. Hypertext Transfer Protocol – HTTP/1.1, RFC 2616: <http://tools.ietf.org/html/rfc2616>
- d. hData, Gerald Beuchelt, et al, MITRE Corporation: www.projecthdata.org,
- e. DoDAF v2 Volume 1: <http://cio-nii.defense.gov/docs/DoDAF%20V2%20-%20Volume%201.pdf>
- f. *RESTful Web Services*, Leonard Richardson & Sam Ruby, 2007

2. Interface Definition

This ICD specifies the interface(s) between respective services. The interface is therefore the connection between the two services and the ports are their respective connection points. The described interfaces are therefore between two ports.

This section provides an overview of the context in which these services interact, their respective ports, and the introduction of interface requirements that are common to both consumer and producer services.

2.1. System Overview

The VLER DAS serves as a proxy between consumers and producers of VLER data. Consumers initiate all data requests. The VLER DAS interacts with Identity Management and VAP to ensure that all such data transactions maintain the integrity of personal data associated with individuals and enforce all related security and privacy policies. Figure 1 introduces the VLER DAS interface relationships of the services described in this ICD. The interfaces with Identity Management and VAP are described their respective ICDs.

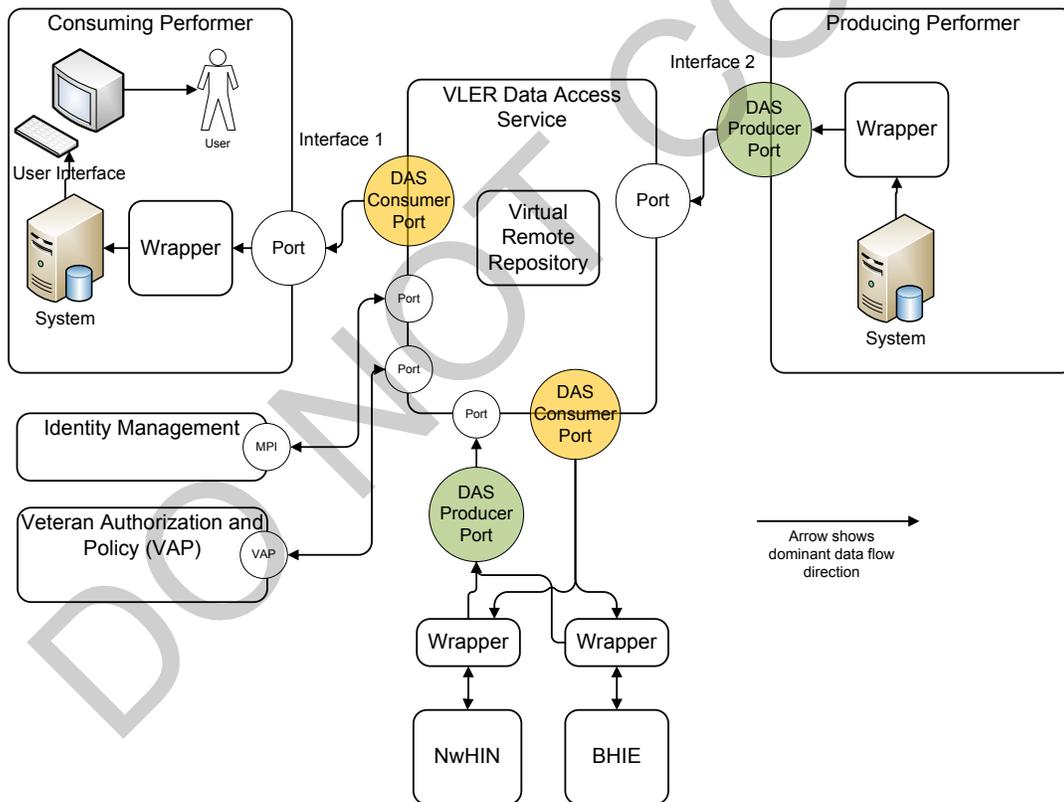


Figure 1 - VLER DAS Context Diagram

VLER data for any individual is made available via a standard set of hierarchical HTTP resources (see Appendix A). The entire VLER dataset for joint VA-DoD interoperability is contained under a base URL. The base URL will be denoted as *baseURL* throughout this document. This allows VLER DAS to provide

interactions with VLER using RESTful APIs and a set of services common to web feeds around the world.

All other interfaces with VLER DAS are supporting interfaces to the VLER DAS's primary role as serving as a VLER proxy between consumer and producer services. An interface with Identity Management ensures that the VLER DAS is handling data that is associated with a given individual. An interface with VAP ensures that proper security and privacy policies that may apply to a transaction are being appropriately applied.

2.2. Interface Overview

The VLER DAS consumer and producer ports with are based on Atom and RESTful principles. In serving a consumer request the VLER DAS acts as web server and the consumer as web client. Similarly, when interacting with a producer service, the VLER DAS acts as a web client and the provider service as a web server.

Though at present, the VLER DAS only supports GET requests at the consumer and producer ports, the following description uses PUTs and POSTs to convey the quality of RESTful statelessness. A RESTful service is stateless if the web server never stores any application state¹. In a stateless application, the web server considers each web client request in isolation and in terms of the current resource state². If the web client wants any application state to be taken into consideration, the web client must submit it as part of the request. The web client manipulates resource state within web servers by sending a representation of the resource as part of the PUT or POST request. The web server manipulates client state by sending resource representations in response to the client's GET requests.³

Consumers, acting as web clients, initiate transactions on VLER data entities or collections of VLER data entities. VLER collections are Atom collections and can be thought of as folders that contain an Atom feed document that describes the collection, data/document records (hereafter referred to as subject documents), and/ or other collections. The Atom feed document contains metadata about the collection and an entry for each item in the collection. An entry may wholly contain the item if the content is text, html, xhtml, or XML. An entry can also reference a subject document that is a resource separate from the feed doc. Each collection has one atom feed document and every VLER atom feed document uses the filename, section.xml.

Subject documents may be text, pdf, or XML. Those that are XML may have been generated in compliance with XML Schema Documents (XSD). The entire expression of VLER (the collection structure, the Atom feed documents, the XSDs, and the subject documents) will be referred to as vData. vData XML Schemas serve to guide the production of VLER subject documents.

