

Repositories

Health Data Repository 3.8

System Design Document Volume 1

(Volume 1 of 2)

Version 0.9



March 2015

Revision History

Date	Version	Description	Author
09/08/2014	0.1	This is an initial document that captures the high level details of the scope changes for HDR 3.8	
9/16/2014	0.2	Tables 17, 19, and 20 Updated. Table 21 Deleted, Renumbered subsequent tables.	
10/03/2014	0.3	Updated Section 5; renumbered figures; updated footer to include Vol 1.	
10/17/2014	0.4	Updated TRM table with most current version	
10/29/2014	0.5	Updates to Interface section for HMP/eHMP client.	
11/02/2014	0.6	Updates from peer review throughout. Updated Data Flow Diagram with CDW added and VSSC and DALC connection.	
11/06/2014	0.7	Updates to Sections 1.3 Scope; 2.2/2.2.7 added new data extract process and flow diagram; 4.3 Network Architecture updates for CDW and SOA diagram	
11/06/2014	0.8	Update to TRM Table for Maven and Cache	
11/13/2014	0.9	Signatures applied for this release of SDD	

Table of Contents

1. Introduction.....	9
1.1. Purpose of the SDD.....	10
1.2. Identification	10
1.3. Scope	10
1.4. Constraining Policies, Directives and Procedures.....	10
1.5. User Characteristics.....	11
1.6. Relationship to Other Documents and Plans.....	11
1.7. Definitions, Acronyms, and Abbreviations.....	11
1.8. References	11
2. Background.....	12
2.1. Overview of the System	13
2.1.1. CDS.....	13
2.1.2. Pathways	14
2.1.3. FPDS	14
2.1.4. Aggregate Read Service (ARS)	14
2.1.5. CDS Message Mediator	14
2.1.6. Socket Adapter.....	14
2.2. Overview of the Business Process	15
2.2.1. Create, Update, and Delete Process	15
2.2.2. Read Request Process	17
2.2.3. Get Schema Process	21
2.2.4. Get Patient Identifiers	21
2.2.5. Query Data	22
2.2.6. Handle Errors	23
2.2.7. Data Extract Process	23
2.3. Business Benefits	25
2.4. Assumptions and Constraints	26
2.4.1. Design Assumptions	26
2.4.2. Design Constraints	26
2.4.3. Design Trade-offs	26
2.5. Overview of Significant Requirements	27
2.5.1. Overview of Significant Functional Requirements.....	27
2.5.2. Overview of Functional Workload / Performance Requirements.....	27
2.5.3. Overview of Operational Requirements	27
2.5.4. Overview of Technical Requirements	28
2.5.5. Overview of Security or Privacy Requirements	28

2.5.6.	Overview of System Criticality and High Availability Requirements	28
2.5.7.	Single Sign-on Requirement	29
2.5.8.	Requirement for Use of Enterprise Portals	29
2.5.9.	Special Device Requirements	29
2.6.	Legacy System Requirement.....	29
3.	Conceptual Design	29
3.1.	Conceptual Application Design	29
3.1.1.	Application Context	29
3.1.2.	High-Level Application Design	36
3.1.3.	Application Locations	36
3.2.	Conceptual Data Design	36
3.2.1.	Project Conceptual Data Model	36
3.2.2.	Database Information	37
3.2.3.	User Interface Data Mapping.....	37
3.3.	Conceptual Infrastructure Design.....	37
3.3.1.	System Criticality and High Availability.....	37
3.3.2.	Special Technology.....	38
3.3.3.	Technology Locations.....	38
3.3.4.	Conceptual Infrastructure Diagram.....	39
4.	System Architecture.....	39
4.1.	Hardware Architecture	39
4.2.	Software Architecture	39
4.3.	Network Architecture	39
4.4.	Service Oriented Architecture / ESS	42
4.5.	Enterprise Architecture	43
5.	Data Design.....	53
5.1.	DBMS Files	53
5.2.	Non-DBMS Files.....	53
5.3.	Data View.....	53
6.	Detailed Design.....	53
6.1.	Hardware Detailed Design	53
6.2.	Software Detailed Design and all subsections	53
6.3.	Network Detailed Design	53
6.4.	Service Oriented Architecture / ESS Detailed Design	53
6.4.1.	FPDS JMS Interface	54
6.4.2.	FPDS REST Web Service Clients	54
6.4.3.	ARS SOAP Web Service Clients.....	55
6.4.4.	CDS REST Web Service Clients	55

6.4.5.	CDS SOAP Web Service Clients.....	55
6.4.6.	Socket Adapter Clients	56
6.4.7.	CDS Message Mediator Clients.....	57
7.	External System Interface Design.....	57
7.1.	CDS Modules Overview	58
7.1.1.	CDS Internal Access	58
7.1.2.	UDDI – CDS Directory Service	60
7.2.	Mechanisms to interface CDS with external Clients	64
7.2.1.	Data Exchanges.....	64
7.3.	Client Interface Detailed Design	65
7.3.1.	CDS Data Service Interface	66
7.3.2.	VIM Reads	66
7.3.3.	JSON Reads	70
7.3.4.	ARS Reads	73
7.3.5.	VIM Writes	73
7.3.6.	JSON PGD Writes	75
7.3.7.	HL7 Writes.....	75
7.4.	3.8 Client Interfaces	76
7.4.1.	HTH Client.....	76
7.4.2.	HMP/eHMP VPR Client.....	77
8.	Human-Machine Interface	77
8.1.	Interface Design Rules	77
8.2.	Inputs	78
8.3.	Outputs	78
8.4.	Navigation Hierarchy	78
9.	Security and Privacy	78
9.1.	Security.....	78
9.2.	Privacy.....	79
Appendix A.	Additional Information	81
Appendix B.	Data Elements	81
Attachment A.	Reviews and Approval Signatures	118
Attachment B.	Signature Verification.....	119

Table of Figures

Figure 1 HDR 3.8 Clients and Interactions.....	13
Figure 2 CUD Process Flow	17
Figure 3 Read Process Flow	20
Figure 4 Get Schema Process Flow	21
Figure 5 Get Patient Identifiers Process.....	22
Figure 6 Query Data Process	23
Figure 7 Handle Error Process	23
Figure 8: Data Extract Process.....	25
Figure 9 HDR 3.8 Proposed Production System	30
Figure 10 Repositories SOA	40
Figure 11 HDR GSS	42
Figure 12: Repositories SOA Architecture	43
Figure 13 CDS interactions with services/components/data repositories	58
Figure 14 UDDI configuration:	61
Figure 15 SOAP and REST server-side implementation, configuration of delegating objects....	62
Figure 16 SOAP calls without UDDI:	63
Figure 17 Read Response Template	67
Figure 18 Patient Identification in a Filter	68
Figure 19 Entry Point Section of a Filter	69
Figure 20 HMP/eHMP XML Response with Error Section	70
Figure 21 HMP/eHMP JSON Read Response Format{"sites": [.....	71
Figure 22 Write Request Template	75
Figure 23 Allergies Write Request Template	83
Figure 24 ADL Data Write Template	84
Figure 25 HTH DMP Response Write Template.....	84
Figure 26 Write Clinical Data Response Template	85
Figure 27 Read Response Template	85
Figure 28 OP Read Response Templte	86
Figure 29 Lab CH Read Response Template.....	87
Figure 30 HTH Surveys Read Response Template	88
Figure 31 Non-VA Medications Read Response Template.....	89

Figure 32 Immunization Read Response Template	90
Figure 33 Skin Test Read Response Template	91
Figure 34 Problem List Read Response Template.....	92
Figure 35 PSS Schematic.....	93
Figure 36 ADL Schematic	94
Figure 37 Appointment Read Response Template	95
Figure 38 DMP Generic Response	96
Figure 39 DMP Generic Response with Scores.....	97
Figure 40 TIU Domain Read Filter.....	97
Figure 41 Non-VA Medications Filter schematic.....	98
Figure 42 Immunizations Filter Schematic.....	98
Figure 43 Skin Test Filter Schematic.....	99
Figure 44 Problem List Filter Schematic	99
Figure 45 AppointmentSinglePatient Filter Schematic	100
Figure 46 RequestsandExamsSinglePatient Filter Schematic	101
Figure 47 HTH Survey Reads Request Filter	102
Figure 48 MHV Allergies Read Filter	102
Figure 49 MHV Lab Read Filter.....	103
Figure 50 RDI Allergies and OP Reads Data Filter	104
Figure 51 Mobile Health Appointments Read Data Filter.....	105
Figure 52 Error Section in Read request Response Message	106
Figure 53 HMP/eHMP JSON response with error section{"sites": [.....	113
Figure 54 HMP/eHMP XML format read response	114
Figure 55 HMP/eHMP JSON response with error section{"sites":.....	115
Figure 56 HMP/eHMP XML response with error section.....	116

Table of Tables

Table 1 Scope Inclusions	10
Table 2 Scope Exclusions	10
Table 3 Create, Update, and Delete Process	15
Table 4 Read Request Process	18
Table 5 Functional Requirements	27

Table 6 Workload and Performance Requirements	27
Table 7 Operational Requirements	28
Table 8 Technical Requirements	28
Table 9 Security Requirements	28
Table 10 Proposed Legacy Retirements	29
Table 11 Object.....	31
Table 12 Interfaces Internal to OIT	33
Table 13 Externally Shared Data Stores	34
Table 14 Application Locations.....	36
Table 15 Database Inventory	37
Table 16 Technology Location Details.....	38
Table 17 HDR TRM Technologies / Tools.....	44
Table 18 FPDS JMS Clients	54
Table 19 FPDS REST Web Service Clients	54
Table 20 ARS SOAP Web Service Clients.....	55
Table 21 CDS REST Web Service Clients	55
Table 22 CDS SOAP Web Service Clients.....	55
Table 23 Socket Adapter Clients	57
Table 24 CDS Message Mediator Clients.....	57
Table 25 CDS Data Sources	59
Table 26 Filters between CDS and client	64
Table 27 Template IDs for Read transactions.....	65
Table 28 Template IDs for Write transactions.....	65
Table 29 HTH Census Client.....	76
Table 30 HMP/eHMP Read Client	77
Table 31 Security Control Status	79
Table 32 Exceptions and Errors for CDS 3.8	107

1. Introduction

The Repositories/Health Data Repository (HDR) system encompasses a relational database, HDR Database (HDR DB), as well as Simple Object Access Protocol (SOAP) and Representational State Transfer (REST) standards-based web service interfaces for retrieving data from HDR DB or Veterans Health Information Systems and Technology Architecture (VistA) or both HDR DB and VistA any time from within the VA intranet. The system also includes asynchronous Java Messaging Service (JMS) interfaces for enabling storage of data to the HDR Database any time from within the VA intranet. The web services enable VistA federation and data aggregation when data is retrieved from VistA.

Repositories services include Clinical Data Service (CDS), Pathways, Aggregate Read Service (ARS), and the Federated Patient Data Service (FPDS). All services are based on a Java 2 Platform, Enterprise Edition (Java EE) framework and differ mainly in the data domains and exchange formats supported. CDS, Pathways and ARS support Extensible Markup Language (XML) for data interchange while FPDS supports JavaScript Object Notation (JSON) for data interchange. JMS interfaces accept payload in HL7 format from CDS Socket Adapter and the Vitria Interface Engine and accept payload in JSON format from HMP/eHMP PGD application.

HDR 3.8 functionality includes:

- Extension of Health Management Platform (HMP/eHMP) Virtual Patient Record (VPR) support capabilities to subscribe patients to VPR, cancel patient subscriptions to VPR, and retrieve object for subscribed patients.
- Extend Home Telehealth (HTH) Data Extracts to include a new interface to feed VHA Support Service Center (VSSC) and Denver Acquisition & Logistics Center (DALC).

It is anticipated that all clients using previous versions of HDR will migrate to the 3.8 version of the HDR. Clients using any revisions of HDR 3.x will not be impacted in any way other than the fact that there is additional functionality available to them as outlined above.

Remote Data Interoperability (RDI) service requests on the CDS 3.x service for Allergy and Outpatient Pharmacy data and My HealtheVet (MHV) service requests on 3.x service for Laboratory data will continue to return VistA data stored in the HDR DB so as to preserve the current response time Service Level Agreement (SLA) that is in place with RDI and MHV. For all other clients requesting VistA data from HDR system, HDR 3.x service requests will make use of the VistA federation implementation in HDR 3.x to access VistA data directly rather than returning VistA data that is stored in the HDR DB.

The HDR system along with its services (CDS, Pathways, FPDS and ARS) has additional capabilities, such as providing clinical and non-clinical data from a federation of VistA systems, as well as enabling storage and retrieval of Clinical Health Data Repository (CHDR) Allergy and Outpatient Pharmacy (OP), HTH Vitals, HTH Survey responses, HTH Disease Management Protocols (DMPs) and HTH Census Reports. This aspect of the system is not discussed in this System Design Document (SDD) since these are existing capabilities that have been outlined in the *HDR 3.7 System Design Document* and other related documents.

1.1. Purpose of the SDD

The purpose of this document is to describe in sufficient detail how the proposed system is to be constructed. The SDD translates the requirements specifications into a document from which the developers can create the actual system. It identifies the top-level system architecture, and identifies hardware, software, network and interface components.

Note: The HDR 3.8 SDD has been divided into two separate volumes. This is Volume 1 and consists of everything except Sections 6.1 and 6.2 which are addressed in Volume 2.

1.2. Identification

This SDD document applies to the HDR 3.8 version of the HDR system.

1.3. Scope

Table 1 lists the high level features included in the HDR 3.8 release.

Table 1 Scope Inclusions

Scope Inclusions
Extend HMP/eHMP support to add REST service interface to subscribe and cancel subscriptions to patient VPR and to fetch VPR data from a site for subscribed patients using the VPR 1.2 API implemented by the VistA team and extend the HDRDAT stored procedure wrapper to call the VPR API that is deployed at the VAMC until all VistA systems have the most recent version of the VPR API.
Extend Home Telehealth (HTH) Data Extracts to include a new interface to feed VHA Support Service Center (VSSC) and Denver Acquisition & Logistics Center (DALC) via Corporate Data Warehouse (CDW).

Table 2 lists what is out of scope for the HDR 3.8 release.

Table 2 Scope Exclusions

Scope Exclusions
Any architecture and design information regarding external or internal client interactions, dependent system component connections, and clinical or non-clinical domain Read and Write operations not included in the specified Requirements Specification Documents (RSDs) falls outside of the scope of this SDD.

1.4. Constraining Policies, Directives and Procedures

This document is created to comply with Enterprise Systems Engineering (ESE) ProPath directives and captures the HDR 3.8 system design. The contents of this document adhere to the policies outlined in Product Development (PD) ProPath. After document review, approval, and signature, any changes to the document shall follow the procedural guidelines specified in the PD ProPath directive.

1.5. User Characteristics

The primary client base of the HDR 3.x system are software applications that are configured as required in order to access any one of the interfaces exposed by the HDR 3.x services, JMS modules and socket adapter component. The developers of the client applications must have knowledge about creating web-service client modules that will allow them to consume services exposed by the HDR system as required in order to meet their data access requirements. Any changes that are required by the client applications on the HDR system are largely data-related which sometimes results in changes to the request and response schemas that are used for exchanging information.

1.6. Relationship to Other Documents and Plans

The HDR 3.8 system design is based on the following requirements, and Software Quality Assurance (SQA) plan documents, and uses the specifications and contents in these documents as a guide to the development deliverables:

- *HDR 3.8 Baseline Requirements RSD*
- *HDR 3.8 HMP Reads Enhancements RSD*
- *HDR 3.8 HTH Census Data Exports RSD*

1.7. Definitions, Acronyms, and Abbreviations

The abbreviations, acronyms, and definitions referenced in this document are provided in the “Repositories Abbreviations, Acronyms, and Definitions” section of the Technical Services Project Repository (TSPR) located at the following URL:

<http://tspr.VistA.med.va.gov/warboard/ProjectDocs/HDR/RepositoriesAbbreviationsAcronymsDefinitions.htm>

If you do not have VA network or TSPR access, contact the author of the document or your Repositories point of contact to request access.

1.8. References

The artifacts referenced in this document are located in the “Documentation Reference Library” in the Technical Services Project Repository (TSPR) at the following link:

<http://tspr.vista.med.va.gov/warboard/ProjectDocs/HDR/HDR3-8Release.htm>

If you do not have VA network or TSPR access, contact the author of the document or your Repositories point of contact to request access.

2. Background

The HDR system serves as the data service layer in the VA enterprise. The HDR system provides Create, Read, Update and Delete (CRUD) services support for data stored in HDR DB and Read service support for data stored in VistA.

The HDR system provides clinical, administrative and other data to its clients in Health Level Seven (HL7), XML and JSON formats. Clients access HDR data via SOAP, REST, and JMS interfaces. Data returned to clients may include data from the HDR DB, VistA or both. The HDR system also supports storage of non-VistA data in the HDR DB as outlined below.

The HDR system implements the Data Federation Design Pattern (DFDP) to facilitate parallel access to the data sources and the aggregation of data retrieved from the various data sources. HDR integrates with the Identity Management System (IdM) to obtain corresponding local identifiers when a National Identifier is supplied on Read requests and uses this information to determine the VistA systems from which to extract data.

The core service framework, called the CDS framework, is a Java EE-based stateless EJB application. The CDS framework encapsulates request processing, identity correspondence, data source location, persistence, data retrieval, data federation, error handling, audit logging, data aggregation and response generation. All services that are part of the HDR system make use of the CDS framework to satisfy client requirements. The web services that are based off of the CDS framework differ either in the data domains supported, payload types supported, data sources accessed, data transfer protocols supported or a combination thereof.

The core database component of the HDR system is HDR DB, an Oracle relational database. HDR DB houses allergies and outpatient pharmacy data from VistA, as well as allergies and outpatient pharmacy data from Department of Defense (DoD) via CHDR for Active Dual Consumer (ADC) patients. HTH data that includes Vitals, Survey responses, and Disease Management Protocols and Patient Census are stored in the HDR DB. HDR 3.7 included additions to HDR DB for storage of PGD data from HMP/eHMP. HDR 3.8 will provide:

- Extension of Health Management Platform (HMP/eHMP) Virtual Patient Record (VPR) support capabilities to subscribe patients to VPR, cancel patients from VPR subscription, and fetch subscribed patients' VPR data from a site.
- Extend Home Telehealth (HTH) Data Extracts to include a new interface to feed VSSC and DALC.

Another core component of the HDR system is the HDRDAT, which includes a set of classes for exposing VistA files through a Structured Query Language (SQL) interface. HDRDAT is a data access component deployed to each VistA instance that enables VistA to be queried via SQL. HDRDAT also includes stored procedures and custom functions that are used to obtain VistA data via Cache Object Script (COS) classes, methods, and routines. The core CDS framework interacts with HDRDAT for retrieving data from VistA.

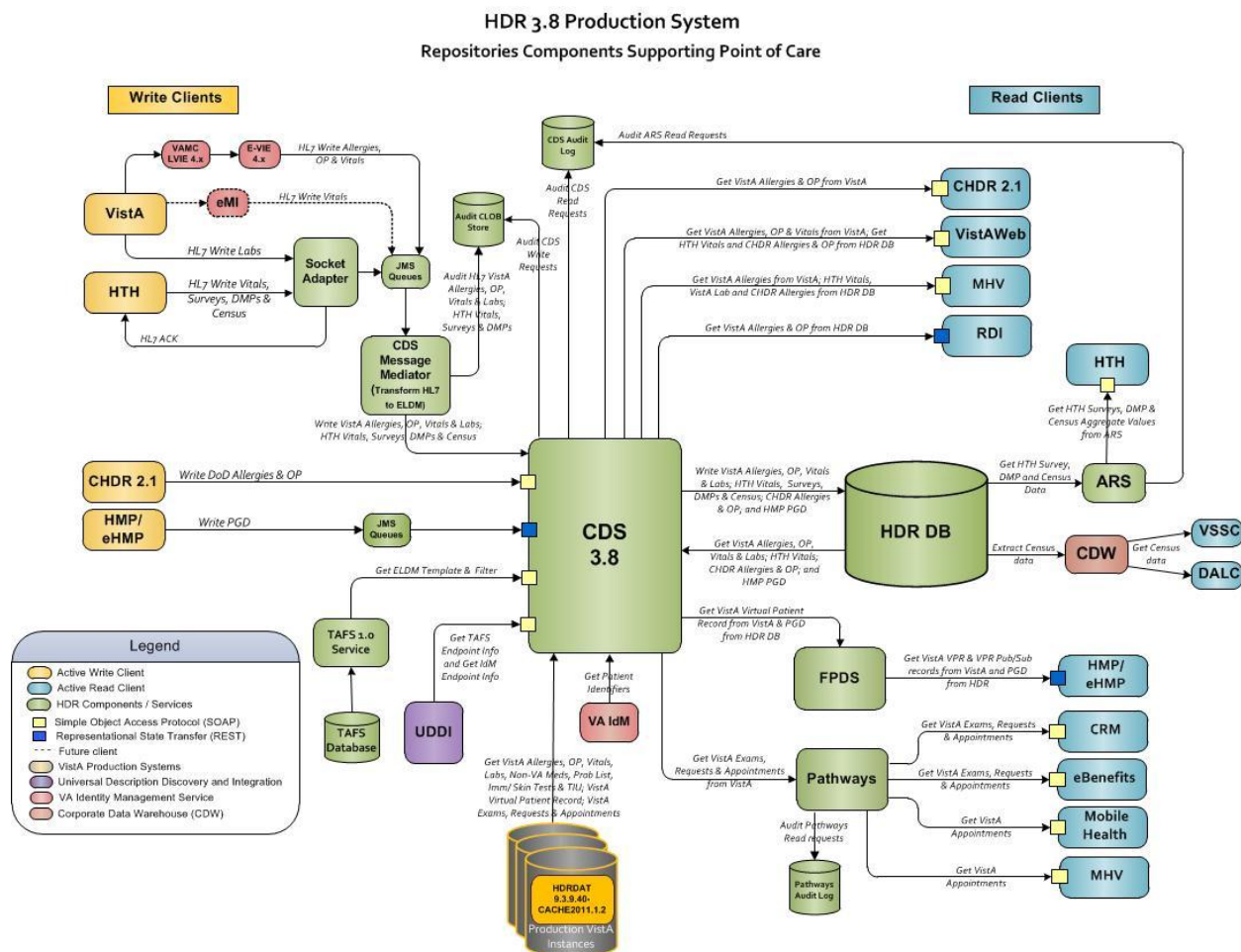
The CDS framework data interchange and data processing is based on XML data payloads. The XML data payload schemas, known as 'templates', follow the Enterprise Logical Data Model (ELDM) where available and can be used on both requests and responses to specify data domain specific detail. The CDS framework Read interface accepts XML data, known as 'filters', that

are transformed into query parameters for filtering data from the various data sources. The templates and filters are stored in a Template and Filter Schema in the HDR DB and can be accessed through a Template and Filter Service (TAFS). The CDS framework uses TAFS to access the templates and filters during validation and response generation processes.

2.1. Overview of the System

The HDR system is comprised of many services and interfaces, each of which was created to meet the requirements of specific clients. See Figure 1 for the various services and clients that interact with the system.

Figure 1 HDR 3.8 Clients and Interactions



2.1.1. CDS

The CDS service is a SOAP and REST-based web service interface that supports Create, Retrieve, Update, and Delete (CRUD) operations against HDR data stores over secure Hypertext Transfer Protocol (HTTPS). The CDS service provides clinical data from the HDR DB, VistA or both, in XML format. CDS interfaces are generated directly from the CDS framework

components using Apache CXF. CDS is accessed by Home TeleHealth, HMP/eHMP, CHDR, RDI, VistAWeb and MHV for reading and writing clinical data.

2.1.2. Pathways

The Pathways service is another SOAP/REST web service interface accessed via HTTPS that provides administrative data from VistA in XML format. The Pathways service interface is derived directly from the CDS framework components using Apache CXF. Customer Relationship Management (CRM), eBenefits, Mobile Health, and MHV use Pathways to obtain patient Appointments and Exam Request and Exam Status-related information from VistA.

2.1.3. FPDS

The FPDS is a REST web service accessed via HTTPS that supports reading Virtual Patient Record (VPR) data from Vista, HMP/eHMP subscription, cancellation and patient data fetches from VistA, and storage and retrieval of PGD data in the HDR. The FPDS interface supports JSON payloads and transforms incoming JSON requests to XML templates and filters before delegating the requests to the CDS framework for further processing. HDR 3.8 will add support for enhanced querying that specifies the amount of data to retrieve in one call and also allows the VPR to obtain updates to patient data without running the query again.

The FPDS supports asynchronous writes via JMS queues that are used for processing requests and acknowledging receipt of PGD data. Acknowledgements for the asynchronous Write requests are sent back to the calling application through either a request specified Reply-To destination or the default JMS response queue. The FPDS Message Driven Bean (MDB) will be configured to watch the JMS queue for new requests. In the event of a new message, the MDB transforms the JSON request data into XML and delegates the processing of the Write request to the CDS framework for storage in the HDR. HMP/eHMP/eHMP/eHMP is currently the only client that makes use of FPDS for accessing VPR data and will remain the only client for storing and retrieving PGD from HDR DB.

2.1.4. Aggregate Read Service (ARS)

The ARS is a SOAP web service exposed over HTTPS that provides an aggregated (report) view of HTH data stored in the HDR DB.

2.1.5. CDS Message Mediator

The CDS Message Mediator facilitates the processing of HL7 messages sent to the HDR by transforming the HL7 data into VIM XML data that the CDS service and framework can process, delegates the request to the CDS framework, and responding to clients in cases where acknowledgement of receipt of messages is required. The CDS Message Mediator is currently used by clients to perform asynchronous create operations. The CDS Message Mediator is a Java EE application that utilizes the JMS facilities of the application server to facilitate asynchronous communications. Messages are sent to the CDS Message Mediator by VistA by way of the Vitria interface engine infrastructure or through the HDR Socket Adapter (described below).

2.1.6. Socket Adapter

The HDR Socket Adapter application is a standalone Java application that accepts Transmission Control Protocol (TCP) connections and transmission of HL7 data from HL7 Minimal Lower Layer Protocol (MLLP) clients, verifies the message source and type, and then forwards the

message to the CDS Message Mediator for further processing. Currently, there are two clients sending HL7 to the HDR via the HDR Socket Adapter: HTH and VistA Lab Chemistry and Hematology.

2.2. Overview of the Business Process

The Create, Update, and Delete (CUD) Process illustrated in Figure 2 below, together with the Read Process illustrated in Figure 3, form the backbone of the CDS framework in the HDR 3.8 system. Different aspects of this framework are exposed through different interfaces by the services.

All aspects of this framework (CRUD) are exposed by the CDS 3.8 service through a stateless synchronous interface. The Create and Update aspects of the framework are also exposed through an asynchronous interface on the CDS 3.8 service via the CDS Message Mediator and the planned 3.8 extensions of the FPDS service. The Pathways service and FPDS service expose the Read aspects of the framework through a stateless synchronous interface.

HDR 3.8 adds a data extract process as illustrated in Figure 8. The data extract aspects are available directly on the HDR database. Stored procedures that execute SQL statements on HDR tables are exposed to users with specific roles. The Corporate Data Warehouse (CDW) uses the extract stored procedures for extracting data from specific tables in HDR. The user role determines the tables that can be extracted from HDR by the CDW user.

2.2.1. Create, Update, and Delete Process

The Write operations on the HDR 3.8 system break down to CUD operations. All of the Write operations follow a similar flow as shown in Figure 2. Figure 2 also shows the HDR 3.8 Write processing that creates an XML write template wrapper around the JavaScript Object Notation (JSON) payload on FPDS as well as the transformation of the XML acknowledgement or error message to JSON. The main aspects of the Write operations flow include the following steps noted in Table 3:

Table 3 Create, Update, and Delete Process

Step	Description	Alternate Process
1	Once the HDR 3.8 system receives the Read request synchronously through the web service, the application determines whether transformations are required prior to processing the message. If transformations of the request are required, the appropriate transformations are applied to the incoming request.	An unsuccessful transformation causes an error to be logged to the HDR DB and an error response is returned to the client that initiated the request indicating the cause of error.
2	Upon successful auditing, the system loads the filter schema that corresponds to the filter ID specified in the Read request from a local filter cache managed by the application and validates the filter information in the request against the filter schema.	If the filter does not exist in the filter schema, the filter is loaded from the TAFS. CDS uses a Universal Description, Discovery, and Integration (UDDI) registry to locate the TAFS service. TAFS is invoked with the filter ID to load the filter into the application filter cache. If the filter schema could not be located in TAFS or if the TAFS web

Step	Description	Alternate Process
		service is unavailable, an error is logged in the HDR DB and an error response is returned to the client that initiated the request indicating the cause of the error.
3	After the request filter has been validated against the filter schema, Patient information is retrieved from the filter. If the filter contains a resolvable national identifier, the VA Identity Management service is invoked to get the identifiers associated with the national Patient ID.	If there are no patient identifiers in the request or if the UDDI or IDM services are unavailable, or if the Patient's national identifier is not known by the VA IDM Service, an error is logged in the HDR DB and an error response is returned to the client that initiated the request indicating the cause of the error.
4	Once all patient identifiers have been resolved by the system, the requested data type requests along with these identities are utilized to determine the sources that should be queried to retrieve the requested patient data.	If no data sources are available for the requested data type or patient, an error is logged in the HDR DB and an error response is returned to the client that initiated the request indicating that no data could be found.
5	Once data sources are determined, connections are made to each source in parallel to retrieve the information relevant to the request.	If a connection to one or more of the sites could not be made, an error is logged in the HDR DB and an error is added to the response indicating that the response is incomplete.
6	Once all queries have completed, the response template is loaded from the internal application template cache and the results from all data sources are aggregated and sequenced per the template schema to form a single response.	If UDDI or TAFS services are unavailable, an error response is returned to the client that initiated the request indicating that the response could not be created. If the response template schema corresponding to the request could not be loaded from TAFS, an error is logged in the HDR DB and an error response is returned to the client.
7	After the response message has been created, information about the response is audited to the HDR DB.	If auditing of the request fails, an error is logged in the HDR DB and an error response is returned to the client.
8	If further transformation of the request is required, the appropriate transformations are applied to the response message prior to returning the response message back to the client.	If the transformation fails, an error is logged in the HDR DB and an error response is returned to the client indicating a transformation error.

The CDS 3.8 Read request processing is illustrated in Figure 3. The details of the processing and the processing requirements remain the same as CDS 3.7. The CDS 3.8 read support consists of a

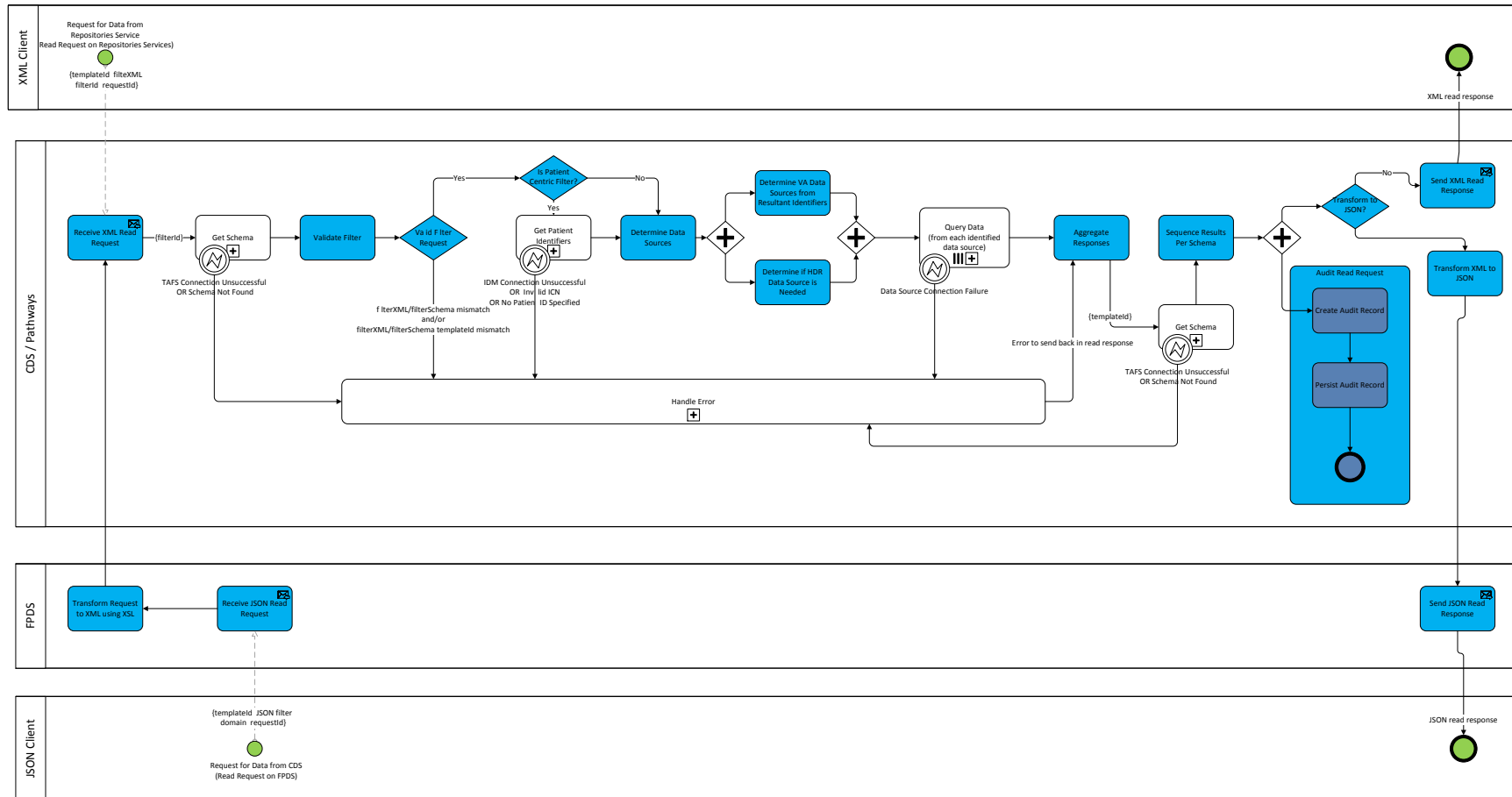
synchronous, stateless web service interface that uses information provided in the Read request to identify the data to be sent back to the client in the response.

