Department of Veterans Affairs

Real Time Location System (RTLS)
National Enterprise Data Architecture

October 17, 2014

Version 9.3

This page intentionally left blank

Revision History

Date	Version	Description	Author
October 17, 2014	9.3	Subsequent Submission to VA: • For the NDR DW Data Dictionary (Sec. 3.4.2), modified Unique Column to False for composite primary keys. • All other embedded Data Dictionaries remain unchanged from the October 8 EDA Delivery (V 9.2).	HP
October 8, 2014	9.2	Subsequent Submission to VA	HP
September 19, 2014	9.1	Subsequent Submission to VA	HP
August 19, 2014	9.0	Subsequent Submission to VA	HP
March 26, 2014	8.1	Subsequent Submission to VA	HP
March 19, 2014	8.0	Subsequent Submission to VA	HP
March 7, 2014	7.0	Subsequent Submission to VA	HP
December 13, 2013	6.0	Subsequent Submission to VA	HP
October 4, 2013	5.0	Subsequent Submission to VA	HP
September 6, 2013	4.0	Subsequent Submission to VA	HP
July 26, 2013	3.0	Subsequent Submission to VA	HP
July 12, 2013	2.0	Subsequent Submission to VA	HP
June 4, 2013	1.0	Initial Submission to VA	HP

This page intentionally left blank

Table of Contents

List of Tables

Table of Figures

This page left intentionally blank.

1 Introduction

The Department of Veterans Affairs (VA) is implementing an integrated, enterprise-wide solution for a Real Time Location System (RTLS). RTLS will initially be used to track equipment, supplies, medical/surgical instruments, and provide the ability to monitor environmental conditions such as humidity and temperature. Future uses for the system may include locating staff, patients, paper documents, and the monitoring of carbon dioxide (CO_2).

The Veterans Health Administration (VHA) has a requirement to implement RTLS in all of its Medical Centers and certain other facilities over the next several years. Each VHA Veterans Integrated Service Network (VISN) will have an implementation of RTLS, which will be capable of exchanging data with VA information systems (e.g., VistA). Data from the RTLS systems will be available within a single National Data Repository (NDR). The NDR will provide sophisticated business intelligence, predictive analytics, and reporting capabilities used for process improvement, business and financial analytics, workflow analysis, and research analysis.

1.1 Purpose

This document describes the enterprise data architecture to support the implementation of RTLS and the NDR. It can be thought of as a “user’s guide” to the underlying data for the RTLS and NDR ecosystems. This document will define the data captured, stored, and used within all data stores that comprise RTLS and the NDR.

1.2 Scope

The scope of this document covers the initial RTLS Enterprise Data Architecture (EDA) and the data architecture for Phase 1 of the RTLS National Data Repository (NDR). The RTLS systems described cover the four primary applications that were identified by the National RTLS Initiative. These applications include Asset Tracking (AT), Cardiac Catheterization Laboratory Supply Management (CL), Sterile Processing Workflow (SPW), and Temperature Monitoring (TM).

As depicted in Table 1, the RTLS and NDR data architectures are comprised of multiple views, each of which are described in subsequent sections of this document. It is important to note that the RTLS Enterprise Data Architecture is implemented via an ecosystem of Commercial-Off-The-Shelf (COTS) products and is therefore contextually the logical composite of all the data schemas of each of these COTS systems.

The RTLS Enterprise Data Architecture includes a set of data standards for RTLS data systems as a vision or a model of the eventual interactions between the component parts of RTLS. The RTLS Enterprise Data architecture consists of policies, rules, and standards that govern which data is collected, how it is stored, arranged, integrated, and used in the RTLS ecosystem and across the VHA. This guidance ensures a higher quality and completeness of specific data. The opportunity for reuse and consolidation are greater when this process is followed. Data integration with the Automated Engineering Management System/Medical Equipment Reporting System (AEMS-MERS), for example, should be dependent upon data architecture standards to insure data interactions between RTLS and AEMS-MERS are consistent with other RTLS interactions.

The RTLS Enterprise Data Architecture addresses data at rest and at an introductory level, data in motion; descriptions of data stores, data groups, and data items; and, it addresses mappings of those data artifacts to data qualities, applications, locations, and how data is processed, stored, and used in a given system. It provides criteria for data processing operations that make it possible to design data flows in the system. The NDR data architecture addresses data at rest and at an introductory level, data in motion plus descriptions of data stores, data groups, and data items. A more detailed description of data in motion (i.e., data transformations) is available in the RTLS NDR ICD. The NDR data architecture provides information about the repository's data structure needed to design the aggregate data stores and reporting structures for enterprise reporting.

The RTLS and NDR data architectures are designed to describe categories of data from conceptual to physical state. This approach uses a five-layer design, as shown in the following table.

Table 1 RTLS Enterprise Data Architecture Five-Layer Design

Layer	View	Data (What)
1	Context	List of details and architectural standards important to the business
2	Conceptual	Conceptual Data Model
3	Physical	Physical Data Model
4	Data Specification	Data Definitions (Data Dictionaries)
5	Data Flow	Data Movement

This document addresses all the data stores that are part of the RTLS and NDR solution. The RTLS data stores are depicted in the RTLS Data Flow Diagram. These databases aggregate data from all the VA Medical Centers (VAMCs) and Community-Based Outpatient Clinics (CBOCs) RTLS systems belonging to the VISNs or all the aggregated Consolidated Mail Outpatient Pharmacies (CMOPs). The NDR data stores are depicted in the diagram in Section 3.1.1.

1.3 Related Documentation

- Real Time Location System (RTLS) System Design Document (SDD)
- RTLS (Real Time Location System) Interface-NDR (National Data Repository) Interface Control Document (RTLS-NDR ICD)
- AEMS-MERS/Intelligent InSites RTLS Interface Control Document (AEMS-MERS ICD)
- Generic Inventory Package (GIP)/WaveMark Real Time Location System Interface Control Document (GIP ICD)
- NEW PERSON File/WaveMark Real Time Location System Interface Control Document (Employee ICD)
- VistA PATIENT File/WaveMark Real Time Location System Interface Control Document (Patient ICD)
- RTLS_National_RSD

- Cardiovascular Assessment, Reporting, and Tracking System for Catheterization Laboratories/WaveMark System Interface Control Document (CART-CL ICD)
- Intelligent InSites Online Help
- Intelligent InSites 2.0 Application Programming Interface (API) RESTful Web Services
- Censis Technologies, Inc. API Documentation
- CenTrak Streaming Services Guide
- OAT Foundation Suite Customer / partners public API
- WaveMark IntegratorConnect – Interface Specification
- CDW’s DBA Handbook
- CDW Guide: Introduction and Policies
- CDW Guide: Projects and Development
- CDW’s Project Team ETL Developers Guide

1.4 Points of Contact

Points of contact for the RTLS Enterprise Data Architecture (EDA) will be the Hewlett Packard Enterprise Services (HPES) and the VHA RTLS Program Management Office (PMO).

2 Data Architecture for RTLS Data Stores

2.1 Layer 1 – RTLS Data Context

The data for Phase 1 of the RTLS implementation will reside in Commercial Off-The-Shelf (COTS) products which were developed at private expense by the owners.

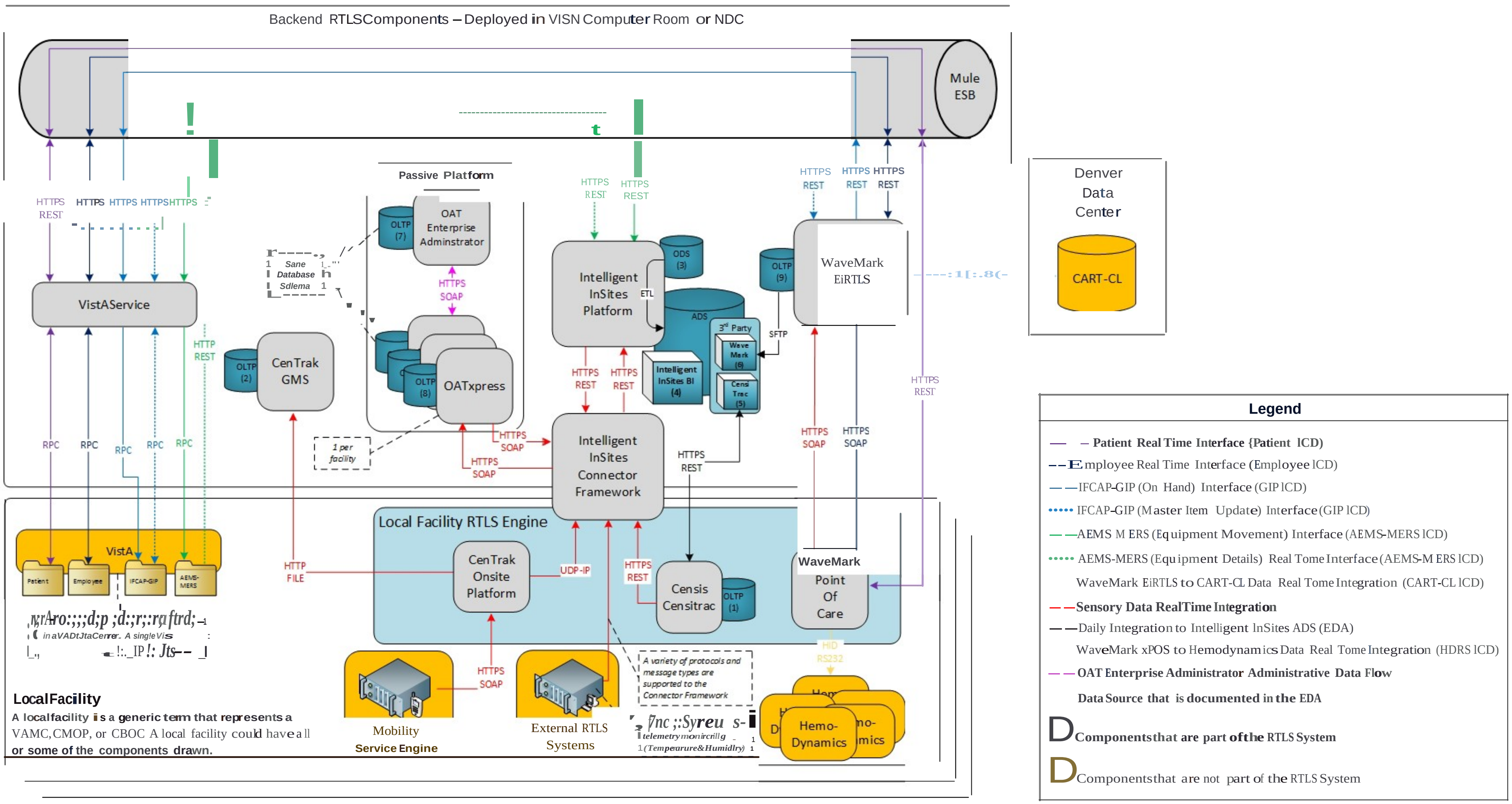
The Performance Work Statement Enterprise Systems Engineering Task Order (PWS ESE TO) includes requirements for a common data model. Since the PWS was issued, that term has evolved into "RTLS Data Model (RDM)." RDM will be used in this section instead of "common data model."

2.1.1 RTLS Data Stores

The RTLS system is a system of systems. As a system of systems, data flows between multiple components both for transactional and for analytical purposes. The following diagram focuses solely on how data flows from one data store to another data store within the RTLS system. For a more in-depth description of RTLS data flows, see Section 6 of the System Design Document (SDD). Section 6 of the SDD contains information pertaining to how sensory data flows from various hardware components to the various software components within the RTLS system (e.g., how multi-mode active tag data flows throughout the RTLS system). The data stores in the following diagram that are blue in color are part of the RTLS Data Model.

.

Figure 1: RTLS Data Flow



Title: RTLS Data Flow **Version:** 1.00 **Date:** 3/18/2014

Notice--Notwithstanding any other lease or license agreement that may pertain to, or accompany the delivery of, this computer software, the rights of the Government regarding its use, reproduction and disclosure are as set forth in Government Contract No. VA118A-12-D-0025.

A PDF version of the diagram above is embedded in this document here: RTLS Data Flow 03182014.pdf

Table 2 (following) provides a list of the data stores from the diagram above. The columns in the RTLS Data Flow Data Store Description table are defined as follows:

- Data Store Name (# in diagram) – this is the name of the data store; the number corresponds to the number in the diagram above.
- Brief Description – an explanation of the data store.
- Native DBMS – the data store's underlying database management system type (i.e., Oracle).
- Company – the company that designed and provided the physical structure of the data store.
- Schema Type – this indicates whether the data store is an online transaction processing (OLTP) or online analytical processing (OLAP) database.
- Scope – this indicates the type of operations for which the data store is used. Valid values include:
    - Facility – indicates that the data store contains facility level data.
    - VISN – indicates that the data store contains VISN level data. VISN level data stores contain facility level data for the facilities within that VISN.
    - National – indicates that the data store retains National level data. National level data stores contain all facility level data in the RTLS enterprise.
    - VISN/National – indicates that the data store contains VISN level data in a VISN deployment model and contains National level data in a National deployment model.
- Supported Applications (version) – indicates what application, including its version, is supported for the current data model and data dictionary contained within the EDA.
- Receives Data From – provides a list of components that send data to the data store. Components can either be a hardware component (i.e., Smart Cabinet), software component (i.e., CenTrak Onsite Platform) or a data store (i.e., Intelligent InSites Business Intelligence). Not all components are present in the diagram. Components with a number in parentheses indicate a data store that is present in the diagram.
- Sends Data To – provides a list of components that receive data from this data store. Components can either be a hardware component (i.e., Smart Cabinet), software component (i.e., CenTrak Onsite Platform) or a data store (i.e., Intelligent InSites Business Intelligence). Not all components are present in the diagram. Components with a number in parentheses indicate a data store that is present in the diagram.
- Detailed Info in Section – which section within the EDA provides more details about the data store.
- Notes – this is used to provide additional information about the data store.