The requests made by the consumers are always in reference to a VLER web resource without having to offer any producer specifics. The VLER DAS, in turn, accesses web feeds established by the producers. The VLER DAS, having knowledge of how the data from producer services map into the HTTP representation of VLER data, interacts appropriately with any number of producers needed to satisfy the incoming request from the consumer. Content-based resources (i.e. data records, documents, etc) are to

¹ Application state is information about the state of a use case path being taken within a client.

² Resource state is information about the resources. Resources are anything that can be referenced with a URI such as a data record.

³ RESTful Web Services, Leonard Richardson & Sam Ruby, 2007

comply with vData XML Schemas to promote interoperability. The VLER DAS acts as an aggregator of producer data and responds to the original consumer request with an aggregated response.

2.3. Operations

VLER DAS supports HTTP operations on:

- *baseURL*
- *baseURL/aa/{PID}*
- *baseURL/aa/{PID}/service.xml*
- *baseURL/ aa/{PID}/collectionpath*
- *baseURL/ aa/{PID}/collectionpath/section.xml*
- *baseURL/ aa/{PID}/collectionpath/documentname.*

The *baseURL* represents the entire VLER collection.

baseURL/aa/{pid} is the pathname to a particular individual with a VLER where *aa* represents the Assigning Authority OID of the *pid* (Person identifier). The VA's VLER DAS internally uses VA Identity Control Numbers (ICNs) as the primary *pid*; however, consumers and producers may use either ICNs or the DoD EDI-PIs. Consumers may make requests of the VLER DAS using either ICNs or EDI-PIs. VLER DAS will make requests of producers using the producers preferred *pid*, be it ICNs or EDI-PIs.

The assigning authority representations for the VA and DoD are the following OIDs:

- VA Assigning Authority: 2.16.840.1.113883.4.349
- DoD Assigning Authority: 2.16.840.1.113883.3.42.10001.100001.12

The root.xml document [see section 6.3] defines the collections, document types, and content profile templates that comprise the VLER collection for any person.

collectionpath is the part of a specific URL to collections and subcollections that comprise VLER. For example, *aa/{pid}/health/clinicalDocuments/conditions/* may contain all the problem list records (conditions) for an individual referred to by *aa/{pid}*. The hierarchy of collections is defined in Appendix A – VLER vData Collection Hierarchy.

The *section.xml* document serves as the Atom feed document for a collection. The feed document contains the Atom entries.

documentname is a part of a URL for a subject document contained within a collection. For example, *123abc.xml* may be a specific condition (i.e. problem list) record for an individual. Subject documents are stored within the appropriate collection. Subject documents may be structured or unstructured. When structured, they are compliant with a defined XML Schema. Document names include its MIME suffix (e.g. .xml, .txt, .pdf).

Table 3 provides a summary of the supported operations described in greater detail in sections 2.3.1-2.3.4. All consumer services must have the proper authority for any particular operation or else the operation will be denied.

Table 3 - Summary of Operations

	GET	POST	PUT	DELETE
<i>baseURL</i>	Returns standard HTTP representation of resource contents (individuals with a VLER)	Future capability	Undefined	Undefined
<i>root.xml</i>	Returns a representation of the root.xml doc	Illegal operation	Illegal operation	Illegal operation
<i>baseURL/aa/{pid}</i>	Returns a standard HTTP representation of the resource contents for an individual. (i.e. the highest level collection feeds)	Future capability	Undefined	Undefined
<i>collectionpath</i>	Returns a set of URI references of the resources contained within the collection path	Future capability	Undefined	Future capability
<i>Section.xml</i>	Returns a representation of the collection feed document	Future capability	Undefined	Undefined
<i>documentname</i>	Returns doc representation	Future capability	Future capability	Future capability

Note: The HTTP operations POST, GET, PUT, and DELETE are analogous to Create, Read, Update, and Delete, respectively.

2.3.1. HTTP Operations on the *baseURL*

The *baseURL* represents the entire VLER collection.

The *baseURL* is <https://vler.va.gov/das/v1/>

Note: this *baseURL* is notional until the first implementation of VLER DAS.

GET

If there are no vData structures at the *baseURL*, the server should return a 404 – Not found status code.

If there is a vData structure at the baseURL, the server will respond with a standard HTTP representation of the resource contents. At the baseURL level, the contents are URI representations of all individuals with vData representations. (ie. all the *aa/{pid}* representations contained within the *baseURL*.)

HTTP Status Code Responses: 200, 404.

2.3.2. HTTP Operations on the *baseURL/aa/{pid}/root.xml*

The root.xml document defines the collections, document types, and content profile templates that comprise the VLER set of collections for any person. It is an hData representation of an Atom service document and the elements of the document are defined in Section 6.3, Root.xml.

GET

This operation returns the current root document, as defined by this specification. This XML document can be used to discover how the VLER is structured and where vData XML Schemas are stored.

HTTP Status Code Responses: 200, 404

2.3.3. HTTP Operations on the *baseURL/aa/{pid}/*

baseURL/aa/{pid} is the pathname to a particular individual with a VLER where *aa* represents the Assigning Authority OID of the *pid* (Person identifier). Consumers may make requests of the VLER DAS using either VA ICNs or DoD EDI-PIs. The VLER DAS shall internally maintain only one set of feeds per individual, not separate feeds for ICNs and another set for EDI-PIs. Producers may use either ICNs or EDI-PIs as their *baseURL/aa/{pid}* and the VLER DAS must be able to aggregate such producers into a single set of VLER DAS feeds for that individual.

GET

The server will respond with a standard HTTP representation of the resource contents.

HTTP Status Code Responses: 200, 404

2.3.4. HTTP Operations on the *baseURL/aa/{pid}/collectionpath*

collectionpath is the specific URL pathnames to collections and subcollections that comprise VLER. The section.xml document serves as the Atom feed document for a collection. The feed document elements are defined in Section 6.1, Feed Document.

GET

The server will respond with a representation of all child resources (by URI) contained within the *collectionpath*. It only returns the next tier of resources, not resources beyond the next level.

HTTP Status Code Responses: 200, 404

2.3.5. HTTP Operations on the *baseURL/aa/{pid}/collectionpath/section.xml*

The section.xml document serves as the Atom feed document for a collection. The feed document elements are defined in Section 6.1, Feed Document.

GET

The server will respond with a representation of the entire section.xml document.

HTTP Status Code Responses: 200, 404

GET with Query Parameters

The standard GET on a *baseURL/aa/{PID}/collectionpath* may be appended with a query string of form below.

baseURL/aa/{PID}/collectionpath?{query filters}

Query filers are TBD.