Table 4 Read Request Process

Step	Description	Alternate Process
1	Once the HDR 3.8 system receives the Read request synchronously through the web service, the application determines whether transformations are required prior to processing the message. If transformations of the request are required, the appropriate transformations are applied to the incoming request.	An unsuccessful transformation causes an error to be logged to the HDR DB and an error response is returned to the client that initiated the request indicating the cause of error.
2	Upon successful auditing, the system loads the filter schema that corresponds to the filter ID specified in the Read request from a local filter cache managed by the application and validates the filter information in the request against the filter schema.	If the filter does not exist in the filter schema, the filter is loaded from the TAFS. CDS uses a UDDI registry to locate the TAFS service. TAFS is invoked with the filter ID to load the filter into the application filter cache. If the filter schema could not be located in TAFS or if the TAFS web service is unavailable, an error is logged in the HDR DB and an error response is returned to the client that initiated the request indicating the cause of the error.
3	After the request filter has been validated against the filter schema, Patient information is retrieved from the filter. If the filter contains a resolvable national identifier, the VA Identity Management service is invoked to get the identifiers associated with the national Patient ID.	If there are no patient identifiers in the request or if the UDDI or IDM services are unavailable, or if the Patient's national identifier is not known by the VA IDM Service, an error is logged in the HDR DB and an error response is returned to the client that initiated the request indicating the cause of the error.
4	Once all patient identifiers have been resolved by the system, the requested data type requests along with these identities are utilized to determine the sources that should be queried to retrieve the requested patient data.	If no data sources are available for the requested data type or patient, an error is logged in the HDR DB and an error response is returned to the client that initiated the request indicating that no data could be found.
5	Once data sources are determined, connections are made to each source in parallel to retrieve the information relevant to the request.	If a connection to one or more of the sites could not be made, an error is logged in the HDR DB and an error is added to the response indicating that the response is incomplete.

Step	Description	Alternate Process
6	Once all queries have completed, the response template is loaded from the internal application template cache and the results from all data sources are aggregated and sequenced per the template schema to form a single response.	If UDDI or TAFS services are unavailable, an error response is returned to the client that initiated the request indicating that the response could not be created. If the response template schema corresponding to the request could not be loaded from TAFS, an error is logged in the HDR DB and an error response is returned to the client
7	After the response message has been created, information about the response is audited to the HDR DB.	If auditing of the request fails, an error is logged in the HDR DB and an error response is returned to the client
8	If further transformation of the request is required, the appropriate transformations are applied to the response message prior to returning the response message back to the client.	If the transformation fails, an error is logged in the HDR DB and an error response is returned to the client indicating a transformation error.

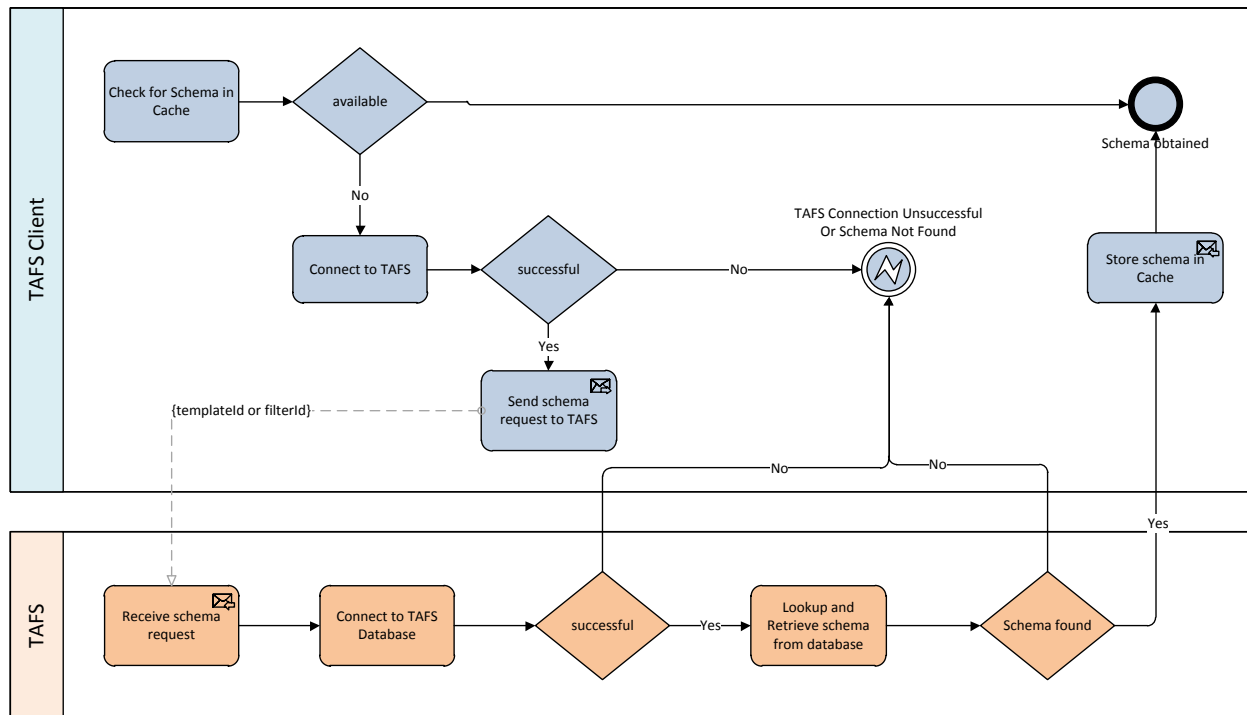
Figure 3 Read Process Flow



2.2.3. Get Schema Process

The ‘Get’ Schema is a sub-process used in both CUD and Read operations. Figure 4 illustrates the process which spans both the TAFS service and its clients (CDS and Pathways). The process includes checking for the schema in the parent cache and only calling out to the TAFS service if the schema does not exist in the cache as seen in Figure 4. Errors that occur due to unsuccessful connection to the service or inability to locate the schema are handled by the parent process.

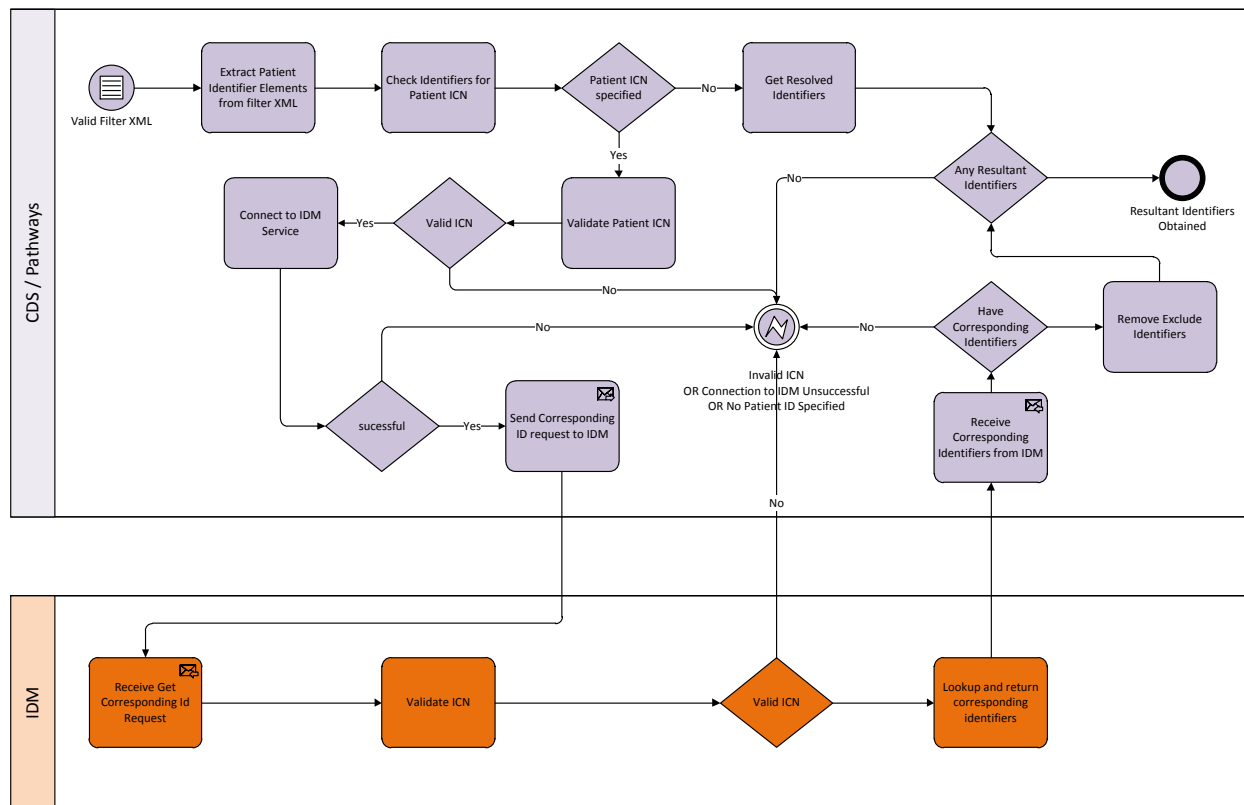
Figure 4 Get Schema Process Flow



2.2.4. Get Patient Identifiers

The ‘Get Patient Identifiers’ illustrated in Figure 5, is a sub-process of the Read process that extracts identifiers from the request filter. In case an ICN is specified in the filter, the VA IdM service is used for obtaining corresponding identifiers. Any exclusions specified by the client in the filter are removed from the corresponding identifiers list. This process returns “resultant identifiers” that are used to determine the sources of data and for querying data from the data sources.

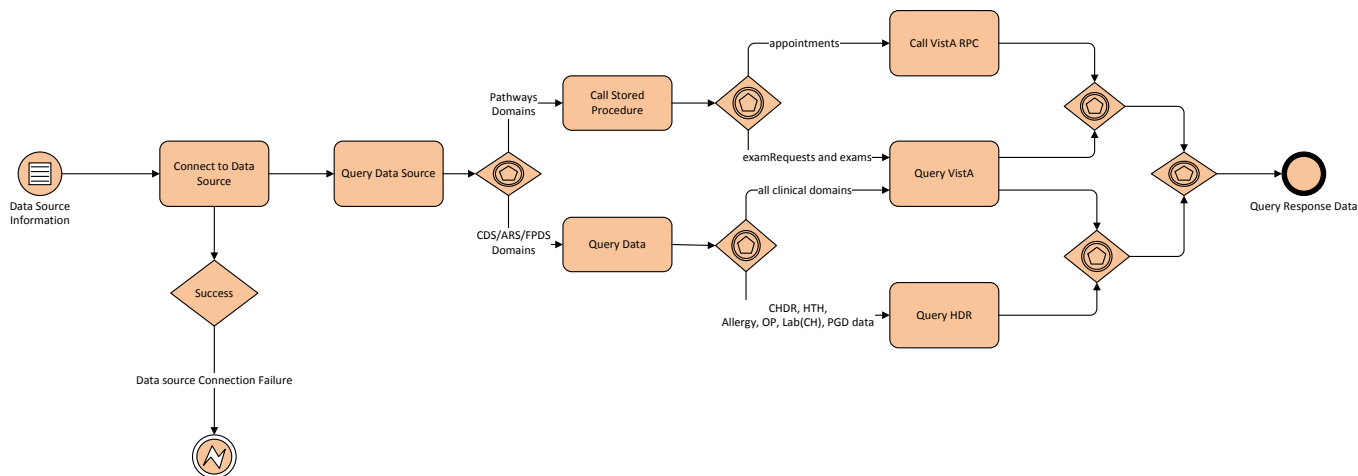
Figure 5 Get Patient Identifiers Process



2.2.5. Query Data

Query data is another sub-process of the Read process and is illustrated below in Figure 6. Based on the resultant identifiers and the data that needs to be obtained, appropriate connections are made to the various data source in parallel to retrieve the required information. Depending on the data domain, either a stored procedure or an SQL query is executed and the various responses from all the databases queried are returned to the Read process for aggregation, sequencing and transformation.

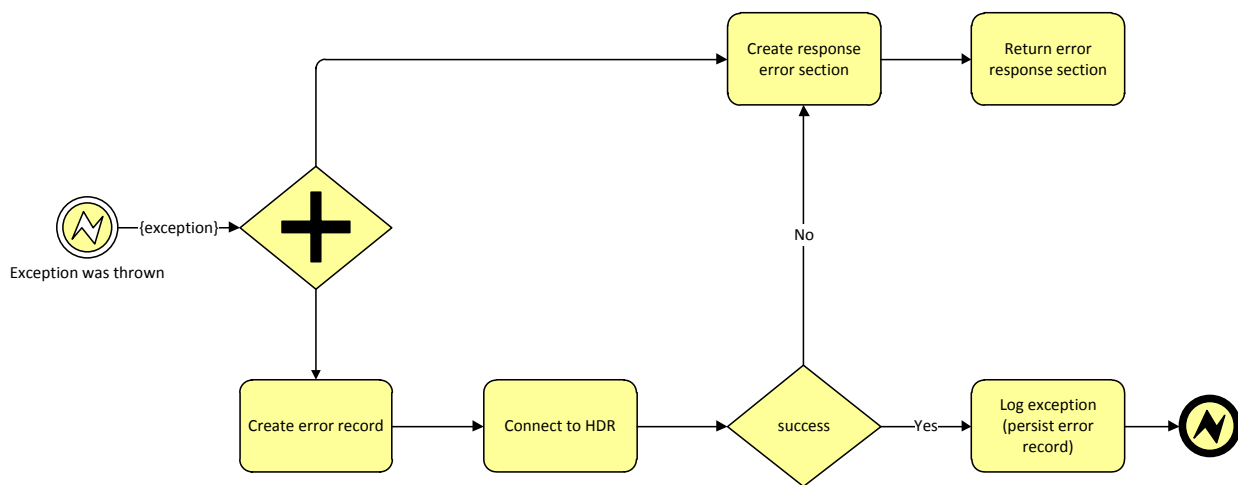
Figure 6 Query Data Process



2.2.6. Handle Errors

Figure 7 illustrates the Handle Error, which is an over-arching process that is used by all processes, except audit processing, to log any exceptions that occur during processing and generate an appropriate error response element or document to be included in the response. See Figure 7 for details of the error handling process.

Figure 7 Handle Error Process



2.2.7. Data Extract Process

The data extract process provides the ability for CDW to extract and copy the data from specific HDR tables. Clients requiring HDR data for analytics purposes should then connect to CDW for executing analytic queries and obtain reports. User roles in HDR database determine the HDR stored procedures that the CDW user will be able to access and execute. Figure 8 shows the details of the implementation and use data extraction process. A stored procedure is created in

HDR for extracting the data from the appropriate HDR tables. User roles that allow access to the stored procedure are created. CDW schedules data extraction and storage execution for extracting and storing the data in CDW using the HDR stored procedure. Clients that require the data, login to CDW and execute queries on CDW tables to meet their data analytics and reporting requirements.

Table 5 Data Extract Process

Step	Description	Alternate Process
1	Once the HDR database development team receives the requirements for analytics data requirements, the team determines the appropriate SQL to develop and creates a stored procedure that can be used for extracting the data.	If the stored procedure already exists, no further action is taken by the HDR database development team.
2	HDR database administrator creates a user with a user role that has permissions to access and execute the stored procedure.	If the user and user role already exist, no further action is taken
3	The stored procedure and user are tested and deployed to production.	
4	CDW users login to HDR database and execute the stored procedure for extracting the data and storing it into CDW tables.	
5	Clients requiring the data, login to CDW and execute analytic queries.	

Figure 8: Data Extract Process

2.3. Business Benefits

HDR, together with its services, provides data services as defined by VA's evolving Service Oriented Architecture (SOA) and can be traced to the data service layer of VA's enterprise architecture. HDR 3.8 follows the HDR application established data service pattern and offers the capability to:

- Store and retrieve data, specifically in HDR 3.8, any type of VPR data and patient generated data
 - Support for querying is enhanced to specify the amount of data to retrieve in one call and also allows the VPR to obtain updates to patient data without running the query again.
- Report on data stored in HDR, specifically in HDR 3.8, reports required by HTH for VSSC and DALC support
- Continue to provide a standard service interface for easy integration
- Audit and log all requests
- Integrate with Identity Management Service for Patient correlation
- Use payloads that adhere to the enterprise standards and thus establish an enterprise wide standard schema for all supported clinical and non-clinical data domains

- Monitor health of its servers to ensure continuity of operations and meet established service level agreements for every request
- Provide interfaces that accept VA enterprise supported data formats such as HL7, XML and JSON
- Adhere to modular component-based architecture that promotes reuse

2.4. Assumptions and Constraints

2.4.1. Design Assumptions

The HDR 3.8 system assumptions include the following:

- All requirements are valid and are as stated in the RSDs identified in Section 1.6
- All functional and non-functional requirements that were part of the previous revisions of the HDR system apply to HDR 3.8
- The new features have to be implemented within the existing framework and reuse HDR framework components where feasible

2.4.2. Design Constraints

Design constraints are subject to the enterprise VA Technical Reference Model and Standards Profile (TRM/SP), which is a guide for the use of tools and programming languages. The TRM/SP includes guidance and policy regarding operating systems, database servers, and application servers. The PD ProPath process provides guidelines for software development. Any variance from these guidelines shall be approved by appropriate waiver.

2.4.3. Design Trade-offs

- This HDR 3.8 version of the system will continue making use of materialized views that are created and populated / refreshed during off-peak hours to perform calculations and generate reports in support of HTH reporting requirements in order to mitigate the impact of calculations load. HDR 3.8 will extend this capability in support of the new HTH requirements.
- HDR 3.8 will make use of the updated VPR API that is enhanced to specify the amount of data to retrieve in one call and also allows the VPR to obtain updates to patient data without running the query again. This provides the ability to restrict the number of records retrieved from the VPR for a patient. This will reduce load impacts to both the servers and the network by allowing the size of the payloads to be restricted.
- HDR 3.8 will continue to support the use of the non-enhanced VPR API which does not have the ability to specify the amount of data to be retrieved in one call. At the HDRDAT layer, prior to calling the VPR API, the HDRDAT VPR stored procedure will inspect the VistA instance to determine the VPR API that is supported at the site and use the API that is available to get the VPR data from VistA. This parallel support of both the new and the existing VPR API will allow HDR to support HMP/eHMP at all VA sites irrespective of the version of the VPR API that is deployed in VistA. This also allows VistA systems to update and release the new VPR API in a planned release.

- HDR 3.8 version will continue to make use of application level JMS modules to reduce complexities of manual JMS configurations in multiple environments. This will also allow the JMS module to be packaged with the corresponding Message Driven Bean (MDB), thus ensuring that the required resources are always available and accessible from within the application context.
- HDR 3.8 will continue storing PGD data in CLOB structure as JSON to isolate HDR from changes to the internal PGD data structure changes. While this reduces the ability to perform fine grained data retrieval on PGD data, the reduction in time to integrate with HDR for new patent generated data is of higher significance at this stage for the VA enterprise.

The rest of the design of HDR 3.8 remains the same as the previous revisions of HDR 3.x.

2.5. Overview of Significant Requirements

2.5.1. Overview of Significant Functional Requirements

The requirements listed in Table 5 represent the high level functionality included in the HDR 3.8 release. HDR 3.8 Baseline Requirements document includes all other baseline requirements that are applicable to clients. Refer to the respective HDR 3.8 HMP and HTH RSDs referenced in Section 1.6 for specific functional requirements.

Table 5 Functional Requirements

Requirement
Extend HMP/eHMP support to add REST service interface to subscribe and cancel subscriptions to patient VPR and to fetch VPR data from a site for subscribed patients using the VPR 1.2 API implemented by the VistA team and extend the HDRDAT stored procedure wrapper to call the VPR API that is deployed at the VAMC until all VistA systems have the most recent version of the VPR API.
Extend Home Telehealth (HTH) Data Extracts to include a new interface to feed VHA Support Service Center (VSSC) and Denver Acquisition & Logistics Center (DALC) via Corporate Data Warehouse (CDW).

2.5.2. Overview of Functional Workload / Performance Requirements

RDI is the only HDR client that has a specific performance requirement and expects to receive responses to data requests from CDS within 10 seconds. The following in Table 6 are standard performance requirements that apply to all releases of HDR.

Table 6 Workload and Performance Requirements

ID	Requirement
PERF 01	The system shall respond to all requests for HDR data with an average response time of 3 seconds
PERF 02	The system shall support 800 requests per second

2.5.3. Overview of Operational Requirements

The Repositories direction is to store and read data from HDR. Clients requiring data from VistA which is not currently available in HDR, may access the VistA data through HDR services for

data domains supported by the HDR services. The HDR database and hardware procurements have been approved by Enterprise Infrastructure Engineering (EIE). Table 7 outlines the operational requirements.

Table 7 Operational Requirements

ID	Requirement
OPS 01	Availability – dependent on availability of AITC – service is available when all servers in AITC are available and operational
OPS 02	Resource Utilization – Memory and CPU usage will vary based on the number of simultaneous requests processed and will not exceed 80% of the capacity on each application and database servers on which the Repositories application is deployed
OPS 3	Growth – Annual growth figures from Introscope can be provided by AITC.

2.5.4. Overview of Technical Requirements

In the 3.8 release, HMP/eHMP/eHMP/eHMP and HTH support shall comply with established Repositories guidelines with regard to development, implementation and release constraints, data constraints as well as database design and content constraints. Refer to Section 4.5 Enterprise Architecture.

Other technical requirements are as listed in Table 8 below.

Table 8 Technical Requirements

ID	Requirement
TR 01	Provide a web service interface for all data access
TR 02	Support asynchronous write requests
TR 03	VSSC and DALC will read census extract data from Corporate Data Warehouse (CDW)
TR 04	Web service interfaces will support XML payloads by default
TR 05	HMP/eHMP payloads shall be JSON objects, that shall remain opaque to Repositories processing

2.5.5. Overview of Security or Privacy Requirements

HDR production platforms run entirely within the Austin Information Technology Center (AITC) environment, and as such there are no specific security requirements for the interface. HDR services and interfaces can be accessed from within the VA network.

Table 9 Security Requirements

ID	Requirement
	Not Applicable.

2.5.6. Overview of System Criticality and High Availability Requirements

The Repositories services are used by RDI for performing order checks when prescribing medications to VA patients. Thus, the system has high availability requirements that are met through the use of application and database clusters. Additionally, Disaster Recovery (DR) servers at Hines Information Technology Center (HITC) ensure that there is continuity of operations in the case of a disaster. In order to meet RPO/RTO of 15 minutes / 1 hour, the

database servers in the disaster recovery environment at HITC are updated instantaneously as they are configured as ‘active’ hot stand-by nodes, thus ensuring minimal loss of data.

2.5.7. Single Sign-on Requirement

Not Applicable. HDR functions as a back end system and does not include any user interfaces.

2.5.8. Requirement for Use of Enterprise Portals

Not Applicable.

2.5.9. Special Device Requirements

Not Applicable.

2.6. Legacy System Requirement

HDR implements the functional requirements to support a single deployed HDR system based primarily on the architecture outlined in previous sections.

Table 10 Proposed Legacy Retirements

Legacy System or Legacy System Component	System Retired or Workload Reduced	Quantify the Workload Reduction
Not applicable.		

3. Conceptual Design

3.1. Conceptual Application Design

This section provides the conceptual design of the application that is being produced by this project.

3.1.1. Application Context

Figure 9 represents the context in which the HDR 3.x application will exist. The green components shown in Figure 8 represent the HDR 3.x system. Refer to Tables 11, 12, and 13 which describe the components of the figure below.

Figure 9 HDR 3.8 Proposed Production System

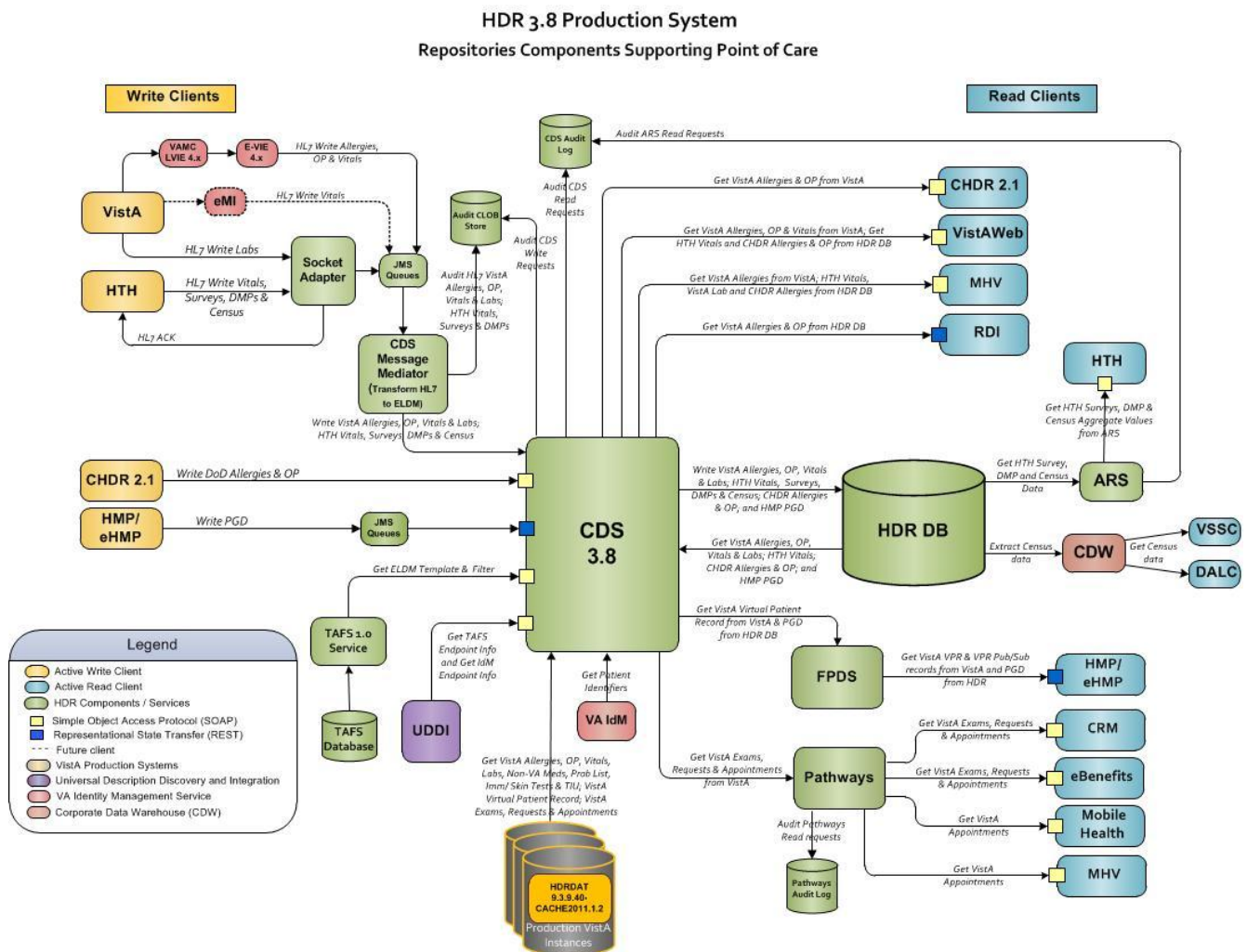


Table 11 Object

Name	Description	Interface Name	Interface System
Write Clients			
Synchronous CDS Clients – XML Messages			
CHDR	CHDR sends allergy and outpatient pharmacy data over a web service interface for active dual consumers to HDR 3.x system.	createOrUpdateClinicalData	CDS
Synchronous CDS Framework Clients – XML Messages			
MessageMediator	MessageMediator reads HL7 messages off of its queues, transforms them to XML format using XSL and calls CDS framework for storing the data in HDR DB. Responses from CDS are placed on a specified response queue or on a default 'reply-to' queue.	createOrUpdateClinicalData	CDS
Asynchronous FPDS Clients – JSON Messages			
HMP/eHMP	HMP/eHMP sends PGD to HDR 3.x system in JSON format asynchronously through JMS modules defined on its FPDS service interface. Success indicator or an error message in case of failures during the storage operation on HDR 3.x will be sent back to HMP/eHMP over preconfigured 'reply-to' queue.	send (see JMS API for creating client connections to JMS Qs)	JMS - FPDS HMP/eHMP Q
MessageMediator Clients – HL7 Messages			
SocketAdapter	SocketAdapter places the messages it receives on an appropriate MessageMediator Queue for processing and storage	send (see JMS API for creating client connections to JMS Qs)	JMS - MessageMediator Q
VistA/VIE	Allergy and Outpatient Pharmacy data from VistA is transmitted to VIE that then places the HL7 messages on HDR 3.x JMS queues for processing and storage in HDR DB	send (see JMS API for creating client connections to JMS Qs)	JMS - MessageMediator Q
Socket Adapter Clients – HL7 Messages (TCP/IP Socket connections)			
HTH	HTH uses HDR 3.x TCP/IP socket interface to send Vital signs data from HTH devices, Survey responses, DMPs and Census reports in HL7 message to HDR 3.x system. Acknowledgements from HDR 3.x will be sent back to HTH over preconfigured 'reply-to' queue.	(Non)BlockingConnection write (Non)BlockingConnection receive (See XSocket API documentation for details)	XSocket – SocketAdapter port

Name	Description	Interface Name	Interface System
VistA Lab	VistA Lab package uses HDR 3.x TCP/IP socket interface to transmit Lab Chemistry and Hematology Result messages in HL7 format to HDR 3.x system	NonBlockingConnection write (See XSocket API documentation for details)	XSocket – SocketAdapter port
Read Clients – Synchronous Clients			
ARS Clients – XML			
HTH	HTH accesses ARS web service interface using HTTP/SOAP for requesting and receiving aggregate reports on Survey Responses, DMPs and Census reports from HDR 3.x system.	readAggregateData	ARS
CDS Clients – XML			
CHDR	CHDR uses CDS 3.x web service interface on HTTP(s)/SOAP to request for and receive VistA allergies and Outpatient Pharmacy data for active dual consumers.	readClinicalData	CDS
MHV	MHV uses CDS 3.x web service interface on HTTP(s)/SOAP to request for and receive VistA allergies, HTH Vitals and Lab Chemistry and Hematology results data.	readClinicalData	CDS
RDI	RDI connects to CDS using HTTP/REST to obtain CHDR and VistA allergies and outpatient pharmacy data stored in HDR DB for drug checks during prescription ordering process.	readClinicalData	CDS
VW	VS uses CDS 3.x web service interface on HTTP(s)/SOAP to request for and receive CHDR allergies and HTH Vitals data.	readClinicalData	CDS
CDS Framework Clients – XML - POJO			
FPDS	FPDS transforms JSON requests to XML using XSL and delegates to the CDS framework for processing the read request. FPDS transforms the XML response from CDS framework to JSON before sending the response back to the client.	readClinicalData	CDS
Pathways	Pathways uses the CDS framework to extract non-clinical data from VistA.	readData	CDS
FPDS Clients – HTTP(s)/REST - JSON			
HMP/eHMP	HMP/eHMP uses FPDS REST interface to request for PGD Data from HDR and VPR data from VistA.	Get	FPDS

Name	Description	Interface Name	Interface System
Pathways Clients – HTTP(s)/SOAP - XML			
CRM	CRM uses Pathways SOAP web service interface to obtain patient Appointments as well as C&P Exam Request and Exam Status information in order to deter the status of their claims	readData	Pathways
eBenefits	eBenefits uses Pathways SOAP web service interface to provide upcoming Appointments as well as C&P Exam Request and Exam Status information to its users	readData	Pathways
MHV	MHV will use Pathways SOAP web service interface to notify clients of their upcoming Appointments	readData	Pathways
MobileHealth	MobileHealth uses Pathways SOAP web service interface to notify clients of their upcoming Appointments	readData	Pathways
TAFS Clients – HTTP(s)/SOAP - XML			
CDS	TAFS is an internal service used by CDS service to access schema files that represent request filters and response templates in its read and write processing for validating the payloads and requests and for generating the responses.	getSchema	TAFS
HDRDAT Clients – JDBC			
CDS	CDS uses HDRDAT, an SQL projection layer on VistA for extracting data from VistA. In addition to SQL tables that are SQL projections of VistA FileMan Files, HDRDAT also consists of extension tables that include data columns for improved cross-table performance and stored procedures that wrap up calls to VistA APIs.	HDRDAT	HDRDAT/VistA
Extract Client - SQL			
CDW	CDW will login and extract HTH census and survey data from HDR using SQL. Extracted data from HDR will be used by VSSC and DALC for analytic purposes. HDR will restrict access to CDW for specific tables.	Oracle	HDR

Table 12 Interfaces Internal to OIT

Name	Related Object	Input Messages	Output Messages	External Party
createOrUpdateClinicalData	CDS	Data Domain specific XML	Contains the unique active	CHDR

Name	Related Object	Input Messages	Output Messages	External Party
createClinicalData updateClinicalData		that consists of data specific for each domain such as allergy, vitals, census, PGD etc. The XML follows domain specific template schemas – see Appendix B for details on the templates	recordId from HDR DB against which the data is inserted or updated	HTH HMP/eHMP VistA
readAggregateData	ARS	Filter data that is used in SQL to filter and query the data sources – see Appendix A for details on the filters	Aggregate information on HTH surveys, census and DMPs stored in HDR DB	HTH
readClinicalData	CDS	Filter data that is used in SQL to filter and query the data sources – see Appendix A for details on the filters	Data domain specific XML that consists of data specific to the requested domains. The XML follows domain specific template schemas – see Appendix A for details on the templates	CHDR HTH MHV
readData	Pathways	Filter data that is used in SQL to filter and query the data sources – see Appendix A for details on the filters	Data domain specific XML that consists of data specific to the requested domains. The XML follows domain specific template schemas – see Appendix A for details on the templates	CRM eBenefits MobileHealth
GET	FPDS	Path and query parameters that are used to determine the data being requested	JSON object that consist of either VPR or PGD data as per the request	HMP/eHMP

Table 13 Externally Shared Data Stores

Name	Data Stored	Owner	Access
VistA	VistA deployed at 130+ VA sites stores all operational health	VAMC	Read

Name	Data Stored	Owner	Access
	and administrative data.		
VSSC	VA's Corporate Data Warehouse (CDW), which is SQL Server	CDW	Read for customers; Read/Write for HDR
DALC	VA's CDW, which is SQL Server	CDW	Read for customers; Read/Write for HDR

3.1.2. High-Level Application Design

Section 3.1.1 includes both the major as well as decomposed components of the HDR 3.x system.