Table 2 RTLS Data Flow Data Store Descriptions

Data Store Name (# in diagram)	Brief Description	Native DBMS	Company	Schema Type	Scope	Supported Applica-tions (version)	Receives Data From	Sends Data To	Detailed Info in Section	Notes
Censis Censitrac (1)	The Censitrac data store handles the storage and retrieval of medical/ surgical instrument inventory and tracking data.	SQL Server	Censis Technologies	OLTP	Facility	Censitrac 3.6.5.1	Censis Censitrac Workstations	Intelligent InSites ODS (3) Intelligent InSites Censitrac Analytical (5) Censis Host Server	2.3.3.1 2.4.3.2	
CenTrak GMS (2)	CenTrak Global Monitoring System (GMS) data store handles the storage and retrieval of current and summary active tag data.	SQL Server	CenTrak	OLTP	VISN/ National	CenTrak 5.11 SP4	CenTrak Onsite Platform	None	2.3.3.1 2.4.3.4	The data received by this data store is also sent to the Intelligent InSites ODS (3) via the CenTrak Onsite Platform.
Intelligent InSites ODS (3)	Intelligent InSites ODS (Operational Data Store) is the online transaction environment (Operational Data Store) for	SQL Server	Intelligent InSites	OLTP	VISN	Intelligent InSites Platform 4.3.3	OATxpress (8) Censis Censitrac (1) AEMS-MERS (VistA)	Intelligent InSites Business Intelligence (4) AEMS-MERS	2.3.3.1 2.4.3.1	

Data Store Name (# in diagram)	Brief Description	Native DBMS	Company	Schema Type	Scope	Supported Applica-tions (version)	Receives Data From	Sends Data To	Detailed Info in Section	Notes
	unstructured data that supports the RTLS business functions and user interface.						External RTLS Systems CenTrak Onsite Platform	(VistA)		
Intelligent InSites Business Intelligence (4)	Intelligent InSites Business Intelligence is the reporting environment (Analytical Data Store) that contains a copy of the unstructured data from the Intelligent InSites ODS data store. This data store transforms unstructured data to structured data that will be used for reporting and analytical purposes.	MySQL	Intelligent InSites	OLAP	VISN	Intelligent InSites Platform 4.3.3	Intelligent InSites ODS (3)	None	2.3.3.2 2.4.4.1	This data is presented to user via the reporting and analytical features of the Intelligent InSites Business Intelligence Platform.
Intelligent InSites Censitrac Analytical	Intelligent InSites Censitrac Analytical (OLAP – 3rd Party) are the data tables	MySQL	Intelligent InSites	OLAP	VISN	Intelligent InSites Platform 4.3.3	Censis Censitrac (1)	None	2.3.3.2 2.4.4.3	This data is presented to user via the reporting and analytical

Data Store Name (# in diagram)	Brief Description	Native DBMS	Company	Schema Type	Scope	Supported Applica-tions (version)	Receives Data From	Sends Data To	Detailed Info in Section	Notes
(5)	imported from Censis Censitrac data store that mirror reports generated from the Censitrac application. This data store is used to provide analytics on medical/ surgical instrument reprocessing.									features of the Intelligent InSites Business Intelligence Platform.
Intelligent InSites WaveMark Analytical (6)	Intelligent InSites WaveMark Analytical (OLAP – 3rd Party)—are the data tables imported from WaveMark interface engine infrastructure HL7 messaging interface. This data store is used to provide analytics on supply data within the Cardiac Catheterization Laboratory.	MySQL	Intelligent Intelligent InSites	OLAP	VISN	Intelligent InSites Platform 4.3.3	WaveMark EiRTLS (9)	None	2.3.3.2 2.4.4.2	This data is presented to user via the reporting and analytical features of the Intelligent InSites Business Intelligence Platform.

Data Store Name (# in diagram)	Brief Description	Native DBMS	Company	Schema Type	Scope	Supported Applica-tions (version)	Receives Data From	Sends Data To	Detailed Info in Section	Notes
OAT Enterprise Administrator (7)	The OAT Enterprise Administrator data store is the master data repository for all the OATxpress data stores across the enterprise. This data store maintains and distributes passive tagging configuration items such as locations and product information. This data store is identical to the OATxpress (8) data store.	SQL Server	OATSystems	OLTP	VISN/ National	OAT Enterprise Administrator 8.1	OATxpress (8)	OATxpress (8)	2.3.3.1 2.4.3.5	This data store is the same database schema as the OATxpress (8) database schema.
OATxpress (8)	OATxpress data store is used to capture, filter, and manage passive tagged asset's sensor data within a single facility.	SQL Server	OATSystem	OLTP	Facility	OATxpress 8.1	OAT Enterprise Administrator (7) Intelligent InSites ODS (3) Fixed and Handheld	OAT Enterprise Administrator (7) Intelligent InSites ODS (3)	2.3.3.1 2.4.3.5	OATxpress is deployed in VISN and National locations on a per facility basis This data

Data Store Name (# in diagram)	Brief Description	Native DBMS	Company	Schema Type	Scope	Supported Applications (version)	Receives Data From	Sends Data To	Detailed Info in Section	Notes
							Passive tag readers			store is the same database schema as the OAT Enterprise Administrator (7) database schema.
WaveMark EiRTLS (9)	The WaveMark EiRTLS data store collects, stores, and retrieves information from WaveMark's RFID Smart Cabinets and Point of Use Stations. The data stored in this data store allows for the efficient management of Catheterization Laboratory supplies.	Oracle	WaveMark	OLTP	VISN/ National	WaveMark EiRTLS 5.57.4.0	Smart Cabinets Point of Care Employee (VistA) IFCAP-GIP (VistA) WaveMark Online	Intelligent InSites WaveMark Analytical (6) CART-CL IFCAP-GIP (VistA)	2.3.3.1 2.4.3.3	Patient data from the Patient (VistA) interface is not directly stored in the EiRTLS data store. It is stored on the xPOS using a request and receive service using the ID of a patient's bracelet as a parameter.

2.1.2 Specified Evaluations

The PWS ESE TO Section 5.2.2, Table 2 specifies that when developing the RDM, that Houston, TX and Columbus, OH be evaluated for their data characteristics concerning Operating Room Flow and Patient Tracking, respectively. Operating Room Flow and Patient Tracking are not part of the four base applications that are being addressed by the current RTLS task orders. When applications for operating room flow and patient tracking are added, HP will evaluate the appropriate data characteristics from these locations and update the EDA to include the requested analysis.

The PWS ESE TO Section 5.2.2 also specifies that during the development of the RDM:

“The Contractor shall analyze the following standards to form part or all of the foundation of the data model:

1. VA AViVA system standards
2. GS1
3. GLN.”

HP did evaluate AViVA. At the time the AViVA assessment was done, the program did not have a data model. AViVA was at that time still very immature and as such did not require a model. This was indicated in the assessment.

The VA has working groups that are reviewing GS1 (global standard for equipment identification) and GLN (Global Location Number). These groups are investigating the feasibility of implementing and deploying these standards within the VA network of systems as well as developing a data governance policy to version changes to these standards. At this time, these groups have indicated that the roll-out of these standards is a future endeavor.

2.1.3 Data Design Decisions

The RDM is designed to track assets, record and report real-time events quickly and efficiently, and support data mining and reporting. Each component of the RDM has its own data store that was developed prior to inclusion in the RDM. The following products make up the RDM:

- Censitrac – designed to track and report on medical/surgical instruments and their containers as well as sterilization. This system is designed to be used at the facility level where medical procedures are performed or sterilization activities take place.
- CenTrak – designed to track and report on active tags.
- Intelligent InSites Visibility Platform– designed to be the primary user interface at the VISN level for tracking and reporting assets. The other 4 systems provide data to Intelligent InSites.
- OATSystems – designed to track and report on passive tags.
- WaveMark EiRTLS – designed to track and report on expendable supplies. These systems have been modified, where needed, to support the exchange of data with other RDM components (see the data flow diagram) to meet the requirements of the PWS ESE TO. Where necessary, the data stores have been extended to handle VA-specific data.

2.1.4 Data Value Standards

Data standards are documented agreements on representation, format, and definitions for common data. Data standards are important to the RTLS enterprise architecture as a means to promote the efficient sharing of RTLS data among VA regions, VISNs, and local facilities and to improve data quality for better decision-making. Data standards allow data to be aggregated at the VISN, region, and national levels for analytics and work flow analysis.

Under the ESE TO, HP along with the RTLS partners, the VA Data Work Group, and other VA subject matter experts worked together to establish a set of data standards for the VA's RTLS enterprise. These standards were prepared and delivered as an excel spreadsheet in the EDA v 8.1. The VA is currently developing a Data Standards Governance process to formalize the management of the data standards.

2.1.5 Naming Conventions

A single naming convention does not exist across all data stores as the RDM is a system of systems. Within individual data dictionaries, table and field naming patterns have been identified where appropriate.

2.1.6 Time Synchronization

RTLS platform policy requires that all virtual machines in the RTLS solution be configured to synchronize with VA's Network Time Protocol (NTP) server so that the local time kept by each host platform is consistent nationwide. The RTLS platform will use the time synchronization service from the NTP servers (ntp.va.gov) which is managed by the VA Network Security Operations Center (NSOC). All RTLS applications hosted on these virtual machines rely on this synchronized server local time to ensure that all RTLS-related events are recorded in the proper sequence at the correct time regardless of where they geographically reside.

2.1.7 Constraints

The RTLS Data Model is deployed on multiple platforms:

- Microsoft SQL Server
- Oracle Database
- MySQL

Each platform has a variety of constraints for the applications that are built on top of them. The table below provides a list of constraints for the RTLS Data Model. The columns in the RTLS Data Model Constraints table are defined as follows:

- Constraints – the database platform restriction
- Impact – the impact on the design of the application from the constraint
- Data Store Name - this is the name of the data store

Table 3 RTLS Data Model Constraints

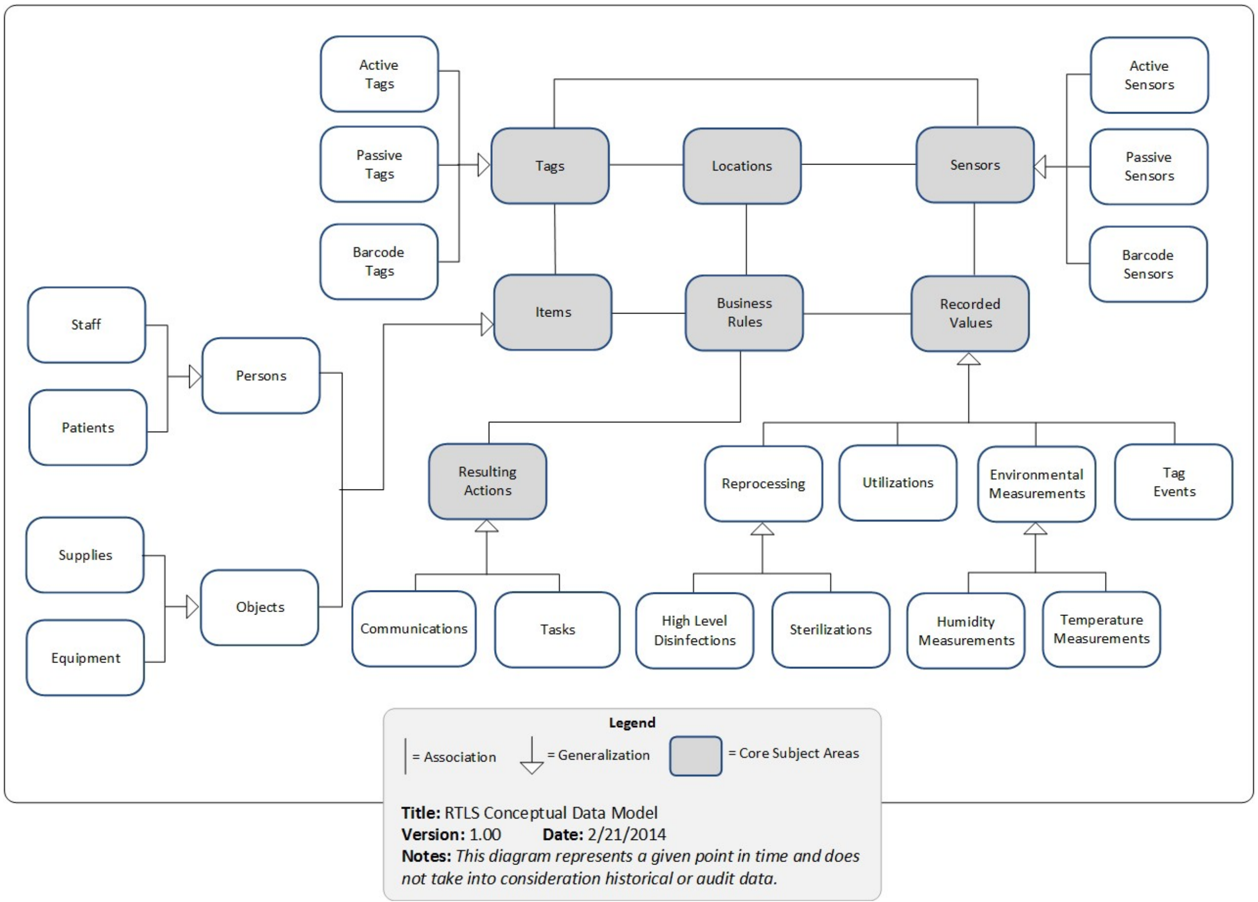
Constraint	Impact	Data Store Name
Sequential number used for ID fields (not GUID).	Database cannot be split over physically separate databases.	Censis Censitrac Intelligent InSites Foundation Intelligent InSites Business Intelligence Intelligent InSites Censitrac Analytical Intelligent InSites WaveMark Analytical WaveMark EiRTLS OAT Enterprise Administrator OATxpress
Character set limited to Latin based.	Non-Latin character sets (like Chinese or Arabic) are not supported.	Censis Censitrac Intelligent InSites Foundation Intelligent InSites Business Intelligence Intelligent InSites Censitrac Analytical Intelligent InSites WaveMark Analytical OAT Enterprise Administrator OATxpress
Some fields do not allow special characters.	Name fields, as an example, can only contain letters or numbers.	Intelligent InSites Foundation Intelligent InSites Business Intelligence Intelligent InSites Censitrac Analytical Intelligent InSites WaveMark Analytical OAT Enterprise Administrator OATxpress
Character fields do not allow for TAB's.	Users may not enter TAB's in data fields.	Censis Censitrac
Varchar fields contain arrays of foreign keys.	These fields do not impact database integrity and have no schema relevance. Typically, record keys are stored in integer fields. However, the data in these fields is strictly used to estimate location accuracy, not to enforce table relationships. Refer to the CenTrak data dictionary and	CenTrak GMS