The server will respond with a representation of the section.xml document with only entries that comply with the query parameters.

HTTP Status Code Responses: 200, 404

2.3.6. HTTP Operations on the *baseURL/aa/{pid}/collectionpath/documentname*

documentname a specific subject document resource contained with a collection. For example, 123abc.xml may be a specific problem list record for an individual.

GET

This operation returns a representation of the document that is identified by *documentname* within the collection identified by *collectionpath*. The *documentname* is typically assigned by the underlying system and is not guaranteed to be identical across two different systems. Implementations may use identifiers contained within the metadata of the document, as *documentnames*.

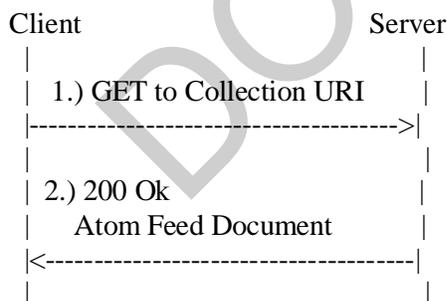
If no document of name *documentname* exists, the server must return a HTTP status code 404.

HTTP Status Code Responses: 200, 404

2.4. Data Transfer

Data transfers are accomplished using standard HTTP protocols. The figure below illustrates the data transfer associated with a client requesting an atom feed document for a collection (i.e. a section.xml document).

To list the Members of a Collection, the client sends a GET request to the URI of a Collection. An Atom Feed Document is returned whose Entries contain the URIs of Member Resources (i.e. subject documents). The returned Feed may describe all, or only a partial list, of the Members in a Collection.



1. The client sends a GET request to the URI of the Collection.
2. If present, the server responds with an Atom Feed Document containing the IRIs of the Collection Members.
3. If the collection is not present (in error), the server responds with HTTP status 404.

2.4.1. HTTP Headers

HTTP version 1.1 Header Constructs must be used. The following header elements are required in making and responding to VLER VDAS HTTP operations.

- Date
- Server
- Cache-Control
- Expires
- Last-Modified
- ETag
- Content-Length
- Content-Type

2.4.2. Caching

The default maximum age for cached items within VLER DAS before they are considered stale will be 6 months. A Producer who directs otherwise using the Cache-Control HTTP header field shall override the VLER DAS default settings.

2.5. Transaction Types

Within the context of a VA ICD, a transaction is a sequence of interface operations that result in something of value to the initiator of the transaction.

All VLER DAS transactions are initiated by the consumer service using any of the standard variations of the HTTP GET command, such as Conditional GETs. Figure 2 illustrates a transaction whereby the consumer service requests a feed document (i.e. a section.xml document) available for a given collection of the VLER on a specific individual. This figure also illustrates a key feature of the VLER DAS in that it aggregates the responses of multiple producers into a single Atom feed document for the collections that it manages.

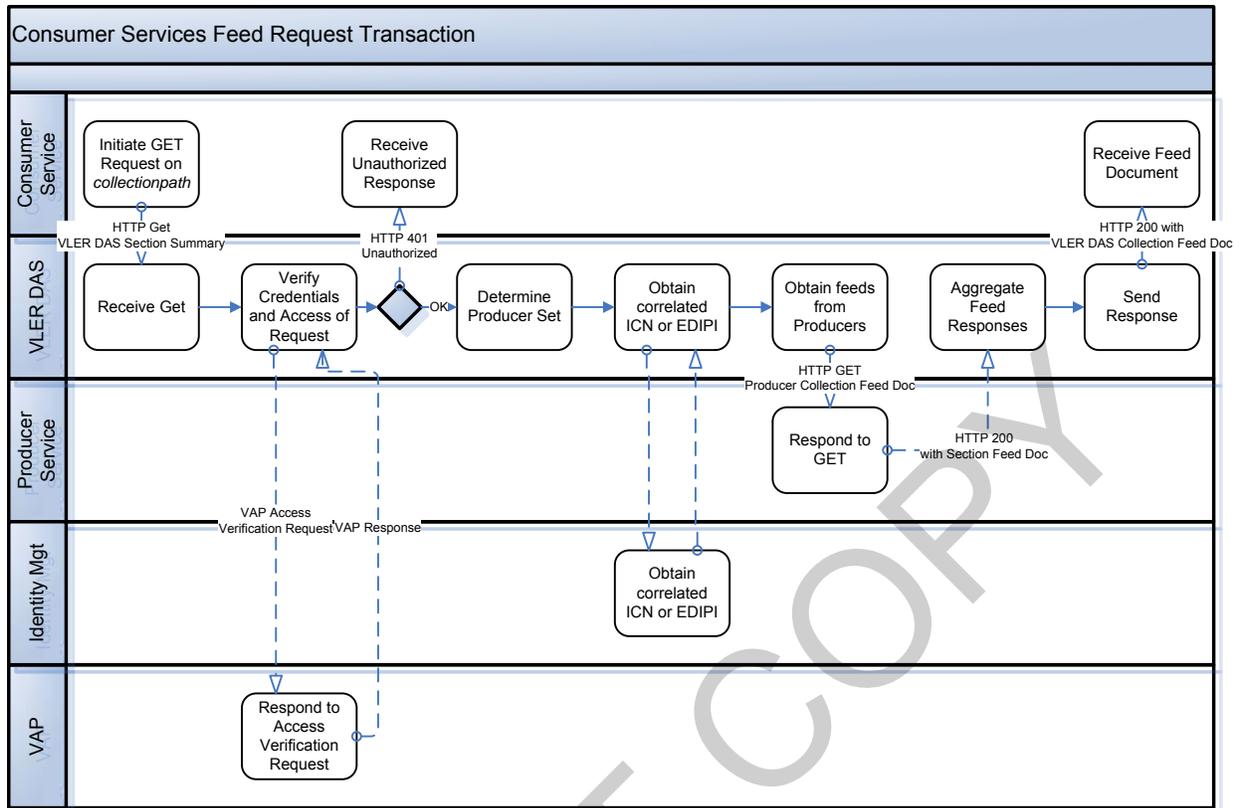


Figure 2 - GET Collection Feed Document Transaction

Figure 3 (below) illustrates a transaction whereby a consumer service requests a specific document within a given collection of the VLER on a specific individual. This request commonly follows a GET of the collection feed document. This figure also illustrates a key feature of the VLER DAS in that consumers will request a specific *collectionpath/documentname* within the VLERDAS *baseURL* and the VLER DAS will ascertain from the collection feed document which producer maintains the associated subject document.