3.1.3. Application Locations

All HDR components and database except for HDRDAT will be hosted at AITC. HDRDAT is hosted on the Cache systems that host VistA. HIDU push is used to push the HDRDAT changes to VistA systems at the Veterans Administration Medical Centers (VAMCs) in production. See Table 14 for details.

Table 14 Application Locations

Application Component	Description	Location at Which Component is Run	Type
HDRDAT	SQL Projection layer that includes relational representation of VistA FileMan files, extension tables that improve performance of cross-table joins and stored procedures that wrap VistA Application Programming Interface (API) calls.	VAMCs – InterSystems Cache	Data Layer
HDR DB	Oracle Relational Database that houses data from ‘write’ clients identified in Figure 8.	AITC – Oracle RAC	Data Layer
All components except for HDRDAT	Components that are comprised of the HDR 3.x core CDS framework and all services and interfaces that interact with CDS framework	AITC – Weblogic cluster	Service Layer Application Layer

3.1.3.1. Application Users

The HDR 3.x system does not have any direct human users. All access to its services and interfaces is through client applications. The Users interact with the client applications which in turn uses the HDR 3.x system to satisfy data requirements to meet the needs of the users. The User roles are determined by the client applications. The HDR 3.x system does not distinguish access based on user roles and determines the data to be returned based on the information specified in the request filters. Similarly, when storing data the system uses the information specified in the templates to identify the HDR DB table in which to store the data. Refer to Tables 11 – 13 which illustrate the CDS application use at current clients outlined in Section 1 Introduction.

3.2. Conceptual Data Design

3.2.1. Project Conceptual Data Model

The HDR 3.x data model consists of entities that are patterned after the enterprise logical data model (ELDM). Data domains represent subject areas and there are no relationships maintained in the model between subject areas. A majority of the data domains are de-normalized and all

data is captured in a single entity. For further information see *HDR 3.7 System Design Document* and other related documents.

3.2.2. Database Information

The HDR DB will add appropriate schemas for supporting HMP/eHMP and HTH clients as needed. The database inventory will be completed once the detailed storage requirements for HDR 3.8 have been finalized.

Table 15 Database Inventory

Database Name	Description	Type	Steward
HDR DB			HDR

3.2.3. User Interface Data Mapping

Not Applicable.

3.2.3.1. Application Screen Interface

Not Applicable.

3.2.3.2. Application Report Interface

Not Applicable.

3.2.3.3. Unmapped Data Element

Not Applicable.

3.3. Conceptual Infrastructure Design

3.3.1. System Criticality and High Availability

The HDR 3.x system and associated services are implemented as JEE Enterprise JavaBeans (EJB) that are exported as web services and are deployed in a WebLogic cluster environment. All operations supported by HDR services are stateless to reduce system complexity and resilience to system failure. High availability and scalability are provided through the use of WebLogic clustering technology.

Hardware network load balancers will manage connections to the WebLogic managed nodes of the cluster. It can further scale by adding more hardware and managed nodes to the WebLogic cluster. The cluster environment provides the ability to stop and restart one or more managed servers with little to no effect on the processing of requests.

HDR 3.x will use HDRDAT to access VistA data by means of Cache persistent objects that provide SQL access via JDBC connections. HDRDAT is installed on all of the VistA systems and runs on the Caché database server in Engineering Change Proposal (ECP) configurations.

HDR 3.x uses the HDR database, which utilizes Oracle Real Application Clusters (RAC) that provides high availability and scalability.

Disaster Recovery (DR) for the HDR 3.x system is provided at HITC and is a replica of the production system at the AITC with an RTO/RPO of 15 minute to 1 hour. SAN to SAN

replication is used for replicating production data from AITC to HITC with no data loss during transition/fail-overs/fail-backs.

3.3.2. Special Technology

Not Applicable.

3.3.3. Technology Locations

The CDS application is supported in the Development, Dev Preview, Performance and Load Testing, SQA, and Production environments. The Development environment is used by CDS developers for unit testing of the CDS application. Once tested here, the CDS application is deployed in the SQA and Dev Preview environments for client integration testing. The Developer systems in Salt Lake City (SLC) is comprised of the developer's laptop or Linux-based virtual desktop connected to WebLogic servers and database servers that are installed in a Linux environment. The virtual desktop environment is located in SLC along with the CDS XX deployed environment and VistA test accounts that are utilized by the CDS application. It is a non-cluster WebLogic environment in which the CDS application is deployed. The Development environment currently uses two VistA shadow instances. The virtualized desktops specifications are as follows:

- 2 CPU's
- 4 GB Memory
- 40 GB HDD space

Table 16 Technology Location Details

Technology Component	Location
Development VDT	SLC
Workstations	SLC
Special Hardware	SLC
VistA Shadow Test Accounts	SLC
HDR Database	SLC
Application Servers	SLC
Development Laptop	SLC/Remote
Workstations/Laptop	SLC/Remote
Special Hardware	N/A
VistA Shadow Test Accounts	SLC
HDR Database	Developer Laptop
Application Servers	Developer Laptop

After the Dev Preview, CDS is deployed in the SQA environment. This environment is used for system integration testing including the regression testing of CDS and its associated services/components. It is a WebLogic cluster environment in which the CDS application is deployed. This environment uses two copies of seven VistA accounts in two separate SQA environments. The WebLogic cluster environment contains one cluster with four managed nodes. The database servers utilize Oracle RAC software. This environment currently uses three VistA shadow instances.

The production environment is located at the AITC. Please see Section 3.3.4.2 Conceptual Production String Diagram, and the *HDR 3.8 Installation and Deployment Guide* for details.

3.3.4. Conceptual Infrastructure Diagram

3.3.4.1. Location of Environments and External Interfaces

Refer to Section 3.3.3 Technology Locations.

3.3.4.2. Conceptual Production String Diagram

Figure 10 in Section 4.3 illustrates a detailed production deployment environment showing an instance of the HDR 3.x system deployed to a WebLogic managed node within a cluster, and the cluster's relationship with the Oracle RAC and Cache cluster configurations.

4. System Architecture

4.1. Hardware Architecture

See Figure 10 for Hardware Details.

4.2. Software Architecture

The complete description of the software architecture of the HDR 3.x system and its subsystems for data requests and responses is detailed in Volumes 1 and 2 of this SDD. The current architecture does not vary from the previous releases of the system.

The specifics that will be added in HDR 3.8 include:

- FPDS will be enhanced to extend HMP/eHMP/eHMP/eHMP support to add a REST service interface to access the paging feature implemented in VPR API by the VistA team and extend the HDRDAT stored procedure wrapper to call the VPR API that is deployed at the VAMC until all VistA systems have the most recent version of the VPR API.
- FPDS will also be enhanced to support querying to specify the amount of data to retrieve in one call and also allow the VPR to obtain updates to patient data without running the query again.
- CDW will pull census extract data weekly and make it available to VSSC and DALC.

4.3. Network Architecture

Figure 10 illustrates the position of a Service Oriented Architecture (SOA) in the delivery of data to Repositories customers / clients. Repositories clients are not end users but applications that the Clinicians (end users) depend upon in delivering care to the Veterans. Centralized data centers are required as the Clinicians that use our clients' products are located across the entire VA wide area network (from Puerto Rico to Manila Philippines).

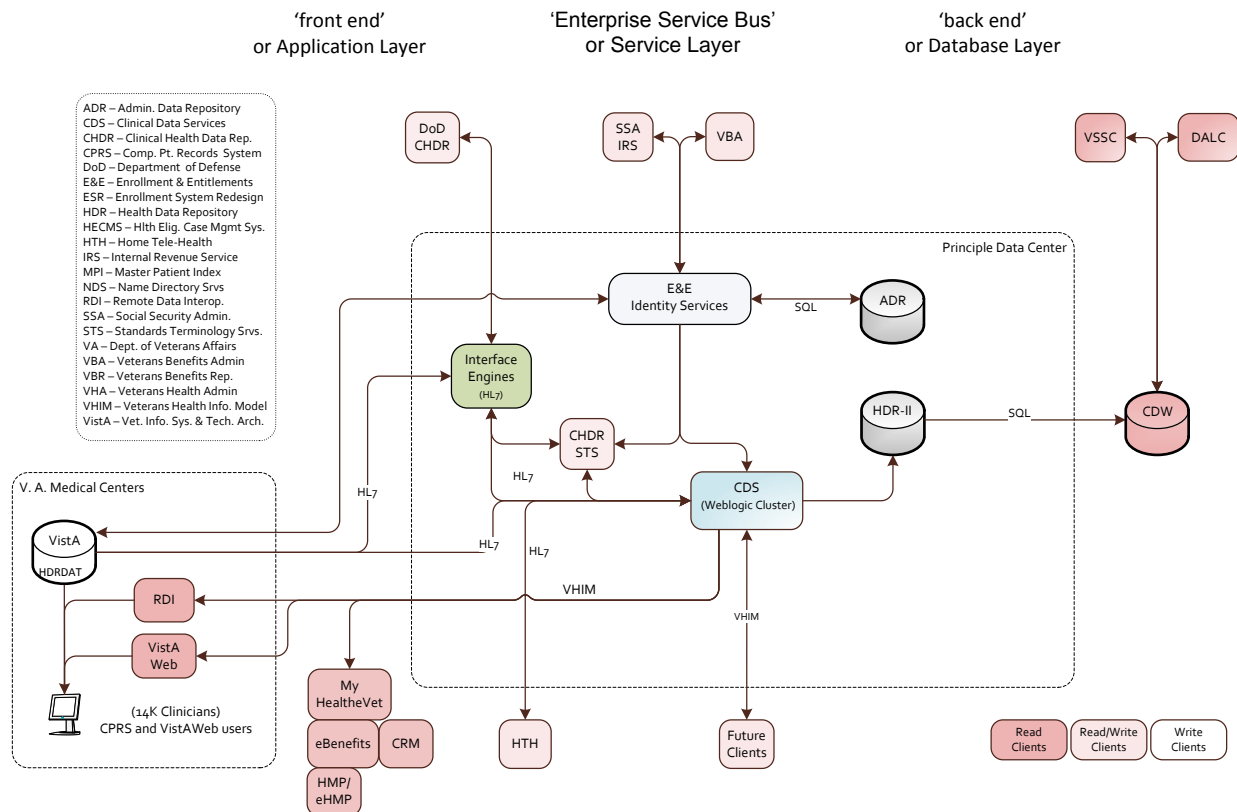
This illustration shows the RDI and VistA Web clients at the VA Medical Centers. This is because these applications are specifically for clinical use but they may actually reside in other locations.

The CDS application that provides an SOA is part of the HDR program. In HDR-CDS communications we show that HL7 is supported, however our primary service will change the HL7 communications into VIM or Veterans Health Information Model format. When a new client is brought on to HDR-CDS, they are provided the SOA model (in VIM) that their SOAP-based communications can read or write to. VistA is a noSQL database written in Mumps/Cache'. The HDRDAT interface gives HDR access to all of the VistA API's. Thus, HDR-CDS can provide SOA access to all VistA data.

The first Department of Defense (DoD) clinical records interoperation with the Veterans Health Administration was accomplished through CDS early in 2005.

CDW will login and extract HTH census and survey data from HDR using SQL. Extracted data from HDR will be used by VSSC and DALC for analytic purposes. HDR will restrict access to CDW for specific tables.

Figure 10 Repositories SOA

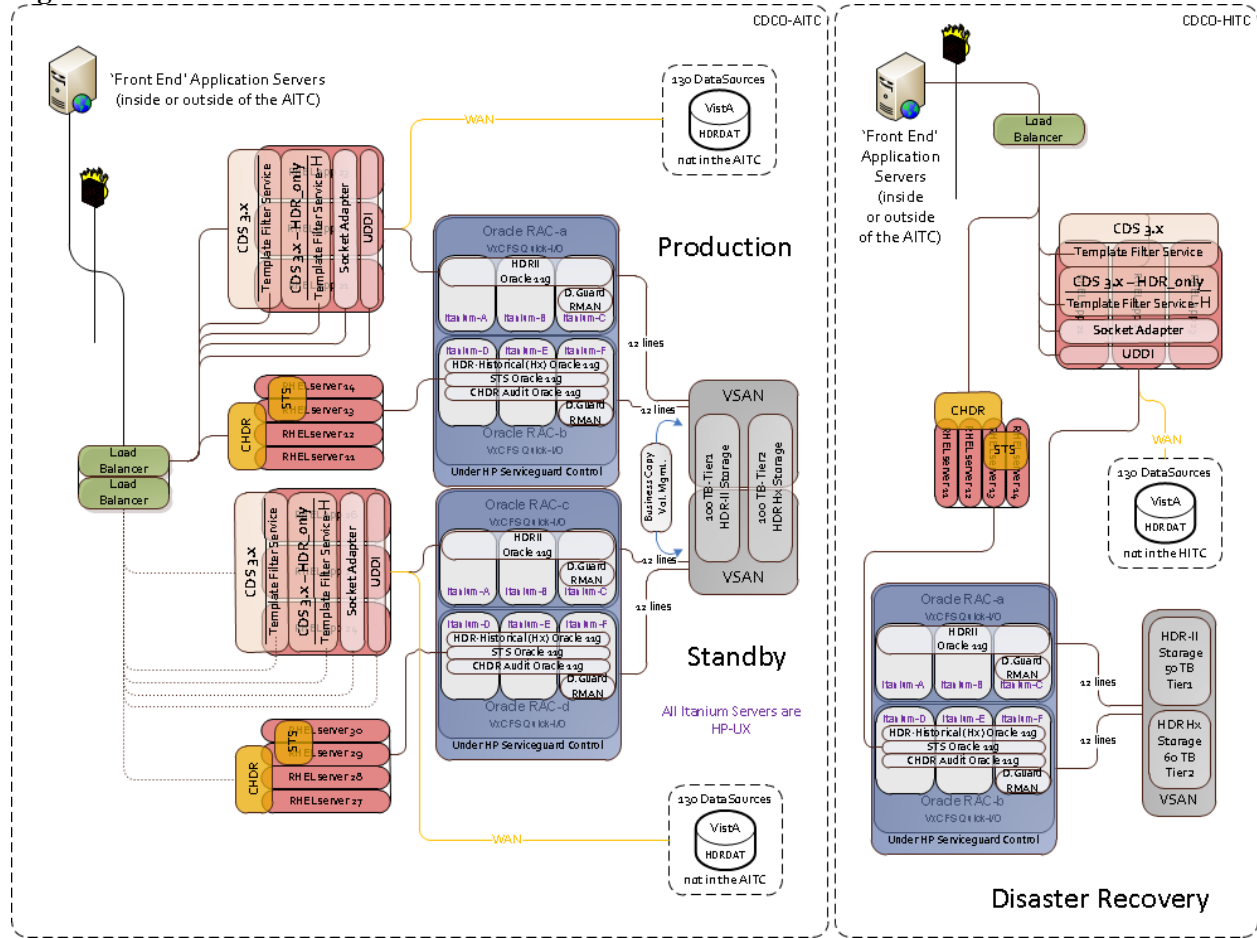


The HDR-CDS has three environments available for Production use. The Standby and Disaster Recovery environments are copies of the Production database. Due to the fact that these database copies are in a down state (no connections during replication); there is no active duplication of the CDS applications. All three environments are located in what we call the GSS or Generally Serviceable Systems (synonymous with production).

Figure 11 illustrates that the CHDR and STS programs use HDR hardware even though they are separate development teams.

The red Linux servers function as a cluster because of the design of the applications (many are on WebLogic). The blue systems are put together as Serviceguard clusters in HP-UX. On top of these hardware clusters is the Oracle Real Application Clusters (RAC's). The RAC-a uses tier 1 storage and the RAC-b uses tier 2 storage. These storage units (on a fibre channel Storage Area Network – FCSAN) have the capability of making rapid copies of the volumes (called Business Copy) at the AITC. If at any time the Standby copy gets too far out of sync, the storage unit can make a rapid snap of the Production volumes and they can be used as a new starting point for replication. There is also storage replication going on between the AITC and the HITC. The Disaster Recover copies of the databases are bit for bit copies of the Production databases (just not turned on).

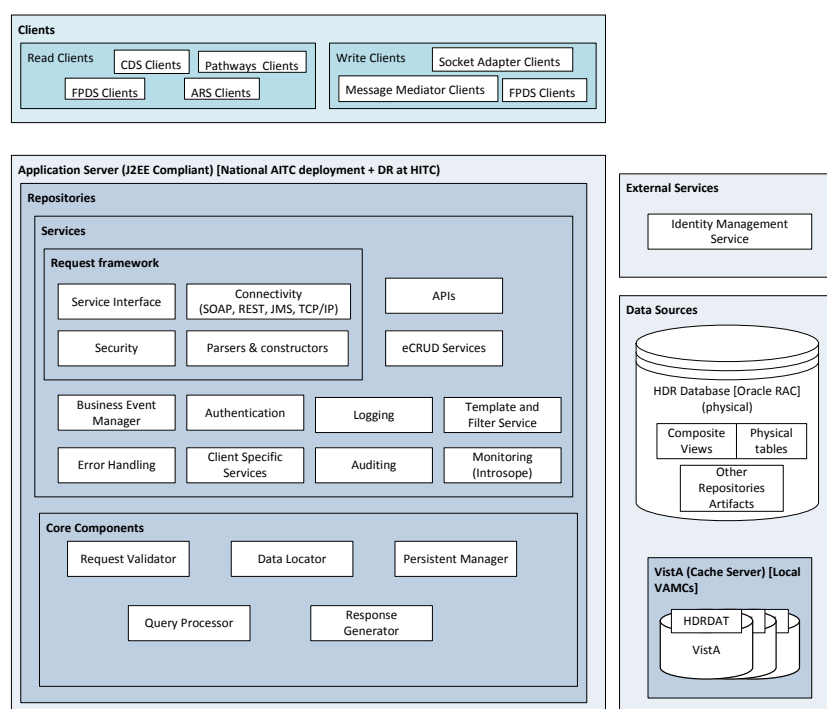
Figure 11 HDR GSS



4.4. Service Oriented Architecture / ESS

Repositories program architecture is an SOA that enables maintaining logical divisions in a complex system. It is designed and implemented using modular component-based principles that promotes loose coupling. Modules contain their own discreet and encapsulated purpose and interact with one another on a contract basis and can be maintained and modified independently of one another as long as the contract remains unchanged. HDR 3.8 continues this modular SOA architecture to realize the requirements outlined in section 2.5. Figure 12 below shows the architecture layers, modules and components along with the external services accessed by the Repositories components.

Figure 12: Repositories SOA Architecture



4.5. Enterprise Architecture

At its core, HDR 3.x system is a Java EE framework that takes advantage of Java Spring and dependency injection to extend and implement parallel components required to support the differing needs of its clients. The features the system provides can be accessed by its clients through various means as suited to the client, such as synchronous SOAP or REST service calls, asynchronous messaging through JMS and TCP/IP connections over sockets. The Java EE framework interacts with the various data sources using JDBC and Hibernate. Internal support for logging, auditing and error handling are implemented as services. Patient identity correlation, primarily used for identifying the VistA locations at which the patient was seen, is achieved using SOAP calls to the external Identity Management Service.

The HDR 3.x system adheres to the VA Enterprise Architecture (EA) and makes use of approved tools, standards and rules as identified by the VA Technical Reference Model (TRM) / Standards Profile (SP) located at <http://trm.oit.va.gov>. By using these standards, the programs take advantage of the current technologies that are adapted by VA and promotes interoperability, portability and adaptability within systems and promotes quality assurance. In addition to using SOA principles as defined by the VA EA team, HDR 3.x uses TRM approved tools such as WebLogic 12.x and Oracle 11g as its application server and database server respectively.

The HDR 3.x team continuously monitors TRM and upgrades the software, tools and libraries that are used to build the system on a regular basis to ensure compliance with VA EA. Table 17 describes the technologies used by HDR.

Table 17 HDR TRM Technologies / Tools

Tool Name	Version	Purpose	Comments/Actions	Approved	Approved w/Constraints	Version Upgrade Required
Another Tool for Language Recognition (ANTLR)	2.7.7	Language Parser for Java		X		
AOP Alliance	1.0	Aspect Oriented Programming library.	This is a transitive dependency of Spring AOP (Approved).		X	
Apache Commons BeanUtils	1.7.0	Common Utility Classes	Awaiting TRM Assessment Results; Upgrade to version 1.9.2, target to use latest version available			X
Apache Commons Codec	1.4	Common Utility Classes	Awaiting TRM Assessment Results; Upgrade to version 1.9, target to use latest version available			X
Apache Commons Collections	3.2, 3.2.1	Common Utility Classes	Awaiting TRM Assessment Results; Upgrade to version 4.0, target to target to use latest version available			X
Apache Commons Database connection pooling (DBCP)	1.2.2, 1.4	Common Utility Classes	Awaiting TRM Assessment Results; Upgrade to version 2.0.1, target to use latest version available			X
Apache Commons Digester	1.8	Common Utility Classes	Awaiting TRM Assessment Results; Upgrade to version 3.2, target to use latest version available			X

Tool Name	Version	Purpose	Comments/Actions	Approved	Approved w/Constraints	Version Upgrade Required
Apache Commons Lang	2.1	Common Utility Classes	Awaiting TRM Assessment Results; Upgrade to version 3.3.2, target to use latest version available	X		X
Apache Commons Logging	1.1, 1.1.1	Common Utility Classes	Awaiting TRM Assessment Results; Upgrade to version 1..2, target to use latest version available			X
Apache Commons Pool	1.3, 1.4	Common Utility Classes	Awaiting TRM Assessment Results; Upgrade to version 2.2, target to use latest version available			X
Apache CXF	2.7.3	Web Service Framework for Java	Upgrade to version 2.7.7		X	X
Apache MINA	2.0.0	Multipurpose Infrastructure for Network Application	Replaced xSocket	X		
Apache Scout	1.2.7	Implementation of JAXR API.		X		
Aspectjweaver	1.5.3, 1.6.8	Aspect Oriented Programming library.	Approved under AspectJ; Upgrade to version 7(1.7.x) or 8(1.8.x). Aspectjweaver is a component in versions compatible to AspectJ 7 or 8.	X		X
Attachmate Reflection	4.0.6	Windows SSH Client	Awaiting TRM Update Results (asking TRM list to include X Advantage product RXA); Replaced PuTTY, NoMachine RemoteDesktop, & xRDP	X		

Tool Name	Version	Purpose	Comments/Actions	Approved	Approved w/Constraints	Version Upgrade Required
Axis2	1.4	Web Services Engine	Upgrade to version 1.6	X		X
Byte Code Generation Library (CGLIB)	2.2	Code Generation Library	May delete as this is include in Spring Framework Upgrade to version 3.1		X	X
Cache	2014.1.1.702	JDBC Driver for Connecting to IS Cache.		X		
Cloverleaf Secure Courier	4.4	Test / Code Coverage			X	
Common Annotations for the Java Platform aka: jsr-250	2.4.0	Common provides annotations for common semantic concepts in the Java 2 Platform, Standard Edition (J2SE) and Java 2 Platform, Enterprise Edition (J2EE) platforms	Approved under Java EE; Upgrade to version 2.7 Also part of Java SE	X		X
dom4j	1.6.1, 1.6.1-PATCH01	Flexible XML Framework for Java		X		
e(Core)	2.4.0	Eclipse EMF Support	Approved under Eclipse Modeling Framework (EMF)	X		
Easymock	2.4, 3.1	Library for Generating Mock Test Objects	Ensure all upgraded to version 3.2	X		X

Tool Name	Version	Purpose	Comments/Actions	Approved	Approved w/Constraints	Version Upgrade Required
Eclipse Classic aka: Eclipse Kepler SR2	4.3	Integrated Development Environment	aka: Eclipse Kepler SR2	X		
Extensible Markup Language (XML) Schema Definition aka: xsd	2.2.2	Utility Library for Working with XML Schema Definitions	Approved under Eclipse Modeling Framework (EMF); Upgrade to version 2.9	X		X
Hermes JMS	1.14	Java Messaging Service API		X		
Hibernate Object/Relational Mapping (ORM)	3.2.0.Final 3.3.1.GA, 3.6.7.Final	Object Relational Mapping Library	Upgrade to version 4.3	X		X
HL7 Application Programming Interface (HAPI)	0.5.1	HL7 API for Java	TRM Approved 11/3/14, Upgrade to version 2.2	X		X
Java API for XML Registries (JAXR)	1.004	Java API for XML Registries	Approved under Java EE	X		
Java Architecture for XML Binding (JAXB)	2.0, 2.2.7	Java API for XML Binding	Approved under Java EE; Ensure upgrade to version 2.2.7	X		X
Java EE 6 aka: javaee-api	6.0	Java 2 Enterprise Edition API	Approved under Java EE; Upgrade to version 7	X		X

Tool Name	Version	Purpose	Comments/Actions	Approved	Approved w/Constraints	Version Upgrade Required
Java SE	1.6, 1.7	Java Development	Ensure all upgraded to version 2.7 (aka 1.7)		X	X
Java Transaction API (JTA)	1.0.1B	Java Transaction API (J2EE)	Approved under Java EE	X		
Javassist	3.18.0.GA	Bytecode editor			X	
Jenkins (Hudson) Continuous Integration Server	1.554	Continuous Integration	Upgrade to version 3.1.0	X		X
JetBrains Annotations	9.0	Java annotations support library.		X		
jsr311-api	1.1-ea	JAX-RS (RESTful Web Service) API	Approved under Java EE	X		
JUnit	4.1.10, 4.8.1, 4.11	Java unit test development and testing framework	Ensure all upgraded to version 4.11	X		X
LinkChecker	9.3	LinkChecker is a free, GPL licensed website validator. LinkChecker checks links in web documents or full websites.	Awaiting TRM Assessment Results; It runs on Python 2 systems, requiring Python 2.7.2 or later. Python 3 is not yet supported.			
Log4J	1.2.13, 1.2.16	Logging Utility for Java	Upgrade to version 1.2.17		X	X

Tool Name	Version	Purpose	Comments/Actions	Approved	Approved w/Constraints	Version Upgrade Required
Maven	3.0.3	Build Automation & Dependency Management	Need to change to version 3, 3.0.5, or 3.1.1. Version 3.0.3 is not listed in TRM.		X	X
Mockito	1.6.8	Mock Testing Framework	Upgrade to version 1.9.5	X		X
Nexus	2.10.0-02	Infrastructure and services for organizations that use repository managers to obtain and deliver software	Need version 2.2 or 2.7; version 2.10.0-02 is not listed. aka: Sonatype Nexus Replaced Archiva	X		X
ojdbc6	11.1.0.7.0	Oracle JDBC Client Library	Approved Under Oracle Database; Upgrade to version 11.2.x		X	X
Oracle Database aka: 11g r2	11.2.x	Relational Database Management System	aka: 11g r2		X	
Powermock	1.4.12	Java Class Mocking Framework	Upgrade to version 1.5. Supports and extends Easymock	X		X
Rational Team Concert	4.0	Source Control	Replaced Perforce	X		

Tool Name	Version	Purpose	Comments/Actions	Approved	Approved w/Constraints	Version Upgrade Required
Remote Server Administration Tools/Windows AdminPack aka: Windows Remote Desktop	Win7	Remote Desktop	Approved under Windows (Client)		X	
SAXON	9.1.0.2, 9.1.0.8	XSLT engine and Xquery processor	Upgrade to version 9.5	X		X
SLF4J (includes: slf4j-api, slf4j-log4j12)	1.5.8/1.5.0, 1.5.2	Simple Logging Façade for Java	Upgrade to version 1.7.0		X	X
Spring Framework	3.1.2.RELEASE	Comprehensive infrastructure supporting the development of Java Applications	Upgrade to version 4.0.5 or 4.0.6		X	X
Toad for Oracle	11.6	Database Development	Versions 10.0 and 11.0 Supported thru CY2015, Q4	X		
WebLogic Maven Plugin	10.3.4	Maven Plugin for Deploying J2EE Applications to WebLogic Server	Approved under Maven; Upgrade to version 12c		X	X
WebLogic Server	10.3.6, 12.1.2	J2EE Application Server	Upgrade to WebLogic Server version 12c (12.x).		X	X

Tool Name	Version	Purpose	Comments/Actions	Approved	Approved w/Constraints	Version Upgrade Required
WinSCP	4.2.x	FTP / SFTP Client for Windows	Upgrade to version 5.5.x		X	X
wlclient	10.3.5, 12.1.2	WebLogic Client Library	Approved under WebLogic Server; Ensure upgrade to version 12.1.2		X	X
wlfullclient	10.3.5, 12.1.2	WebLogic Client Library	Approved under WebLogic Server; Ensure upgrade to version 12.1.2		X	X
wljmsclient	10.3.5, 12.1.2	WebLogic JMS Client Library	Approved under WebLogic Server; Ensure upgrade to version 12.1.2		X	X
wlthint3client	10.3.5, 12.1.2	Weblogic T3 Protocol Client Library	Approved under WebLogic Server; Ensure upgrade to version 12.1.2		X	X
xdb	11.1.0.7.0	Oracle XML DB Support Library	Approved under Oracle Database	X		
Xerces2	2.6.2, 2.11.0	Xerces2 Java is a library for parsing, validating and manipulating Extensible Markup Language (XML) documents	Ensure upgrade to version 2.11.0	X		X
xmlparserv2	11.1.0.7.0	Oracle XML DB Support Library	Approved under Oracle Database	X		
XMLUnit for Java	1.0	Support Library for Testing Java Classes with XML	Upgrade to version 1.5		X	X

5. Data Design

5.1. DBMS Files

As part of HDR 3.8 enhancements, the current existing schema remains unchanged.