Constraint	Impact	Data Store Name
	the field descriptions for StarSeen and StarCount.	
Primary/foreign key relationships are not enforced in the database.	Relationships are enforced in the business logic layer (software) to allow for analytical and transactional processing of data within the context of maintaining a single instance of the data.	CenTrak GMS
Datetime fields capture data in integer format.	These integer values are used in the business logic layer (software) and do not affect data integrity.	CenTrak GMS
All character and string columns are nvarchar (SQL Server) or nvarchar2 (Oracle).	The application data store utilizes Unicode character encoding to support internationalization and localization.	OAT Enterprise Administrator OATxpress
The application _property tables (dms_property, physical_property, etc.) store non-string values as string.	These values are maintained by application logic and are displayed as binary if selected from the database by a general purpose tool.	OAT Enterprise Administrator OATxpress

2.2 Layer 2 – RTLS Conceptual Data Model

A conceptual data model is a data model viewed at the highest level of abstraction. This aspect of the model represents the strategic information requirements of the enterprise within the scope of the RTLS initiative. The conceptual data model is coarse grained and is intended to show the broad set of entities and relationships under consideration. The conceptual data model provides a high-level view of the information universe under the RTLS umbrella. The conceptual data model below covers the RTLS Data Model across the enterprise. The terminology used in this model is general in nature and not necessarily intended to reflect official VA definitions.

Figure 2: RTLS Conceptual Data Model



The following subject areas are included in the figure above:

- Business Rules – describe the operations, definitions, policies, and constraints that apply to an item. Within the RTLS system, business rules can be applied to groups of items or individual items based on recorded values and locations. A series of resulting actions may be executed based on the results of the completed business rules. The status of an item may be altered by the execution of one or more business rules.
- Items – are any entities that are tracked within the RTLS system. The RTLS system will track the following types of items:
  - Objects – physical tangible things. An object within the RTLS system is extended to:
    - Equipment –are physical assets (other than land or buildings) that are used in the VA's mission. The movement (physical locations), utilizations, and reprocessing of equipment within a facility will be recorded.
      - *Expendable Equipment* is equipment defined by VA Directive 7002 (2011) that is considered disposable. An example of expendable equipment is Medical/Surgical instruments.
      - *Non-Expendable Equipment* is equipment defined by VA Directive 7002 (2011) which retains its original identity and characteristics during its useful life. Non-

expendable equipment requires accountability/control regardless of cost, life expectancy, or maintenance requirements. Non-expendable equipment will have a Catalog Stock Number (CSN) assigned; non-expendable equipment with assigned CSNs will be found in VA Catalog Number 3, section V. Examples of non-expendable equipment are IV Pumps, beds, endoscopes and case carts.

    - Supplies – are consumable items that are single use items utilized in the course of patient care. The following will be recorded in relation to supplies:
        - The consumption and quantities of supplies in the Cardiac Catheterization Laboratory; their utilizations
        - The rooms where supplies reside (locations)
        - The staff members who were present during the procedure
        - The patient who received the supplies
  - Persons – human beings. The RTLS system extends person entities into Staff and Patients.
    - Staff – staff members may be located (in future functionality), via an RTLS badge. The movement (physical locations) of staff members such as doctors and nurses may be recorded.
    - Patients – patients will be tracked through their visit (in future functionality). Movement of patients throughout the facility (i.e., physical locations) will be recorded by using an RTLS bracelet.
- Locations – particular places or positions. Within the RTLS system, locations of items are determined and updated using a combination of information from the sensors and the tags that are affixed to the items. Business rules executed based on the location of an item may alter the status of an item.
- Recorded Values –the recorded data of items (equipment, patients, staff, or supplies). Items can have many recorded values of various types:
  - Reprocessing – details about the procedure of cleaning, disinfecting, and/or sterilizing of reusable medical equipment and medical/surgical instruments. This is comprised of a series of steps that when performed based on the device manufacturer's recommendations readies the items for reuse.
    - Sterilizations – data retrieved from the process of sterilization. All the records related to the sterilization processes are retained. These include the recorded values as it pertains to the sterilization that the medical/surgical instrument or equipment went through (e.g., 200 cycles of steam sterilization at temperature of 135° Celsius).
    - High-Level Disinfections – data related to the processing of semi-critical items that come in contact with mucous membranes or non-intact skin. High-Level Disinfection traditionally is defined as complete elimination of all microorganisms in or on medical/surgical instrument, except for small numbers of bacterial spores with the goal of making objects safe for reuse. "High-Level Disinfections are performed on items such as equipment that cannot withstand the rigors of sterilization.
  - Utilizations – includes details about process change of items. This includes the consumption of supplies and the number of times medical/surgical instruments and equipment have been used.

- Environmental Measurements – are the recording of metrics and other data collected at remote or inaccessible points and transmitted to receiving apparatus for monitoring. The RTLS system records the following environment measurements of items:
    - Temperature Measurements – the measurement of heat
    - Humidity Measurements – the measurement of moisture in the air
  - Tag Events – use of information from the tag to record events other than locations and environmental measurements. A tag event includes actions such as a tag being removed from an item causing the tamper switch to be activated.
- Resulting Actions – this generalization set includes information about activities caused from the execution of business rules on items or the recorded values of items. Multiple resulting actions can be triggered based on one or more business rules. For example, if a recorded value such as a temperature monitored item goes past a preset threshold, a communication such as an alert could be sent to a technician to address the situation. Another example is a recorded value where a temperature is recorded that is within the prescribed parameters, and therefore no resulting event is triggered.
  - Communications – details about the interchange of information. Information can be exchanged via alerts (need immediate human or systematic attention), notifications (inform individuals or systems), and messages (a simple exchange of information). Information exchanges happen via person-to-system, system-to-person, and system-to-system (e.g., the results of a sterilizer load will be sent from one system [Censis Censitrac] to another system [Intelligent InSites Foundation]). The following are examples of communications:
    - Within the equipment tracking domain, an alert could trigger a security alarm when a piece of equipment has been moved outside a secured area. Within the supply domain, an example of an alert would be when a supply is nearing its expiration date. Within the sterilization process of medical/surgical instruments, a staff member could receive an alert that medical/surgical instruments failed to be sterilized due to an error in the sterilizer.
    - A nurse could receive a notification via email or SMS text message when a patient has reached a certain location and is ready to be checked in.
    - If a piece of equipment has been broken and needs maintenance, a user can send a message to the maintenance staff about repairing the out of service equipment.
  - Tasks (in future functionality) – are used as an interchange when a human or system is requesting assistance or an activity needs to be performed. For example, inventory levels are reaching restock levels, so a task is sent to a staff member to restock the inventory.
- Sensors – devices that collect values to be recorded in the RTLS system. The values collected include item locations, environment levels such as temperature and humidity, and data from tags used to identify supplies. A tag within the RTLS system will be classified as one of the following:
  - Active Sensors – devices used to sense when an active tag is within its range. Active tags possess internal power that enables them to originate signals to communicate with active sensors. Examples of active sensors include:
    - Active Communication Devices which include Wi-Fi access points and 900 MHz stars
    - Active Supplemental Devices which include monitors/exciters

- Passive Sensors – devices used to retrieve information from a passive tag. A passive sensor captures information from the passive tag by sending a signal to the tag which energizes it. Once the passive tag is energized, it reflects back to the passive sensor some of the sensor's energy that is encoded with tag data for the sensor to collect. Examples of passive sensors include handheld RFID readers, equipment cabinet readers, and passive portal devices.
- Barcode Sensors – devices used to retrieve information from barcodes. Barcode sensors consist of a light source, a lens, and a light sensor translating optical impulses into electrical ones. Barcode image data is captured by the sensor and sends the barcode's content to the sensor's output port.

- Tags – this generalization includes information about how items will be tracked and/or monitored through the RTLS system. A tag within the RTLS system will be one of the following:
    - Active Tags – a RTLS tag that initiates transmissions to a sensor and has an internal energy source such as a battery.
    - Passive Tags – a RTLS tag that receives energy from an external source (i.e., portal or handheld reader) and reflects data back to the source. Passive tags do not contain batteries. The portal or handheld reader causes the passive tag to relay information back to it.
    - Barcodes Tags – optical machine-readable representations of data relating to the object(s) to which they are attached. The barcodes are printed on or affixed to the object using an adhesive. Barcodes include the following types:
        - 1D Barcodes –a pattern of parallel lines of varying widths.
        - 2D Barcodes – expanding upon a 1D barcode, the 2D barcode stores information in both width and height. The 2D barcode consists of an arrangement of small dots or squares, marked as either a square or rectangle.
        - Markings – two dimensional Electrochemical/Electro-erosion markings applied into the surface of items such as medical/surgical instruments. The table below provides a cross walk between the conceptual data model subject areas and which transactional data stores use each entity within the RTLS Systems, and which, if any specializations of the core subject areas are used within a given data store.

Table 4 Conceptual Data Model Crosswalk

		Transactional Data Stores				
		Censis Censitrac	CenTrak GMS	Intelligent InSites ODS	OATxpress/ OAT Enterprise Administrator	WaveMark EiRTLS
	Business Rules	Business Rules	N/A	Business Rules	N/A	Business Rules
	Items Locations	Equipment	N/A	Equipment	N/A	Supplies
	Recorded Values	Locations	Locations	Locations	Locations	Locations
		Sterilizations	Humidity Measurements	Humidity Measurements	N/A	Patients
eas		High Level Disinfections	Tag Events	Tag Events		Staff
		Utilizations	Temperature Measurements	Temperature Measurements		Utilizations
	Resulting Actions	Communications	Communications	Communications	Communications	Communications
						Tasks
	Sensors	Barcode Sensors	Active Sensors	Active Sensors	Passive Sensors	Barcode Sensors
				Passive Sensors		Passive Sensors
	Tags	Barcode Tags	Active Tags	Active Tags	Passive Tags	Barcode Tags
				Barcode Tags		Passive Tags
				Passive Tags		

2.3 Layer 3 – RTLS Data Model

The Common Data Model defined in the VA RTLS ESE PWS will be referred to as the RTLS Data Model and it includes all data stores purchased as part of the RTLS System. Normally when a database is being developed for a solution, both a logical and a physical model provide value. Discussions with VA determined that only physical models are necessary since these data stores are mature products. The following data stores are part of the RTLS Data Model:

- Intelligent InSites Foundation
- CenTrak GMS
- WaveMark EiRTLS
- Censis Censitrac HL
- OATSystems OATxpress[1]
- OATSystems Enterprise Administrator[2]

[1] OATSystem's OATxpress and Enterprise Administrator are exactly the same database schema, just configured and used for different business purposes.
[2] OATSystem's OATxpress and Enterprise Administrator are exactly the same database schema, just configured and used for different business purposes.

VA118A-12-D-0025.

- Intelligent InSites Business Intelligence
- Intelligent InSites Censitrac Analytical[3]
- Intelligent InSites WaveMark Analytical[4]

For each data store, a data model will be provided. A data model is a representation of entities and their attributes and how those entities are related to other entities in the physical landscape of a database. The union of all data store models addresses the RTLS Phase One Applications:

- Asset Tracking
- Cardiac Catheterization Laboratory supply tracking
- Sterile Processing Service (SPS) workflow and medical/surgical instrument tracking
- Temperature Monitoring

2.3.1 System Configuration

The RTLS System will utilize several vendors' databases. For online-transaction-processing (OLTP) databases, Intelligent InSites, Censis, CenTrak and OAT will leverage the MS SQL Server 2008R2 database, and WaveMark will leverage the Oracle Database. Intelligent InSites will use MySQL for their online-analytical-processing (OLAP) database.

The RTLS System will be configured through a variety of tools. The MS SQL Server instances used by Intelligent InSites, Censis, CenTrak, and OAT will be managed by system administrators through the use of a tool called MS SQL Server Management Studio. Likewise the Oracle database is managed through the use of Oracle Enterprise Manager.

Through the use of Management Studio and Enterprise Manager, the administrator of the database manages backups/restores, re-indexing, etc., for general database maintenance. The database hardware and software are maintained by the HPES team during the contract period.

The versioning of the installed RTLS software which includes the data stores listed above is validated through the use of the BMC Software's RemedyForce tool. BMC Software's RemedyForce is a configuration management tool that handles the deployments of the COTS RTLS software to the various physical environments (development, test, pre-production and production).