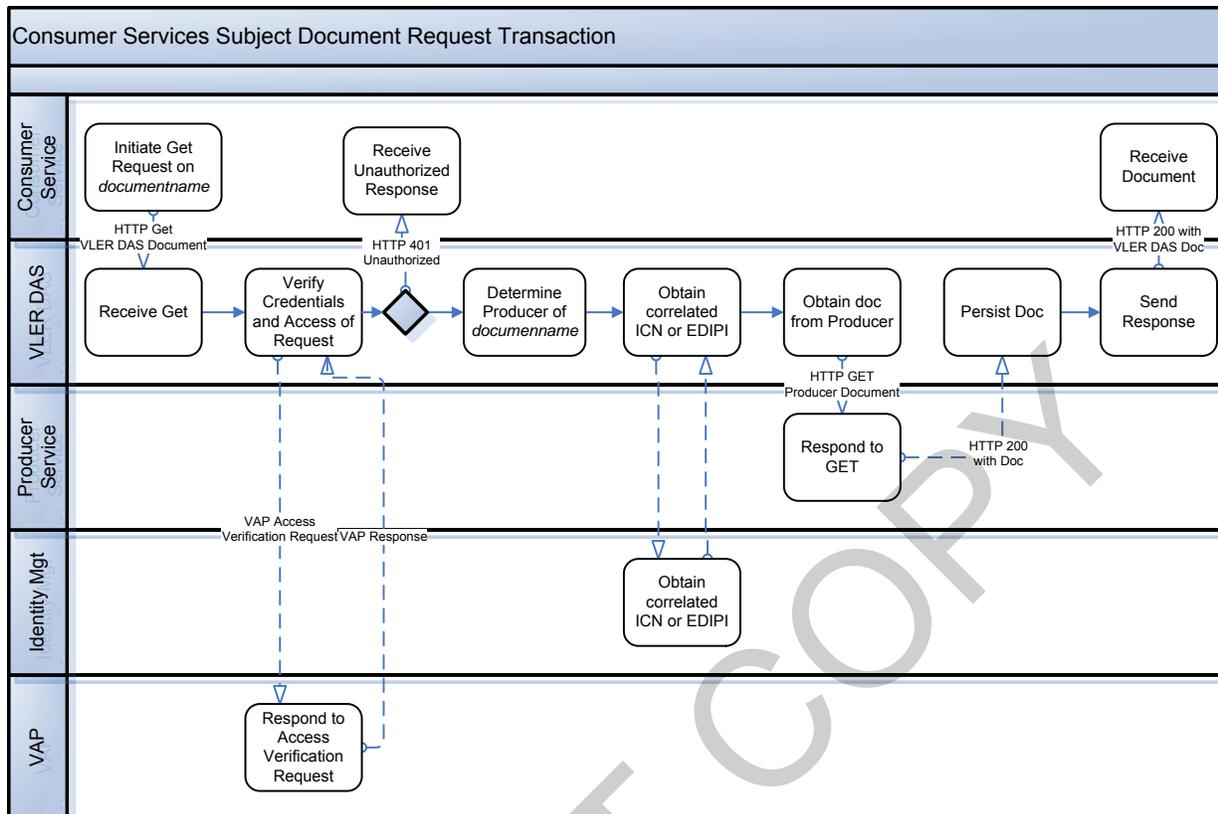


Figure 3 - Get Document Transaction

2.6. Data Exchanges

VLER DAS data exchanges with consumer and producer services are HTTP GET requests that focus on the retrieval of summary lists of resources associated with a specific individual within a specific collection or obtaining a specific resource. Resources are all media types allowed by VLER; however, most will be XML documents, text, or pdf files. The collections and specific XSDs for document types are defined in Appendix A and B, respectively.

All standard variations of the HTTP GET are supported, such as Conditional GETs. All responses are standard HTTP status codes. The entity body of the 200 (“OK”) status code includes a representation of the requested resource.

2.7. Precedence and Criticality

All operations shall be processed with equal priority. VLER DAS shall give all Consumer operation requests equal priority and Producers shall give all VLER DAS operation requests equal priority.

2.8. Communications Methods

All interfaces between the VLER DAS and Consumers and Producers shall use HTTP 1.1 communication methods supplemented by Security and Privacy considerations described below (see section 2.10).

Error Messages

The individual values of the numeric status codes defined for HTTP/1.1, and an example set of corresponding Reason-Phrase's, are presented below. The reason phrases listed here are only recommendations -- they MAY be replaced by local equivalents without affecting the protocol.

Sections represent the section of the RFC 2616 HTTP/1.1.

Table 4: HTTP 1.1 Error Codes

Status Code / RFC 2616 Section Reference / Reason Phrase
"100" ; Section 10.1.1: Continue
"101" ; Section 10.1.2: Switching Protocols
"200" ; Section 10.2.1: OK
"201" ; Section 10.2.2: Created
"202" ; Section 10.2.3: Accepted
"203" ; Section 10.2.4: Non-Authoritative Information
"204" ; Section 10.2.5: No Content
"205" ; Section 10.2.6: Reset Content
"206" ; Section 10.2.7: Partial Content
"300" ; Section 10.3.1: Multiple Choices
"301" ; Section 10.3.2: Moved Permanently
"302" ; Section 10.3.3: Found
"303" ; Section 10.3.4: See Other
"304" ; Section 10.3.5: Not Modified
"305" ; Section 10.3.6: Use Proxy
"307" ; Section 10.3.8: Temporary Redirect
"400" ; Section 10.4.1: Bad Request
"401" ; Section 10.4.2: Unauthorized
"402" ; Section 10.4.3: Payment Required
"403" ; Section 10.4.4: Forbidden
"404" ; Section 10.4.5: Not Found
"405" ; Section 10.4.6: Method Not Allowed
"406" ; Section 10.4.7: Not Acceptable

Status Code / RFC 2616 Section Reference / Reason Phrase
"407" ; Section 10.4.8: Proxy Authentication Required
"408" ; Section 10.4.9: Request Time-out
"409" ; Section 10.4.10: Conflict
"410" ; Section 10.4.11: Gone
"411" ; Section 10.4.12: Length Required
"412" ; Section 10.4.13: Precondition Failed
"413" ; Section 10.4.14: Request Entity Too Large
"414" ; Section 10.4.15: Request-URI Too Large
"415" ; Section 10.4.16: Unsupported Media Type
"416" ; Section 10.4.17: Requested range not satisfiable
"417" ; Section 10.4.18: Expectation Failed
"500" ; Section 10.5.1: Internal Server Error
"501" ; Section 10.5.2: Not Implemented
"502" ; Section 10.5.3: Bad Gateway
"503" ; Section 10.5.4: Service Unavailable
"504" ; Section 10.5.5: Gateway Time-out
"505" ; Section 10.5.6: HTTP Version not supported

2.9. Performance Requirements

End to end response times for Consumer requests shall be less than 5 seconds (for 90% of all Consumer requests). Table 5 depicts the performance allocation for transactions described in section 2.5, Transaction Types.