5.2. Non-DBMS Files

Not Applicable.

5.3. Data View

Pending any changes during agile development, no additional data views will be required.

6. Detailed Design

The Hardware and Software Detailed Design subsections of Section 6 are included in HDR 3.8 SDD, Volume 2. Volume 2 describes the major components of the HDR 3.x system, including an overview, module design, processing, data structure, configurations, process flow diagrams and other diagrams as needed. HDR provides back end processing only and does not include user interfaces or interactions.

The following Subsections are described in detail in the *HDR 3.8 System Design Document, Volume 2*:

6.1. Hardware Detailed Design

Please refer to Volume 2.

6.2. Software Detailed Design and all subsections

Note: Volume 2 contains detail design that is not likely to change from release to release. In the event there are modifications to any of the detailed design, it will be indicated to the reviewer. For HDR 3.8, there are no changes to the detailed design described in the HDR 3.7 SDD.

6.3. Network Detailed Design

Refer to previous diagrams in Section 4.3 Network Architecture.

6.4. Service Oriented Architecture / ESS Detailed Design

See section 4.4 for an overview of the HDR 3.x SOA/ESS design. Data exchanged by HDR 3.x services and other interfaces is computable data and is primarily Patient Health Information (PHI). Other types of information such as survey results are also exchanged and stored in HDR.

HL7 and VistA terminology standards apply to the information exchanged by the HDR 3.x clients with the HDR 3.x services and interfaces. The following sub-sections show the clients that consume each of the HDR 3.x services and/or interfaces. These sections provides details that can be used by Enterprise Service Registry and Repository (ESRR) for exposing the service and interfaces to the VA enterprise. The individual files and links to the actual services are not included in this table due to security constraints and will be provided directly to ESRR.

6.4.1. FPDS MS Interface

The HDR 3.8 revision continues to support a JMS interface to FPDS for supporting asynchronous write requests. Refer to HDR 3.8 SDD Volume 2, Section 6.2.1.28 for detailed information on FPDS. The information exchange details for the interaction are as specified in the following table.

Table 18 FPDS MS Clients

HDR 3.x Client System	Data Domains	Information Exchange Description	Information Element (Payload)
Health Management Platform (HMP/eHMP)	PGD	HMP/eHMP places a JSON payload consisting of PGD data onto JMS queue that is processed by the FPDS Message Driven Bean (MDB). See HMP/eHMP specifications for details on the JSON object structure. The FPDS MDB creates an XML payload from the JMS parameters and encapsulated the JSON payload for storing as a CLOB in the HDR database.	Per HMP/eHMP Message Exchange Standard: PGD JSON Object Payload Format: JSON

6.4.2. FPDS REST Web Service Clients

FPDS provides a REST interface to be used by clients for retrieving data. See section 6.2.1.28 for FPDS details. This information exchange details for the interaction are as specified in the following table.

Table 19 FPDS REST Web Service Clients

HDR 3.x Client System	Data Domains	Information Exchange Description	Information Element (Payload)
Health Management Platform (HMP/eHMP)	Multiple Per VPR. See VistA VPR API for details on supported domains.	HMP/eHMP uses FPDS REST interface to request PGD Data from HDR and VPR and VPR data from VistA. The path parameters and the query parameters are used to construct an XML filter per GenericListDataFilter2.xsd as required by the CDS framework. The XML response from CDS that conforms to the GenericObservationRead2.xsd is transformed to JSON prior to returning to client.	GenericListDataFilter2.xsd ; GenericObservationRead2.xsd Message Exchange Standard: VistA VPR JSON Object Payload Format: JSON

6.4.3. ARS SOAP Web Service Clients

ARS provides a SOAP interface to be used by clients for accessing aggregated data. This interface is used by HTH as shown in the following table.

Table 20 ARS SOAP Web Service Clients

HDR 3.x Client System	Data Domains	Information Exchange Description	Information Element (Payload)
Home TeleHealth (HTH)	DMPs, Patient Census, and Survey report-related aggregations	HTH uses ARS to obtain aggregated reports on survey reports, patient census, and DMP data stored in HDR.	HTReportFilter.xsd; HTResponse.xsd Message Exchange Standard: VA ELDM Payload Format: XSD

6.4.4. CDS REST Web Service Clients

CDS provides a REST interface to be used by clients for accessing data. This interface is used by RDI as shown in the following table.

Table 21 CDS REST Web Service Clients

HDR 3.x Client System	Data Domains	Information Exchange Description	Information Element (Payload)
Remote Data Interoperability (RDI) component of Vista Computerized Patient Record System (Vista CPRS)	Allergy Pharmacy	CPRS uses readClinicalData1 operation on CDS Web Service through RDI for order checks. - Gets both CHDR and Vista allergies and medication data from HDR database for a patient - Operation parameters include an XML string that specifies data domain and filter attributes along with the response payload schema name. Response is an XML schema payload	RDIIICRXSinglePatientFilter.xsd,RDIIntoleranceConditionPharmacyRead40010 Message Exchange Standard: VA ELDM (VIM) Payload Format: XSD

6.4.5. CDS SOAP Web Service Clients

CDS provides a SOAP interface to be used by clients for storing and accessing data. This interface is used by various clients for obtaining clinical data as shown in the following table.

Table 22 CDS SOAP Web Service Clients

HDR 3.x Client System	Data Domains	Information Exchange Description	Information Element (Payload)
Clinical Data Repository/ Health Data Repository (CHDR)	Allergy Pharmacy	Call to readClinicalData1 operation on CDS Web Service on HTTP/SOAP to get Vista allergies and outpatient pharmacy data. 1 call per domain - varies by the filter and template schema parameters specified and used to create the payloads	CHDRAllergySinglePatientFilter.xsd,IntoleranceConditionRead40010, CHDR_RXSinglePatientFilter.xsd,PharmacyRead40010 Message Exchange Standard: VA ELDM (VIM)

HDR 3.x Client System	Data Domains	Information Exchange Description	Information Element (Payload)
			Payload Format: XSD
Clinical Data Repository/Health Data Repository (CHDR)	Allergy Pharmacy	Call to createOrUpdateClinicalData operation for storing CHDR Allergy and CHDR Pharmacy data for Active Dual Consumer patients in HDR. 1 call per data domain	IntoleranceConditionCreateOrUpdate40060, PharmacyCreateOrUpdate40060 Message Exchange Standard: VA ELDM (VIM) Payload Format: XSD
My HealtheVet - Personal Health Record - Online Viewing (MHV)	Allergy Laboratory, HTH Vitals	Service: CDS; Operation: readClinicalData1 - 1 call per patient per data domain - Uses filters to specify query parameters and specifies response template to use on web service operation. MHV uses HDR/CDS via MDWS for following data from HDR database: Vitals (HTH) Lab C/H (VistA) Intolerance condition (CHDR + VistA)	MHV VitalSinglePatientFilter.xsd, MHV VitalsignsRead40010, IC SinglePatientAllDataFilter.xsd, MHV IntoleranceConditionRead40011, LabSinglePatientAllDataFilter.xsd, MHV LabRead40011 Message Exchange Standard: VA ELDM (VIM) Payload Format: XSD
VistA Web (VW)	Allergy Pharmacy HTH Vitals	Call readClinicalData1 on CDS web service for HTH Vitals, CHDR Allergies and CHDR Pharmacy Data	VW AllergySinglePatientFilter.xsd, VW AllergiesRead40010, RX SinglePatientAllDataFilter.xsd, VW PharmacyRead40010, VW VITALS SinglePatientFilter.xsd, VW VitalsignsRead40010 Message Exchange Standard: VA ELDM (VIM) Payload Format: XSD

6.4.6. Socket Adapter Clients

HDR 3.x provides a Socket Adapter interface that clients can connect to using TCP/IP for storing data in HDR. When the Socket Adapter sees a message, it places the message on the appropriate CDS Message Mediator JMS queue for further processing. HTH and VistA are the primary Socket Adapter clients that send HL7 messages which are then transformed by the CDS Message Mediator component into XML payload for processing and storing to HDR, as shown in the following table.

Table 23 Socket Adapter Clients

HDR 3.x Client System	Data Domains	Information Exchange Description	Information Element (Payload)
Home TeleHealth (HTH)	Vitals Measurement; Patient Satisfaction Surveys; Veterans Rand 12 Item Health Survey (VR-12); DMPs; Patient Census	HTH Connects to HDR/CDS Socket Adapter Port and sends HL7 data over TCP/IP; receives acknowledgements and CA and CE messages back 1 call per data domain	HL7 2.4 messages Survey, Census and DMP are XML strings wrapped in HL7 Message Exchange Standard: HL7 2.4 Payload Format: HL7
VistA	Lab	Vista connects to the HDR/CDS Socket Adapter to send VistA lab results (Chemistry and Hematology) for storing in the HDR database.	HL7 2.4 messages Message Exchange Standard: HL7 2.4 Payload Format: HL7

6.4.7. CDS Message Mediator Clients

HDR 3.x provides a JMS interface through the CDS Message Mediator component. This interface is used by the CDS Socket Adapter and VIE to process messages. See section 6.2.1.27.6 for further details on the processing of the HL7 messages received on the JMS queues by Message Mediator. The interface exchange details are as shown in Table 24.

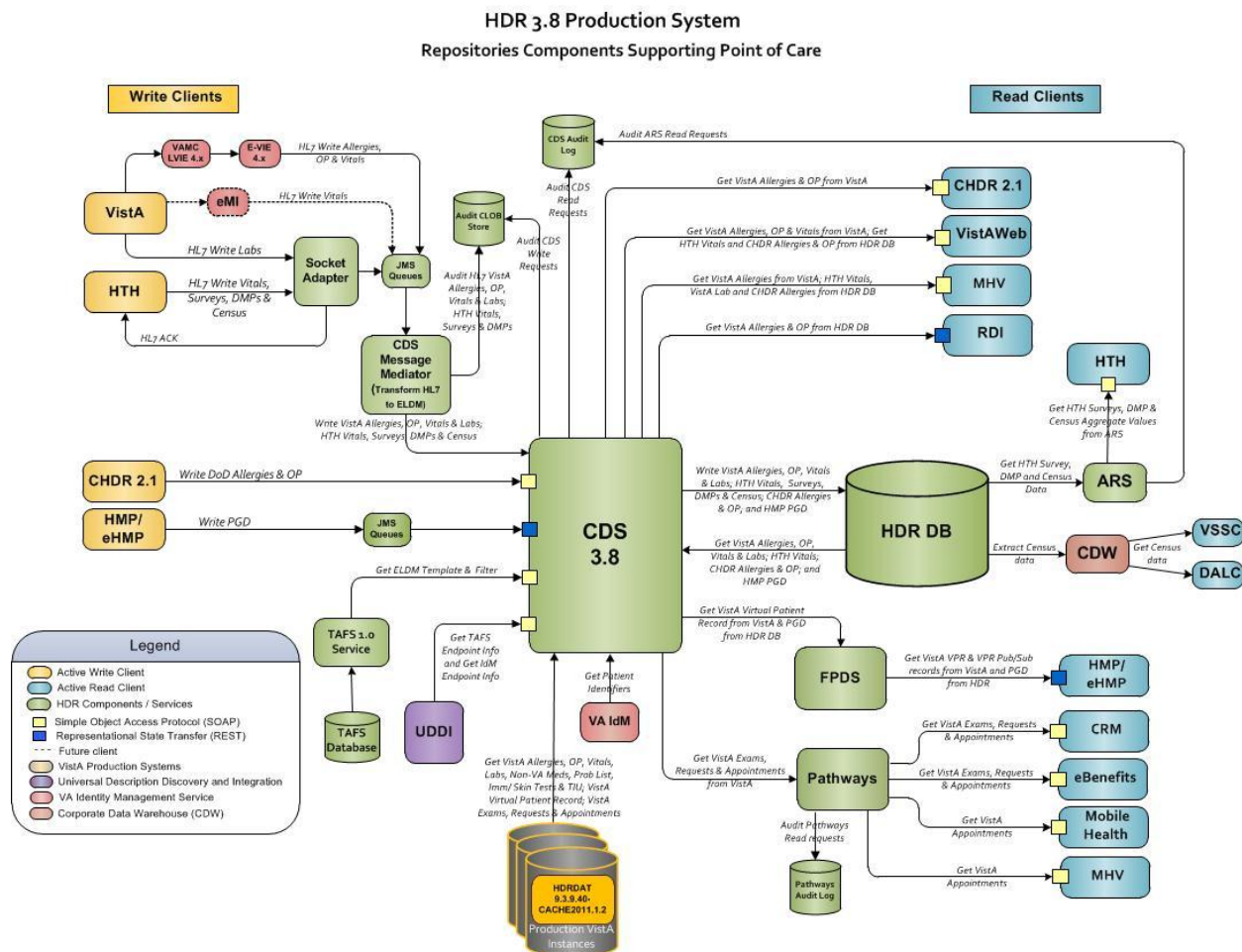
Table 24 CDS Message Mediator Clients

HDR 3.x Client System	Data Domains	Information Exchange Description	Information Element (Payload)
VIE	Allergies, OP, Vitals	VIE connects to the HDR/CDS Message Mediator component to send VistA Allergies, OP, and Vitals data for storing in HDR.	HL7 2.4 messages Message Exchange Standard: HL7 2.4 Payload Format: HL7
Socket Adapter	Vitals Measurement; Patient Satisfaction Surveys; Veterans Rand 12 Item Health Survey (VR-12); DMPs; Patient Census	All data that arrives on the socket connection is placed on the appropriate JMS queues for processing by the Message Mediator	HL7 2.4 messages Survey, Census and DMP are XML strings wrapped in HL7 Message Exchange Standard: HL7 2.4 Payload Format: HL7

7. External System Interface Design

The CDS process flow requires interaction with various enterprise and internal web services, data repositories, data access components and data store components as illustrated the following figure.

Figure 13 CDS interactions with services/components/data repositories



7.1. CDS Modules Overview

The HDR ecosystem is comprised of CDS and supporting components, including: ARS, FPDS, Pathways, HDRDAT, and the HDR database itself. Other internal services include: TAFS, Auditing and Logging, and Error Handling services. The paradigm of Service Oriented Architecture is not only employed for interaction with other VA Enterprise Systems, it is extended for interaction within HDR's various modules. The phrase CDS External Access is being used to explain the mechanism by which external VA Enterprise Systems interact with CDS, and the phrase CDS Internal Access to explain interaction amongst various modules within HDR.

7.1.1. CDS Internal Access

CDS accesses the data repositories (HDRDAT and HDR) using the data sources configured in the WebLogic environment. CDS queries several instances of HDRDAT and queries/stores in HDR for the client request clinical data. CDS utilizes the Audit CLOB Store component to log the audit of the Read request along with the cursory information of the Read response in HDR. CDS is a Java EE application which will be deployed in a WebLogic cluster environment. The WebLogic cluster will be configured with data sources of HDRDAT and HDR. CDS uses

Hibernate to retrieve/persist HDRDAT/HDR using the JDBC connections provided by the data sources. The Audit CLOB Store will use Hibernate to persist audits of the Read request/response using the JDBC connections provided by the HDR data sources.

The following table illustrates the data source details used by CDS in order to access the HDR/HDRDAT.

Table 25 CDS Data Sources

Data Store	Interface	Input Messages	Output Messages
IdM / TAFS web services	UDDI registry	CDS connects to the UDDI service using JAXR, a standard API for interacting with web service repositories. Once the endpoint is resolved from the UDDI service, a web service client is instantiated by the HDR client and injected with the endpoint Uniform Resource Identifier (URI) for the service.	The webservice endpoint is used by CDS to evoke the required business API.
HDRDAT	JDBC data source pool for each of HDRDAT database instance is maintained in the WebLogic cluster in which CDS is deployed.	CDS will use Spring to obtain the data source via JNDI lookup by passing the HDRDAT data source specific JNDI information as configured in WebLogic environment where CDS application is deployed.	The data source which provides the JDBC connection to HDRDAT database. Upon obtaining the JDBC connection to HDRDAT database, CDS will use Hibernate to query for clinical data by passing Vista site id/patient (local) identifier of the patient ICN and start and end dates specified in the clinical data request filter XML received as part of the client read request. The query response from HDRDAT will contain the patient (local) identifier-specific clinical data whose clinically relevant date is between the date ranges specified in the query.

Data Store	Interface	Input Messages	Output Messages
HDR	JDBC data source pool for HDR database maintained in the WebLogic cluster in which CDS is deployed.	CDS will use Spring to obtain the data source via JNDI lookup by passing the HDR data source specific JNDI information as configured in WebLogic environment where CDS application is deployed.	The data source which provides the JDBC connection to HDR database. Upon obtaining the JDBC connection to HDR database, CDS will use Hibernate to query for clinical data by passing Vista site id/patient (local) identifier of the patient ICN and start and end dates specified in the clinical data request filter XML received as part of the client read request. The query response from HDR will contain the patient (local) identifier specific clinical data whose clinically relevant date is between the date ranges specified in the query.

7.1.2. UDDI – CDS Directory Service

HDR Service Endpoint Resolution is supported via Universal Description, Discovery, and Integration (UDDI). HDR components interact with UDDI using the standard Java API for XML Registries (JAX-R API).

CDS gains access to the enterprise (IdM) and internal services (TAFS) by getting their connection information from UDDI. According to the enterprise requirements, the providers and consumers of the services must register/access the services from a registry service. UDDI is a service registry that hosts the services and administers the run-time and deployment information of the services.

HDR incorporates the use of a UDDI service for internally resolving service endpoint information at runtime for services consumed by HDR components. HDR components needing to perform an operation on a remote web service connect to the UDDI service using JAXR, a standard API for interacting with web service repositories. Once the endpoint is resolved from the UDDI service, a web service client is instantiated by the HDR client and injected with the endpoint Uniform Resource Identifier (URI) for the service. Currently, the UDDI registry is hosted by HDR. In the future, the enterprise is expected to host an enterprise service registry that will also support JAXR-based clients. Phase 2 of the UDDI integration will be to encourage all clients to migrate to alternative service lookup services ending with the termination of NDS services hosted by the HDR.

UDDI is just another directory service which contains end point information. It is not essential for CDS' existing clients to look up CDS end point information from UDDI. The existing clients of CDS can connect to CDS directly without having to look up CDS info in UDDI. The VA enterprise did encourage as part of an enterprise wide effort to move all projects to a Service Oriented Architecture and register themselves in a single directory. This solution was never completed and an enterprise wide directory service has never been stood up. A lack of enterprise directory service resulted in each project standing up their own.

CDS is a Java EE application which will be deployed in a WebLogic cluster environment. WebLogic cluster will be configured with data sources of HDRDAT and HDR. CDS retrieves data from HDRDAT and retrieves/persists data in HDR using the JDBC connections provided by the data sources.

CDS accesses the Audit CLOB Store and HDR Oracle data stores using an Oracle JDBC RAC-enabled data source configured within the WebLogic server system. CDS accesses VistA data via the HDRDAT using JDBC data sources configured within the WebLogic server system.

7.1.2.1. SOAP requests with UDDI lookup

The SOAP remote procedure call (RPC) over HTTP client leverages the HTTP POST protocol and initiates SOAP structured message requests on the server using a well-known URL. The web service then delegates the request to a stateless session bean for request fulfillment. The SOAP service is registered in UDDI.

When a read request arrives at the CDS interface component for processing, the application server receives the request over the well-known URL and identifies the operation name value; it passes it to the appropriate method on the web service interface. These requests are internally processed by CDS EJB component that is deployed on the application server as a stateless session bean. The EJB delegates the incoming requests to CDS framework.

7.1.2.2. SOAP requests without UDDI lookup

The SOAP remote procedure call (RPC) over HTTP client leverages the HTTP POST protocol and initiates SOAP structured message requests on the server using a well-known URL. The web service then delegates the request to a stateless session bean for request fulfillment. This interface is exposed as a coarsely grained Web service to all clients. However, the Service Interface is not registered in UDDI. It provides clients functionality to read non-clinical data using the ELDM-formatted payloads for data retrieved from VistA systems. The component delegates client requests to framework objects which implement the logic to parse and fulfill the data request as specified by the ELDM filter payload.

7.1.2.3. REST web service

The REST web service client uses the HTTP protocol, and leverages a collection of server side operations using the HTTP methods GET. The client can initiate an HTTP request to a server, which in turn processes the request and returns an appropriate response based on the HTTP method submitted. As with the SOAP client, the REST web service implementation logic on the server delegates to a stateless session bean for request fulfillment.

Service registration in UDDI: The endpoint of the REST service is registered in UDDI that is hosted on the Weblogic application server. The endpoints configuration is as follows and is in the classpath of weblogic environment in which the services are deployed.

Figure 14 UDDI configuration:

```
<bean id="jaxrUtil" class="gov.va.med.cds.registry.JAXRUtility">
  <property name="connectionProperties">
    <props>
      <prop key="javax.xml.registry.queryManagerURL">http://islvhdwls06.fo-
slc.med.va.gov:7101/uddi/uddilistener</prop>
    </props>
  </property>
  <property name="factoryClass" value="org.apache.ws.scout.registry.ConnectionFactoryImpl"/>
</bean>
```

```

<bean id="TAFS_UDDI_SVC_ID" class="java.lang.String">
    <constructor-arg value="c621cae5-7692-4f19-80a0-e2af53374483"/>
</bean>

<bean id="IDM_UDDI_SVC_ID" class="java.lang.String">
    <constructor-arg value="732abcf1-70c7-4c07-8bcb-ab6077adaafb"/>
</bean>

<bean id="MESSAGE_MEDIATOR_UDDI_SVC_ID" class="java.lang.String">
    <constructor-arg value="08c68960-7210-4933-8f82-a3e75c1fc9e3"/>
</bean>

<bean id="CDS_UDDI_SVC_ID" class="java.lang.String">
    <constructor-arg value="7f4163ef-ea7b-495e-9457-970bb2e37717"/>
</bean>

```

Figure 15 SOAP and REST server-side implementation, configuration of delegating objects

```

<jaxws:endpoint id="cdsService"
    implementorClass="gov.va.med.cds.ejb.ClinicalDataServiceSynchronousSession"
    implementor="#cdsStatelessSession"
    address="/cds-service"
/>

<jaxrs:server id="createServiceProxy" address="/cds-service/createClinicalData">
    <jaxrs:serviceBeans>
        <bean
class="gov.va.med.cds.webservice.RestClinicalDataServiceCreateProxy">
            <property name="clinicalDataService" ref="cdsStatelessSession"/>
        </bean>
    </jaxrs:serviceBeans>
</jaxrs:server>

<jaxrs:server id="deleteServiceProxy" address="/cds-service/deleteClinicalData">
    <jaxrs:serviceBeans>
        <bean
class="gov.va.med.cds.webservice.RestClinicalDataServiceDeleteProxy">
            <property name="clinicalDataService" ref="cdsStatelessSession"/>
        </bean>
    </jaxrs:serviceBeans>
</jaxrs:server>

<jaxrs:server id="readServiceProxy" address="/cds-service/readClinicalData1">
    <jaxrs:serviceBeans>
        <bean class="gov.va.med.cds.webservice.RestClinicalDataServiceReadProxy">
            <property name="clinicalDataService" ref="cdsStatelessSession"/>
        </bean>
    </jaxrs:serviceBeans>
</jaxrs:server>

<bean id="cdsStatelessSession"
    class="org.springframework.ejb.access.LocalStatelessSessionProxyFactoryBean"

```

```

        lazy-init="true">
        <property name="lookupHomeOnStartup" value="false" />
        <property name="jndiName">

<value>comp/env/ejb/local/gov/va/med/cds/ClinicalDataServiceSynchronousSession</value>
        </property>
        <property name="jndiEnvironment">
                <props>
                        <prop
key="java.naming.factory.initial">weblogic.jndi.WLInitialContextFactory</prop>
                </props>
        </property>
        <property name="businessInterface">
                <value>gov.va.med.cds.client.ClinicalDataServiceInterface</value>
        </property>
</bean>

```

The `RestClinicalDataServiceCreateProxy` is passed the request by the application server, and delegates to the `clinicalDataServiceSynchronous` object, which is an implementation of the `ClinicalDataServiceSynchronousSession` EJB.

Figure 16 SOAP calls without UDDI:

```

<jaxws:endpoint id="pathways"
        implementorClass="gov.va.med.cds.rpc.ejb.ClinicalDataServiceRpcSynchronousSession"
        implementor="#pathwaysStatelessSession"
        address="/pathways"
/>

<jaxrs:server id="readServiceProxy" address="/pathways/readData">
        <jaxrs:serviceBeans>
                <bean
class="gov.va.med.pathways.webservice.RestPathwaysServiceReadDataProxy">
                        <property name="pathwaysService" ref="pathwaysStatelessSession"/>
                </bean>
        </jaxrs:serviceBeans>
</jaxrs:server>

        <bean id="pathwaysStatelessSession"
class="org.springframework.ejb.access.LocalStatelessSessionProxyFactoryBean" lazy-init="true">
                <property name="lookupHomeOnStartup" value="false" />
                <property name="jndiName">

<value>comp/env/ejb/local/gov/va/med/cds/ClinicalDataServiceRpcSynchronousSession</value>
                </property>
                <property name="jndiEnvironment">
                        <props>
                                <prop
key="java.naming.factory.initial">weblogic.jndi.WLInitialContextFactory</prop>

```

```

        </props>
    </property>
    <property name="businessInterface">
        <value>gov.va.med.repositories.Pathways</value>
    </property>

```

7.2. Mechanisms to interface CDS with external Clients

CDS interfaces with its clients through SOAP, REST, JMS and Socket-based services. A Write request submitted by the client application can be a VIM write XML instance, a JSON document, or an HL7 payload. The Read request can be a filter XML, a JSON document, an HL7 query message or a REST query string. The clinical data exchanged between CDS and its clients takes the form of XML documents representing VIM template instances or JSON documents. The format of these XML instances is governed by XSD schema definitions for VIM templates that are specific to the clinical domain. Refer to Data Exchanges in the following section.

- CDS exchanges VIM-based XML clinical/questionnaire results/expressions data with client applications as defined by the templates and filters.
- Pathways exchanges VIM-complaint non-clinical data with client applications as defined by the templates and filters.
- FPDS exchanges XML and JSON clinical and non-clinical data with client applications.
- For HMP/eHMP, CDS exchanges JSON formatted clinical data with HMP/eHMP.
- CDS exchanges XML HTH Census, DMP, ADL and Surveys data with HTH as defined by the templates and filters.
- HTH Census, DMP, ADL and PSS data are generated by HTH vendors in HL7 format. They are sent to HDR via VA HTH system to be stored in the HDR data stores.
- Data exchange between CDS and client applications is confined to specific templates and filters. The following filters are used for Read requests between CDS and client applications..

7.2.1. Data Exchanges

The following filter IDs are used for Read transactions between CDS and client applications and determine what data is returned to the requesting client:

Table 26 Filters between CDS and client

Read Filter Name	
AASinglePatientAllDataFilter.xsd	CHDRALLERGYSinglePatientFilter.xsd
AllergySinglePatientAllDataFilter.xsd	CHDRRXSinglePatientFilter.xsd
VWALLERGYSinglePatientFilter.xsd	CHDRLabSinglePatientFilter.xsd
VWVITALSinglePatientFilter.xsd	LabSinglePatientAllDataFilter.xsd
VWICSinglePatientFilter.xsd	TiuSinglePatientListDataFilter.xsd

Read Filter Name	
VitalSinglePatientAllDataFilter.xsd	TiuSinglePatientDetailDataFilter.xsd
ICSinglePatientAllDataFilter.xsd	ImmunizationSinglePatientFilter.xsd
RDIICRXSinglePatientFilter.xsd	SkinTestSinglePatientFilter.xsd
RXSinglePatientAllDataFilter.xsd	ProblemListSinglePatientFilter.xsd
RXSinglePatientSingleRecordFilter.xsd	AppointmentsSinglePatientFilter.xsd
HTReportFilter.xsd	RequestsAndExamsSinglePatientFilter.xsd
PGDSINGLEPATIENTFILTER	GENERICVISTALISTDATAFILTER
PGDUNIQUEIDFILTER	

Table 27 Template IDs for Read transactions

Read Template Name	
AllergiesRead40010.xsd	VWPharmacyRead40010.xsd
IntoleranceConditionRead40010.xsd	VitalsignsRead40010.xsd
IntoleranceConditionReducedRead40011.xsd	VWVitalsignsRead40010.xsd
Allergy AssessmentRead40010.xsd	MHVLabRead40010.xsd
MHVIntoleranceConditionRead40011.xsd	TiuDocumentListRead2.xsd
VWIntoleranceConditionRead40010.xsd	TiuDocumentDetailRead2.xsd
VWAllergiesRead40010.xsd	NonVAMedicationsRead3
IntoleranceConditionPharmacyRead40010.xsd	ImmunizationRead3
RDIIntoleranceConditionPharmacyRead40010.xsd	SkintestRead3
PharmacyRead40010.xsd	ProblemListRead2
VWPharmacyDetailRead40010.xsd	GenericObservationRead1.xsd
HTResponse.xsd	RequestsAndExamsRead1.xsd
PGDRead1.xsd	AppointmentsRead1.xsd

The following template IDs are used for Write transactions between CDS and client applications:

Table 28 Template IDs for Write transactions

Write Template Name	
PharmacyCreateOrUpdate40060.xsd	CensusSurveyCreate40024
IntoleranceConditionCreateOrUpdate40060.xsd	DMPSurveyCreate40024
Allergy AssessmentCreateOrUpdate40060.xsd	ADLSurveyCreate40024
VitalsignsCreateOrUpdate40060.xsd	PSSurveyCreate40024
LabCreateOrUpdate40060.xsd	PGDCreateOrUpdate1

7.3. Client Interface Detailed Design

The HDR system provides a synchronous Java Application Programming Interfaces (API) for XML-based Web Services (JAX-WS) and Java API for RESTful Web Services (JAX-RS) web services using the Apache CXF open source services framework; an asynchronous application programming interface using the Java Messaging Service (JMS) and a TCP/IP socket interface using XSOckets.

- HDR system clients can connect to the Java API services using either a Simple Object Access Protocol (SOAP) or a Representational State Transfer (REST) Web service interface. CDS, ARS and Pathways clients can connect to the service using SOAP web service interface and FPDS clients can connect to the service using a REST web service interface.
- VIM clinical data Read clients connect synchronously to CDS and receive a VIM response payload on the same connection. CDS, Pathways and ARS Read clients use this feature of the HDR system for obtaining VIM data from HDR DB and/or VistA.
- JSON Read clients connect synchronously to the HDR system and FPDS Read clients obtain JSON data from the HDR database (for PGD) or VistA for VPR data.
- Clinical data Write clients are able to persist clinical data to HDR for the following VistA clinical domains: Allergies, Vitals, Outpatient Pharmacy (OP), and Laboratory. The clients use either the JMS interface or the Socket interface for data exchange. CDS Message Mediator and FDPS expose a JMS interface while the CDS Socket adapter exposes a socket interface to clients.
- HL7 write clients connect to either the CDS Message Mediator or CDS Socket Adapter for persisting data to HDR database. The Socket Adapter is a client of the CDS Message Mediator and places all HL7 messages it receives on the CDS Message Mediator JMS queues. The HL7 message is transformed to VIM by the CDS Message Mediator prior to persistence to the HDR database.
- JSON write clients connect to FPDS JMS interface for persisting data to HDR database. The JSON payload is embedded into VIM to enable processing by the CDS framework and is persisted as a CLOB in the HDR database.