The validation of the configuration is critical to meet the requirements of uptime and high availability.

To prevent outages, the RTLS System will use VMware failover and the installed Storage Area Network (SAN) for high availability. If a piece of hardware within one of the rack servers or a software instance crashes, the instance(s) will fail over to the secondary server. To achieve high availability requirements, the Oracle databases within the RTLS System will use Oracle Real Application Clusters (RAC) and MS SQL Server will use a dual-instance active-active cluster configuration.

The MySQL data stores will be managed by MySQL Workbench by administrators of the database. MySQL Workbench allows the administrator of the database to perform backups/restores, re-indexing, etc. for general database maintenance. The server is configured in a single instance on VMWare. The

[3] The Intelligent InSites Censitrac Analytical Database sits inside of an Intelligent InSites database called Intelligent InSites Third Party Analytics.

[4] The Intelligent InSites WaveMark Analytical Database sits inside of an Intelligent InSites database called Intelligent InSites Third Party Analytics.

server is installed on MS Server 2008, and virtualized so that it is managed across the HP hardware solution.

Additional configuration information (including sizing estimates) can be found in section 5 of the RTLS System Design Document. The columns in the RTLS System Database Configuration table are defined as follows:

- Vendor – the company that built and maintains the underlying RDBMS.
- Version – the software version of the vendor's database
- Software/Modules – the name of the software/modules that are used to maintain and monitor the underlying RDBMS.

Table 5 RTLS System Database Configuration

Vendor	Version	Software/Modules
Microsoft	SQL 2008R2	MS SQL Server Management Studio
MySQL	5.5	MySQL Workbench
Oracle	11g	Oracle Enterprise Manager

2.3.2 Data Retention

Data retention defines the policies of persistent data and records management for meeting legal and business data archival requirements. The RTLS Data Model will meet the requirements in the VA approved RTLS Requirements Specification (RSD):

RDB-140	Data retention shall adhere to the Veterans Health Administration Records Control Schedule ([redacted]vhapublications/rcs10/rcs10-1.pdf).

2.3.3 Data Models

A data model for each data store is included in this section in two formats, to include a zip file of a directory providing a HTML version and a zip file of a XML version.

- HTML – The HTML format is a directory structure of folders designed to support a JavaScript enabled browsers interface to a hyperlinked data dictionary. Unzip the HTML folder to your local disk and browse the index.htm file to start the data dictionary HTML application. Of importance in this format is the Entity Relationship Diagram (ERD).
- XMI 1.1 – The XMI 1.1 format is a zipped XML file generated from the Sparx Enterprise Architect modeling tool. XMI 1.1 is an interchange format for metadata that is defined in terms of the Meta Object Facility (MOF) standard www.omg.org/mof. XMI 1.1 allows for modeling tool interoperability by providing a flexible information interchange format. There are many modeling tools capable of metadata interchange using the XMI 1.1 format.
- Each data model will include table names, column names, column data type, primary keys, foreign keys, and including an ERD diagram. These models may contain redundant textual descriptions with the vendor data dictionaries. The vendor data dictionary is the document of record for textual descriptions of the tables and columns.

2.3.3.1 Online Transaction Processing Data Models

This section provides the models that make up the Online Transaction Processing components of the RTLS System.

Table 6 RTLS OLTP Data Models

Data Store Name	HTML/XMI 1.1
Censis Censitrac OLTP Data Model	20140307 HTML Censitrac OLTP Data Model.zip
	20140307 XML Censitrac OLTP Data Model.zip
CenTrak OLTP Data Model	20140307 HTML CenTrak OLTP Data Model.zip
	20140307 XML CenTrak OLTP Data Model.zip
Intelligent InSites OLTP Full Data Model. The representation includes a list of every table in the Intelligent InSites data model.	20140307 HTML InSites OLTP Full Data Model.zip
	20140307 XML InSites OLTP Full Data Model.zip
Intelligent InSites OLTP VA Used Data Model. The representation includes a list of only those tables in the Intelligent InSites data model that are used in RTLS Phase One Applications.	20140307 HTML InSites OLTP Partial Data Model.zip
	20140307 XML InSites OLTP Partial Data Model.zip
OATSystems OLTP Data Model	20140307 HTML OAT OLTP Data Model.zip
	20140307 XML OAT OLTP Data Model.zip
WaveMark EiRTLS OLTP Data Model	20140307 HTML WaveMark EiRTLS OLTP Data Model.zip
	20140307 XML WaveMark EiRTLS OLTP Data Model.zip

2.3.3.2 Online Analytical Processing Data Models

This section provides the models that make up the Online Analytical Processing Data Models of the RTLS System.

Table 7 RTLS Online Analytical Processing Data Models

Data Store Name	HTML/XMI 1.1
Intelligent InSites OLAP Data Model. The models include fact and dim tables only.	20140307 HTML InSites OLAP Data Model.zip
	20140307 XML InSites OLAP Data Model.zip
Intelligent InSites-Censitrac Analytical Data Model	20140307 HTML InSites-Censitrac Analytical Data Model.zip
	20140307 XML InSites-Censitrac Analytical Data Model.zip
Intelligent InSites-WaveMark OLAP Data Model	20140307 HTML InSites-WaveMark OLAP Data Model.zip
	20140307 XML InSites-WaveMark OLAP Data Model.zip

2.3.4 Data Model Extension Instructions

These sections describe how to extend each of OLTP data models that are part of the RTLS Data Model.

2.3.4.1 Intelligent InSites

The Intelligent InSites Data Model is a COTS solution and provides a feature that allows consumers of the API to add arbitrary data to resources within Intelligent InSites. This feature is part of the Intelligent InSites Snap-In Framework and allows service consumers to add custom attributes to existing resources. The custom attributes can be either single values, or collections of values. This is achieved by defining an attribute in the data section and deploying the Snap-In to any existing location resources.

The Snap-In attribute functionality is supported for the following resource types in the initial four base applications:

- equipment
- equipment-type

- location
- location-type
- resource-list

For detailed information about extending the Intelligent InSites data model, see the Intelligent InSites Online Help.

2.3.4.2 CenTrak/Censis/WaveMark

CenTrak, Censis, and WaveMark each have a database that is Structured Query Language (SQL) based. SQL is an international standard (and American National Standards Institute – ANSI) for definition and manipulation of databases. SQL is based on the relational model, which is a mathematical model for data management and manipulation. Extending SQL databases requires the use of the SQL Data Definition Language. Data Definition Language (DDL) is a powerful vocabulary that describes the portion of SQL that allows one to create, modify, and remove database objects such as tables, indexes, and users. Common DDL statements are CREATE, ALTER, and DROP. The VA will work with HPES to identify and create extensions that will be deployed using DDL statements.

As an example, this DDL demonstrates creating two tables with fields of different types and linking them together by using the primary key from one table as the foreign key for the other:

```
CREATE TABLE Instrument
(
      InstrumentID LONG, InstrumentName VARCHAR2(50), InstrumentTypeID LONG
)
CREATE TABLE InstrumentType
(
      InstrumentTypeID LONG CONSTRAINT PK_ InstrumentType PRIMARY KEY,
      TypeName VARCHAR2 (50)
)
ALTER TABLE Instrument
ADD CONSTRAINT MyInstrumentTypeIDRelationship
FOREIGN KEY (InstrumentTypeID) REFERENCES InstrumentType (InstrumentTypeID)
```

2.3.4.3 OATSystems

OATSystems has methodology to extend their data model. OATSystems uses custom properties, which are a way to extend the OATSystems data model without adding tables or fields. These model extensions are available for all types, including assets and locations, to name a few. The three tables associated with this capability are (MD is short for Metadata):

1. CUSTOM_PROPERTY_MD – this table will contain one entry for each type that the VA wants to extend. Section 5.3.5 Application-Specific Terminology - Product contains the data entries for 2 extension types (DOVAasset and DOVATagUpAsset).
2. CUSTOM_PROPERTY_GROUP_MD – this table is the parent for each custom property that is associated to a type. This table contains the attribute names from the attached file.
3. CUSTOM_PROPERTY_LOOKUP – this table contains the name of the column in the Object_State table where the values associated with the custom property are stored. The specified column in the Object_State table can contain a list of allowable values for this property and can thus be used to control what the user can enter through the UI.

Extension requests should be submitted to OATSystems Inc. and the VA should NOT perform changes on the data store directly.

2.4 Layer 4 – RTLS Data Dictionaries

By definition, a data dictionary is a collection of descriptions of the data objects or items in a data model for the benefit of programmers and others who need to refer to them. When developing programs that use the data model, a data dictionary can be consulted to understand where a data item fits in the structure, what values it may contain, and, in essence, what a data item means in real-world terms.

An established data dictionary provides numerous benefits:

- Improved data quality
- Easy access to trusted data
- Improved documentation and control
- Reduced data redundancy
- Reuse of data
- Consistency in data use
- Easier data analysis
- Enforcement of standards
- Better means of estimating the effect of change

The template for the data dictionary is divided into the following two sections:

- Table description
- Data column description

In addition, when a data element is part of a system integration with a VistA system, or other system, the Interface Control Document (ICD) contains additional information about the data element such as data element mapping, business rules, and metadata.

2.4.1 Data Dictionary Table Description

Table 8 Data Dictionary Table Description

Table Property	Description
Table Name	A single or multi-word designation used as the primary means of identification of a database table.
Table Description	A definition statement that expresses the essential nature of a data table and its differentiation from all other data tables. In the future, this may include the date this table was started to be used by RTLS if after the original implementation date.

2.4.2 Data Dictionary Data Column Description

Table 9 Data Dictionary Data Column Description

Data column property	Description
Primary Key (PK)	An index which represents a row identification value that ensures table level row uniqueness. An indication as to whether this column is the Primary Key for the table. True = yes, it is the primary key. False = no, it is not the primary key.
Foreign Key (FK)	A column or a combination of columns that is used to establish and enforce a link between two tables. The foreign key is one or more fields in one table that uniquely identifies a row in another table. True = yes, it is a foreign key. False = no, it is not a foreign key.
Name	A single or multi-word designation used as the primary means of identification of a data element.
Type	The data type, or format, used for the collection of letters, digits, and/or symbols, to depict values of a data element, determined by the operations that may be performed on the data element. The list of Types and their meanings are: **Bigint** - Integer (whole number) data from -2^63 (-9,223,372,036,854,775, 808) through 2^63-1 (9,223,372,036,854,775,807). **Bit or Boolean** - Integer data with either a 1 or 0 value. **Char [(n)] -** Fixed-length, non-Unicode data. The (n) defines the string length and can be a value from 1 through 8,000. **Decimal or Numeric** – Numeric data types with fixed precision and scale. When maximum precision is used, valid values are from - 10^38 +1 through 10^38 - 1. **Double** - Double-precision floating point number **Date** - variables use 3 bytes to store a date only (with no time information) in the range January 1, 0001 through December 31, 9999. **DateTime** - Date and time data from January 1, 1753, through December 31, 9999, with an accuracy of three-hundredths of a second or 3.33 milliseconds. **Float** – a small number with a floating decimal point. Maximum size is 4 bytes. **Int or Integer** - Integer (whole number) data from -2^31 (-2,147,483,648) through 2^31 - 1 (2,147,483,647). **Money** - Monetary data values from -2^63 (-922,337,203,685,477.5808) through 2^63 - 1 (+922,337,203,685,477.5807), with accuracy to a ten-thousandth of a monetary unit. **Number** - Data type to store integers (negative, positive, floating) of up to 38 digits of precision. The NUMBER data type can store numbers in the range of 1.0E-130 to 1.0E126 **Nvarchar [(n | max)] -** Variable-length Unicode data. The (n) defines the string length and can be a value from 1 through 4,000. The (max) indicates that the maximum storage size, which is 2^31-1 bytes (2 GB).

Data column property	Description
	Nvarchar2 [(n)] - Variable-length Unicode data. The (n) defines the string length and can be a value from 1 through 4,000. Real - Single-precision floating point number **Smallint** - Integer (whole number) data from -2^15 (-32,768) to 2^15-1 (32,767) **Text** – Variable-length non-Unicode data in the code page of the server and with a maximum string length of 2^31-1 (2,147,483,647). **Timestamp** - Variables are automatically populated by SQL Server with the time that a row is created or modified. The timestamp value is based upon an internal clock and does not correspond to real time. **Tinyint** - Integer data from 0 through 255. **Uniqueidentifier** - The uniqueidentifier data type stores 16-byte binary values that operate as globally unique identifiers (GUIDs).. The main use for a GUID is for assigning an identifier that must be unique in a network that has many computers at many sites. A GUID value for a uniqueidentifier column is usually obtained by one of the following ways: • In a Transact-SQL statement, batch, or script by calling the NEWID function. • In application code by calling an application API function or method that returns a GUID. **Varchar [(n | max)]** - Variable-length non-Unicode data. The (n) defines the string length and can be a value from 1 through 8,000. The (max) indicates that the maximum storage size, which is 2^31-1 bytes (2 GB).
Not Null	This constraint enforces a column to always contain a value. This means that you cannot insert a new record, or update a record without adding a value to this field. True = mandatory. False = optional.
Unique	Are the values in this column unique within the table? True = yes, the values are unique. False = no, the values are not unique.
Len	Contains the maximum length for that column when the data types are Varchar, nVarchar, nVarchar2 and Char. Otherwise, the value is blank.
Init	The default value for this column.
Field Description	This field contains the definition for the data column, making sure to describe the essential nature of the data element and its differentiation from all other data elements within the context of the table. In the future, this may include the date this data element was introduced into the RTLS system if that occurs after the original implementation date. It may contain values for enumerated types if such a list can be identified. Field descriptions may be blank if: An universal field description is established in the Overview section of the data dictionary The field description exists in another data dictionary and the reference to the source data dictionary is provided.