Table 5- Transaction Response Performance Allocation

System	Transaction Subset (see Figure 2 and 3)	Response Time (msec) / 90% of transactions
VLER DAS	Start: Receive Consumer Request End: Send Request to all related Producers	<2000 (includes Identity Mgt queries, VAP query, and network latency)
VAP	Respond to access verification request	<250
Identity Mgt	Respond to ICN-EDIPI correlation request	<250

System	Transaction Subset (see Figure 2 and 3)	Response Time (msec) / 90% of transactions
Producer	Respond to VLER DAS request	<1000
VLER DAS	Start: Receive response from last related Producer End: Send response to Consumer	<1000
Reserve	For network latency and pending actual test results	1000

2.10. Security and Privacy

Security and privacy will be enforced at multiple levels. Https and certificate management through Federal Bridge and FIPS 140-2 must be used.

3. Interface Requirements

This section provides additional interface requirements unique to each interface. Where appropriate, interface requirements describe in Section 2 are referenced. The interfaces described are:

- Consumer Services – VLER DAS Interface
- VLER DAS – Producer Services Interface

3.1. Consumer Services – VLER DAS Interface

Consumer services are any system, application, or service that needs to retrieve data from servers that persist VLER data. VLER DAS acts as a proxy to these servers providing a consistent interface for consumer services.

This interface conforms to Section 2 with any special requirements described below.

3.1.1. Interface Processing Time Requirements

VLER DAS will respond to consumer requests within 5 seconds for 90% of the requests. Large files being transferred that can not be completely sent within the 5 seconds due to network speeds may take longer; however, the transfer will begin within the specified time.

3.1.2. Message/File Requirements

The VLER DAS response represents an aggregation from all provider services that the VLER DAS connects to.

The schemas for these messages are defined in Appendix B, section 7.4, vData Content Profiles.

3.1.3. Communication Methods

The communication methods described in Section 2.8 applies to this interface.

3.1.4. Security Requirements

The security methods described in Section 2.10 applies to this interface.

DAS will take a number of attributes including the User Name, Origin of the request, Role, Purpose of Use and the targeted resource requested (including the person Id) and will check whether that request can be made. If it obtains VAP authorization the request will continue. If not, the DAS will return “403 Forbidden”.

Note that any caching performed in the DAS will be stored with the credentials of Role and Purpose of Use under which it was obtained and further access to that data in the cache will validate that the credentials of the user match those of the data in the cache.

3.1.5. Physical Requirements

Not applicable.

3.1.6. Structural Requirements

Not applicable.

3.1.7. Mechanical Requirements

Not applicable.

3.1.8. Electrical Requirements

Not applicable.

3.1.9. Thermal Requirements

Not applicable.

3.1.10. Special Fluid Requirements

Not applicable.

3.1.11. Signal Requirements

The signal requirements are a subset of the communication method standards and require no further clarification.

3.2. VLER DAS – Producer Services Interface

Producer services are any system, application, or service that offers VLER data persistence in support of consumer services. VLER DAS acts as a proxy between these servers and consumer services thereby providing a consistent interface to consumer services.

This interface conforms to Section 2 with any special requirements described below.

3.2.1. Interface Processing Time Requirements

Producers will respond to VLER DAS requests within 1 second for 90% of the requests (excludes network latency). Large files being transferred that can not be completely sent within the 1 second time due to network speeds may take longer; however, the transfer will begin within the specified time.

3.2.2. Message/File Requirements

The schemas for these messages are defined in Appendix B, section 7.4, vData Content Profiles.

3.2.3. Communication Methods

The communication methods described in Section 2.8 applies to this interface.

3.2.4. Security Requirements

The security methods described in Section 2.10 applies to this interface.

It is not expected that the Producer will try to re-authenticate the user and is not expected to participate in a single sign on mechanism.

It is expected that a DAS producer will inspect the Role and Purpose of Use to determine whether access is allowed.

If access is not allowed the DAS Producer will return “403 Forbidden”

3.2.5. Physical Requirements

Not applicable.

3.2.6. Structural Requirements

Not applicable.

3.2.7. Mechanical Requirements

Not applicable.

3.2.8. Electrical Requirements

Not applicable.

3.2.9. Thermal Requirements

Not applicable.

3.2.10. Special Fluid Requirements

Not applicable.

3.2.11. Signal Requirements

The signal requirements are a subset of the communication method standards and require no further clarification.

DO NOT COPY

4. Interface Verification

TBD

DO NOT COPY

5. Appendix A – VLER vData Collection Hierarchy

The VLER DAS acts as a [Feed Aggregator](#) for each Atom collection that comprises the VLER.

Aggregators reduce the time and effort needed to regularly check web servers for updates, creating a unique information space or "personal newspaper". Once subscribed to a feed, an aggregator is able to check for new content at determined intervals or on an ad hoc basis and retrieve the updates. The content is sometimes described as being "[pulled](#)" to the subscriber, as opposed to "[pushed](#)" with email or IM.

The aggregator provides a consolidated view of the content within its own feed document for each collection. The VLER vData collection hierarchy can be thought of as a hierarchical folder structure where each folder is an Atom collection populated with content from producers.