7.3.1. CDS Data Service Interface

The CDS component features the CDS Data Service Interface component that is:

- exposed to service requests for clinical data from VistA systems/HDR. This interface is exposed as a coarsely grained Web service to all clients. The Service Interface is registered in UDDI.
- exposed to service requests for non-clinical data from VistA systems. This interface is exposed as a coarsely grained Web service to all clients. However, the Service Interface is not registered in UDDI. It provides clients functionality to read non-clinical data using the VIM-formatted payloads for data retrieved from VistA systems. The component delegates client requests to framework objects which implement the logic to parse and fulfill the data request as specified by the VIM filter payload.
- exposed as a coarsely grained RESTful Web Service and is accessed as a URL.

7.3.2. VIM Reads

VIM clinical data Read clients submit Read requests for clinical data results. The Read request is maintained for auditing purposes. All applicable data sources are queried and results are returned to the client in the appropriate payload structure.

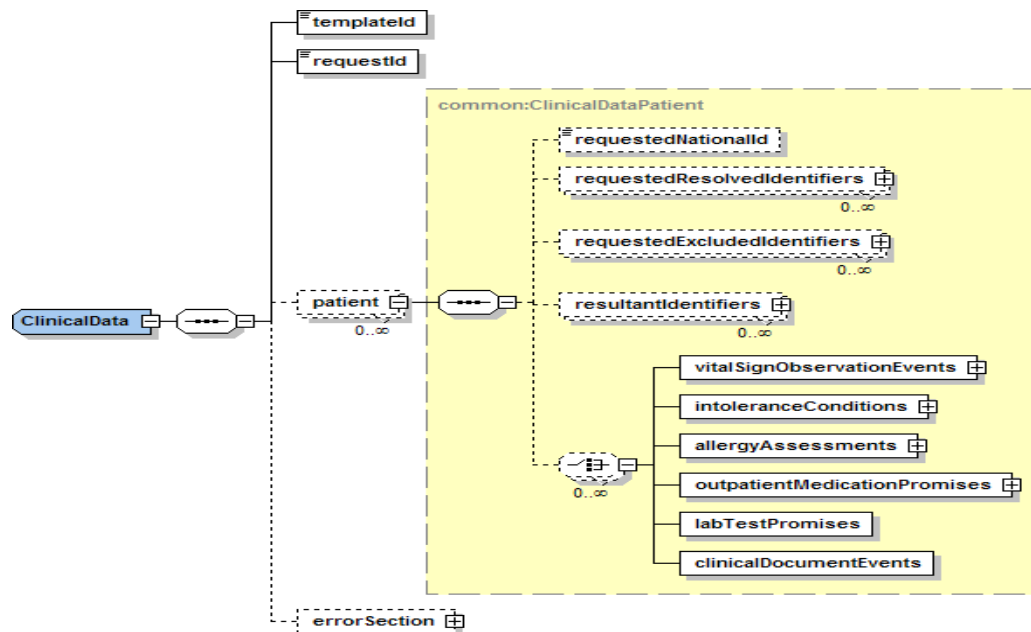
7.3.2.1. CDS Read Request

A Read request is used by the VIM read clinical data client to obtain patient-centric clinical data. The input parameters for the read request are as follows:

- The template ID specifies the type of data payload to be returned to the client application. Template IDs are defined for each supported clinical domain. These include the OP, Allergies, Vitals, Text Integration Utilities (TIU), and Laboratory (Chemistry/Hematology) clinical domains.
- The filter request is an XML document used by CDS and Pathways to identify the specific data retrieval conditions. For read requests, the filter request includes the patient identification information, as well as an optional date range.
- The filter ID designates the schema used by CDS and Pathways to define and validate filter requests.
- The request ID is a unique identifier provided by the requesting site that is used to identify the read request within the system.
- The string value returned by this request is an XML document that is an instance of the VIM template ID input parameter.

The Read response template contains the clinical data returned from the data sources and the errors generated while processing the request. Below is a general example of a Read response, a basic pattern that all specific templates follow.

Figure 17 Read Response Template



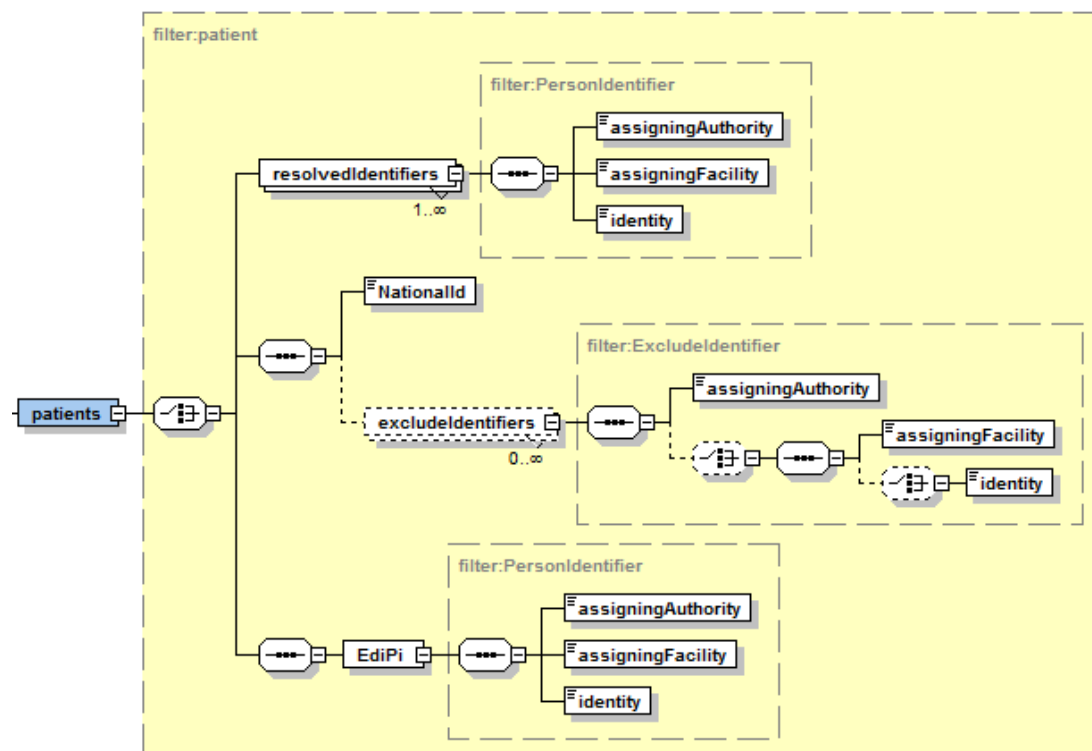
7.3.2.2. Filters

The filter determines which records are to be returned to the client. The filter is presented as an XML document. The first filter section defines the patient identification information that is used to qualify the data to be returned in the response.

- Patient identifiers may be specified by a single national identifier. CDS 3.x supports both the Integration Control Number (ICN) or the Electronic Data Interchange Person Identifier (EDIPI) as the national identifier. CDS 3.x or Pathways will perform a lookup to resolve the national identifier to a list of all local identifiers. Optionally, a list of exclude (local) identifiers can be specified to further restrict the information returned in the response.

The following figure shows the patient identification section in a filter:

Figure 18 Patient Identification in a Filter



The second section specifies one or more entry point filters, which consist of the following:

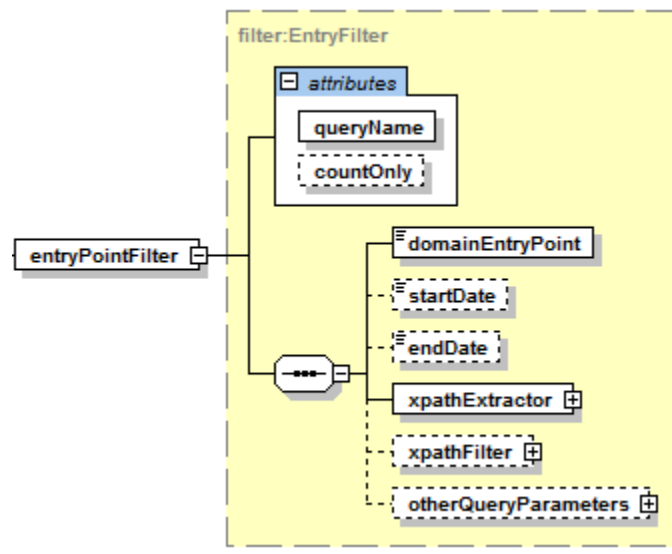
- **Domain Entry Point:** A mandatory parameter that determines the type of data to be returned in the result document, illustrated in the following table. If the domainEntryPoint in the request filter is null/blank, CDS 3.x cannot determine the proper schema to include in the response document and a generic response will be sent to the client. Refer to Appendix B.1 for CDS Supported Domain Entry Points.
- **Start Date:** An optional parameter that represents the beginning point in time to be used to constrain the results.

- **End Date:** An optional parameter that represents the ending point in time to be used to constrain the results.
- **Record Identifier:** An optional list of parameters that is used to return a specific set of one or more records in the result.
- **XPath Query:** An optional parameter that consists of an XPath expression specified in the filter that can be applied to the resultant document to constrain the result.
- **Other Query parameters:** An optional list of one or more parameters that can be used to constrain the returned results. These optional query parameters can be used for ad-hoc query capability.
- **Query Timeout (Seconds):** The optional queryTimeoutSeconds parameter in the Read request filter, which specifies how many seconds CDS allows to pass before the Read request times out.

Note: In the event that a reasonable date range is not specified in the filter, the responsiveness of the request could decrease and the size of the payload may be extremely large.

The following figure shows the entry point section of a filter:

Figure 19 Entry Point Section of a Filter



7.3.2.3. Error Responses

When processing results in errors that interrupt and abort Write or Read requests, such as validation, system errors, missing parameters, etc., the interface to CDS and FPDS will return a response that includes an error section. The error section contains detailed error explanations of the issues encountered. A sample HMP/eHMP XML response with an error section is illustrated in the following figure.

Figure 20 HMP/eHMP XML Response with Error Section

```
<?xml version="1.0" encoding="UTF-8"?>
<clinicaldata:ClinicalData xmlns:clinicaldata="Clinicaldata">
  <templateId>GenericObservationRead1</templateId>
  <requestId>65756756</requestId> -<patients> -<patient>
    <requestedResolvedIdentifiers/> -<resultantIdentifiers> -<resultantIdentifier>
      <identity>xxxxxxxxxx</identity>
      <assigningFacility>xxx</assigningFacility>
      <assigningAuthority>USVHA</assigningAuthority>
    </resultantIdentifier>
  </resultantIdentifiers>
  <genericObservations/>
</patient>
</patients> -<errorSection> -<fatalErrors> -<fatalError>
  <errorId>65756756</errorId>

  <exception>gov.va.med.repositories.fpds.validator.DefaultReadRequestValidator$InvalidParameterValidationError</exce
ption>

  <exceptionMessage>FPDS_ERROR</exceptionMessage>
  <errorCode>FPDS_ERROR</errorCode>
  <displayMessage>Request parameter clientName was not provided or was
empty.</displayMessage>
  </fatalError>
</fatalErrors>
<errors> </errors>
<warnings> </warnings>
</errorSection>
</clinicaldata:ClinicalData>
```

7.3.3. JSON Reads

The JSON client sends a Read request to CDS as a transform REST request URL (Uniform Resource Locator). The URL request is processed by FPDS, which is a thin web-service adapter layer to CDS and transforms incoming URL and HTTP query parameter information REST requests into filter Read requests that can be processed by CDS and the underlying CDS framework. The query call contains mandatory elements which comprise the URL, and optional elements which comprise the query parameters.

7.3.3.1. JSON Reads – VistA Virtual Patient Record (VPR)

FPDS features a RESTful web service façade, aggregation of data across VistA sites, and the option to return data in JavaScript Object Notation (JSON) or a JSON wrapped in XML

- The FPDS web service module is a RESTful implementation hosted along with CDS providing access to HMP/eHMP's Generic Stored Procedures
- The URL and optional query parameters are parsed by an XSL or dom4j library in the FPDS web -service to generate a filter XML
- The patient-centric Read results are queried from VistA systems and the Read results from various VistA sites are aggregated in the Read response returned to HMP/eHMP
- The site-centric Read results are also queried from VistA systems and the Read results from a single site for multiple subscribed patients is returned in the Read response to eHMP/eHMP.
- The FPDS service supports subscribing and cancelling subscriptions for a patient. A patient can be subscribed by ICN at all sites they are known or at a single site by their site DFN – cancellations follow the same essential logic.

- Patient lookup information is an Integration Control Number (ICN), whereby the ICN is correlated to its respective DFNs by CDS calling the VA Identity Management (IdM) web service
- The FPDS service has the option to return the response as a JSON document, or wrap an XML around the JSON document
- The HMP/eHMP client does not send VIM template and filter to CDS. Instead, the HMP/eHMP client provides only the names of the template and filter in the URL.
- HMP/eHMP can request the format of the Read response to be either JSON or XML. The FPDS web service takes the response template data returned by the CDS service and transforms as appropriate to either JSON or XML data prior to returning to the HMP/eHMP FPDS Read client. By default, the Read response will be in JSON format if the format type is not specified in the REST web-service read request URL.

A sample JSON Read response is illustrated below.

Figure 21 HMP/eHMP JSON Read Response Format {sites: [

```
{
  "apiVersion": "1.01",
  "params": {
    "domain": " SLC ",
    "systemId": "D"
  },
  "data": {
    "totalItems": 0,
    "items": []
  }
},
{
  "apiVersion": "1.01",
  "params": {
    "domain": "SLC",
    "systemId": "I"
  },
  "data": {
    "updated": "20140130231928",
    "totalItems": 2,
    "items": [
      {
        "displayName": "BP",
        "facilityCode": 500,
        "facilityName": "R",
```

```

    "high": "210/110",
    "kind": "Vital Sign",
    "localId": 4,
    "locationName": "2TE",
    "locationUid": "u",
    "low": "100/60",
    "observed": 200304041518,
    "result": "138/72",
    "resulted": 20030404151847,
    "summary": "BLOOD PRESSURE 138/72 mm[Hg]",
    "typeCode": "u",
    "typeName": "BLOOD PRESSURE",
    "uid": "u",
    "units": "mm[Hg]"
  },
  {
    "displayName": "P",
    "facilityCode": 0,
    "facilityName": "R",
    "high": 120,
    "kind": "Vital Sign",
    "localId": 2507,
    "locationName": "20 MINUTE",
    "locationUid": "u",
    "low": 60,
    "observed": 200304041518,
    "result": 72,
    "resulted": 20030404151847,
    "summary": "PULSE 72 /min",
    "typeCode": "u",
    "typeName": "PULSE",
    "uid": "u",
    "units": "/min"
  }
}

```

7.3.3.2. **JSON Read s – PGD from HDR Database**

The JSON PGD Read client submits a Read request for PGD stored in the HDR database. The Read request is maintained for auditing purposes.

- Mandatory request path parameters include the templateID, FilterID and the Identifier, which is either the patient identifier to return collection type results for the patient, or the UID, which returns the specific patient record.
- The HTTP query parameters, which are mandatory, include the client name as 'HMPPGD' and client initiation time.
- Until the HMP/eHMP client has an MVI-correlated patient identifier, the Read request will contain the Patient Facility ID as '200HMP/eHMP' and the Patient Assigning Authority as 'USVHA'.
- HDR will return the JSON document CLOB as is was persisted in the Write request.
- Read results will be based on patient or UID only. No query date ranges are being implemented for this release.

7.3.4. ARS Reads

ARS provides aggregate reports on Census data stored in the HDR DB. ARS currently supports aggregate read services for Patient Satisfaction Surveys and DMPs. The ARS is a SOAP web service exposed over HTTPS that provides an aggregated (report) view of HTH data stored in the HDR DB. The support for Census reads through ARS consists of:

- Generating reports on HTH Census data stored in the HDR database.
- Providing calculations on the stored data and returning counts and values in the results report.
- Materialized view to be developed in the database layer.
- Report filter and report template that are part of the production HDR system will be used for all Census aggregate reads

7.3.5. VIM Writes

VIM clinical data Write clients submit Write requests to create or update clinical data to the Repositories data sources. The Write requests are audited and validated before being persisted to the HDR database.

7.3.5.1. CDS Write Request

A VIM Write request is used by the VIM Write clinical data client to store clinical data for a patient into the HDR DB. The input parameters for the Write request are as follows:

- The create request contains a VIM template XML document that represents the clinical data to be stored.
- The template ID specifies the VIM write template schema to be used to validate the VIM template XML document create request. Template IDs are defined for each supported clinical domain. These include OP, Allergy, Vitals, and Laboratory (Chemistry/Hematology).
- The request ID is unique to the sending site and is used to identify the Write request within the system.

- The optional clientName is used for auditing and monitoring the Create/Update requests. The clientName is limited to 100 characters. If the clientName is more than 100 characters, CDS shall return a validation exception to the clients.
- The optional clientRequestInitiationTime contains the time at which the Create/Update request is initiated at the client's instance.

The string value returned by this request is an XML document that is an instance of the CDS response schema. This tells the requester whether CDS successfully stored the data into HDR and identifies any error conditions in the case of failure. Clinical data is not included in this instance, just status and error information (if any).

Write clients use VIM to store data in the HDR database. In the case of HTH, all HL7 Write requests are submitted to a domain-specific Java Messaging Service (JMS) queue via a socket adapter, which are transformed to VIM by the CDS Message Mediator component for persistence. An Update request is used by the DBQ client to update the existing data records in the HDR database.

An Update request is identical to a Write request, except for an additional data element that contains the record identifier of the record being updated. An Update request is identical to a Write request, except for an additional data element that contains the record identifier of the record being updated.

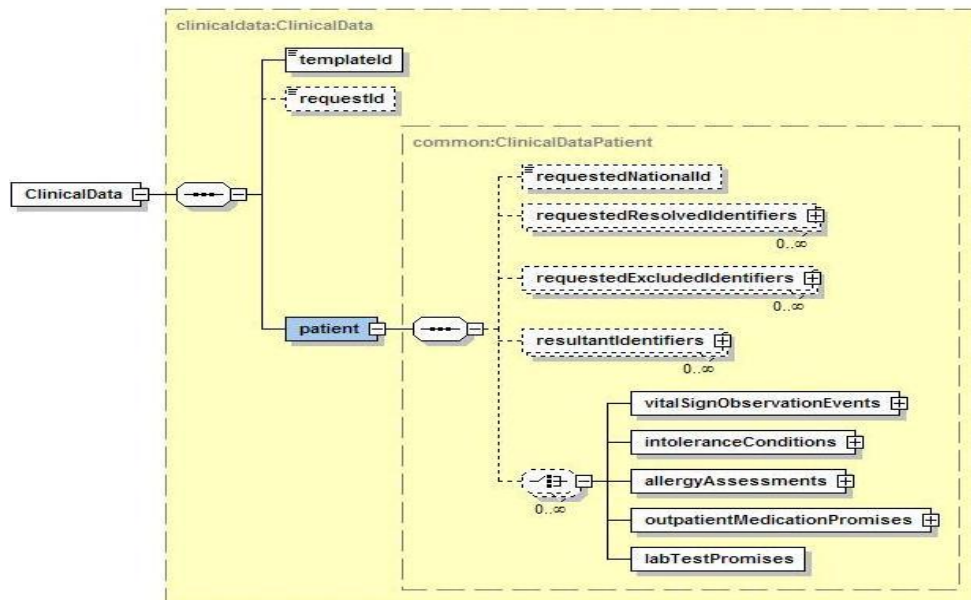
7.3.5.2. Write Request Response

When a Write (Create) is successful, the response that is sent has an empty error section. If issues occur during a Create that result in the record not being written into the HDR then a response is sent with an error section containing a detailed error and explanation. An example of the write or create response is illustrated in the Figure below.

7.3.5.3. Write Request Template

A Create or Update clinical data template is used by client applications to persist clinical records to the HDR DB. At the schematic view, they are nearly identical to a Read template, however, the meta data differs and the error section is missing (a separate response template is used for writes). Write templates define a single entry point – with the exception of HTH Surveys which can define two survey entry points. An example of a Write template is illustrated in the Figure below.

Figure 22 Write Request Template



7.3.6. JSON PGD Writes

The JSON PGD Write client submits an asynchronous write message to create or update PGD to the HDR database. The Write requests are audited and validated before being persisted to the HDR database. This was functionality added in the HDR 3.7 release. HDR provided a simple service interface and mechanisms for persisting (including backups) of PGD in the HDR System of Record (SOR).

- The Create and Update request contains a self-describing JSON document with the data to be stored, such as patient-entered medications.
- The Create and Update request contains a JMS Reply-To destination for a Write response or a default destination.
- In the interim, until the HMP/eHMP PGD client sends an MVI-correlated patient identifier, the Patient Facility ID will default to '200HMP/eHMP' and the Patient Assigning Authority will default to 'USVHA'
- The Unique Record Identifier (UID) is unique to the sending site and is used to identify the Write request within the system.
- The client Collection Type specifies the domain of data being persisted to the HDR database.

7.3.7. HL7 Writes

The HL7 Write clients connect to either the CDS Message Mediator or CDS Socket Adapter when persisting clinical data to the HDR database. The HL7 message is transformed to VIM by the CDS Message Mediator prior to persistence to the HDR database.

- HL7 messages from VistA come through the CDS Message Mediator and include: Allergies, Outpatient Pharmacy, Laboratory Chemistry and Hematology (Lab can come through the Socket Adapter and the CDS Message Mediator), and Vitals
- HL7 messages from HTH come through the Socket Adapter and include: Vitals, multi-patient Census Records, Patient Satisfaction Surveys, DMPs and ADLs.

7.4. 3.8 Client Interfaces

7.4.1. HTH Client

With the HDR 3.8 release, census extract data will be made available to the HTH clients VSSC and DALC via CDW.

Table 29 HTH Census Client

System	Details
Title	Home Telehealth
Abbreviation	HTH
Point of Contact	George Blankenship
Vendor	VA
HTH Reads	
The census data that resides in HDR will be pulled from HDR by CDW on a regular basis and will be written into CDW. The VSSC and DALC teams will be given the ability to query the census extract data as they require.	
HTH Writes	
HTH submits Write requests for multiple-patient Census data and single patient DMP, ADL and PSS data and HDR returns the response. The Write request is in HL7 format and goes to the Socket Adapter component of CDS. Socket Adapter interacts with the CDS Message Mediator component of CDS via a JMS asynchronous interface. The CDS Message Mediator component of CDS extracts the HTH data presented in the HL7 message and transforms the data into a VIM write template that is sent to HDR for storage. CDS returns a Write response in HL7 format. <ul style="list-style-type: none"> • CDS supports HTH Census, DMP, Activities of Daily Living and Patient Satisfaction Survey data XML Write /Read requests 	
HTH Read Filters and Templates	
HTH uses the following filter-template combination for retrieval of HTH Survey data from the HDR database. <ul style="list-style-type: none"> • HTReportFilter.xsd • HTResponse.xsd The following filter is used for a Read request between ARS and the HTH client application: <ul style="list-style-type: none"> • HTReportFilter.xsd 	

HTH Write Templates

The template IDs used by CDS to Write HTH ADL, PSS, Census and DMP data to HDR include:

- ADLSurveyCreate4002
- PSSurveyCreate40024
- CensusSurveyCreate40024
- DMPSurveyCreate40024

7.4.2. HMP/eHMP VPR Client

Support for the HMP/eHMP VPR Client will be extended.

Table 30 HMP/eHMP Read Client

System	Details
Title	Health Management Platform
Abbreviation	HMP/eHMP VPR 1.2
Point of Contact	Deb Migliore
Vendor	VA
eHMP/eHMP Reads	
<p>The improved VPR API 1.2 provides the ability to subscribe for patient data at VistA sites, fetch the data for all subscribed patients from VistA sites and cancel subscription for patient data at VistA sites. The subscription for patient data at a VistA site triggers a VistA routine that will start gathering available patient data in the site and place it into the VPR HMP/eHMP namespace. The subsequent data fetch calls for data at the VistA site to include a maximum number of records parameter that is used to restrict the number of records fetched in the call. The data fetch call returns VPR data for all subscribed patients at the site. Each data fetch call not only returns the specified number of records but also returns the number of records that were not yet fetched. Multiple data fetch calls have to be made in order to get all the VPR data for all patients from each site</p>	
eHMP/eHMP Read Template and Filter	
<p>HMP/eHMP uses the following template-filter combinations for retrieval of HMP/eHMP VPR from the HDR database:</p> <ul style="list-style-type: none">• Filter information will be added for final of SDD	

8. Human-Machine Interface

Not Applicable. HDR is a back end system only with no user interface.

8.1. Interface Design Rules

Not Applicable.

8.2. Inputs

Not Applicable.

8.3. Outputs

Not Applicable.

8.4. Navigation Hierarchy

Not Applicable.

9. Security and Privacy

9.1. Security

HDR is a major application and is classified at FISMA HIGH security category as it stores PII and PHI. HDR SCA was performed against the System Security Plan (SSP) and all supporting documentation for HDR. The SSP was created based on NIST 800-53 rev. 3 security controls consisting of 18 controls, plus the 8 privacy controls. The SSP discusses the management, operation and technical requirements and shows details and status each of the controls. The following table summarizes the security control status summary:

Table 31 Security Control Status

NIST SP800-53 Rev3	Implemented	Planned	Compensating	Not Implemented	Not Applicable	Risk Based Decision
Access Control	19	0	0	0	2	5
Audit and Accountability	12	0	0	0	0	4
Awareness And Training	1	0	0	0	0	0
Configuration Management	3	0	0	0	0	1
Contingency Planning	4	0	0	0	0	0
Identification and Authentication	7	0	0	0	1	6
Incident Response	1	0	0	0	0	0
Maintenance	1	0	0	0	0	0
Media Protection	0	0	0	0	0	0
Personnel Security	1	0	0	0	0	0
Physical And Environmental Protection	0	0	0	0	0	0
Planning	3	0	0	0	0	0
Program Management	0	0	0	0	0	0
Risk Assessment	2	0	0	0	0	0
Security Assessment And Authorization	0	0	0	0	1	0
System And Communications Protection	19	0	0	0	1	3
System And Information Integrity	11	0	0	0	0	1
System And Services Acquisition	8	0	0	0	0	0
Total	92	0	0	0	5	20

All of the accreditation documentation has been completed along with the supporting documents for the SSP. For access to the SSP, PTA and other accreditation and supporting documents, please contact the HDR ISO.

9.2. Privacy

HDR is a privacy sensitive IT system. A Privacy Threshold Analysis (PTA) was performed and approved in April 2014. A Privacy Impact Assessment (PIA) was completed in August of 2011 and is publicly available at <http://www.privacy.va.gov/privacyimpactassessment.asp>. There have been no major changes since that time and thus the privacy officers have determined that

the PTA performed in April is sufficient. The eight (8) Privacy control families from NIST 800-53 rev. 4 have been documented in the SSP:

- Authority and Purpose (AP);
- Accountability, Audit, and Risk Management (AR);
- Data Quality and Integrity (DI);
- Data Minimization and Retention (DM);
- Individual Participation and Redress (IP);
- Security (SE);
- Transparency (TR); and,
- Use Limitation (UL).

All of the accreditation documentation has been completed along with the supporting documents for the SSP. For access to the SSP, PTA and other accreditation and supporting documents, please contact the HDR ISO.

Appendix A. Additional Information

Appendix A.1. RTM

The RTM will be posted in the HDR 3.8 TSPR site listed in Appendix A.5.

Appendix A.2. Packaging and Installation

Refer to the HDR 3.8 Installation Guide that will be posted in HDR 3.8 TSPR site.

Appendix A.3. Design Metrics

Refer to HDR 3.8 SDD, Volume 2 for detailed software design.

Appendix A.4. Acronym List and Glossary

The abbreviations, acronyms, and definitions referenced in this document are provided in the “Repositories Abbreviations, Acronyms, and Definitions” section of the Technical Services Project Repository (TSPR) located at the following URL:



If you do not have VA network or TSPR access, contact the author of the document or your Repositories point of contact to request access.

Appendix A.5. Required Technical Documents

The artifacts referenced in this document are located in the “Documentation Reference Library” in the Technical Services Project Repository (TSPR) at the following link:



If you do not have VA network or TSPR access, contact the author of the document or your Repositories point of contact to request access.