2.4.3 Transactional Data Dictionaries

2.4.3.1 Intelligent InSites Transactional Data Dictionaries

InSites ODS VA USED Data Dictiona	InSites ODS FULL Data Dictionary.docx

2.4.3.2 Censitrac HL Data Dictionary

Censitrac HL Data Dictionary.docx

2.4.3.3 WaveMark EiRTLS Data Dictionary

WaveMark EiRTLS Data Dictionary.docx

2.4.3.4 CenTrak Data Dictionary

CenTrak Data Dictionary.docx

2.4.3.5 OATSystems Data Dictionary

OATSystems_Data_Dictionary.docx

2.4.4 Analytical Data Dictionaries

2.4.4.1 Intelligent InSites Business Intelligence Data Dictionary

InSites BI Data Dictionary.docx

2.4.4.2 Intelligent InSites Third Party WaveMark Data Dictionary

InSites OLAP WaveMark Data Dict

2.4.4.3 Intelligent InSites Third Party Censitrac Data Dictionary

InSites Censitrac Analytical Data Dicti

2.5 Layer 5 – RTLS Data Flow

The data flow descriptions identify the source and destination of data items that are exchanged between partner systems in the RTLS Data Model.

The columns in the data flow tables are defined as follows:

- **Source Table** – The database table where the data element originated.
- **Source Field** – The database column where the data element originated.
- **Target Table** – The database table destination for the data element.
- **Target Field** – The database column destination for the data element.
- **Data Directionality** – An indication of the flow of data. If this column is not present, the data flows in one direction only and is described above the table.

2.5.1 Transactional to Transactional

2.5.1.1 Censitrac to Intelligent InSites

This section describes the integration of Censitrac data into the Intelligent InSites transactional data store at the database level. Its purpose is to show the source and destination of data items that are exchanged between partner systems in the RTLS solution.

The Censitrac system sends event data to Intelligent InSites. A Censitrac event is equivalent to a single message sent from Censitrac to Intelligent InSites. As various events occur in Censitrac, messages with specific data about the event are sent to the Intelligent InSites API, which processes and stores the data in Intelligent InSites' transactional database. For example, when assets (i.e., carts, trays, and instruments) reach the decontamination area within the sterilization process workflow and are scanned, Censitrac sends a message to Intelligent InSites. All data elements represent data flowing from Censitrac to Intelligent InSites.

The Intelligent InSites External System Key feature is used to ensure the two systems are communicating about the same equipment items, equipment types, and locations. An individual equipment item in Censitrac can have two external references in Intelligent InSites: the EE number (unique AEMS-MERS identifier entered manually into the Censitrac data store) and the Censitrac unique identifier. For more information about how the External System Key feature works, see the Intelligent InSites Online Help.

Table 10 Integration of Censitrac Data into Intelligent InSites ODS

Source Table	Source Field	Target Table	Target Field
HL_Set	Set_name	taggable	name
HL_Vendor Product	Vendor_product_name		
Derived from information about the asset. See note below.		taggable_type	name
HL_Set	Location_id	taggable	current_location_id
HL_Item	Location_id		
HL_Set	Set_id	taggable_external_reference	external_key
HL_Item	item_id		
HL_Set	Update_timestamp	equipment_location_history	entered
HL_Item	Update_timestamp		

Source Table	Source Field	Target Table	Target Field
HL_Item	asset_id	taggable_external_reference	external_key

Note: The method of determining which asset type is being sent from Censitrac to Intelligent InSites is part of the underlying business logic in Censitrac and involves both the function being performed through the UI and the pattern of the unique identifier of the asset.

2.5.1.2 CenTrak to Intelligent InSites

This section describes the integration of CenTrak data into the Intelligent InSites transactional data store at the database level. Its purpose is to show the source and destination of data items that are exchanged between partner systems in the RTLS solution.

Stars are network access points that communicate all the information gathered by active tags and monitors to the network. The CenTrak multi-mode tag can communicate with both stars (using 900 MHz) and with Wi-Fi access points using Cisco Compatible Extensions (CCX). In a clinical grade setting the multi-mode tags will transmit to the stars on average every three seconds. Data is streamed continuously to the CenTrak Platform Server over Transmission Control Protocol/Internet Protocol (TCP-IP) from the star. Outside of a clinical grade setting, the multi-mode tags will communicate through the Wireless Network to the CenTrak Platform Server using HTTPS/SOAP via the Cisco Mobility Services Engine. The CenTrak Platform Server will send sensory data to the Intelligent InSites Connector framework using the UDP-IP protocol. As the tag’s location changes, the Connector Framework updates the tag’s location that’s stored in cache and then sends this data to the Intelligent InSites Platform using REST over HTTPS.

All data elements represent data flowing unidirectionally from CenTrak (source system) to Intelligent InSites (target system). This is a direct mapping from the CenTrak Global Monitoring System (GMS) database to the Intelligent InSites Online Transaction Processing (OLTP) database.

Table 11 Integration of CenTrak Data into Intelligent InSites ODS

Source Table	Source Field	Target Table	Target Field
tblActive Tags	TagId	tag_external_reference	external key
tblActive Tags	TagId	tag	label
tblActive Tags	Objectid	tag	currentxymap_id
tblActive Tags	X	tag	current_x
tblActive Tags	Y	tag	current_y
tblActivespiders	Monitorid	tag	current_location_id
tblActive Tags	Starid	tag	rf_zone
tblActive Tags	LastSeen	tag	time_entered

Source Table	Source Field	Target Table	Target Field
tblActive Tags	Temp	equipment_telemetery_history	value
tblActive Tags	Humidity	equipment_telemetery_history	value
tblActive Tags	LBIValue	tag	low_battery

2.5.1.3 OATSystems to Intelligent InSites

This section describes the integration of OATSystems data with the Intelligent InSites transactional data store at the database level by showing the source and destination of data items that are communicated.

The delivery of all OATSystems tracking data is handled by the OAT provided EPCIS data delivery service, which allows a consumer application such as Intelligent InSites to subscribe via web services to certain EPCIS event notifications. The delivery mechanism was designed to be inherently reliable and to preserve the original order of notifications.

OATSystems will send the following event driven data to Intelligent InSites based on events detected in real time by the system. For example, a passive tag is excited by a wall-mounted, fixed RFID reader triggering a data notification event to be sent to Intelligent InSites.

Table 12 Integration of OATSystems Data with Intelligent InSites ODS

Source Table	Source Field	Target Table	Target Field
OBJECT	EPC	tag_external_reference	external key
OBJECT	EPC	tag	label
LOCATION	DESCRIPTION	taggable	current_location_id
EPC_DETECTION	START_TS	equipment_attribute	value_datetime
OBJECT_STATE	STATE_DATA_0	tag_external_reference	external key
OBJECT_STATE	STATE_DATA_1	taggable_type	name

2.5.2 Transactional to Analytical

2.5.2.1 WaveMark to Intelligent InSites

This section describes the integration of WaveMark data into the Intelligent InSites analytical data store at the database level by showing the source and destination of data items that are exchanged.

The WaveMark system creates files every 24 hours. These files contain the current inventory stored in the SmartCabinets and usage updates captured by the point of use device. Intelligent InSites pulls these files each night from a secure FTP server and adds the data to their analytical database.

All data elements represent data flowing unidirectionally from WaveMark (source system) to Intelligent InSites (target system).

2.5.2.1.1 Inventory

These elements represent the product inventory at a given facility. All of the data flows from WaveMark to Intelligent InSites.

Table 13 Integration of WaveMark Data into Intelligent InSites ADS

Source Table	Source Field	Target Table	Target Field
ENDPOINTPRODUCTGROUP	CONSIGNEDFLG	wavemark_onhand	Consignment Note: The following transformation takes place in an Intelligent InSites transient staging database: If CONSIGNEDFLG = Y, then set Consignment=C If CONSIGNEDFLG = N, then set Consignment=NC
ENDPOINTPRODUCT	CURRENTCOST	wavemark_onhand	Price
DISPOSITION	DISPOSITIONNAME	wavemark_onhand	Disposition
ENDPOINTPRODUCT	MMSITEMNUMBER	wavemark_onhand	HIS_Number
HOSPITALDEPT	HOSPITALDEPTNAME	wavemark_onhand	Department_Name
ROOMVEHICLE	ROOMNAME	wavemark_onhand	Room_Name
PRODUCTITEM	LOT + MFRSERIALNUMBER	wavemark_onhand	Lot_Serial_Number
MANUFACTURER	MFRNAME	wavemark_onhand	Manufacturer_Name
PRODUCT	MFRPRODUCTID	wavemark_onhand	Manufacturer_Model_Number

Source Table	Source Field	Target Table	Target Field
PRODUCT	PRODUCTNAME	wavemark_onhand	Item_Description
PRODUCTTYPE	PRODUCTTYPE	wavemark_onhand	Product_Type
PRODUCTITEM	UPN	wavemark_onhand	UPN
PRODUCTITEM	EXPIRATIONDATE	wavemark_onhand	Expiration_Date

2.5.2.1.2 Usage

The following table shows elements representing the product usage, cost, location and associated encounters at a given facility. All of the data flows from WaveMark to Intelligent InSites.

Table 14 Data Element Flow from WaveMark to Intelligent InSites ADS

Source Table	Source Field	Target Table	Target Field
ENDPOINTPRODUCT	CURRENTCOST	wavemark_encounter_used	Unit_Cost
		wavemark_encounter_wasted	Unit_Cost
HOSPITALDEPT	HOSPITALDEPTNAME	wavemark_encounter_used	Department_Name
		wavemark_encounter_wasted	Department_Name
PRODUCTITEM	EXPIRATIONDATE	wavemark_encounter_used	Expiration_Date
		wavemark_encounter_wasted	Expiration_Date
PRODUCTEVENT	DISPOSITIONDATETIME	wavemark_encounter_used	Transaction_Date
		wavemark_encounter_wasted	Transaction_Date
PRODUCTITEM	LOT	wavemark_encounter_used	Lot_Number
		wavemark_encounter_wasted	Lot_Number
MANUFACTURER	MFRNAME	wavemark_encounter_used	Manufacturer_Name
		wavemark_encounter_wasted	Manufacturer_Name
PRODUCT	PRODUCTNAME	wavemark_encounter_used	Product_Name
		wavemark_encounter_wasted	Product_Name

Source Table	Source Field	Target Table	Target Field
ROOMVEHICLE	ROOMNAME	wavemark_encounter_used	Room_Name
		wavemark_encounter_wasted	Room_Name
PRODUCTITEM	MFRSERIALNUMBER	wavemark_encounter_used	Serial_Number
		wavemark_encounter_wasted	Serial_Number
PRODUCTITEM	UPN	wavemark_encounter_used	Transaction_Code
		wavemark_encounter_wasted	Transaction_Code
PRODUCTTYPE	PRODUCTTYPE	wavemark_encounter_used	Product_Type
		wavemark_encounter_wasted	Product_Type
ENCOUNTER	HOSPENCOUNTERNBR	wavemark_encounter_used wavemark_encounter_wasted	Patient_Account_Number

Note: Patient_Account_Number is not available for reporting.

2.5.2.2 Censitrac to Intelligent InSites

Intelligent InSites uses the Censitrac reporting API to pull Censitrac reports nightly for display within the Intelligent InSites user interface. The following statements apply to the data flow table below as well as to the Intelligent InSites-Censitrac Analytical Third Party Data Dictionary:

- Censitrac reports are transformed into a specialized third party analytical database within Intelligent InSites where the Censitrac Report Name is the Intelligent InSites Target Table Name.
- The Intelligent InSites field name (i.e., target field name) is identical to the Censitrac report parameter name. The target field name is not included in the table below.
- Many report parameters are calculated on demand when the API engine receives the report request. These fields are not represented in the table below because they are not stored in the Censitrac database. These fields are described in the Intelligent InSites-Censitrac Analytical Third Party Data Dictionary.
- Censitrac reports share common report parameters. For example, Facility and Catalog Name are report parameters that are common across multiple reports.
- The data flow table is organized alphabetically by Censitrac report parameter name.
- All data elements flow from Censitrac to Intelligent InSites.

Note: Some report parameters pull data from multiple Censitrac tables because those reports represent multiple underlying business objects in the database. The word “or” is used in the table below in those cases where data could be pulled from more than one source.