The URI structure for the collection hierarchy is defined in the VLER Data Collection Hierarchy spreadsheet, draft 10, dated January 9, 2012. The document is maintained on the VA's VLER SharePoint site within the folder at:

<http://vaww.oed.portal.va.gov/products/vler/Requirements%20Management/Forms/AllItems.aspx?RootFolder=%2fproducts%2fvler%2fRequirements%20Management%2fVLER%20PMO%20Documentation%2fDraft%20Documents&FolderCTID=&View=%7bF41EC702%2dCF52%2d4D14%2dB6F7%2d6E049BDD6CC4%7d>

The spreadsheet is also embedded herein:



VLER Data Collection
Heirarchy Draft10.xls

6. Appendix B – Schemas and Data Elements

6.1. Feed Document

An Atom feed XML document MUST exist for every Atom collection or be created upon demand. The filename of such documents MUST be “section.xml”. Each VLER collection is an Atom collection. The feed schema is a standard Atom entry document schema. Many of the XML elements contained within the feed document are of the same type as the atomEntry document; however, the context is that of the feed. An XSD for the entry document is at <TBD location>.

Atom feed elements are:

- Author
- Category
- Contributor
- Generator
- Icon
- Id
- Link
- Logo
- Rights
- Subtitle
- Title
- Updated
- extensionElement
- Entry

The XML Namespace used for the XML data format within the entry document is:

<http://www.w3.org/2005/Atom>

Table 6: Atom Feed Document Schema

Element	Multi- plicity	Description
Author	*	Author of the feed. (See atomEntry)
Category	*	Feed categories. (See atomEntry)
Contributor	*	Contributor to the feed. (See atomEntry)
Generator	0..1	Identifies the agent used to generate a feed, for debugging and other

Element	Multiplicity	Description
		<p>purposes.</p> <p>Attributes:</p> <ul style="list-style-type: none"> • uri • version <p>uri: the URI of the generating agent</p> <p>version: indicates the version of the generating agent</p> <p>Rules:</p> <ol style="list-style-type: none"> The content of generator, when present, MUST be a string that is a human-readable name for the generating agent When the URI is dereferenced, the resulting URI (mapped from an IRI, if necessary) SHOULD produce a representation that is relevant to that agent
Icon	0..1	<p>An IRI reference that identifies an image that provides iconic visual identification for a feed.</p> <p>Rules:</p> <ol style="list-style-type: none"> The image SHOULD have an aspect ratio of one (horizontal) to one (vertical) and SHOULD be suitable for presentation at a small size
Id	1	Id of the feed. (See atomEntry)
Link	*	Links associated with the feed. (See atomEntry)
Logo	0..1	<p>Content is an IRI reference [RFC3987] that identifies an image that provides visual identification for a feed.</p> <p>Rules:</p> <ol style="list-style-type: none"> The image SHOULD have an aspect ratio of 2 (horizontal) to 1 (vertical).
Rights	0..1	Rights of the feed. (See atomEntry)
Subtitle	0..1	A Text construct that conveys a human readable description or subtitle for a feed.
Title	1	Title of the feed. (See atomEntry)
Updated	1	Update timestamp of the feed. (See atomEntry)
extensionElement	*	Reserved for future capability
Entry	*	One for each Entry in the feed. The entire atomEntry is included (see

Element	Multi- plicity	Description
		section 6.2). Rules: a. If multiple atom:entry elements with the same atom:id value appear in an Atom Feed Document, they represent the same entry. Their atom:updated timestamps SHOULD be different. If an Atom Feed Document contains multiple entries with the same atom:id, Atom Processors MAY choose to display all of them or some subset of them. One typical behavior would be to display only the entry with the latest atom:updated timestamp.

6.2. Atom: Entry Element

An Atom entry XML element is a mixed content element (i.e. an element composed of child elements) that must exist for every entry represented within an Atom feed document.

The atom entry is composed of the following child elements:

- Author
- Category
- Content
- Contributor
- Id
- Link
- Published
- Rights
- Source
- Summary
- Title
- Updated
- extensionElement

The entry element can wholly contain content or can reference a subject document using a URI link. The entry schema is a standard Atom entry element schema with VLER DAS defined extensions. An XSD for the entry document is at <TBD location>.

The XML Namespace used for the XML data format within the entry document is:

<http://www.w3.org/2005/Atom>

The atom:entry child elements are defined in Table 2. The elements make use of the Atom common constructs of text, person, and date defined in Table 3. When an entry is used in conjunction with a subject document created with a VLER XML Schemas, there may be an optional Preview complex element that corresponds to the structured XML Schema.

Previews are a set of elements that directly correlate to elements contained within structured subject documents. The intended purpose of Preview elements is to make information content available to Consumers directly from the atom feed document rather than having to subsequently open structured subject documents. This provides a higher value “first pass” query to Consumers who can make use of the content. Structured documents with VLER Content Profiles (XSD-based) that have previews would have a related XSD to serve as the Preview and the preview XSD would be a subset of the larger subject document. Preview is optional and not all VLER Content Profiles will have a related Preview. Previews are contained within atom:entry:content.

Table 7: Atom Entry Element Schema

Element	Multi- plicity	Description
Author	*	<p>The "atom:author" element is a Person construct that indicates the author of the entry or feed.</p> <p>Rules:</p> <p>Required unless:</p> <ol style="list-style-type: none"> the atom:entry contains an atom:source element that contains an atom:author element or, the atom:feed document contains an atom:author element itself. In this case the feed author applies.
Category	*	<p>The "atom:category" element conveys information about a category associated with an entry or feed. Atom assigns no meaning to the content (if any) of this element.</p>
Content	0..1	<p>Contains content of the entry or links to the content of the entry.</p> <p>Attribute:</p> <ul style="list-style-type: none"> type = “text”, “html”, “xhtml”, application/xml or any atomMediaType src = any URI <p>Rules:</p> <ol style="list-style-type: none"> If the type is “text”, “html”, or “xhtml”, then the content of the content element is the same as defined in the atom text construct (see Table 3). The src attribute MUST NOT be used. If the type is an atomMediaType, it must conform to the syntax of the MIME media type, but it MUST NOT be a composite type. Only an atomMediaType MAY make use of the src attribute. Optionally, an atomMediaType MAY NOT use the src