Appendix B. Data Elements

Appendix B.1. CDS Supported Domain Entry Points

CDS Supported Domain	Domain Entry Point
Intolerance Condition	intoleranceCondition
Allergy Assessment	allergyAssessment
Vital Signs	vitalSignObservationEvent
Outpatient Pharmacy	outpatientMedicationPromise
Lab Chemistry/Hematology	labTestPromise

CDS Supported Domain	Domain Entry Point
Text Integration Utility	clinicalDocumentEvents
Non-VA Medications	nonVaMedicationEvents
Immunizations	immunizationEvents
Skin Tests	skinTestProcedures
Problem List	healthConcerns
HTH Activities of Daily Living	adlSurveyResponse
Patient Satisfaction Survey Request	psSurveyResponse
Census	CensusSurveyResponse
Appointments	appointments
DMP	dmpSurveyResponse dmpSurveyResponseWithScore
Exam	Exam2507 ExamRequest2507
HMP/eHMP	GenericObservation
PGD	PatientGeneratedDataEvent

Appendix B.2. CDS Supported Templates and Filters

CDS Template	Applicable Filter
<i>AllergyAssessmentRead0010</i>	AASINGLEPATIENTALLDATAFILTER
<i>AllergiesRead0010</i>	ALLERGYSSINGLEPATIENTALLDATAFILTER CHDRALLERGYSSINGLEPATIENTFILTER
<i>IntoleranceConditionRead0010</i>	CHDRALLERGYSSINGLEPATIENTFILTER ICSINGLEPATIENTALLDATAFILTER
<i>IntoleranceConditionReducedRead0011</i>	CHDRALLERGYSSINGLEPATIENTFILTER ICSINGLEPATIENTALLDATAFILTER
<i>LabRead0010</i>	CHDRLABSSINGLEPATIENTFILTER LABSSINGLEPATIENTALLDATAFILTER
<i>MHVIntoleranceConditionRead0011</i>	ICSINGLEPATIENTALLDATAFILTER
<i>MHVLabRead0011</i>	LABSSINGLEPATIENTALLDATAFILTER.XSD
<i>MHVVitalsignsRead0010</i>	MHVVitalSinglePatientFilter
<i>PharmacyRead0010</i>	CHDRRXSSINGLEPATIENTFILTER RXSSINGLEPATIENTALLDATAFILTER RXSSINGLEPATIENTSINGLERECORDFILTER
<i>RDIAllergiesPharmacyRead0013</i>	RDIALLERGYRXSSINGLEPATIENTFILTER
<i>TiuDocumentListRead2</i>	TIUSINGLEPATIENTLISTDATAFILTER
<i>TiuDocumentDetailRead2</i>	TIUSINGLEPATIENTDETAILDATAFILTER
<i>VitalsignsRead0010</i>	VITALSSINGLEPATIENTALLDATAFILTER VWVITALSSINGLEPATIENTFILTER
<i>VWAllergiesRead0010</i>	VWALLERGYSSINGLEPATIENTFILTER
<i>VWIntoleranceConditionRead0010</i>	VWICSINGLEPATIENTFILTER
<i>VWPharmacyRead0010</i>	RXSSINGLEPATIENTALLDATAFILTER

CDS Template	Applicable Filter
	RX_SINGLEPATIENT_SINGLERECORD_FILTER
<i>VWPharmacyDetailRead0010</i>	RX_SINGLEPATIENT_ALLDATA_FILTER RX_SINGLEPATIENT_SINGLERECORD_FILTER
<i>NonVAMedicationsRead2</i>	NONVAMEDS_SINGLEPATIENT_FILTER
<i>ImmunizationRead3</i>	IMMUNIZATIONS_SINGLEPATIENT_FILTER
<i>SkintestRead3</i>	SKINTESTS_SINGLEPATIENT_FILTER
<i>ProblemListRead2</i>	PROBLEMLISTS_SINGLEPATIENT_FILTER
<i>OutpatientPharmacyRead1</i>	RX_SINGLEPATIENT_ALLDATA_FILTER
<i>LabChemHemRead1</i>	LABS_SINGLEPATIENT_ALLDATA_FILTER
<i>HTResponse</i>	HTREPORT_FILTER
<i>AppointmentsRead1</i>	APPOINTMENTS_SINGLEPATIENT_FILTER
<i>RequestsAndExamsRead2</i>	REQUESTS_AND_EXAMS_SINGLEPATIENT_FILTER
<i>GenericObservationRead1</i>	GENERIC_VISTALIST_DATA_FILTER
<i>PGDRead1</i>	PGD_SINGLEPATIENT_FILTER PGD_UNIQUEID_FILTER

Appendix B.3. Write Request Examples

Figure 23 Allergies Write Request Template

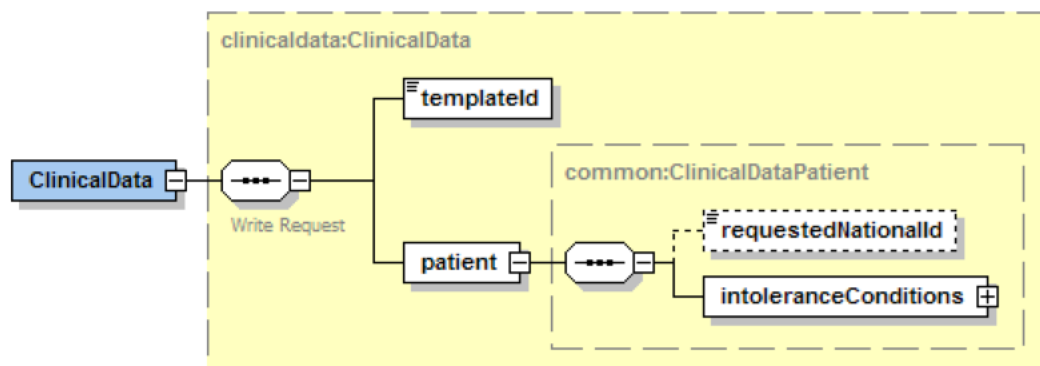


Figure 24 ADL Data Write Template

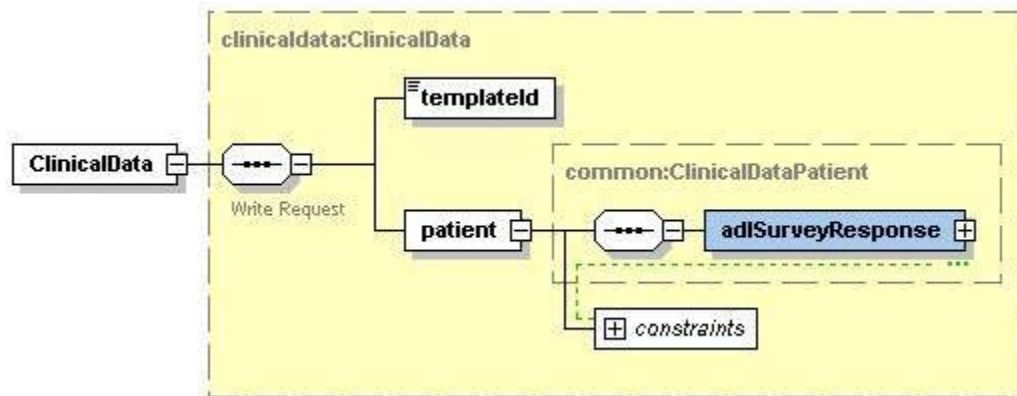


Figure 25 HTH DMP Response Write Template

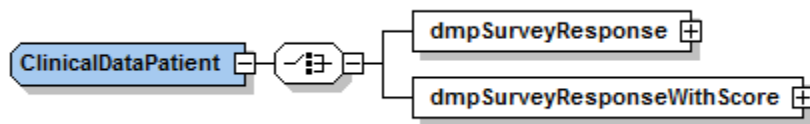
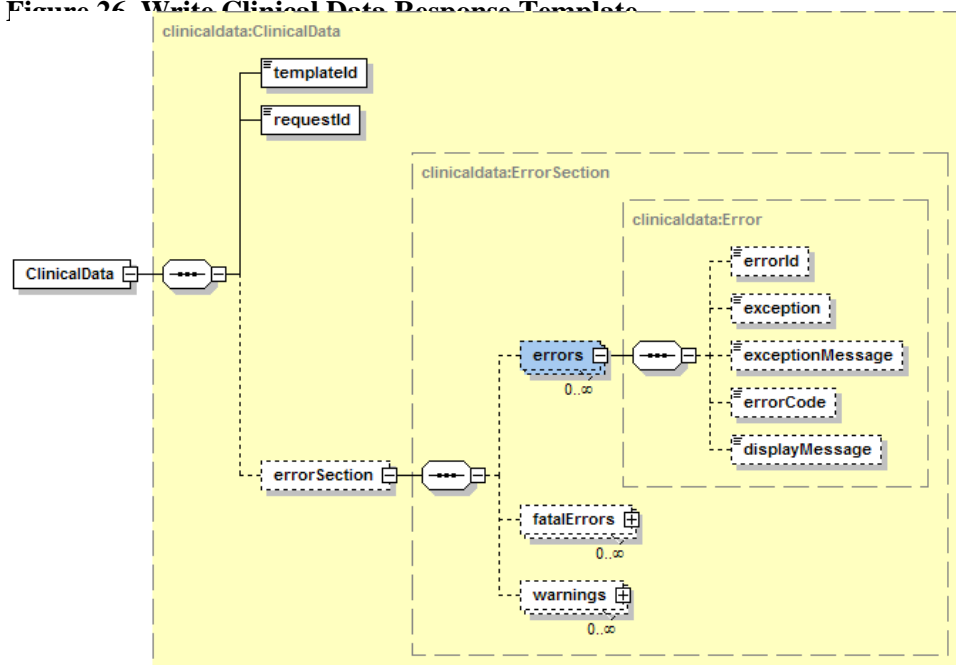


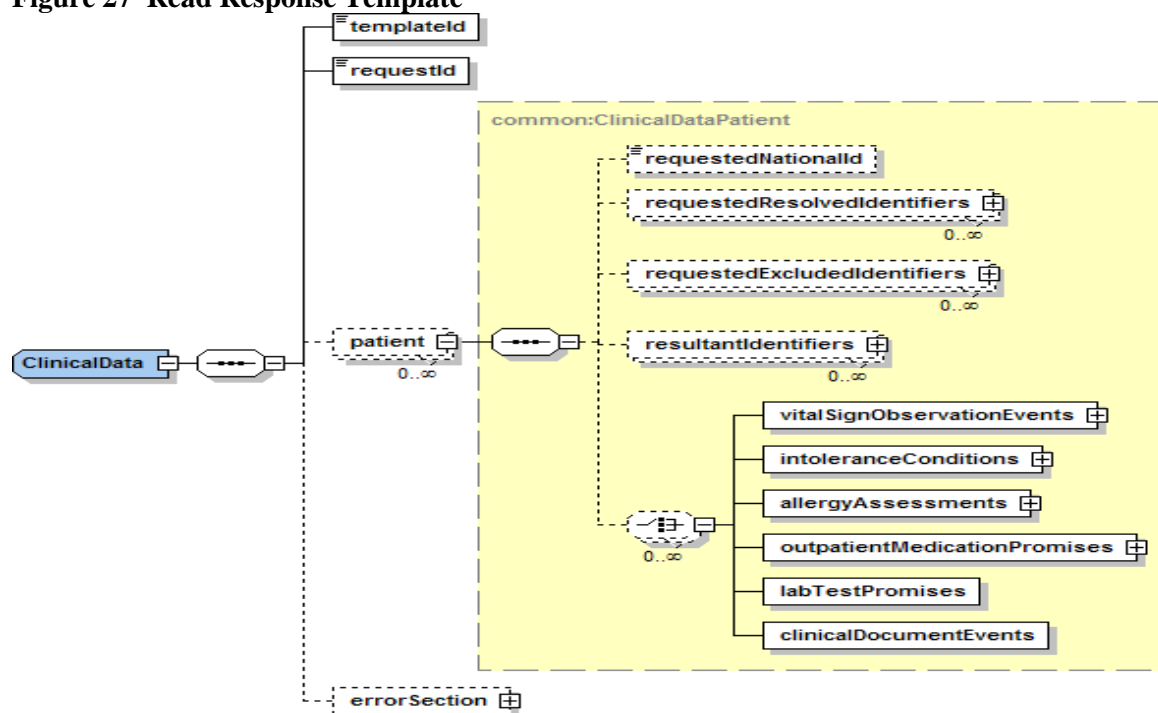
Figure 26 Write Clinical Data Response Template



Appendix B.4. Read Response Examples

The following is the general schematic of a read response – the examples following are the schematic of individual entry points such as vitalSignObservationEvents, intoleranceConditions, labTestPromises, etc.