Table 15 Censitrac to Intelligent InSites ADS

Censitrac Report Parameter	Target Table Name	Source Table Name	Source Field Name
Access Level	censitrac_general_inventory_throughput	AccessLevel	Access_level_name
Actual Assembly Time	censitrac_container_completion_rates	HL_SetHistory	assy_time
Actual Return Date	censitrac_instrument_maintenance_history	HL_ItemRepair	repair_return_date_actual
Alias	censitrac_instrument_inactive_instruments	HL_VendorProduct	product_alias
	censitrac_instrument_inventory_summary		
	censitrac_instrument_maintenance_history		
Area	censitrac_instrument_inventory_summary	HL_Location	location
	censitrac_general_inventory_throughput		
Asset Name	censitrac_general_inventory_throughput	HL_Set or HL_VendorProduct	set_name (if from HL_Set) or vendor_product_name (if from HL_VendorProduct)
	censitrac_general_quality_feedback		
Auto Actual	censitrac_container_cost	HL_Set	disposable_flag
Bio - Class 6 Tests	censitrac_container_inventory	HL_Set	biotest_flag
Building	censitrac_general_inventory_throughput	HL_Location	location
	censitrac_instrument_inventory_summary		
Case	censitrac_sterilizer_load_summary	HL_SterilizerLoad	case_ref
Case Cart	censitrac_container_inventory	HL_Set	set_name

Censitrac Report Parameter	Target Table Name	Source Table Name	Source Field Name
Case Reference	censitrac_general_quality_feedback	OverrideList	override
Catalog Name	censitrac_instrument_inactive_instruments	HL_VendorProduct	vendor_product_name
	censitrac_instrument_maintenance_history		
	censitrac_instrument_inventory_summary		
	censitrac_sterilizer_load_summary		
Catalog Number	censitrac_instrument_inventory_summary	HL_VendorProduct	model_number
	censitrac_instrument_maintenance_history		
	censitrac_sterilizer_load_summary		
	censitrac_instrument_inactive_instruments		
Comments	censitrac_general_quality_feedback	HL_QualityEvent	comment
Container	censitrac_container_average_assembly_times	HL_Set	set_name
	censitrac_container_completion_rates		
	censitrac_container_cost		
	censitrac_container_inventory		
	censitrac_container_processing_flow_times		
	censitrac_instrument_inventory_summary		
	censitrac_sterilizer_load_summary		
	censitrac_instrument_maintenance_history		

Censitrac Report Parameter	Target Table Name	Source Table Name	Source Field Name
Container Status	censitrac_container_inventory	HL_Set	set_status_id
Cost	censitrac_instrument_inactive_instruments	HL_VendorProduct	cost
	censitrac_instrument_inventory_summary		
Cost	censitrac_instrument_maintenance_history	HL_ItemRepair	repair_cost
Cycle	censitrac_sterilizer_load_summary	HL_SterilizerCycle	cycle_type
Date	censitrac_general_inventory_throughput	HL_SetHistory or HL_ItemHistory	build_end_timestamp (if from HL_SetHistory) or update_timestamp (if from HL_ItemHistory)
Duration	censitrac_container_processing_flow_times	ProcessStartHistory	duration
End Date/Time	censitrac_container_completion_rates	HL_SetHistory	build_end_timestamp
Est. Return Date	censitrac_instrument_maintenance_history	HL_ItemRepair	repair_return_date_est
Event Date	censitrac_general_quality_feedback	HL_QualityEvent	event_date
Event Time Delay	censitrac_general_quality_feedback	HL_QualityEvent	event_time_delay
Facility	censitrac_container_processing_flow_times	HL_Facility	facility_name
	censitrac_instrument_inactive_instruments		
	censitrac_instrument_inventory_summary		
	censitrac_sterilizer_load_summary		

Censitrac Report Parameter	Target Table Name	Source Table Name	Source Field Name
Group	censitrac_sterilizer_load_summary	HL_SterilizerGroup	ster_group_name
ID	censitrac_general_quality_feedback	Security	user_id
Item Qty	censitrac_sterilizer_load_summary	HL_SterilizerLoadUnmarked	qty
Last Updated	censitrac_container_inventory	HL_Set	update_timestamp
Last Updated By (ID)	censitrac_general_quality_feedback	Security	user_id
Last Updated By (Name)	censitrac_general_quality_feedback	Security	user_name
Location	censitrac_container_inventory censitrac_general_inventory_throughput censitrac_instrument_inactive_instruments censitrac_instrument_inventory_summary censitrac_sterilizer_load_summary	HL_Location	location
Location Type	censitrac_general_inventory_throughput	HL_LocationType	location_type
Lot	censitrac_sterilizer_load_summary	HL_SterilizerLoad	load_barcode
Mark #	censitrac_instrument_inactive_instruments	HL_Item	item_mark
Owner	censitrac_instrument_inactive_instruments	HL_Owner	owner

Censitrac Report Parameter	Target Table Name	Source Table Name	Source Field Name
	censitrac_instrument_inventory_summary		
Packaging Type	censitrac_sterilizer_load_summary	HL_SetEx	rigidity
Physician	censitrac_container_inventory censitrac_sterilizer_load_summary	HL_Set	physician_name
Pickup Date	censitrac_instrument_maintenance_history	HL_ItemRepair	repair_pickup_date
Pickup Loc.	censitrac_instrument_maintenance_history	HL_Location	location
PO #	censitrac_instrument_maintenance_history	HL_ItemRepair	purchase_order_num
Process	censitrac_container_processing_flow_times	ProcessType	process_name
Process End Date/Time	censitrac_container_processing_flow_times	ProcessStartHistory	end_timestamp
Processing Cost	censitrac_container_cost	HL_Set	processing_cost
Quality Event	censitrac_general_quality_feedback	HL_QualityEventType	quality_event_type
Reason	censitrac_sterilizer_load_summary	SterilizationReason	reason_name
Reason	censitrac_instrument_maintenance_history	HL_ItemRepair	request_reason
Reference Number	censitrac_instrument_inventory_summary	HL_VendorProduct	procurement_ref_id
	censitrac_general_inventory_throughput censitrac_sterilizer_load_summary	HL_Set or HL_VendorProduct	procurement_ref_id
Repair Details	censitrac_instrument_maintenance_history	HL_ItemRepair	request_reason
Repairer	censitrac_instument_maintenance_history	HL_Vendor	vendor_name

Censitrac Report Parameter	Target Table Name	Source Table Name	Source Field Name
Reported By	censitrac_general_quality_feedback	OverrideList or Security	override (if from OverrideList) or user_name (if from Security)
Req. By (ID)	censitrac_instrument_maintenance_history	Security	user_id
Req. By (Name)	censitrac_instrument_maintenance_history	Security	user_name
Req. Date	censitrac_instrument_maintenance_history	HL_ItemRepair	request_date
Responsible Party	censitrac_general_quality_feedback	OverrideList or Security	override (if from OverrideList) or user_name (if from Security)
Scanned By (Access Level)	censitrac_sterilizer_load_summary	AccessLevel	access_level_name
Scanned By (ID)	censitrac_sterilizer_load_summary	Security	user_id
Scanned By (Name)	censitrac_sterilizer_load_summary	Security	user_name
Serial #	censitrac_instrument_maintenance_history	HL_Item	serial_number
Service	censitrac_container_average_assembly_times	HL_Procedure	procedure_name
	censitrac_container_completion_rates		
	censitrac_container_cost		
	censitrac_container_inventory		
	censitrac_instrument_maintenance_history		
	censitrac_general_inventory_throughput		
	censitrac_sterilizer_load_summary		

Censitrac Report Parameter	Target Table Name	Source Table Name	Source Field Name
Shift	censitrac_container_completion_rates	HL_Shift	shift_name
Standard Assembly Time	censitrac_container_average_assembly_times censitrac_container_completion_rates	HL_Set	standard_assembly_time
Standard Time	censitrac_general_inventory_throughput	HL_Set	standard_assembly_time
Start Date/Time	censitrac_container_completion_rates	HL_SetHistory	build_start_timestamp
Status	censitrac_instrument_inactive_instruments censitrac_instrument_inventory_summary	HL_Item	item_status_id
Status	censitrac_general_quality_feedback	HL_QualityEvent	status_id
Status Updated By (ID)	censitrac_general_quality_feedback	Security	user_id
Status Updated By (Name)	censitrac_general_quality_feedback	Security	user_name
Sterilant Batch	censitrac_sterilizer_load_summary	HL_SterilizerLoad	sterilant_batch
Sterilization Method	censitrac_container_inventory	HL_SterilizationMethod	method_name
Sterilized Date/Time	censitrac_sterilizer_load_summary	HL_SterilizerLoad	load_timestamp
Sterilizer Lot	censitrac_general_inventory_throughput	HL_SterilizerLoad	load_barcode
Supplier	censitrac_instrument_inactive_instruments censitrac_instrument_inventory_summary	HL_Vendor	vendor_name

Censitrac Report Parameter	Target Table Name	Source Table Name	Source Field Name
	censitrac_instrument_maintenance_history		
	censitrac_sterilizer_load_summary		
Tech	censitrac_general_inventory_throughput	Security	user_id
Tech ID	censitrac_container_completion_rates	Security	user_id
	censitrac_container_inventory		
	censitrac_general_quality_feedback		
Tech Name	censitrac_container_completion_rates	Security	user_name
	censitrac_container_inventory		
	censitrac_general_inventory_throughput		
	censitrac_general_quality_feedback		
Update Date	censitrac_instrument_inactive_instruments	HL_Item	update_timestamp
Updated Date	censitrac_general_quality_feedback	HL_QualityEvent	update_timestamp
User Name	censitrac_instrument_inactive_instruments	Security	user_name

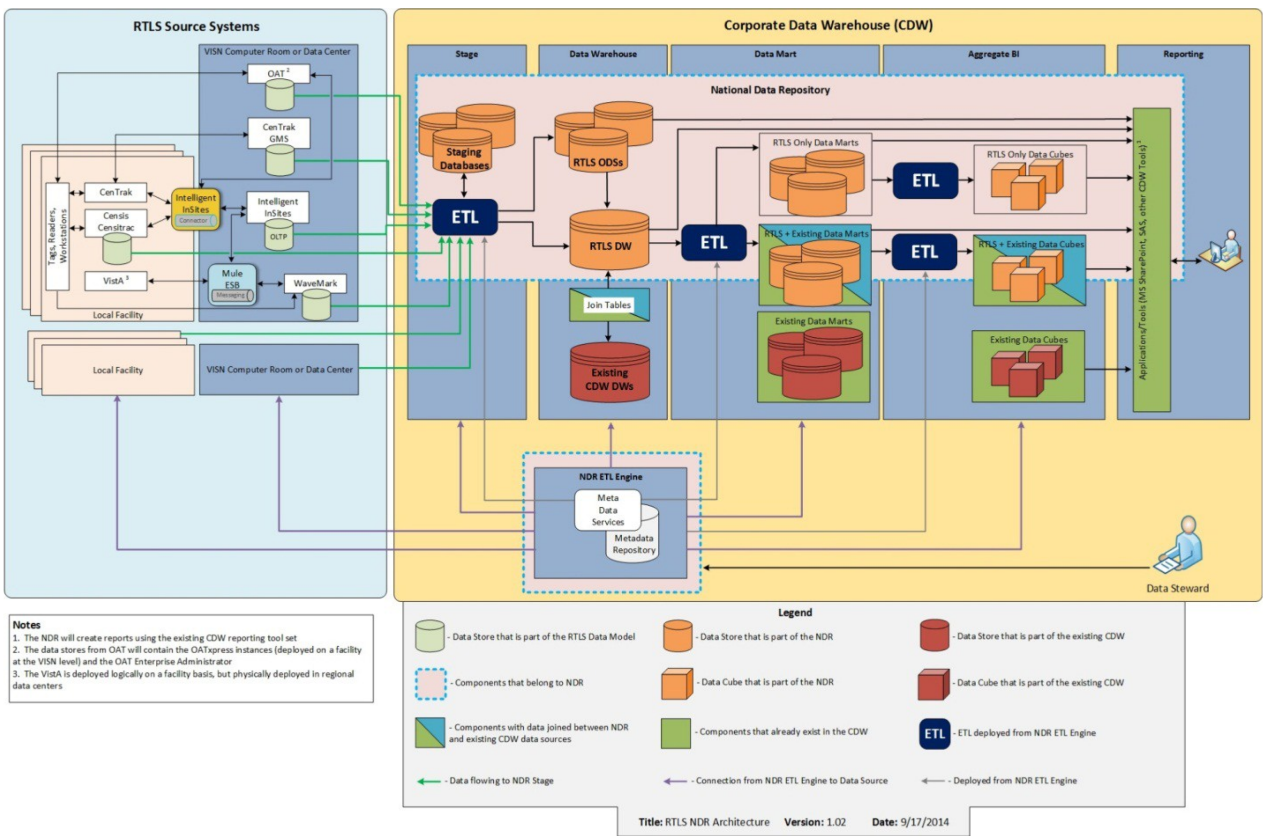
3 Data Architecture for NDR

3.1 Layer 1 – NDR Data Context

3.1.1 Overview of NDR Data Architecture

The proposed solution for the NDR is detailed below. The left side of the diagram provides a list of RTLS data sources; the right side of the diagram represents how the data from the disparate RTLS data sources will be transformed, aggregated, and stored in a single data repository.

Figure 3: Complete NDR Solution



The complete NDR solution is comprised of two phases. NDR Phase 1 covers the design and deployment of the data movement from source systems to a single national repository (i.e., the RTLS DW data store depicted in the diagram). NDR Phase 2 covers a comprehensive logical schema as well as the design and deployment of cubes, data marts, and other analytical structures built from the national repository. This phase is shown as the Data Mart, Aggregate BI, and Reporting sections of the diagram. Figure 3 covers both phases to provide a complete roadmap.

3.1.2 Specified Evaluations

3.1.2.1 Intelligent InSites Functionality at the NDR

The Task Order Performance Work Statement (PWS) for the RTLS NDR specifies that "The Contractor shall determine whether [Intelligent] InSites can be used at a national level as an additional tool within Corporate Data Warehouse (CDW) for meeting the requirements of the NDR." HPES considered using Intelligent InSites at a national level as an additional tool within CDW for meeting the requirements of the NDR. As a result of our analysis, HPES recommends that a loosely coupled approach be maintained and recommends not including Intelligent InSites in the NDR. This will help to isolate changes that occur as a normal practice with COTS business models and allow greater flexibility to accommodate all RTLS vendor products.

3.1.2.2 NDR Reports Analysis

The Task Order Performance Work Statement (PWS) for the RTLS NDR specifies that "the contractor shall analyze the reports being requested in Attachment A – NDR reports." The attachment to the PWS is a Word document (RTLS Reports.docx) which contains a list of 48 reports covering the RTLS base Use Cases for Asset Tracking, Temperature Monitoring, and Catheterization Lab. Reports related to the Sterile Processing Service Use Case are not included. This artifact was first communicated to HP in early 2013 as a roadmap for what VA expected the RTLS system to be able to provide with respect to reporting. Each report contains a title, a description, a list of data elements, and expected reporting formats.