Element	Multi- plicity	Description
		<p>attribute.</p> <p>d. When the type is atomMediaType and the src attribute is not used, the content that follows will be either text or anyElement.</p> <p>e. When the type is atomMediaType and the src attribute is used, no further content is allowed.</p> <p>f. If the type is an XML media type or ends with "+xml" or "/xml" (case sensitive), the content MAY include child elements and should be suitable for handling in the indicated media type. In this situation, when content other than a src attribute follows, the "alternate" atom:entry:link element serves as the referenced subject document (see Link element below in this table).</p> <p>g. If neither the type attribute nor the src attribute is provided, processors MUST behave as though the type attribute present was "text".</p> <p>h. Entries for which there is an associated XML schema that are wholly contained within Content MUST:</p> <ul style="list-style-type: none"> • Set type as "application /xml" • Include XML that is compliant with the XML schema <p>i. Entries for which the Content entry is not used because the entry is completely represented by a subject document to be referenced:</p> <ul style="list-style-type: none"> • Set a Link element (see Link below) reference of ref = "enclosure" with the URI pointing to the subject document <p>j. Content may include a Preview. A Preview is a set of elements that directly correlate to elements contained within structured subject documents (i.e. documents compliant with an associated vData XML Schema). The intended purpose of Preview elements is to make information content available to Consumers directly from the atom feed document rather than having to subsequently open structured subject documents. This provides a higher value "first pass" query to Consumers who can make use of the content. Structured documents with VLER Content Profiles (XSD-based) that have previews would have a related XSD to serve as the Preview and the preview XML would be a subset of the larger subject document. The root element of all Previews is "Preview". Preview is optional and not all VLER Content Profiles will have a related Preview. If so:</p>

Element	Multi- plicity	Description
Contributor	*	<ul style="list-style-type: none"> • Set type as “application /xml” • Include XML that is compliant with the Preview XML schema associated with the subject document • Set a Link element (see Link below) reference of ref = “enclosure” with the URI pointing to the subject document <p>A Person construct that indicates a person or other entity who contributed to the entry or feed.</p>
Id	1	<p>Conveys a permanent, universally unique identifier for an entry or feed.</p> <p>Rules:</p> <ol style="list-style-type: none"> a. MUST be an IRI. Note that the definition of "IRI" excludes relative references. Though the IRI might use a dereferencable scheme, Atom Processors MUST NOT assume it can be dereferenced. b. When an Atom Document is relocated, migrated, syndicated, republished, exported, or imported, the content of its atom:id element MUST NOT change. Put another way, an atom:id element pertains to all instantiations of a particular Atom entry or feed; revisions retain the same content in their atom:id elements. It is suggested that the atom:id element be stored along with the associated resource. c. The content of an atom:id element MUST be created in a way that assures uniqueness d. Because of the risk of confusion between IRIs that would be equivalent if they were mapped to URIs and dereferenced, the following normalization strategy SHOULD be applied when generating atom:id elements: <ul style="list-style-type: none"> • Provide the scheme in lowercase characters • Provide the host, if any, in lowercase characters • Only perform percent-encoding where it is essential • Use uppercase A through F characters when percent-encoding • Prevent dot-segments from appearing in paths • For schemes that define a default authority, use an empty authority if the default is desired • For schemes that define an empty path to be equivalent to a path of "/", use "/"

Element	Multi- plicity	Description
Link	*	<ul style="list-style-type: none"> • For schemes that define a port, use an empty port if the default is desired • Preserve empty fragment identifiers and queries • Ensure that all components of the IRI are appropriately character normalized, e.g., by using NFC or NFKC <p>The "atom:link" element defines a reference from an entry or feed to a Web resource. Atom assigns no meaning to the content (if any) of this element</p> <p>Rules:</p> <ol style="list-style-type: none"> a. atom:entry elements that contain no child atom:content element MUST contain at least one atom:link element with a rel attribute value of "enclosure". This represents the link to the subject document associated with the entry. b. atom:entry elements MUST NOT contain more than one atom:link element with a rel attribute value of "alternate" that has the same combination of type and hreflang attribute values c. atom:entry elements MAY contain additional atom:link elements beyond those described above d. Links to documents that are similar to the subject document, but formatted differently should set the rel attribute to "alternate" e. Links to documents that are cross referenced with the atom:entry (including subject documents) should set the rel attribute to "related"
Published	0..1	<p>The "atom:published" element is a Date construct indicating an instant in time associated with an event early in the life cycle of the entry.</p> <p>Rules:</p> <p>Typically, atom:published will be associated with the initial creation or first availability of the resource.</p>
Rights	0..1	<p>The "atom:rights" element is a Text construct that conveys information about rights held in and over an entry or feed.</p> <p>Rules:</p> <ol style="list-style-type: none"> a. The atom:rights element SHOULD NOT be used to convey machine-readable licensing information b. If an atom:entry element does not contain an atom:rights element, then the atom:rights element of the containing

Element	Multi- plicity	Description
Source	0..1	<p>atom:feed element, if present, is considered to apply to the entry</p> <p>A construct for preserving an entries metadata when copied from one feed to another.</p> <p>Rules:</p> <ol style="list-style-type: none"> If an atom:entry is copied from one feed into another feed, then the source atom:feed's metadata (all child elements of atom:feed other than the atom:entry elements) MAY be preserved within the copied entry by adding an atom:source child element, if it is not already present in the entry, and including some or all of the source feed's Metadata elements as the atom:source element's children. Such metadata SHOULD be preserved if the source atom:feed contains any of the child elements atom:author, atom:contributor, atom:rights, or atom:category and those child elements are not present in the source atom:entry The atom:source element is designed to allow the aggregation of entries from different feeds while retaining information about an entry's source feed. For this reason, Atom Processors that are performing such aggregation SHOULD include at least the required feed-level Metadata elements (atom:id, atom:title, and atom:updated) in the atom:source element.
Summary	0..1	<p>The "atom:summary" element is a Text construct that conveys a short summary, abstract, or excerpt of an entry.</p> <p>Rules:</p> <ol style="list-style-type: none"> atom:entry elements MUST contain an atom:summary element in either of the following cases: <ul style="list-style-type: none"> the atom:entry contains an atom:content that has a "src" attribute (and is thus empty). the atom:entry contains content that is encoded in Base64; i.e., the "type" attribute of atom:content is a MIME media type [MIMEREG], but is not an XML media type [RFC3023], does not begin with "text/", and does not end with "/xml" or "+xml". It is not advisable for the atom:summary element to duplicate atom:title or atom:content because Atom Processors might assume there is a useful summary when there is none.
Title	1	The "atom:title" element is a Text construct that conveys a human

Element	Multiplicity	Description
Updated	1	readable title for an entry or feed. The "atom:updated" element is a Date construct indicating the most recent instant in time when an entry or feed was modified in a way the publisher considers significant.
Extensions	0..1	Extensions to the standard Atom:entry. VLER DAS defined subelements are: <ul style="list-style-type: none"> Confidentiality (0..1) AccessControl (0..1) Consent (0..1) <p><u>Confidentiality</u>: This element contains controls for confidentiality that apply to the entry and subject document</p> <p><u>AccessControl</u>: This element contains controls for access control that apply to the entry and subject document.</p> <p><u>Consent</u>: This element contains controls for consent that apply to the entry and subject document</p>