Figure 27 Read Response Template



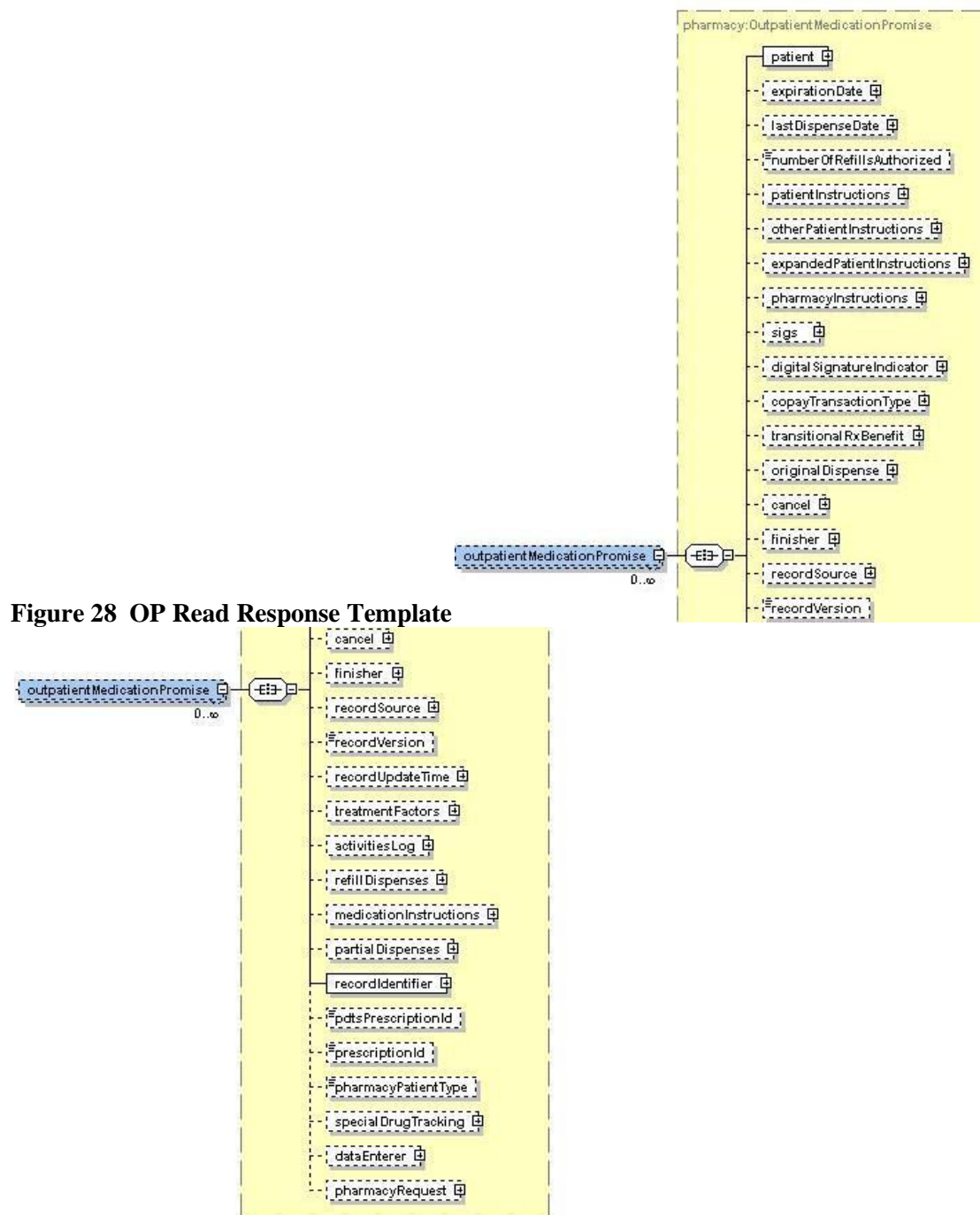


Figure 28 OP Read Response Template

Figure 29 Lab CH Read Response Template

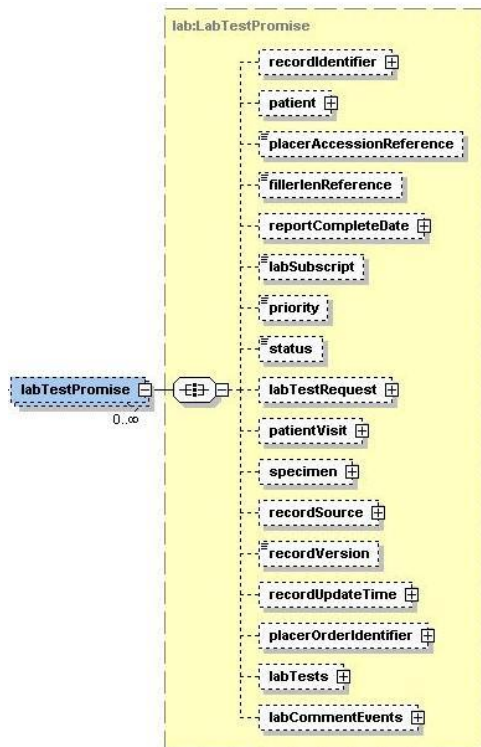


Figure 30 HTH Surveys Read Response Template

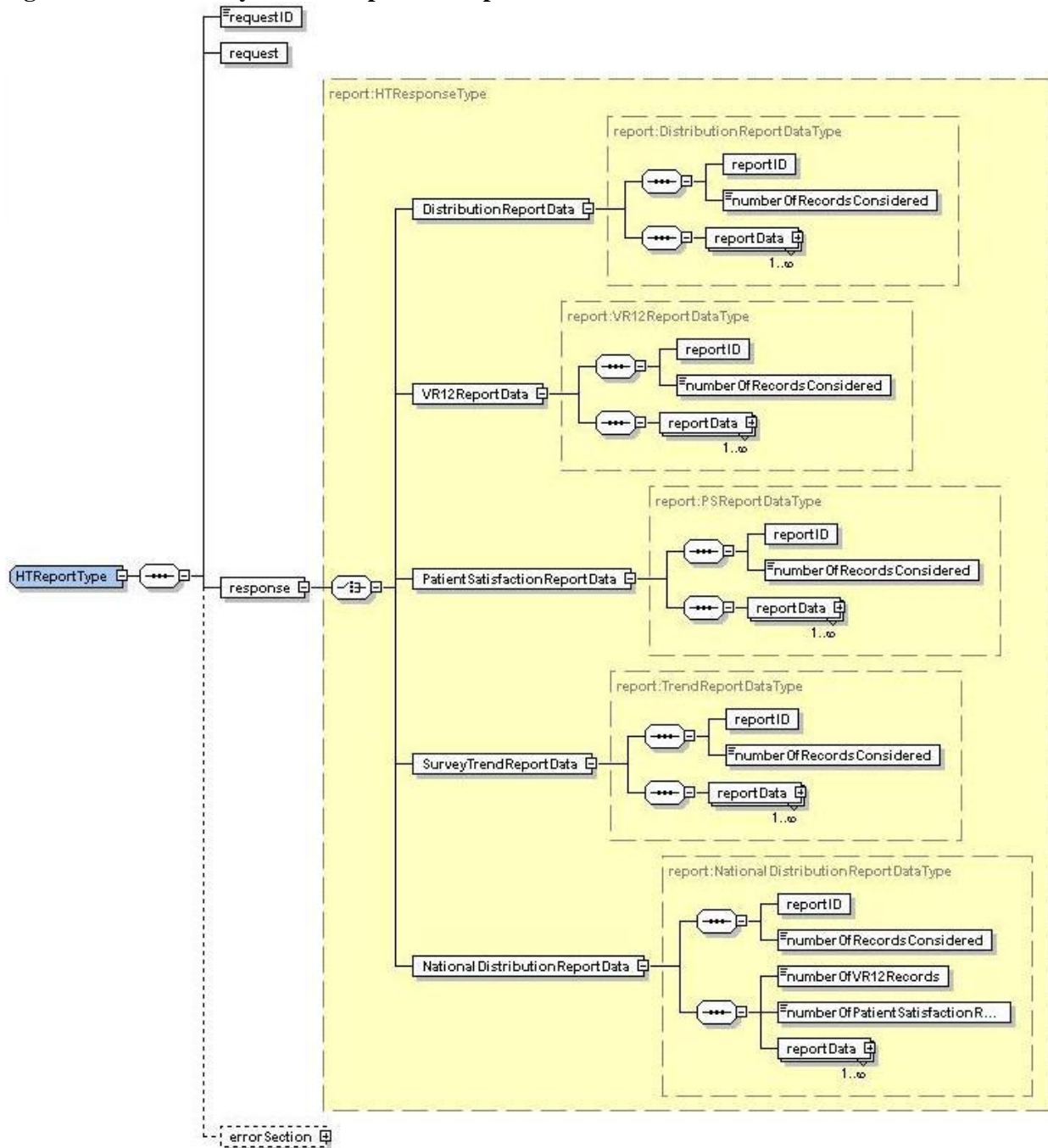


Figure 31 Non-VA Medications Read Response Template

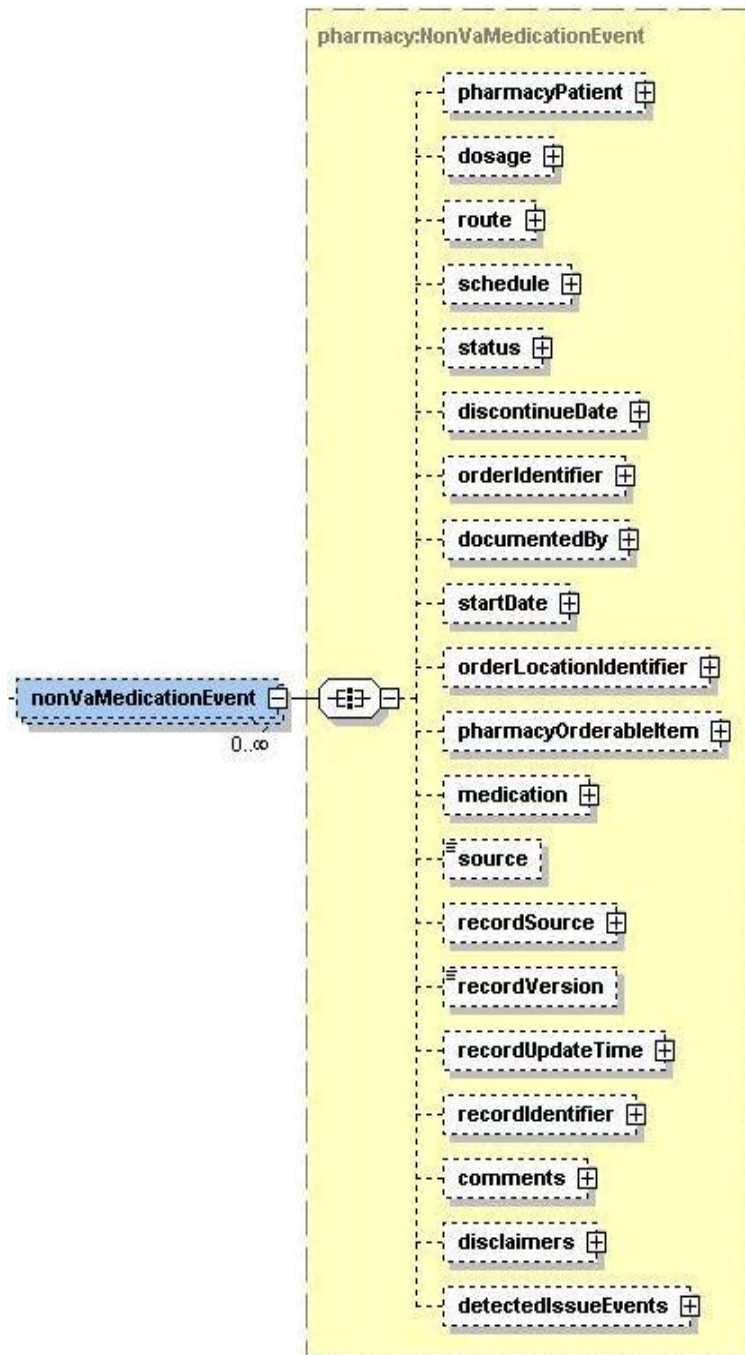


Figure 32 Immunization Read Response Template

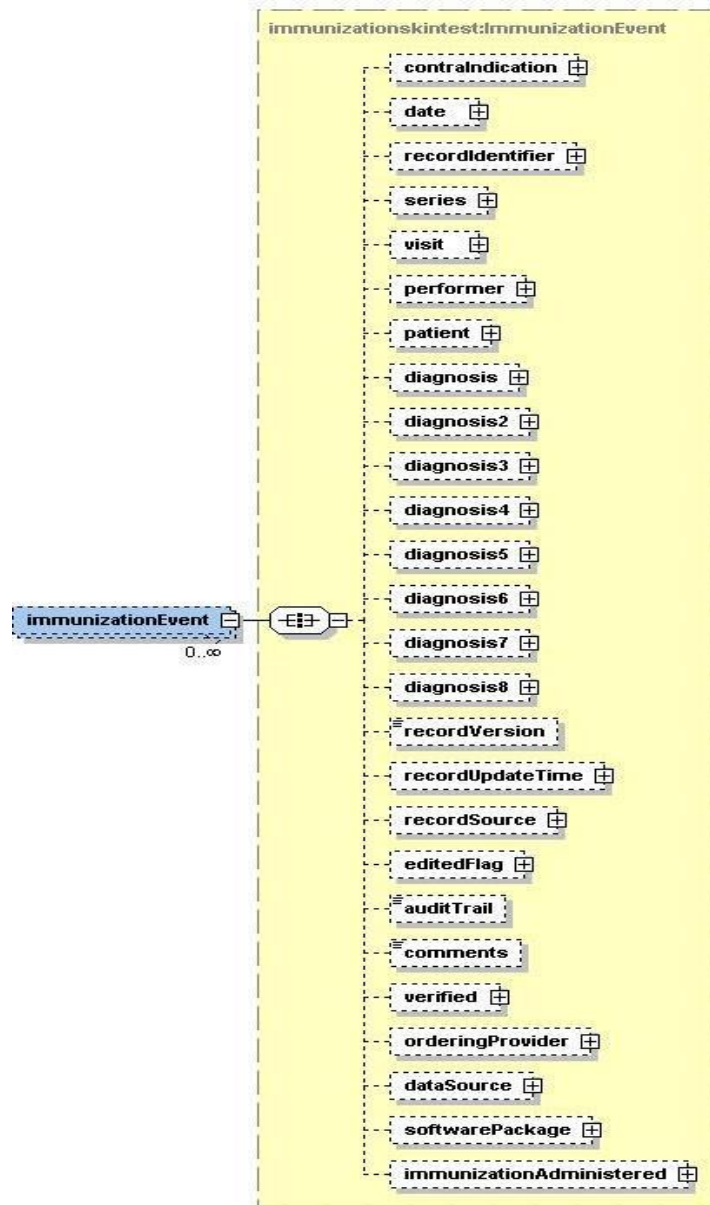


Figure 33 Skin Test Read Response Template

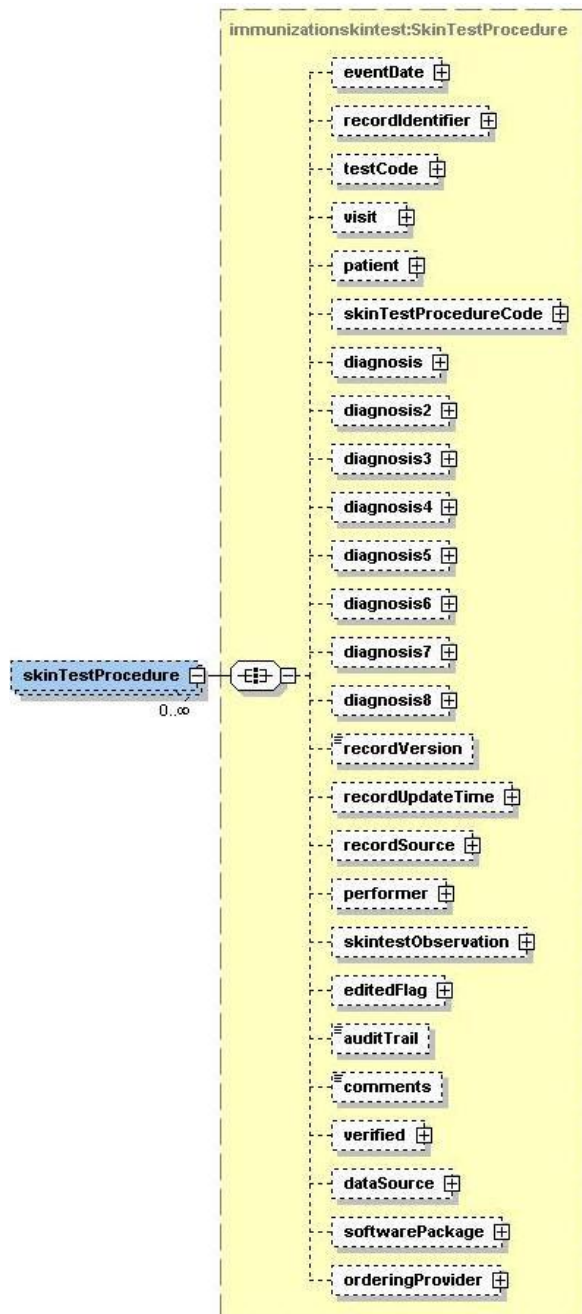


Figure 34 Problem List Read Response Template

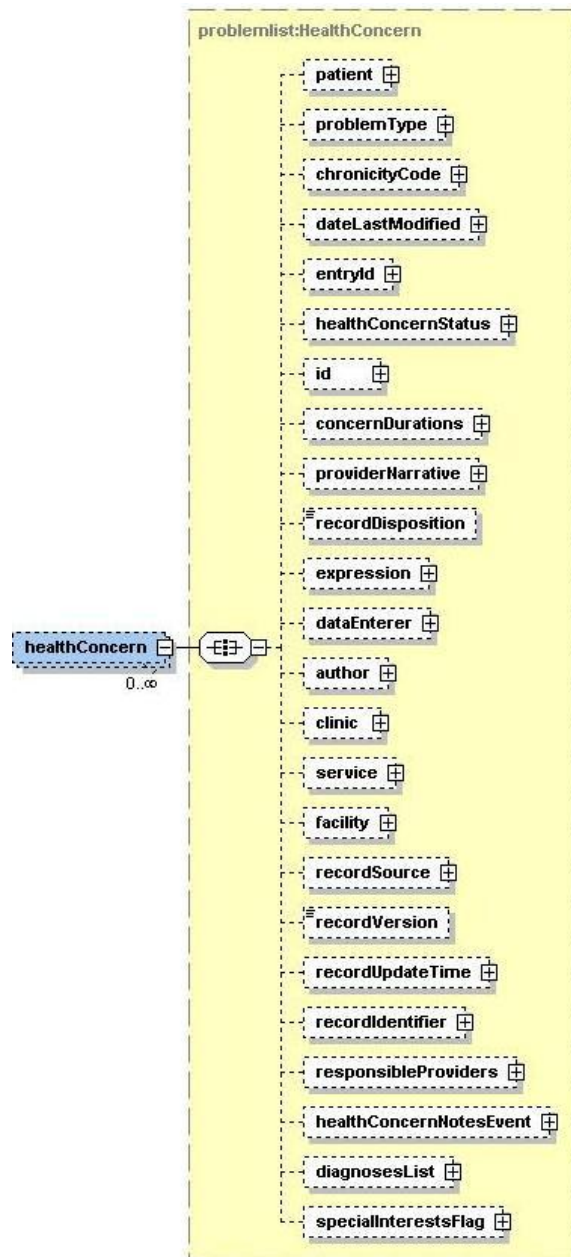


Figure 35 PSS Schematic

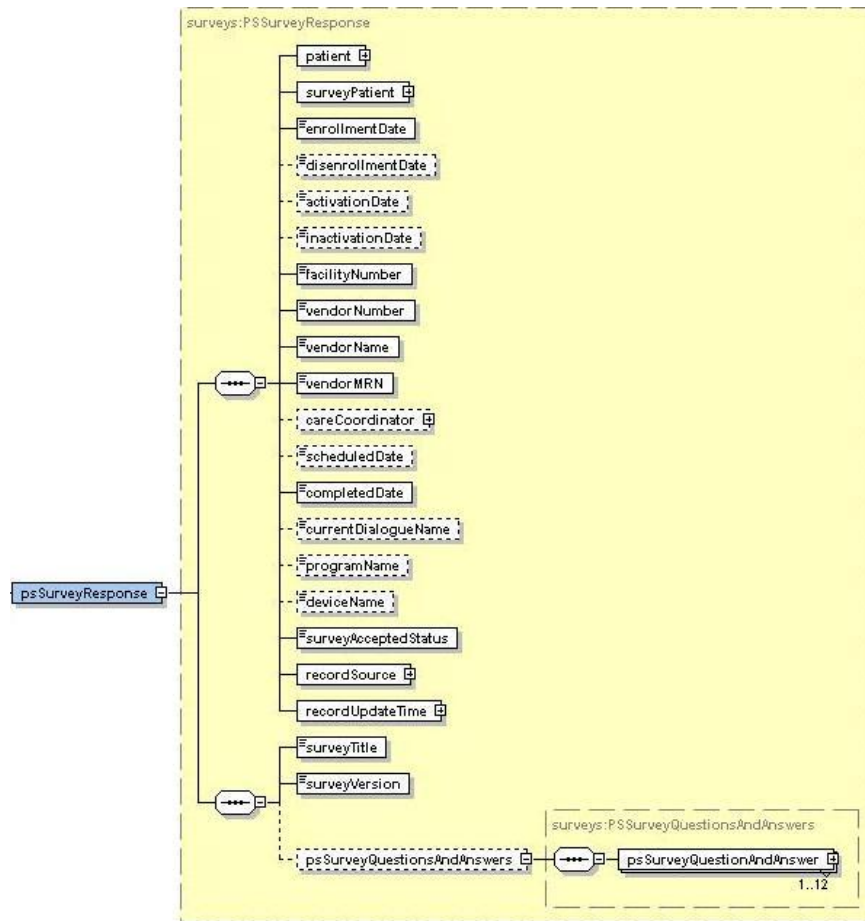


Figure 36 ADL Schematic

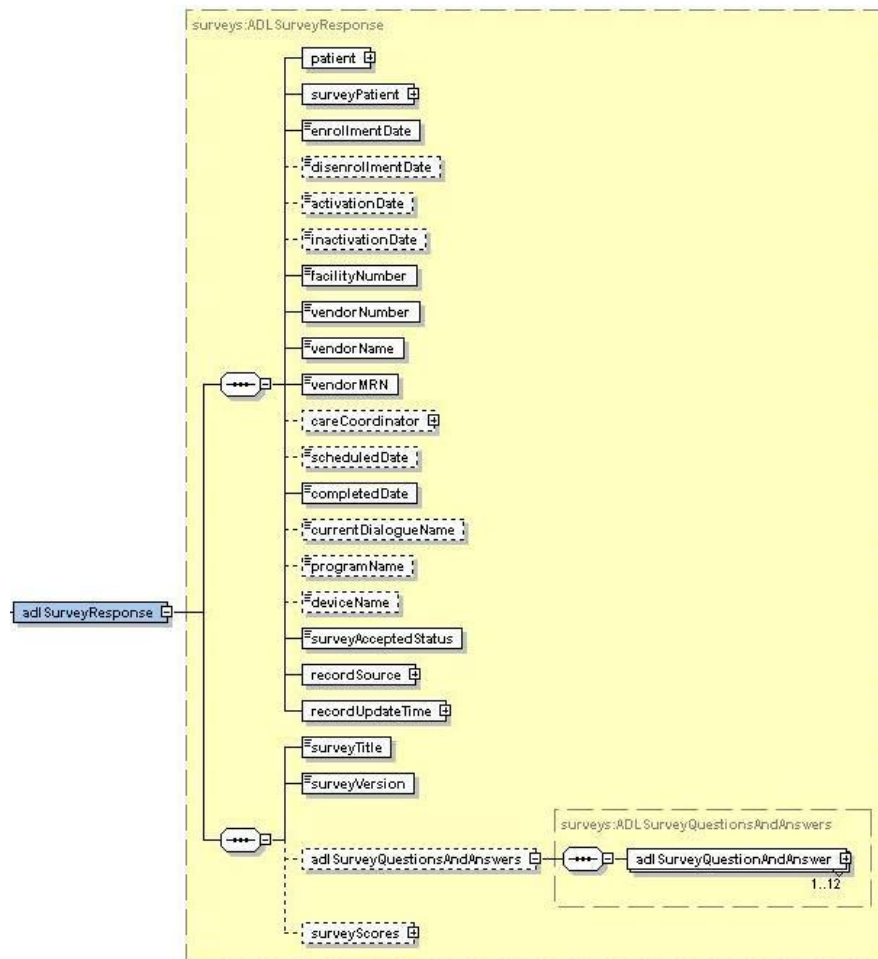


Figure 37 Appointment Read Response Template

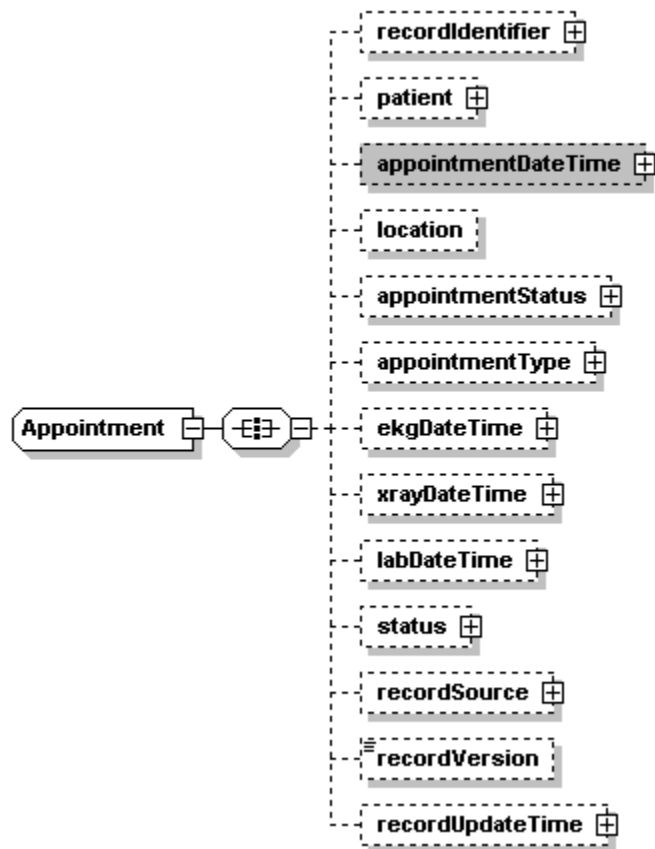


Figure 38 DMP Generic Response

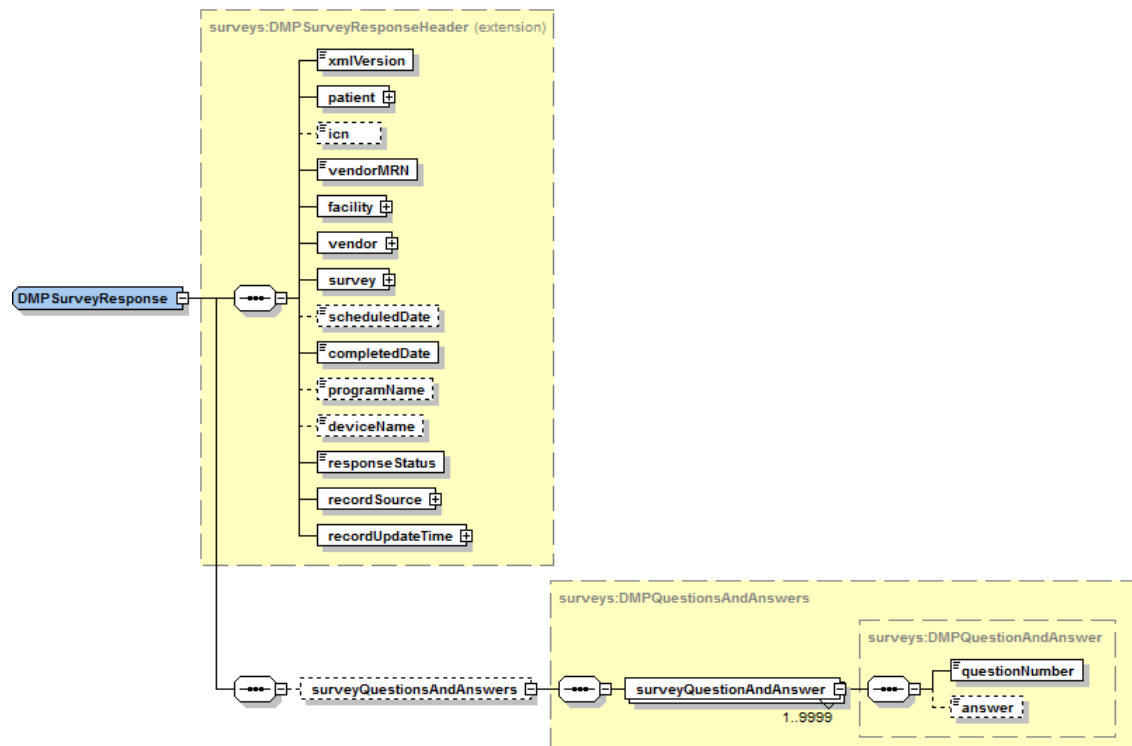
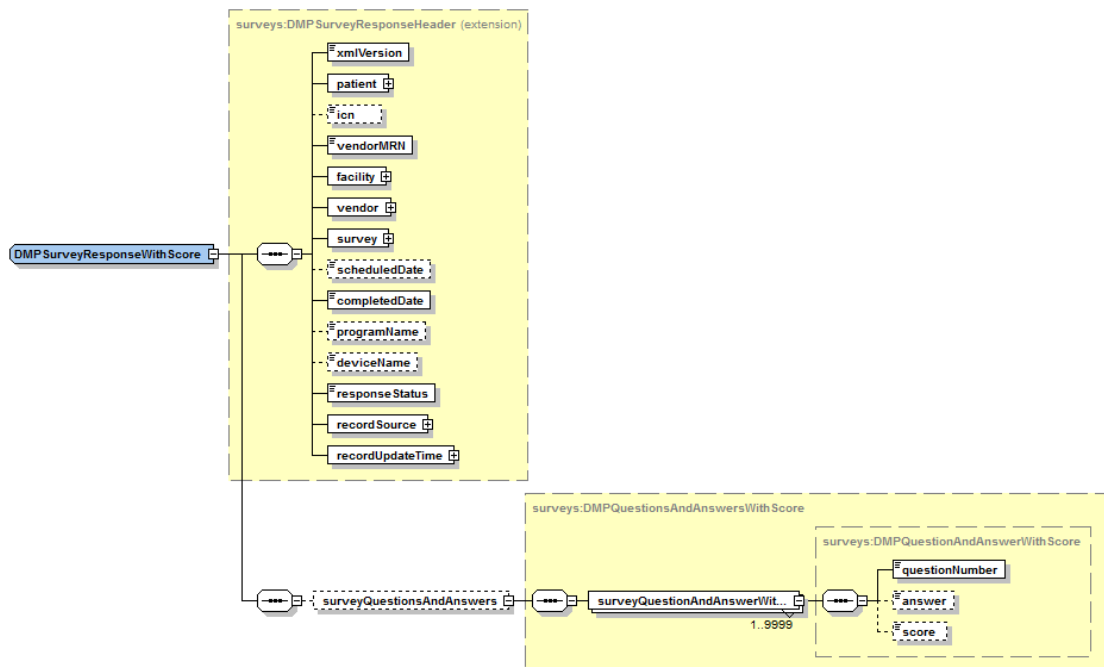
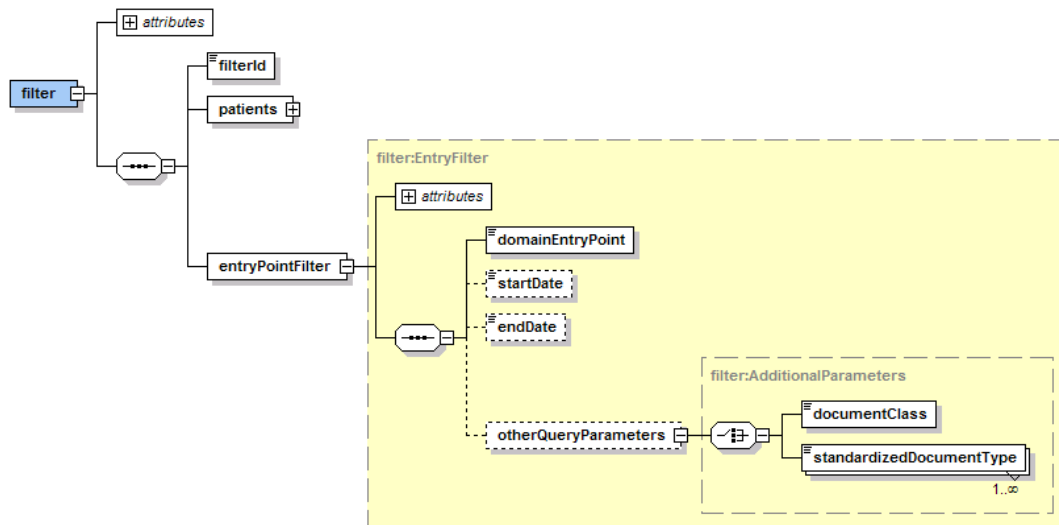


Figure 39 DMP Generic Response with Scores



Appendix B.5. Filter Examples

Figure 40 TIU Domain Read Filter



Note: Due to the hierarchical nature of the document class, and because a resolution of the subclass to the parent class is necessary, querying TIU data by document class will certainly be less responsive than querying TIU data by standardized document type.

Figure 41 Non-VA Medications Filter schematic

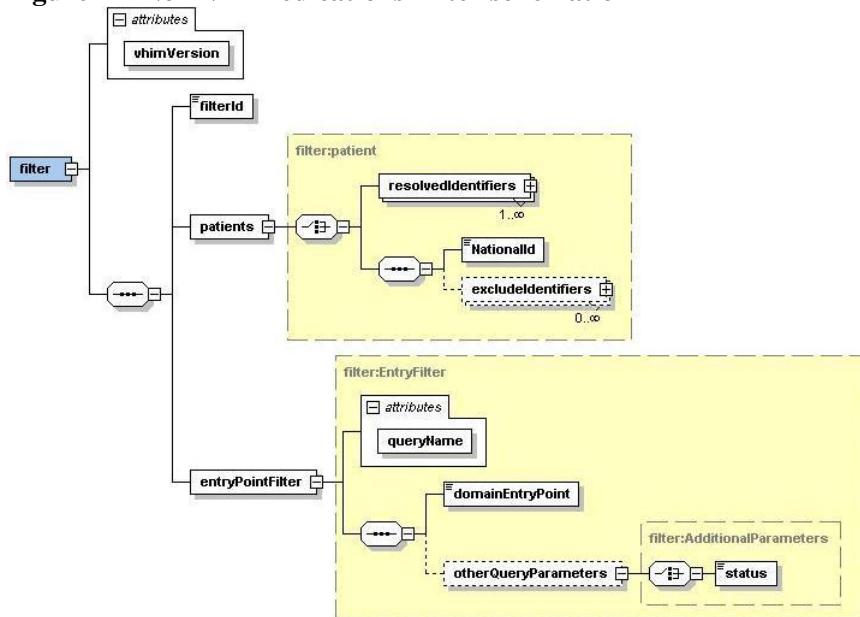


Figure 42 Immunizations Filter Schematic

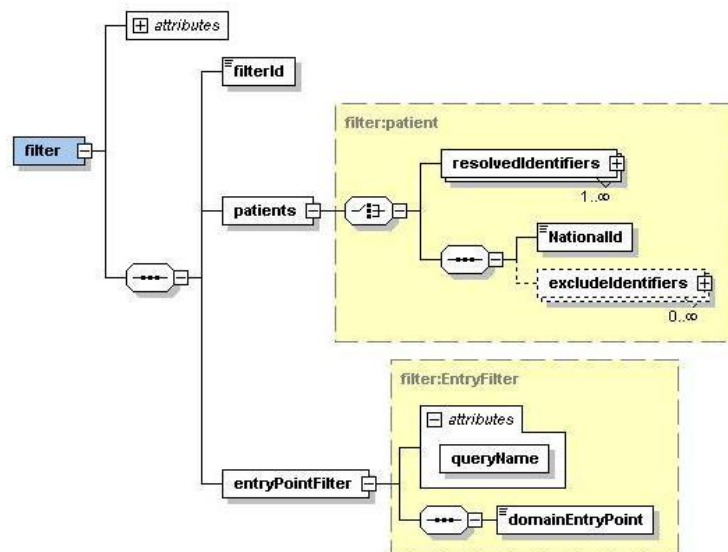


Figure 43 Skin Test Filter Schematic

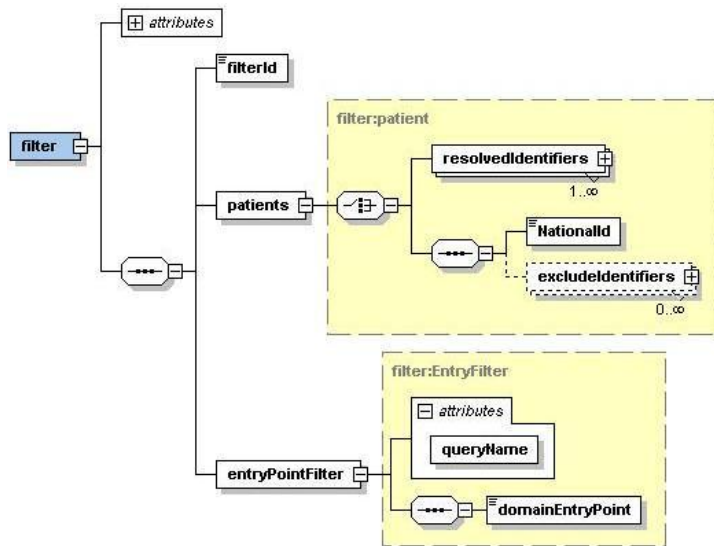


Figure 44 Problem List Filter Schematic

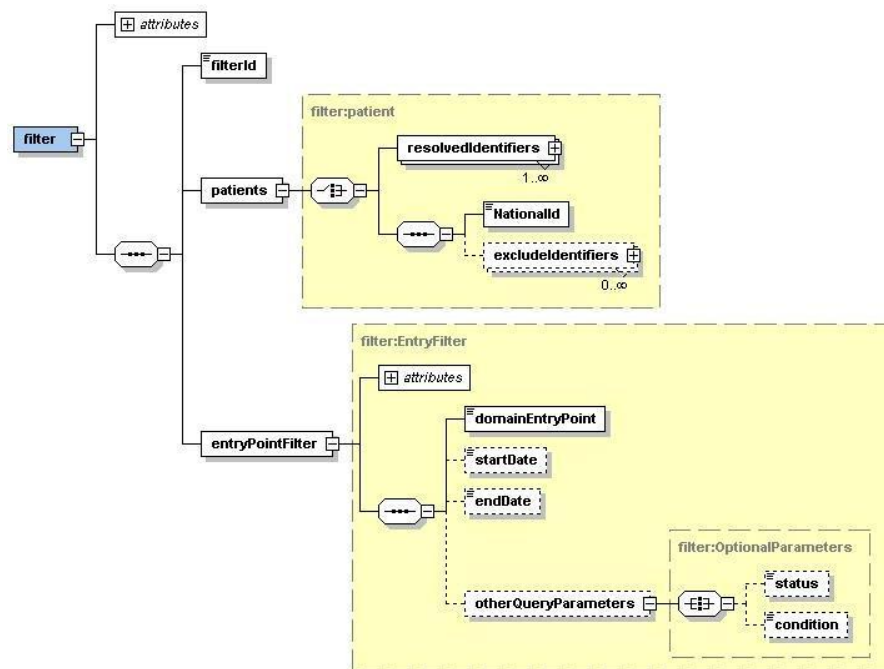


Figure 45 AppointmentSinglePatient Filter Schematic

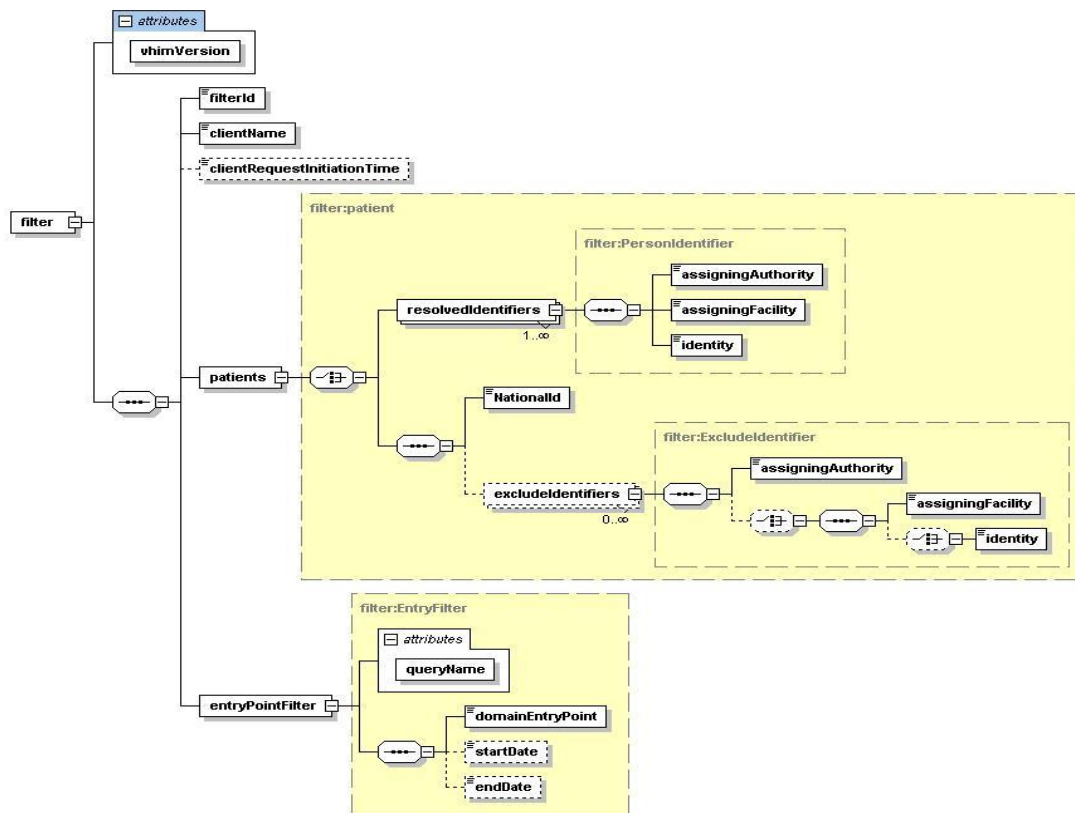


Figure 46 RequestsandExamsSinglePatient Filter Schematic

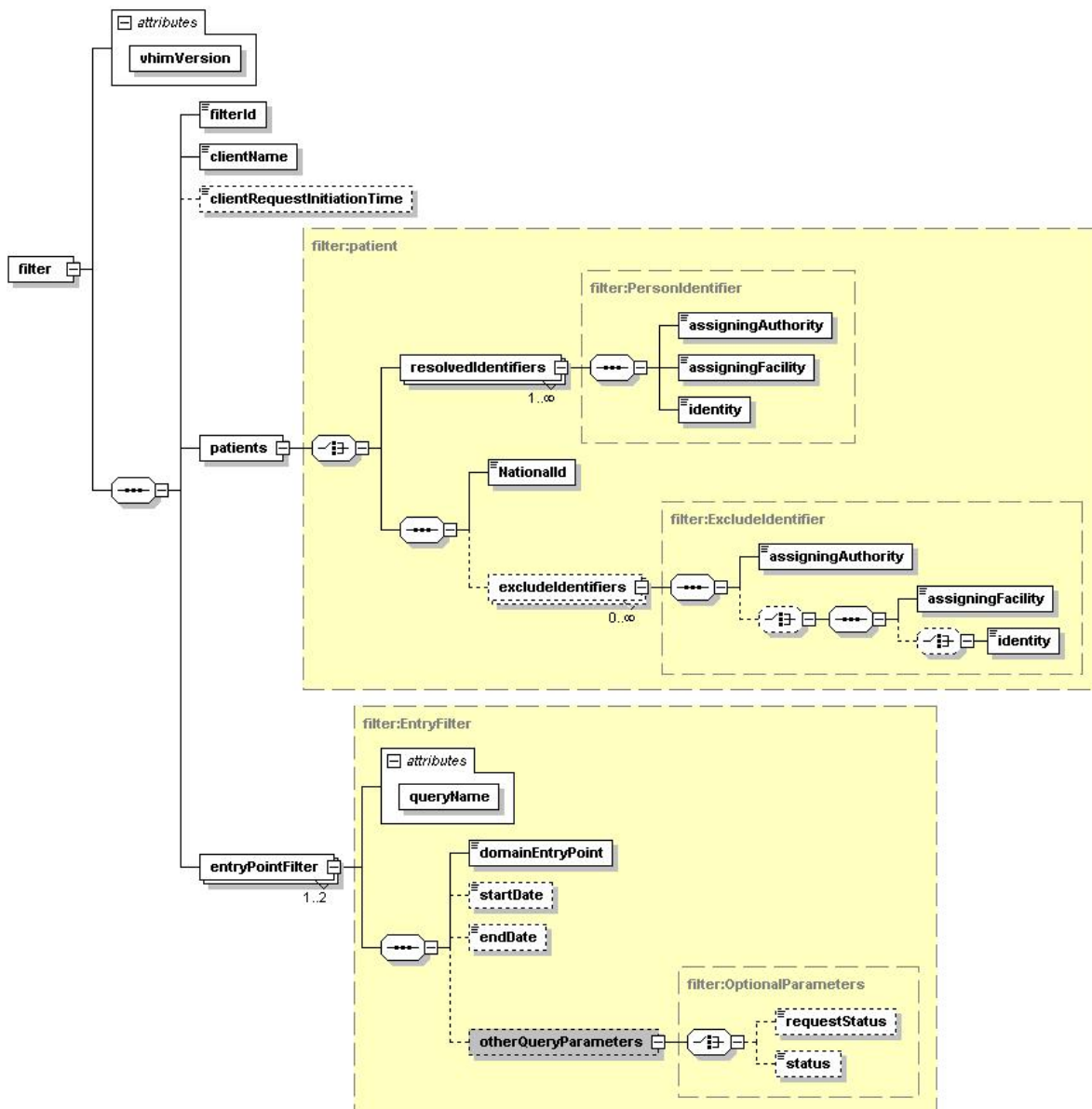


Figure 47 HTH Survey Reads Request Filter

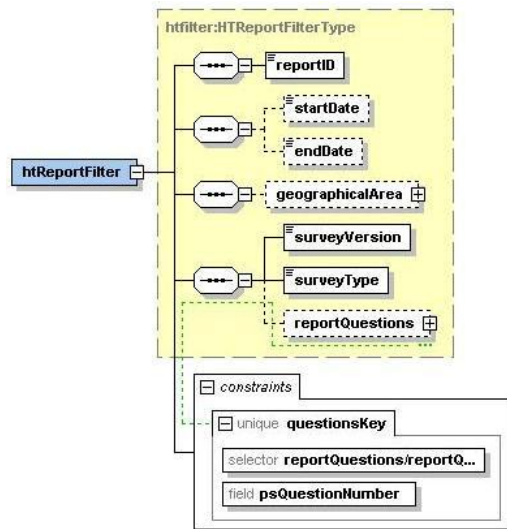


Figure 48 MHV Allergies Read Filter

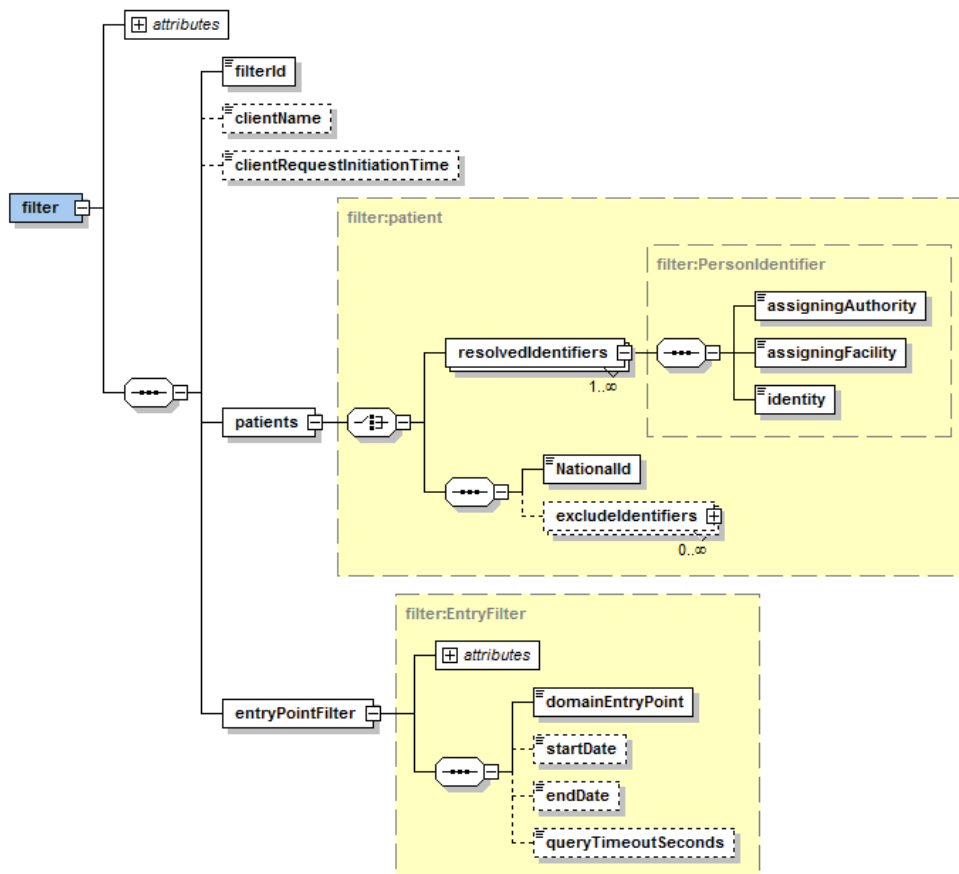


Figure 49 MHV Lab Read Filter

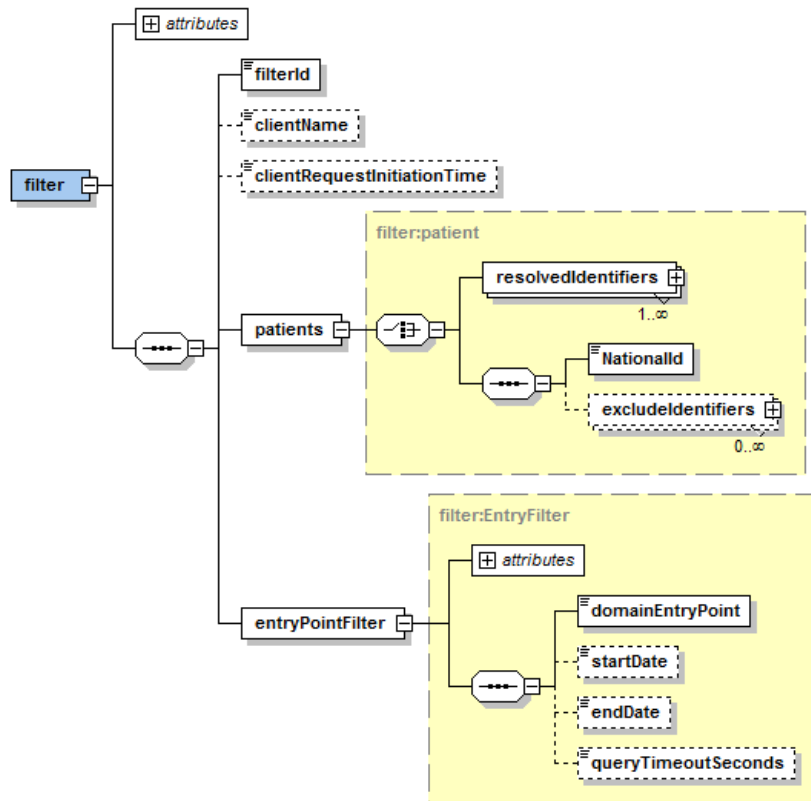


Figure 50 RDI Allergies and OP Reads Data Filter

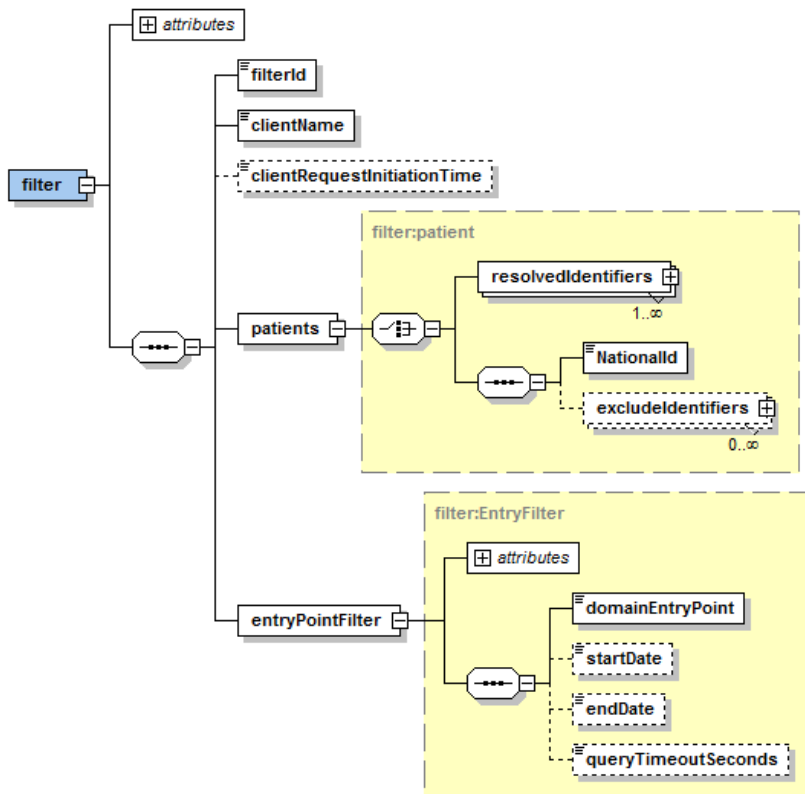
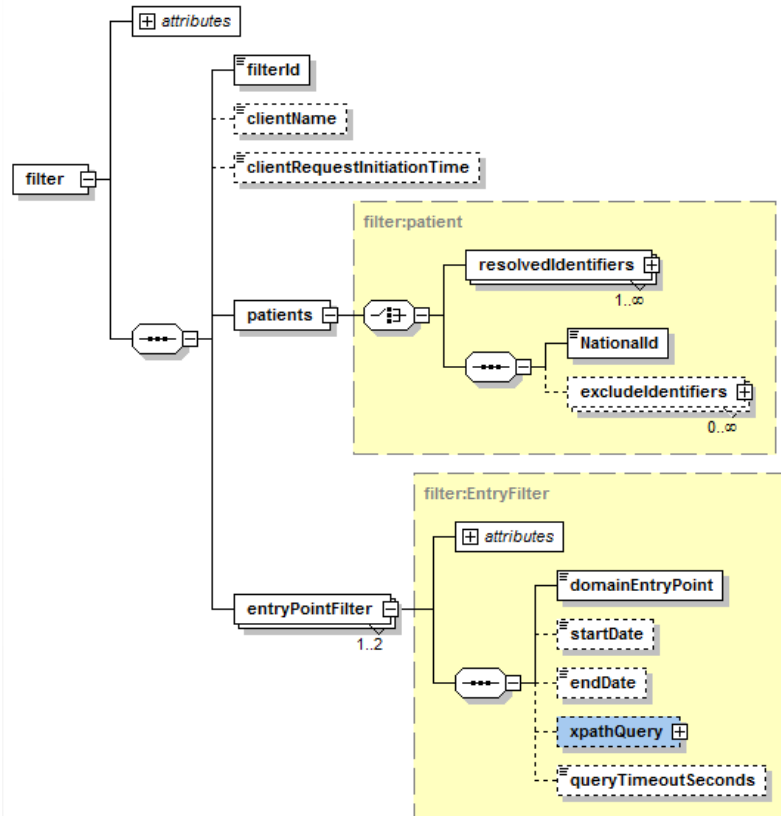
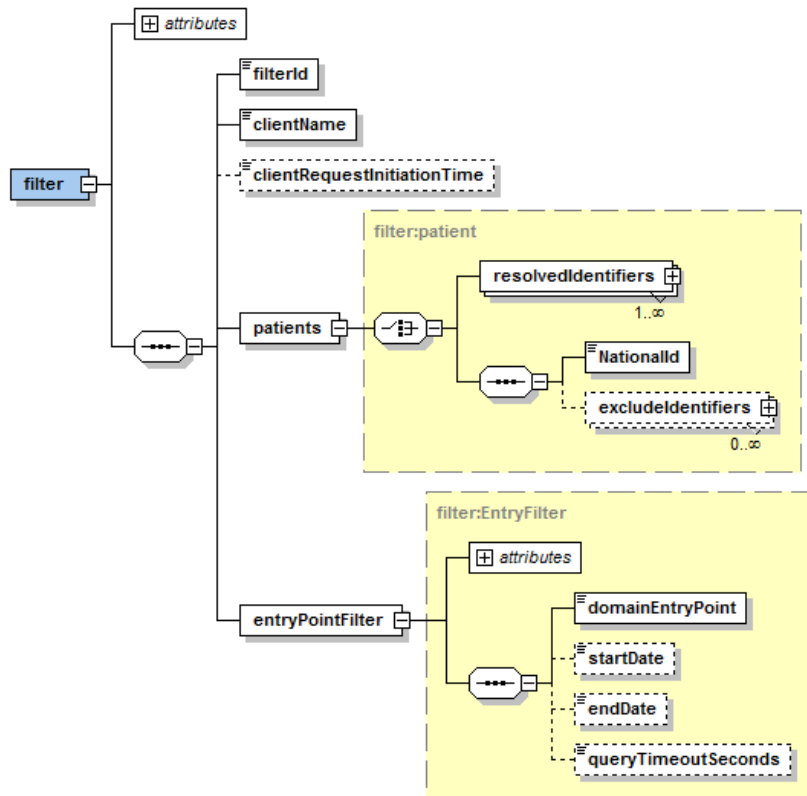