HP analyzed each report and the requested data elements. The artifact is helpful in understanding, at a high level, VA's expectations for reporting. However, due to: the age of the document, the underlying design of the RTLS systems, the new understanding of data elements, and the maturity of the other CDW and related VA analytical data stores—HP recommends that VA update the document before we consider it a complete roadmap for the NDR.

3.1.2.3 Feasibility of VISN 10 and VISN 11

The Task Order Performance Work Statement (PWS) for the RTLS NDR specifies that "The Contractor shall assess the feasibility of incorporating VISN 10 and VISN 11 data into the NDR architecture." HP held a kickoff meeting with VISN 10 and VISN 11 on 7/31/2014. HP has requested more information for the feasibility statement from VISN 10. HP will provide a feasibility statement for both VISN 10 and VISN 11 ten business days after HP receives a full response from VISN 10.

3.1.3 Data Design Decisions

The NDR is built using Data Warehouse and Business Intelligence technologies (DW/BI) that incorporate best practices, a configurable framework, and metadata driven methods. The purpose of the NDR will be to aggregate data from multiple sources and locations throughout the VA RTLS systems of systems. The end goal will be to source analytical data stores and produce Key Performance Indicators (KPI) used to make and enable visibility for the purpose of planning and making strategic business decisions.

Building a DW/BI system uses different data modeling approaches. Unlike transactional systems that are built for a large user base to help manage dynamic data in real-time, a DW/BI system is built to manage extreme data volumes, complex and resource intensive queries, and a smaller user base. In data modeling, analytic is the antonym of transactional. The design and deployment of the NDR takes into account the basic analytic characteristics and requirements of the intended system as well as the following parameters:

- The number of RTLS installations
- The growth and capacity of the storage systems
- The need for a flexible, extensible data model
- The underlying infrastructure and services provided by the hosting organization, namely the CDW
- The features available in the software tools approved for use by the VA
- Best practices in data warehousing and database design

The NDR design will use partitioning for enhancing performance and managing the data volume. When working with large amounts of data, both logical and physical partitioning is needed to make the data more manageable and accessible. These partitions allow the database processes to read and write data in parallel to improve throughput. The processing is called “multi-threaded processing.” The logical partitions will be based on a schema-naming convention to isolate each VISN’s data. Within each of the vendors’ schemas, large transaction tables will be physically partitioned by date. For more information on the partitioning strategy, see the System Design Document (SDD).

The NDR design will relax the source system database constraints. Relaxing the constraints improves the performance of the data loading. In addition, concurrent loading of tables is not possible with foreign key constraints because the foreign keys can cause out-of-order loading of tables to fail. An explanation of the algorithm applied to relax the constraints is available in the NDR Data Dictionary, Section 3.4.2.

3.1.4 Naming Conventions

The main goal of adopting a naming convention for database objects in a data warehouse project is for easy identification of the type and purpose of all objects. A consistent and well-documented naming convention is especially important for the NDR because part of the design involves programmatically naming some of the database objects as they move through the ETL framework. CDW conventions and best practices have been considered in the conventions.

Naming conventions for the NDR are divided into two sections:

- Conventions for database objects created to support the NDR
- Conventions for naming RTLS source system data as it moves through the ETL framework

The two naming conventions are not mutually exclusive. For example, within the design of the data movement through the ETL framework, new fields such as surrogate keys will be created. The naming of the surrogate keys will follow the naming conventions for database objects created to support the NDR.

Throughout the following sections, two different naming conventions are referenced:

- *Camel case* is a naming convention that combines words by capitalizing the first letter of each word <u>except</u> for the first word (e.g., employeeName).
- *Upper camel case* is a naming convention that combines words by capitalizing the first letter of each word <u>including</u> the first word (e.g., EmployeeName). This naming convention is also sometimes called Pascal case.

3.1.4.1 Conventions for Database Objects created to support the NDR

This section provides the naming conventions the development team will follow when creating database objects to support the NDR. For example, a data store containing the metadata for ETL batch runs will have tables and fields that abide by the rules in this section.

3.1.4.1.1 General Guidelines

1. Use names that make sense and are descriptive of their purpose.
2. Avoid using abbreviations. Only use abbreviations when they are in the glossaries or data dictionaries (e.g., max, min, admin, cathLab). Abbreviations should be camel case. Also, do not sacrifice readability for brevity.
3. Only use acronyms when they are well known and documented (e.g., RTLS). Also, they should be in uppercase if used.
4. Do not use spaces or hyphens between name parts, otherwise SQL statements referencing the names will require double quotes or brackets (e.g., select [Location-Hierarchy] from table).
5. Do not use numbers or special characters in names. This can indicate the data has not been properly normalized (e.g., fields Customer2012, Customer2013, Customer2014).
6. Do not use redundant field names (e.g., rpt_locationHierarchyReport).
7. Do not use reserved database words in any name (http://technet.microsoft.com/en-us/library/ms189822.aspx).
8. Do not use plural names; use singular forms.
9. As a best practice, keep the data item names aligned with the names used for common business purposes (e.g., the terms used in the RTLS Conceptual Data Model).
10. The maximum character length of all identifiers is 128 characters.

3.1.4.1.2 Schemas

1. The following convention applies to schemas used to move data through the ETL framework:

   VISNxx_<vendor>
   CMOPxxx_<vendor>

   Where:

   VISNxx represents the VISN identifier (i.e., xx is 01 – 23)
   CMOPxxx represents the CMOP identifier
   vendor represents one of the five RTLS source systems:
   - InSites (shortened from Intelligent InSites for brevity)
   - Censitrac
   - WaveMark
   - OATSystems
   - CenTrak
2. The following convention applies to schemas at the data warehouse:

   NDR_<vendor>

   Where:

vendor represents the same RTLS source systems as above

3.1.4.1.3 Tables

1. All tables should have a primary key constraint.
2. Use upper camel case (e.g., LocationHierarchy).
3. The maximum character length of tables is 128 characters. Try to limit the name to 50 characters (shorter is better), but do not sacrifice readability for brevity.

3.1.4.1.4 Columns/Fields

1. Use upper camel case (e.g., LocationHierarchy)
2. Boolean columns should use the prefix "is" (i.e., IsEnabled instead of Enabled).
3. Do not use other column prefixes like "fld_" or "col_"
4. If the column holds date or time information, a word representing the date or time unit (e.g. date, time, minutes, hours) should appear in the field name. Date fields should be suffixed with the word Date (e.g., CreatedDate, TransmitDate). Examples of other time unit field names are RunTimeHours or ScheduledMinutes.
5. Use the suffix "DW" for the NDR attributes where DW stands for Data Warehouse. These fields are audit columns used in the operation of the NDR. The DW suffix hardens the name so the name will not collide with any of the vendor data item names.

3.1.4.1.5 Constraints

1. Foreign key constraints are references from one table to another table. Foreign keys should be prefixed with a capitalized FK followed by the name of the table plus a suffix of "SID." The syntax is {DimensionTableName}SID or {ParentTableName}SID. For example, a record in the CDW RxOutPat table representing a prescription has a foreign key called LocalDrugSID. LocalDrugSID is used to identify the record in the CDW Dim.LocalDrug table that represents the drug prescribed. Instead of referencing a dimension, a foreign key could reference a parent fact table. For example, in the CDW RxOutpatFill table, the FKRxOutpatSID is the foreign key to the parent RxOutpat fact table.
2. Composite keys (where more than one field makes up the unique value) should use the identity column as the primary key.
3. Unique keys should be prefixed with a capitalized UQ{tableName}ID. For example: UQTaggableID.

3.1.4.1.6 Indexes

1. Use the suffix "Idx" for non-clustered indexes and "Cdx" for clustered indexes.
2. The naming convention syntax for indexes is:

   {table name}{column 1}{column 2}{column n}Idx | Cdx
   For example, a non-clustered index for a table named Location on a column named LocationID would be: LocationLocationIDIdx

3.1.4.1.7 Views

1. Use the suffix "Vw" to distinguish views from regular database tables.
2. Use upper camel case (e.g., AllLocationHistoryVw).

3.1.4.1.8 Stored Procedures

1. Do not use the prefix “sp_” in a stored procedure name. The CDW advises against this “bad practice” because it is less efficient. SQL Server will by default check in the master database for that procedure first and execute one there if it finds it, also causing a performance hit.
2. Use the suffix “Sp”
3. Stored procedures perform an activity (e.g., purge, update, get, put) so the operation performed should be part of the name. If the stored procedure is applicable to a subject area of the application, the subject area comes first followed by the activity (e.g., LocationGetSp). This convention is called noun-verb.

3.1.4.1.9 Functions

1. Use the suffix “Fn”
2. Use the noun-verb form as described for stored procedures.

3.1.4.1.10 Dimensions

1. Use the suffix “Dim”
2. Use upper camel case (e.g., DateDim)

3.1.4.1.11 Facts

1. Use the suffix “Fact”
2. Use upper camel case (e.g., LocationLevelFact)

3.1.4.1.12 User Accounts and Groups

1. The VA will recommend, create and provide user accounts for the NDR effort.
2. The CDW limits the length of user account names to 20 characters.

Note: The solution will adhere to the Local Security View (LSV) schema that is enforced at the CDW.

3.1.4.2 Conventions for RTLS System Data (Like Schema)

The purpose of a naming convention for the data coming from the RTLS source systems is to represent the source systems transaction and reference data in the same way (i.e., Like Schema) but not carry the system configuration tables and other tables not used for reporting. The NDR employs a Like Schema design. In the Like Schema design, the RTLS source system database objects (e.g., tables and fields) are available in the areas of the ETL layers (e.g., staging and Like Schemas) in schemas similar to their original source schemas. This means that a table or field name moving through the ETL framework and rolled up to the regional and national NDR will retain the name assigned by the RTLS source systems as much as possible.

Keeping the table and field names identical to their source systems creates a design pattern for programmatic generation of some of the tables and fields in the stage areas instead of needing to create this by hand. In addition, keeping the table and field names intact provides a familiarity for the business users and makes the system easier to support. This is a similar methodology to how the CDW extracts and aggregates data from the VistA systems.

As data moves from the NDR to the data marts, names will change to represent data being combined across the multiple RTLS schemas. One methodology that will be implemented is name hardening. Name hardening is a systematic approach to applying conventions that ensures uniqueness within the Data Warehouse portion of the NDR. For example, the field named ID from the InSites table Alert will not provide enough meaning at the data mart level. The field might be renamed AlertID.

3.1.4.2.1 General Guidelines

1. Table and field names retain the RTLS source system naming in stage and are distinguished by attribute schema name. For example, WaveMark's Supply Disposition is named DISPOSITION.DISPOSITIONNAME. It would retain the same table and field name in the stage data stores.
2. Map and change names starting from NDR Data Warehouse to data marts. For example, logical entity would come from different physical entity (Equipment and Instrument might become one logical entity Asset).

3.1.5 Constraints

As with any large data warehousing project, the design of the NDR has limitations and trade-offs. The following sections describe, at a high level, the most significant data-related design constraints. For information about the hardware and system constraints and trade-offs, see the System Design Document.

Note: Constraints are also described in the System Design Document and in the RTLS-NDR ICD.

3.1.5.1 Big Bang Approach

The two main data warehouse design strategies are incremental and big bang. With an incremental strategy, the design involves the business experts thinking through what they need, how they want the data represented, and what the reporting requirements are. With this top-down approach, a subset of the entire data set is extracted and transferred for the data warehouse solution.

The Big Bang Approach to data warehousing is a bottom-up design where all data from all source systems is collected and aggregated. This strategy is the approach VA has elected to use.

There are pros and cons to both strategies. The big bang approach ensures no data is excluded in the final representation, but requires transferring and transforming data that may never be used. The incremental approach requires the business experts to understand what analytical and reporting questions the system will answer. In the case of VA's RTLS solution, the underlying RTLS source systems are being installed in conjunction with the design of the NDR, so the solution itself is very new and the business experts have not had concurrent access to real-time RTLS data to develop KPIs for analytics and reporting.

3.1.5.2 Big Data Management

The NDR will be classified as a Very Large Data Base (VLDB) project. The RTLS solution includes extracting data from five COTS products with installations at both the VISN/CMOP and facility levels. Given the size of the project, the need for the solution to be dynamic enough to detect and respond to changes in the underlying schemas and to be extensible to include new use cases, the recommended solution for management of the entire data set is to implement a metadata driven ETL design in conjunction with built-in features of the Microsoft SQL Server technology stack.

3.1.5.3 Methods for data extraction

Data can be extracted from the source systems using different strategies. The sheer volume of data and number of source systems limits the choices. Some strategies, although potentially more effective, are more invasive and could cause performance degradation to the underlying source systems. For example, using SQL Data Manipulation Language (DML) statements that use the same resources as the source system could have this result. To ensure our data extraction strategies do not negatively affect the transactional RTLS systems, our solution only employs strategies that extract the incremental changed data out of the transactional RTLS systems without affecting the performance of the source system. In addition, transactional systems often use audit columns appended to tables to record the date and time a record changes. Some of the RTLS source systems do not employ audit columns across all tables, so our ability to compare the last modified date and time for records is not available in many cases.

The most efficient ETL transfers only the most recent changes from the source to the data warehouse. The NDR solution will implement a design pattern used to determine the data that has changed called Change Data Capture (CDC). The source of changed data is the SQL Server transaction log. As inserts, updates, and deletes are applied to tracked source tables, entries that describe those changes are added to the log. The log serves as input to the CDC process.