Table 8: Common Atom Constructs

Element	Description
Text	<p>A Text construct contains human-readable text, usually in small quantities. The content of Text constructs is Language-Sensitive.</p> <p>Attributes:</p> <ol style="list-style-type: none"> Type = "text", "html", or "xhtml" <p>Rules:</p> <ol style="list-style-type: none"> If the type attribute is not provided then processors will consider the contents to be "text" If the value of type is "text", the content of the Text construct MUST NOT contain child elements. Such text is intended to be presented to humans in a readable fashion. Thus, Atom Processors MAY collapse white space (including line breaks) and display the text using typographic techniques such as justification and proportional fonts. If the value of type is "html", the content of the Text construct MUST NOT contain child elements and SHOULD be suitable for handling as HTML [HTML]. Any markup within MUST be escaped; for example, "
" as "&lt;br>". HTML markup within SHOULD be such that it could validly appear directly

Element	Description
Person	<p>within an HTML <DIV> element, after unescaping. Atom Processors that display such content MAY use that markup to aid in its display.</p> <p>d. If the value of type is "xhtml", the content of the Text construct MUST be a single XHTML div element [XHTML] and SHOULD be suitable for handling as XHTML. The XHTML div element itself MUST NOT be considered part of the content. Atom Processors that display the content MAY use the markup to aid in displaying it. The escaped versions of characters such as "&" and ">" represent those characters, not markup.</p> <p>A Person construct is an element that describes a person, corporation, or similar entity (hereafter, 'person'). There is no significance to the order of the child elements. The child elements are:</p> <ul style="list-style-type: none"> • Name • URI • Email • extensionElement (not used by this spec) <p>Rules:</p> <p>a. Name: conveys a human-readable name for the person. The content of name is Language-Sensitive. Person constructs MUST contain exactly one name element.</p> <p>b. URI: conveys an IRI associated with the person. Person constructs MAY contain an uri element, but MUST NOT contain more than one. The content of uri in a Person construct MUST be an IRI reference.</p> <p>c. Email: conveys an e-mail address associated with the person. Person constructs MAY contain an email element, but MUST NOT contain more than one. Its content MUST conform to the "addr-spec" production in [RFC2822].</p>
Date	<p>A Date construct is an element whose content MUST conform to the "date-time" production in [RFC3339]. In addition, an uppercase "T" character MUST be used to separate date and time, and an uppercase "Z" character MUST be present in the absence of a numeric time zone offset.</p> <p>Examples:</p> <pre><updated>2003-12-13T18:30:02Z</updated> <updated>2003-12-13T18:30:02.25Z</updated> <updated>2003-12-13T18:30:02+01:00</updated> <updated>2003-12-13T18:30:02.25+01:00</updated></pre>

6.3. Root.xml

The root document is at the root of the hierarchy. It MUST comply with the root.xsd schema at <TBD location>. The XML Namespace used for the XML data format within the entry document is:

<https://github.com/projecthdata/hData/blob/master/schemas/2009/06/root.xsd>

It contains the following elements:

Table 9: Root.xml Element Schema

Element	Multiplicity	Description
id	1	This element uniquely identifies the document, e.g. through a textual representation of a UUID. It is RECOMMENDED to not use absolute URIs, but only fragments that may be used within a URI.
version	1	The version of the vData structure used within this document. It corresponds to the version number of this ICD.
created	1	Creation date of the document, using the W3C XML Schema Date data type. This data SHOULD be significant to at least the second.
lastModified	1	Last modification of the document, using the W3C XML Schema Date data type. This data SHOULD be significant to at least the second.
extensions	1	<p>Contains a list of extensions defined to be used within the vData structure. Any extension to be available for use in collections MUST register itself in this document.</p> <p>Subelements:</p> <ul style="list-style-type: none"> • Extension (*) <ul style="list-style-type: none"> ○ contentType (an attribute) ○ extensionId (an attribute) <p>Extension: This text element contains a unique identifier for the extension. It is RECOMMENDED to use a URL. For elements of content type “application/xml”, it is RECOMMENDED that the text element contains a URL that provides a RDDDL document that describes the format of instances of XML document of this extension type by including a <rddl:resource> element with the xlink:role attribute set to the schema definition. For other content types, it is RECOMMENDED that the RDDDL document resolves to an Atom format, such as a PDF or HTML description.</p> <p>contentType: contains the content type of all documents in a</p>

Element	Multi- plicity	Description
sections	1	<p>collection that registers with this extension. If the attribute is not present, the documents in the collection MUST be of content type “application/xml”.</p> <p>extensionId: contains a URI identifier for the extension. It MUST be unique within the root document.</p> <p>This element contains references to all the collections that comprise the vData structure.</p> <p>Subelements:</p> <ul style="list-style-type: none"> • Section (*) <ul style="list-style-type: none"> ○ Path (a required attribute) ○ Name (an optional attribute) ○ extensionId (a required attribute) ○ requirement (an optional attribute) ○ Section (*) <p>Section: a single instance of a section within the vData structure. A section may have subcollections within it as well.</p> <p>Path: This text attribute is a path segment, used to construct the full path to the collection from the root document. Valid characters are [a---z][A---Z][0---9]. The full path to a collection is obtained by starting with a forward slash (“/”), and concatenating the path segments, separated by forward slashes.</p> <p>extensionId: This identifier MUST be equal to the identifier of any of the registered extension elements, as identified by the id attribute of the <extension> element. It describes the default contentType for documents contained in this collection. Note that the metadata for each individual document MAY override the default contentType.</p> <p>Name: Used for a human---friendly name to this collection.</p> <p>Requirement: this attribute indicates if a given collection is required or optional. Valid values are “required” or “optional”. If this attribute is not present, the collection is “required”.</p>

6.4. vData XML Schemas

vData XML Schemas are schemas that define the normalized content of documents or records contained within specific VLER collections. VLER XML Schemas are to be developed for:

- Health (hData)
- Benefits (bData)
- Personnel (pData)
- Other Admin (aData)

The available xsd schemas are maintained at <https://vler.va.gov/das/schemas>.

Note: this URL is notional until the first implementation of VLER DAS.

DO NOT COPY