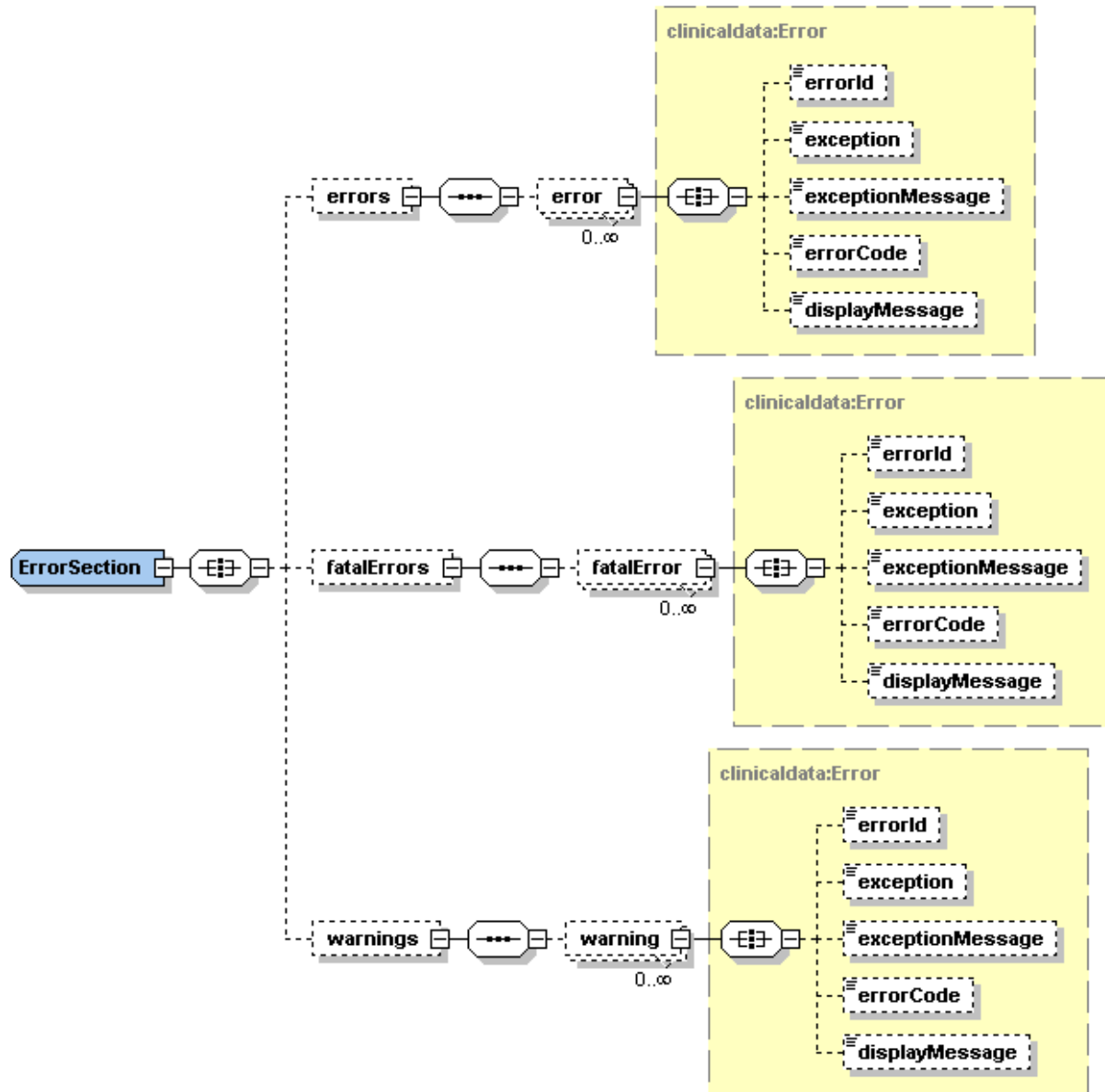
Figure 51 Mobile Health Appointments Read Data Filter



Appendix B.6. Error Messages

Processing errors that occur in the processing of a Read/Write request result in exceptions that generate internal error messages. These internal error messages are returned to the client application as part of the error section in the response message, detailed in the Figure below.

Figure 52 Error Section in Read request Response Message



The definitions of the fields for Error Type are as follows:

- **errorId**: an internally generated sequence value, not guaranteed to be unique, that links the error with internal CDS/Pathways log entries.
- **exception**: defines the general class of error (the Java class name in the CDS/Pathways exception framework).
- **exceptionMessage**: included for future scenarios in which it would be useful to capture stack-trace and external-dependency-generated messages. Currently, this value is usually null.

- **errorCode:** a computable identifier for the error. Currently, it is a string value. In the future it may be converted to another format.
- **displayMessage:** a parameterized text message, hopefully informative to the recipient.

Table 32 Exceptions and Errors for CDS 3.8 lists the possible errors that may be returned to a user in a response from HDR 3.8.

Table 32 Exceptions and Errors for CDS 3.8

Exception	Error Code	Display Message
Persistence Errors	HIBERNATESESSIONFAILURE	Unable to get the Hibernate session for the HDR database.
	HIBERNATEREADFAILURE	Failed to read from the Hibernate layer. The error is {0}.
	HIBERNATEDELETEFAILURE	Failed to delete using the Hibernate layer.
	NONUNIQUEIDENTITY	Could not find unique record with this identity: {0}. The reason is {1}.
	CANTCREATEIDENTITY	Failed to create an IdentityInterface object with this Identity: {0}.
	UNABLETORESOLVEWRITEABLE_PERSISTENCEMANAGERFAILURE	Unable to resolve writeable persistence manager applicable to template meta data.
	HDRIIOperationFailed	{0} Error: {1}. Working with object: {2}.
	HDRIConstraintViolated	HDR Integrity constraint violated: {0}.
	ROOTCAUSEMSG	CDS Persistence Failure Message: {1}.
Persistence Manager Exceptions	READPERSISTENCEMGRNOPERS ONWITHSITEID	There is no person identifier in the list of person identifiers that has a site identifier of {0}.
	INVALIDORUNEXPECTEDQUERY_PARAM	The query parameter '{0}' is not expected or is invalid.
	READPERSISTENCEMGRNULLPERSONIDENTIFIERLIST	The persistence manager {0} requires that the person identifiers list contains values
Model Assembler Exception	MODELASSEMBLEREXCEPTIONNO TALLRESULTSASSEMBLED	Not all results were assembled. The domain is '{0}' and the association name is '{1}'.
Request Error	REQUESTTYPE_MISMATCH_OPERATION_VS_INSTANCE	The API request operation must match the request type defined in the instance XML.
	CDS2XSERVICEFAILURE	Service Call failed on CDS 2xservice. The exception is {0} and the reason is {1}
Write Requests	WRITEREQUESTXMLNULL	Write request XML cannot be null or empty.
	WRITEREQUESTREQUESTIDNULL	Write request requestId cannot be null or empty.
	WRITEREQUESTTEMPLATEIDINVALID	Invalid templateId {0}. The schema is not loaded in cache. The reason is {1}

Exception	Error Code	Display Message
	WRITE_REQUEST_SIMPLE_INVALID_TEMPLATEID WRITE_REQUEST_TEMPLATEID_NULL	Invalid template identifier {0}. Write request template Id cannot be null or empty.
Read Exceptions	UNEXPECTED_READ_EXCEPTION	An unexpected exception occurred while reading data for patient. Assigning Facility is {0}. Patient ID is {1}. Domain entry point is {2}. The message is {3}
	READ_REQUEST_TEMPLATEID_NULL	Read request template id cannot be null or empty.
	READ_REQUEST_FILTERID_NULL	Read request filter id cannot be null or empty.
	READ_REQUEST_FILTER_XML_NULL	Read request filter XML cannot be null or empty.
	READ_REQUEST_REQUESTID_NULL	Read request requestid cannot be null or empty.
	READ_REQUEST_FILTER_VHIM_VERSION_INVALID	Read request filter VHIM version is invalid {0}.
	READ_REQUEST_TEMPLATEID_INVALID	Invalid templateId {0}. The schema is not loaded in cache.
	REQUEST_TYPE_MISMATCH_TEMPLATE_VHIM_VERSION_FILTER_VHIM_VERSION	The requested Read template VIM version {0} must match the filter VHIM version {1}.
	REQUEST_TYPE_MISMATCH_TEMPLATE_ENTRYPOINT_FILTER_ENTRYPOINT	The requested Read template entry points {0} must be a subset of the filter entry points {1}.
	READ_RESULT_INCORRECTLY_ACCESSED	The state of the ReadResult does not support this operation.
	READ_REQUEST_DATA_SOURCE_FAILURE	Assigning Facility is {0}. Patient ID is {1}. Domain entry point is {2}. Database is {3}. The Error reason is {4}, , LoggingSeverity.WARNING.
	READ_REQUEST_ALL_DATA_SOURCES_FAILED	All datasources failed. Unable to get data from applicable datasource(s) due to the following reason:\n {0}.
	READ_REQUEST_WORK_SCHEDULE_EXCEPTION	Could not schedule work, LoggingSeverity.WARNING.
	READ_REQUEST_WORK_WAIT_INTERRUPTED_EXCEPTION	InterruptedException occurred while waiting for scheduled work to finish, LoggingSeverity.WARNING.
	CANT_MATCH_TEMPLATE_TO_DELEGATE	Cannot match template {0} to available service delegate.
	READ_REQUEST_INPUT_PARAMETER_NULL	Read request cannot have empty input parameters(s). Empty input parameter(s) are {0}

Exception	Error Code	Display Message
	WRITEREQUESTINPUTPARAMETER_NULL	Write request cannot have empty input parameters(s). Empty input parameter(s) are {0}
	READQUERYSTRATEGY_NULL_FAILURE	The Read query strategy for the template {0} and the domain entry point {2} is not defined and is expected to be LoggingSeverity.WARNING
	INVALIDASSIGNINGFACILITIESIN_FILTERXMLREQUEST	Invalid Assigning Facility mentioned in the Filter, LoggingSeverity.WARNING
	ALLASSIGNINGFACILITIESINVALIDIN_FILTERXMLREQUEST	All the Assigning Facilities mentioned in the Filter are invalid and they are {0}.
XML Parsing Exception	XSLTHELPERTRANSFORMATION	An error occurred while transforming the data {0}.
XML Aggregation Exceptions	XMLAGGREGATIONZEROINPUTSOURCES	XML aggregation requires at least one XML input source.
	XMLAGGREGATIONVHIMVERSION_INVALID	No XML Aggregator found for VIM version {0}.
	XMLAGGREGATIONFAILED	XML aggregation failed.
Filter Errors	INVALIDXPathQUERY	Invalid xpath query.
	ERRORMARSHALLINGFILTERXML	Error unmarshalling filter XML. Filter XML was {1}. The reason is {0}.
	NOENTRYPOINTSINFILTER	No entry points specified in filter: {0}.
	NOPATIENTIDSREQUESTED	No patient identifiers are requested in the filter.
	NOPATIENTIDSRESOLVED	No patient identifiers could be resolved from the filter
	ALLPATIENTIDSEXCLUDED	All patient identifiers were excluded by the filter.
	PATIENTIDMISSINGASSIGNINGAUTHORITY	No assigning authority value in the patient identifier.
	PATIENTIDMISSINGASSIGNINGFACILITY	No assigning facility value in the patient identifier.
	PATIENTIDMISSINGIDENTITY	No identity value in the patient identifier.
	CANNOTPARSEDATE	Cannot parse the date {0} specified in the filter.
Filter Patient Resolver Exceptions	FILTERPARSERDOMEXCEPTION	Unable to parse the request filter {0}.
	MULTIPLENATIONALIDSINFILTER	Multiple national identifiers specified in the filter {0} (only 1 permitted per patient).
	ERROROBTAININGCORRESPONDINGIDS	Error while obtaining corresponding IDs for National Id {0}.
	ERRORADDINGRESOLVEDIDENTIFIERTO_FILTER	Error adding resolved identifiers to filter.
Schema	SCHEMAVALIDATIONFAILED	Schema validation failed because of {0}.

Exception	Error Code	Display Message
Helper Exceptions	ERROR_VALIDATEING_XML_AGAINST_COMPILED_SCHEMA	Error occurred when validating an XML instance against a compiled schema. The reason is {0}, LoggingSeverity.ERROR.
	SCHEMA_XML_NULL	Schema XML document cannot be null.
	FATAL_ERROR_VALIDATEING_XML_AGAINST_COMPILED_SCHEMA	Fatal error occurred when validating an XML instance against a compiledSchema. Reason is {0}
Template Cache Exceptions	CANNOT_CREATE_TEMPLATE_CACHE_PERSISTENCE	Problem occurred creating template cache persistence: {0}.
	CANNOT_SAVE_TEMPLATE	Problem occurred when persisting a template jar: {0}. The reason is {1}.
	CANNOT_LOAD_TEMPLATE_CACHE	Problem occurred when loading template cache. The reason is {0}.
	CANNOT_LOAD_SCHEMA_INTO_TEMPLATE_CACHE	Problem occurred when trying to load a schema into the template cache: {0}. The reason is {1}.
Template Meta Data Provider Exception	CANNOT_LOAD_TEMPLATE_METADATA	Problem loading template meta data for templateId {0}. The reason is {1}.
SymbolMap Provider Exception	CANNOT_CREATE_TEMPLATE_SCHEMA_SYMBOL_MAP	Problem creating the symbol map for a template schema: {0}. The reason is {1}.
Template Service Exceptions	VTSSERVICE_FAILED	The Template Service (VTS) failed: {0}.
	CANNOT_MANIPULATE_TEMPLATE_DATA	The data in the template cannot be manipulated correctly: {0}.
	CANNOT_LOAD_TEMPLATE_FROM_VTS	Problem loading the template with templateId {0} from the VHIM Template Service. The reason is {1}.
	NOT_TEMPLATE_FOUND_FROM_VTS	Problem loading the template with templateId {0} from the VIM Template Service. No template returned.
Filter Cache Exceptions	CANNOT_LOAD_FILTER_SCHEMA_FROM_VTS	Problem loading the filter with filterId {0} from filter service. The reason is {1}.
	CANNOT_LOAD_FILTER_SCHEMA_FROM_PERSISTENCE	Problem loading the filter with filterId {0} from persistence. The reason is {1}.
	CANNOT_LOAD_FILTER_SCHEMA_FROM_FILESYSTEM	Problem loading the filter with filterId {0} from file system. The reason is {1}.
	FILTER_MEMORY_CACHE_IS_NOT_PROVIDED	The filterMemoryCache passed in is null.
	CANNOT_SAVE_FILTER_SCHEMA	Problem occurred when persisting a filter schema: {0}. The reason is {1}.
	CANNOT_FIND_FILTER_SCHEMA	Filter schema not found: {0}.

Exception	Error Code	Display Message
	INVALIDFILTERSCHEMA	Invalid filterId {0}. The filter schema is NOT loaded in cache. The reason is {1}.
	UNABLETOREADFILTERSCHEMA	Unable to Read filter schema with filterId {0}.
	CANNOTFINDFILTERVALIDATOR	Filter schema validator not found in filter cache for filterId {0}.
	CANNOTDELETEFILTER	Problem occurred when deleting a filter schema from database: {0}. The reason is {1}.
	ERRORDELETINGFILTERDOMAINTABLES	Problem deleting filter domain tables: {0}. The reason is {1}.
Entity Resolver Exceptions	CANNOTGETSTRINGDATA	The entity resolver had a problem while getting the string data.
	OPERATIONNOTSUPPORTED	The method {0} is not supported.
	MISSINGSCHEMAJAR	Schema jar cannot be null. {0}.
	UNABLETOLOADSCHEMASINTOCACHE	Unable to load schemas into the cache for Template name {0}. Reason is {1}.
	UNABLETOLOADSCHEMASINTORESOURCE	Unable to load schema into the resource. {0}. The reason is {1}.
	INVALIDTEMPLATE	Invalid templateId {0}. The schema is NOT loaded in cache. The reason is {1}.
IdM Exceptions	IDMNULLPERSONIDENTIFIEREXCEPTION	Person identifier cannot be null.
	IDMNULLPERSONIDENTITYEXCEPTION	Person identity cannot be null.
	IDMRESPONSEPROCESSINGEXCEPTION	An error occurred while processing the IDM service response.
	IDMERRORRESPONSE	{0}.
Exception Handler Exceptions	TEMPLATEIDNULL	The <templateId> was null.
	TEMPLATEHELPERCONFIGURATIONEXCEPTION	TemplateHelper was not configured for class {0}.
	MISSINGERRORSECTIONHELPEREXCEPTION	No ErrorSectionHelperInterface was defined in the configuration for templateId {0}.
	MISSINGCLINICALDATARESPONSEEXCEPTION	No ClinicalDataResponseInterface was set while configuring class: {0}.
	GUARANTEEDLOGGERNOTCONFIGUREDANDNULL	Guaranteed logger is not configured and is null.
Socket Client Timeout Exception	SOCKETCLIENTIOEXCEPTION	An IO exception has occurred.
	SOCKETCLIENTTIMEOUTEXCEPTION	A socket timeout exception occurred.

Exception	Error Code	Display Message
MOM Exception	MDBONMESSAGEJMSEXCEPTION	Error processing message in MDBs onMessage: {0}.
MLLP Exception	MLLPILLEGALENCODINGEXCEPTION	MLLP illegal encoding.
Encryption Exceptions	ENCRYPTIONMISSINGKEYSTOREPASSWORDEXCEPTION	Missing KeyStore password in System Parameters (required for encryption and decryption).
	ENCRYPTIONMISSINGKEYSTOREFILEPATHEXCEPTION	Missing KeyStore file path (required for encryption and decryption).
	ENCRYPTIONEXCEPTION	Encryption exception.
Audit CLOB Store Errors	CDSAUDITCLOBSTOREERROR	Error in auditing VIM Write request.
	CDSAUDITLOGERROR	Error in auditing VIM Read request.

Appendix B.7. HMP/eHMP REST web-service read request detailed examples

Given below is the REST web-service read request URL format supported by CDS for HMP/eHMP. REST web-service URL supports HTTP GET requests and support multiple path and query parameters. Path parameters are separated from query parameters by '?'. Path parameters are required. The path parameters are separated by '/'. The query parameters are name value pairs and are optional except for the clientName value. Query parameters are separated by '&'.

Example of REST web-service read request URL format that defaults the response to JSON format:

```
http://<host>:<port>/repositories.med.va.gov/fpds//<templateId>/<filterId>/<patientId>/<domain>?clientName=<clientName>
&excludedIdentifier=<excluded Id>&siteId=<excluded Id site>&text=<text>&start=<yyyy-MM-dd>&stop=<yyyy-MM-dd>
&max=<max>&id=<id>&requestId=<requestId>
```

Example of REST web-service read request URL format that needs the read response in VIM XML format:

```
http://<host>:<port>/repositories.med.va.gov/fpds//<templateId>/<filterId>/<patientId>/<domain>?clientName=<clientName>
&excludedIdentifier=<excluded Id>&siteId=<excluded Id site>&text=<text>&start=<yyyy-MM-dd>&stop=<yyyy-MM-dd>
&max=<max> &id=<id>&requestId=<requestId>&_type=xml
```

Path elements:

templateId: The ID of the template that CDS uses internally to represent the response format

filterId: The ID of the filter that CDS uses internally to represent the client specific request

patientId: The patient's ICN

domain: The clinical or non-clinical domain identifier

Query parameters:

clientName – name of the client requesting the data from VistA systems.

excludedIdentifier: The patient ID at the site to exclude data from the response

siteId: The ID of the site to exclude the patient data from

text: Boolean to include document text

start: The start date of search in yyyy-MM-dd format

end: The end date of search in yyyy-MM-dd format

max: The maximum number of items to return

id: The ID of the single record to return

requestId: A unique alpha-numeric string assigned to the request for audit purposes

type: The type of the read response format. Default is JSON. Specify this query parameter only when the read response format is needed in XML format.

Figure 53 HMP/eHMP JSON response with error section {sites: [

```
{
  "apiVersion": "1.01",
  "params": {
    "domain": " SLC ",
    "systemId": "D"
  },
  "data": {
    "totalItems": 0,
    "items": []
  }
},
{
  "apiVersion": "1.01",
  "params": {
    "domain": "SLC",
    "systemId": "1"
  },
  "data": {
    "updated": "20140130231928",
    "totalItems": 2,
    "items": [
      {
        "displayName": "BP",
        "facilityCode": 500,
        "facilityName": "R",
        "high": "210/110",
        "kind": "Vital Sign",
        "localId": 4,
```

```

"locationName": "2TE",
"locationUid": "u",
"low": "100/60",
"observed": 200304041518,
"result": "138/72",
"resulted": 20030404151847,
"summary": "BLOOD PRESSURE 138/72 mm[Hg]",
"typeCode": "u",
"typeName": "BLOOD PRESSURE",
"uid": "u",
"units": "mm[Hg]"
},
{
  "displayName": "P",
  "facilityCode": 0,
  "facilityName": "R",
  "high": 120,
  "kind": "Vital Sign",
  "localId": 2507,
  "locationName": "20 MINUTE",
  "locationUid": "u",
  "low": 60,
  "observed": 200304041518,
  "result": 72,
  "resulted": 20030404151847,
  "summary": "PULSE 72 /min",
  "typeCode": "u",
  "typeName": "PULSE",
  "uid": "u",
  "units": "/min"
}
}
}
}

```

Figure 54 HMP/eHMP XML format read response

```

<?xml version="1.0" encoding="UTF-8"?>
<clinicaldata:ClinicalData xmlns:clinicaldata="Clinicaldata">
  <templateId>GenericObservationRead1</templateId>
  <requestId>6</requestId><-<patients>-<patient>

```

```

<requestedNationalId>1</requestedNationalId><resultantIdentifiers><resultantIdentifier>
  <identity>local_id_1</identity>
  <assigningFacility>5</assigningFacility>
  <assigningAuthority>USVHA</assigningAuthority>
</resultantIdentifier><resultantIdentifier>
  <identity>local_id_2</identity>
  <assigningFacility>6</assigningFacility>
  <assigningAuthority>USVHA</assigningAuthority>
</resultantIdentifier><resultantIdentifier>
  <identity>110</identity>
  <assigningFacility>5</assigningFacility>
  <assigningAuthority>USVHA</assigningAuthority>
</resultantIdentifier><resultantIdentifier>
  <identity>1</identity>
  <assigningFacility>2</assigningFacility>
  <assigningAuthority>USVHA</assigningAuthority>
</resultantIdentifier><resultantIdentifier>
  <identity>174</identity>
  <assigningFacility>2</assigningFacility>
  <assigningAuthority>USVHA</assigningAuthority>
</resultantIdentifier>
</resultantIdentifiers><genericObservations><genericObservation><recordIdentifier>
  <identity>174</identity>
  <namespaceId>2</namespaceId>
</recordIdentifier>
<observationValue><![CDATA{"apiVersion":"1.01","params":{"domain":"SLC","systemId":"D"},"data":{"totalItems":0,"items":[]}}]></o
bservationValue>
  <genericObservation><genericObservation><recordIdentifier>
    <identity>1</identity>
    <namespaceId>2</namespaceId>
  </recordIdentifier>
<observationValue><![CDATA{"apiVersion":"1.01","params":{"domain":"SLC","systemId":"1"},"data":{"updated":"20140205105840","
totalItems":14,"items":[{"displayName":"BP","facilityCode":500,"facilityName":"CAMP MASTER","high":"210V110","kind":"Vital
Sign","localId":2504,"locationName":"20
MINUTE","location":"","low":"100V60","observed":200304041518,"result":"138V72","resulted":20030404151847,"summary":"BLOOD
PRESSURE 138V72 mm[Hg]","typeCode":"4500634","typeName":"BLOOD PRESSURE","units":"Vmin"}]}]></observationValue>
  </genericObservation>
</genericObservations>
</patient>
</patients>
</clinicaldata:ClinicalData>

```

Figure 55 HMP/eHMP JSON response with error section {sites:

```

[[{"errorSection":{
  "fatalErrors":{
    "fatalError":{

      "exceptionMessage":"FPDSE_ERROR",
      "displayMessage":"Response type can only be XML or JSON",
      "exception":"gov.va.med.repositories.fpds.validator.DefaultReadRequestValidator$InvalidParameterVa
lidationError",
      "errorCode":"FPDSE_ERROR",
      "errorId":"65756756"
    }
  }
}}
]]
{"sites":[{"errorSection":{"fatalErrors":{"fatalError":{"
  "exceptionMessage":"IDMERRORRESPONSE",

```

```

    "displayMessage": "IDM service error for the national id x. The IDM error response is <IDMRESPONSE
type='ERROR' requestType='GETCORRESPONDINGIDS'><RESULT type='AR'><ERROR
code='GCID01'><TEXT><![CDATA[ICN/VPID Does Not
Exist:gov.va.med.person.idmgmt.server.profile.Identifier@58ccee7f[NI,IDM,x,200M,USVHA,,,,]]]></TEXT></E
RROR></RESULT></IDMRESPONSE>.",
    "exception": "gov.va.med.cds.exception.IdmPersonServiceException",
    "errorCode": "IDMERRORRESPONSE",
    "errorId": "65756756"
  }
}
}
}
}

```

Figure 56 HMP/eHMP XML response with error section

```

<?xml version="1.0" encoding="UTF-8"?>
<clinicaldata:ClinicalData xmlns:clinicaldata="Clinicaldata">
  <templateId>GenericObservationRead1</templateId>
  <requestId>65756756</requestId> <-patients> <-patient>
    <requestedResolvedIdentifiers/> <-resultantIdentifiers> <-resultantIdentifier>
      <identity>xxxxxxxxxx</identity>
      <assigningFacility>xxx</assigningFacility>
      <assigningAuthority>USVHA</assigningAuthority>
    </resultantIdentifier>
  </resultantIdentifiers>
  <genericObservations/>
</patient>
</patients> <-errorSection> <-fatalErrors> <-fatalError>
  <errorId>65756756</errorId>
  <exception>gov.va.med.repositories.fpbs.validator.DefaultReadRequestValidator$InvalidParameterValidationError</exce
ption>
  <exceptionMessage>FPDS ERROR</exceptionMessage>
  <errorCode>FPDS ERROR</errorCode>
  <displayMessage>Request parameter clientName was not provided or was
empty.</displayMessage>
  </fatalError>
</fatalErrors>
<errors> </errors>
<warnings> </warnings>
</errorSection>
</clinicaldata:ClinicalData>

```

Appendix B.8. HMP/eHMP PGD Write Request and Read Response examples

TBD after completion of implementation.

Appendix B.9. HTH Census HL7 write request

```

MSH^~\|^HTAPPL^200T6~XX.HL7.X.X.GOV~DNS^HT CENSUS^688HT~
XX.HL7.X.X.GOV~DNS^20130923080000-0600^^1M~T02^CensusT2013  -09-21IM -T^P^2.4^^AL^AL^USA
OBX^1^TX^^<HTCensus><Version>2</Version><Segment><Current>1</Current><Last>1</Last></Segment><
VendorFacilityNumber>1X</VendorFacilityNumber><VendorName>Telehealth
Emulator</VendorName><SubmissionDateTime>2006-01-
01T00:00:15</SubmissionDateTime><ReportStartDate>2000-12-25</ReportStartDate><ReportEndDate>2000-12-
31</ReportEndDate><Patient><SSN></SSN><ICN></ICN><VendorMRN>1</VendorMRN><HomeDevice><Mo
del>Home

```

Device</Model><Serial>2</Serial></HomeDevice><Name><Given></Given><Middle>M</Middle><Family>Patient</Family></Name><DOB></DOB><LevelOfCare>1</LevelOfCare><EnrollmentDate>2000-12-15</EnrollmentDate><DisenrollmentDate></DisenrollmentDate><ActivationDate>2000-12-15</ActivationDate><InactivationDate></InactivationDate><FacilityLocation></FacilityLocation><Compliance>.85</Compliance><Responses>6</Responses><Expectations>7</Expectations><DataAge>1</DataAge><CareCoordinator><DUZ>45873</DUZ><FacilityNumber>673</FacilityNumber><Name><Given>Eileen</Given><Middle></Middle><Family>Carenurse</Family></Name></CareCoordinator><ProgramName></ProgramName><DMP>Other-CHF Diabetes Weight</DMP><Measurement>Weight</Measurement><Measurement>Blood Pressure</Measurement><Measurement>Pulse</Measurement><Measurement>Blood Glucose</Measurement><Telecom>Internet</Telecom></Patient></HTCensus>

Sample Census HL7 Write request response

```
MSH^~|\\&^HDR
HTTPS^200HD~[REDACTED]~DNS^HTAPPL^200T3~SENDING_DNS~DNS^20140203092019.984-
0700^^ACK^00000000000000369344^P^2.4^^^AL^NE
```

Sample Census HL7 Write response with error section

```
MSH^~|\\&^HDR
HTTPS^200HD~VAHDRSVWLS01.AAC.VA.GOV~DNS^HTAPPL^200T3~SENDING_DNS~DNS^20140203092019.984-
0700^^ACK^00000000000000369344^P^2.4^^^AL^NE MSA^AR^feb32014409120011123
ERR^feb32014409120011123gov.va.med.cds.xml.schema.SchemaValidationExceptionSCHEMA_VALIDATION_FAIL
EDSCHEMA_VALIDATION_FAILEDSchema validation failed because of cvc-complex-type.2.4.a: Invalid
content was found starting with element 'surveyAcceptedStatus'. One of '{responseStatus}' is
expected.: cvc-complex-type.2.4.a: Invalid content was found starting with element
'surveyAcceptedStatus'. One of '{responseStatus}' is expected.^
```

Attachment A. Reviews and Approval Signatures

This section contains the reviews and approval signatures for the *HDR 3.8 System Design Document Volume 1*.

Peer Review: 10/31/2014

The signatures below indicate agreement and acceptance of the declarations contained in this document.

//es// [redacted]	11/12/14
[redacted]	Date
Project Manager, HDR	

//es// [redacted]	11/6/14
[redacted]	Date
Program Manager, Home Telehealth	

//es// [redacted]	11/13/14
[redacted]	Date
Program Manager, HMP/eHMP PGD	

//es// [redacted]	11/13/14
[redacted]	Date
Director, Data Architecture Division (DA)	

//es// [redacted]	11/9/14
[redacted]	Date
Chief, Health Application Support	

//es// [redacted]	11/13/14
[redacted]	Date
IT Program Manager, Repositories, Integrated Project Team Chair	

Attachment B. Signature Verification

The Signature Verification section is used to verify and document the electronic signatures, concurrence and approval of the *HDR 3.8 System Design Document, Volume 1*.