3.2 Layer 2 – NDR Conceptual Data Model

The NDR conceptual data model will describe subject areas and their broad relationships, as well as key performance indicators used by VA. Based on Phase 2 deliverables, this section will be updated with the Conceptual Data Model. This is not part of Phase 1.

The NDR conceptual data model will be similar to the RTLS conceptual data model in Section 2.2. The RTLS conceptual data model was developed by examining the underlying RTLS source systems in the ESE solution and marrying up the entities and relationships the solution exposed with the terminology already present within VA. The approach to developing the NDR conceptual data model will be from the bottom up to derive common entities within the vendor databases. The NDR conceptual data model will be a representation of the underlying data structures of the NDR. It is expected that the underlying data structures will expose entities and relationships not currently represented in the RTLS conceptual data model. For example, the Persons entity description is focused on using RTLS technology to locate individuals within a facility. As the NDR design is elaborated upon, we may find that tracking access levels in Censitrac, for example, is important to the business, so the Persons entity description may change or a new entity may emerge.

3.3 Layer 3 – NDR Data Model

3.3.1 System Configuration

The NDR will use the vendor databases that comprise the RTLS Data Model as the source for the NDR. See Section 2.3.1 for more information about the system configuration of the source systems.

The NDR design will rely heavily on the Microsoft SQL Server technology stack. The NDR solution will be built using SQL Server 2012 Enterprise Edition, SQL Server Integration Services (SSIS), SQL Server Analysis Services (SSAS), Master Data Services (MDS), and Data Quality Services (DQS). The reporting portal will be built using SharePoint 2013.

The NDR design relies heavily on metadata to extract the underlying schemas and database properties from the source repositories. The source repositories will include a combination of SQL Server and Oracle databases, on servers located in various VA data centers. The connection information will be provided by VA and stored in the NDR; that information will be utilized to gain access to, extract, and capture data that resides within a database purposed at the VISN or facility levels. The ETL framework will use SSIS package templates and metadata to generate the underlying configuration that will drive the automated creation of the ETL logic, which will then transfer the data between endpoints of the NDR Data Model.

Configuration management and stewardship provide one point of accountability for the control and use of all information assets. HP is responsible for designing the infrastructure that supports good configuration management and stewardship procedures, while VA will be accountable for the actual decision-making with respect to changes to the data and policies that govern the data.

The NDR is an enclave within the CDW umbrella. The NDR solution will leverage existing CDW infrastructure, policies, procedures, guidelines, tools, and resources. The CDW physical architecture will provide the capacity for scalability; the infrastructure and technologies will enable high-performance, continuity of operations, disaster recovery, and high availability. The following documents provide information on CDW policies and guidelines:

- CDW’s DBA Handbook
- CDW Guide: Introduction and Policies
- CDW Guide: Projects and Development
- CDW’s Project Team ETL Developers Guide

3.3.2 Data Retention

Data retention defines the policies of persistent data and records management for meeting legal and business data archival requirements. The NDR solution will adhere to the Veterans Health Administration Records Control Schedule (RCS10), following all CDW best practices related to data retention and records management.

3.3.3 Data Models

The final NDR data model will include data from all RTLS source systems.

Table 16 NDR Data Models

Data Store Name	<format type>
ODS	To be supplied in Phase 1.
NDR DW	To be supplied in Phase 1.
Data Marts	To be supplied in Phase 2.

3.3.4 Data Model Modification Instructions

Modifying the data model includes adding and changing database objects in the NDR schema. At the lowest level of SQL, this involves executing DDL statements. Any modification to the NDR data model, even small ones, will require data analysis, a development effort, a testing effort, and will then follow the established configuration management processes at VA. The NDR solution's Like Schema design, metadata strategy, and naming conventions provide a structure which will help define, in documentation targeted for experienced database developers, how to add or modify database objects.

Below are some of the typical scenarios that might be needed to extend the data model:

- Adding a new table into the NDR schema
- Adding a new field to an existing table
- Changing the field length of an existing field
- Modifying DDL to prevent data loss
- Creating a new data mart or data cube
- Adding a new function or procedure to support reporting
- Changing a view

3.4 Layer 4 – NDR Data Dictionaries

The NDR DW data dictionary included in the below table is a prototype of the RTLS data warehouse data store representing the Like Schema design.

Note: The design is still under development and the data dictionary is expected to change.

3.4.1 ODS Data Dictionary

To be supplied in Phase 1.

3.4.2 NDR DW Data Dictionary

NDR_DW_Data_Dictionary.docx

3.4.3 Data Marts

To be supplied in Phase 2.

3.5 Layer 5 – NDR Source to Target Mapping

ETL processes in the NDR solution will extract transactional data from the source RTLS systems residing on SQL Server and Oracle platforms. Data will be transformed and loaded in the NDR staging layer. From the NDR staging layer, data will again be extracted, transformed, and loaded into the NDR. As the tables and fields from the RTLS source systems move through the ETL framework, the names will remain the same, where possible, to conform to a Like Schema design pattern.

Joining data from multiple sources often requires business rules to transform the data. Data transformations included in the NDR solution will be defined into types so the transformations can be applied uniformly and reused. This systematic approach to ETL metadata creates a framework that will help systems administrators manage the large number of ETL processes. The following are examples of potential transformation types:

- Applying new primary keys values to uniquely identify each row or instance of an entity. These keys are known as surrogate keys.
- Creating ETL patterns to be applied to data field types
- Selecting only certain columns to load into the target
- Encoding free-form values (e.g., mapping location status to a lookup value within a slowly changing dimension)
- Translating values (e.g., the source system stores dates in the MM-DD-YYYY format, which then requires conversion to the YYYY-MM-DD format before loading into the warehouse).
- Data validation which results in full, partial or no rejection of the data (e.g., location types do not conform to accepted values published within the Data Standards document)
- Aggregating multiple rows of data (e.g., calculated values summarized by region or VISN)

Some data elements may not require any transformation at all.

Information about the detailed data element mapping along with the data transformations will be described in the RTLS-NDR ICD. The RTLS-NDR ICD includes information about all five vendor source systems and describes how data is extracted, transformed, and loaded into the NDR.

4 Appendix A: Acronyms and Abbreviations

Acronym/Abbreviation	Meaning
AD	Active Directory
ADC	Analog-to-Digital Converted
ADS	Analytical Data Store
AEMS-MERS	Automated Engineering Management System/Medical Equipment Reporting System
AEMS-MERS EE	Automated Engineering Management System-Medical Equipment Reporting System Engineering Entry number
AER	Automated Endoscope Reprocessor
AM	Ante Meridiem
AMS	Alarm Management System
ANSI	American National Standards Institute
API	Application Programming Interface
BI	Business Intelligence, also Biological Indicator
BMC	BMC stands for the first initials of the three founders' last names. The BMC Software founders are: Scott Boulette, John Moores, and Dan Cloer.
CART-CL	Cardiovascular Assessment Reporting and Tracking system for Catheterization Labs
Cath Lab	Cardiac Catheterization Laboratory
CBOC	Community-Based Outpatient Clinics
CCX	Cisco Compatible eXtension
CDC	Change Data Capture
CDM	Conceptual Data Model
CDW	Corporate Data Warehouse
cf	confidence factor
CFR	Code of Federal Regulations
CI	Chemical Indicator
CMM	Canonical Message Model
CMOP	Consolidated Mail Outpatient Pharmacy
CMS	Configuration Management System
COTS	Commercial Off the Shelf

Acronym/Abbreviation	Meaning
CS	Central Sterile
CSN	Category Stock Number
DART	Dynamic Air Removal Test
DBA	Database Administrator
dbm	decibels per milliwatt
DBMS	Database Management System
DDL	Data Definition Language
DG	Dry Germicide
DHCP	Dynamic Host Configuration Protocol
DIM	Dispenser Integrated Monitor
DML	Data Manipulation Language
DMS	Device Management System
DQS	Microsoft's Data Quality Services
ECP	Engineering Change Proposal
EDA	Enterprise Data Architecture
EDM	Enterprise Data Manager
EE	Engineering Entry
EENMS	Enterprise EPC Number Management System
EMS	Event Management System
EO	Ethylene Oxide
EP	Electrophysiology
EPC	Electronic Product Code; structure is defined in the GS1Tag Data Standard (http://www.gs1.org/gsmp/kc/epcglobal/tds)
EPCIS	Electronic Product Code Information Service
EPS	Endoscope Processing System
EPS HLD	Endoscope Processing System High-Level Disinfection
ERD	Entity Relationship Diagram
ERU	Emergency Remote Unit
ESB	Enterprise Service Bus
ESE	Enterprise Systems Engineering
ETL	Extract, Transform, and Load
ETO	Ethylene Oxide

Acronym/Abbreviation	Meaning
FDA	U.S. Food and Drug Administration
FK	Foreign Key
FTP	File Transfer Protocol
G2	Generation 2 (meaning second generation)
GIAI	Global Individual Asset Identifier; Definition maintained by GS1
GIP	Generic Inventory Package
GLN	Global Location Number
GMS	Global Monitoring System
GND	Building\Grounds
GRAI	Global Returnable Asset Identifier; Definition maintained by GS1
GS1	Global Standard for Equipment Identification
GTIN	Global Trade Item Number
GUID	Globally Unique Identifier
HHC	Hand Hygiene Compliance
HIBCC	Health Industry Business Communications Council
HIPAA	Health Insurance Portability and Accountability Act of 1996
HL	Hospital Level
HLD	High-Level Disinfection
HPES	HP Enterprise Services
HPGP	Hydrogen Peroxide Gas Plasma
HTML	HyperText Markup Language
HTTPS	Hypertext Transfer Protocol Secure
HTTPS/SOAP	Hypertext Transfer Protocol Secure/Simple Object Access Protocol
IANA	Internet Numbers Authority
ICD	Interface Control Document
ICN	Integrated Control Number
IDN	Integrated Delivery Network
IEEE	Institute of Electrical and Electronics Engineers
IEN	Internal Entry Number
IEPD	Information Exchange Package Documentation

Acronym/Abbreviation	Meaning
IFCAP-GIP	Integrated Funds Control, Accounting and Procurement- Generic Inventory Package
IP	Internet Protocol
IR	Infrared
IT	Information Technology department
JPG	Joint Photographic experts Group
KPI	Key Performance Indicator
LBI	Low Battery Indicator
LF	Low Frequency
LSV	Local Security View
MAC	Media Access Control address
Mbps	Megabits per second
MD	Medical Doctor
MDS	Microsoft's Master Data Services
MF	Motion Flag
MHz	Megahertz
MM	Multi-Mode
MOF	Meta Object Facility
MS	Microsoft Corporation
MSE	Mobility Services Engine
NDR	National Data Repository
NTP	Network Time Protocol
ODS	Operational Data Store
OFS	OAT Foundation Suite
OLAP	Online Analytical Processing
OLTP	Online Transaction Processing
OPA	ortho-Phthalaldehyde
OR	Operating Room
PA	Peracetic Acid
PCD	Primary Communication Device
PDF	Portable Document Format
PHI	Protected Health Information

Acronym/Abbreviation	Meaning
PK	Primary Key
PM	Preventive Maintenance; Post Meridiem
PMAS	Project Management Accountability System
PMO	Program Management Office
PNG	Portable Network Graphics
POC	Point of Contact
PWS ESE TO	Performance Work Statement Enterprise Systems Engineering Task Order
QA	Quality Assurance
RAC	Oracle's Real Application Clusters
RDBS	Relational Database System
RDM	RTLS Data Model
RF	Radio Frequency
RFID	Radio Frequency Identification
RFP	Request for Proposal
RGB	Red, Green, Blue
RN	Registered Nurse
RSSI	Received Signal Strength Indicator
RTLS	Real Time Location System
RUN	Remote Unit Notification
SA	System Administrator
SDD	System Design Document
SFPP	Steam Flush Pressure Pulse
SGLN	Serialized Global Location Number; Definition maintained by GS1
SGTIN	Serialized Global Trade Item Number; Definition maintained by GS1
SMS	Short Message Service
SOAP	Simple Object Access Protocol
SPS	Sterile Processing Service
SPW	Sterile Processing Workflow
SQL	Structured Query Language
SSAS	Microsoft's SQL Server Analysis Services

Acronym/Abbreviation	Meaning
SSCC	Serial Shipping Container Code; Definition maintained by GS1
SSIS	Microsoft's SQL Server Integration Services
SSN	Social Security Number
SSO	Single Sign On
TCP/IP	Transmission Control Protocol/Internet Protocol
TDS	Tag Data Standard - This standard defines EPC tag data, including how key GS1 key identifiers are encoded on the tag and how they are encoded for use in the information systems layers of the EPC Systems Network. http://www.gs1.org/gsmp/kc/epcglobal/tds
TX	Transmit
UDI	Universal Device Identifier
UDP-IP	User Datagram Protocol-Internet Protocol
UDT	FDA UDI Type
UI	User Interface
UML	Unified Modeling Language
UPN	Unique Product Number
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
USPS	United States Postal Service
UTC	Universal Time Coordinates
VA	Veterans Administration
VAMC	VA Medical Centers
VAMDNS	VA Medical Device Nomenclature System
VHA	Veterans Health Administration
VHP	Vaporized Hydrogen Peroxide
VISN	Veterans Integrated Service Network
VistA	Veterans Health Information System and Technology Architecture
VLDB	Very Large Data Base
WLAN	Wireless Local Area Network
WM	WaveMark
XMI	XML (Extensible Markup Language) Message Interface

Acronym/Abbreviation	Meaning
XML	Extensible Markup Language is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. It is defined in the XML 1.0 Specification produced by the W3C, and several other related specifications, all free open standards. http://en.wikipedia.org/wiki/XML
XSD	XML Schema Definition