|  |  |  |
| --- | --- | --- |
|  |  |  |



**EARTH OBSERVATION**

**DATA MANAGEMENT SYSTEM**

**Software Design Document**

# **Table of Contents**

[**Executive Summary** 1](#_Toc512269410)

[**Introduction** 1](#_Toc512269411)

[**Document Purpose** 1](#_Toc512269412)

[**Intended Audience** 1](#_Toc512269413)

[**Project Background** 2](#_Toc512269414)

[**Purpose of the Business Requirements** 2](#_Toc512269415)

[**Business Goals & Objectives to Achieve** 2](#_Toc512269416)

[**Benefits/Rationale** 3](#_Toc512269417)

[**Stakeholders** 3](#_Toc512269418)

[**Dependencies on Existing Systems** 3](#_Toc512269419)

[**References** 3](#_Toc512269420)

[**Assumptions** 4](#_Toc512269421)

[**1.0 Requirements Scope** 4](#_Toc512269422)

[**1.1 In Scope** 5](#_Toc512269423)

[**1.2 Out of Scope** 5](#_Toc512269424)

[**2.0 Functional Requirements** 6](#_Toc512269425)

[**2.1 Actor Profile Specification** 6](#_Toc512269426)

[**2.2 Essential Use Case Diagram** 6](#_Toc512269427)

[**2.3 Essential Use Case Specifications** 7](#_Toc512269428)

[**2.4 Business Rules** 7](#_Toc512269429)

[**3.0 Data Requirements** 8](#_Toc512269430)

[**3.1 Data Architecture** 8](#_Toc512269431)

[3.1.1 Entity Relationship Diagram 8](#_Toc512269432)

[**3.2 Data Volumes** 10](#_Toc512269433)

[**3.3 Data Conversion and Storage** 10](#_Toc512269434)

[**3.4 Data Retention and Archiving** 13](#_Toc512269435)

[**3.5 Freedom of Information / Privacy Impact Assessment (FOIPA)** 13](#_Toc512269436)

[**3.6 Data Definition Report** 13](#_Toc512269437)

[3.6.1 Entity Definition Report 13](#_Toc512269438)

[**4.0 Non-Functional Requirements** 14](#_Toc512269439)

[**4.1 Security Requirements** 14](#_Toc512269440)

[4.1.1 Authentication 14](#_Toc512269441)

[4.1.2 Authorization and Access Control 14](#_Toc512269442)

[**4.2 Availability Requirements** 15](#_Toc512269443)

[**4.4 System Help Requirements** 15](#_Toc512269444)

[**4.5 Performance Requirements** 16](#_Toc512269445)

[**4.6 Scalability Requirements** 17](#_Toc512269446)

[**4.7 User Scalability** 17](#_Toc512269447)

[**4.8 Application Scalability** 17](#_Toc512269448)

[**4.9 Database Scalability** 17](#_Toc512269449)

[**5.0 Interface Requirements** 18](#_Toc512269450)

[**5.1 User Interface Requirements** 18](#_Toc512269451)

[**5.2 System Interface Requirements** 18](#_Toc512269452)

[5.2.1 Input and Output Requirements 18](#_Toc512269453)

[**6.0 Throw Away Prototypes** 19](#_Toc512269454)

[**7.0 Glossary** 20](#_Toc512269455)

[**9.0 Revision Log** 22](#_Toc512269456)

[**Appendices** 23](#_Toc512269457)

[**Appendix A – Essential Use Case Report** 23](#_Toc512269458)

[**Appendix B – Entity Definition Matrix** 26](#_Toc512269459)

[**Appendix C – User Interview** 29](#_Toc512269460)

[**Authorization** 37](#_Toc512269461)

# **1.0 Introduction**

## **1.1 Document Purpose**

The Software Design Document describes the system design for the Data3 subsystem. It describes the system’s construction by specifying its components, component organization and relation, and the principles of the application’s internal construction.

PFC Database Administrators, Software Developers, and Data Scientists interested in using the subsystem are the document’s intended audience. This document can serve as a system reference manual for a running implementation, and is the primary reference for application migrations, support and maintenance. It provides a sufficiently detailed description of the system, so a project team can build the system, and support staff can use the document to understand the running system.

## **1.2 System Overview**

The Data3 subsystem will attempt to resolve this issue by providing centralized and efficient storage of remotely sensed raster data. The ultimate deliverables of the project are:

* A data layer composed of *Open Data Cube* and *Postgres* database, using Python API for data interaction.
* OGC compliant distributed access layer composed of *Django Data Cube* and *MapServer.*
* Access to system via GIS clients: *OpenLayers* and *QGIS.*

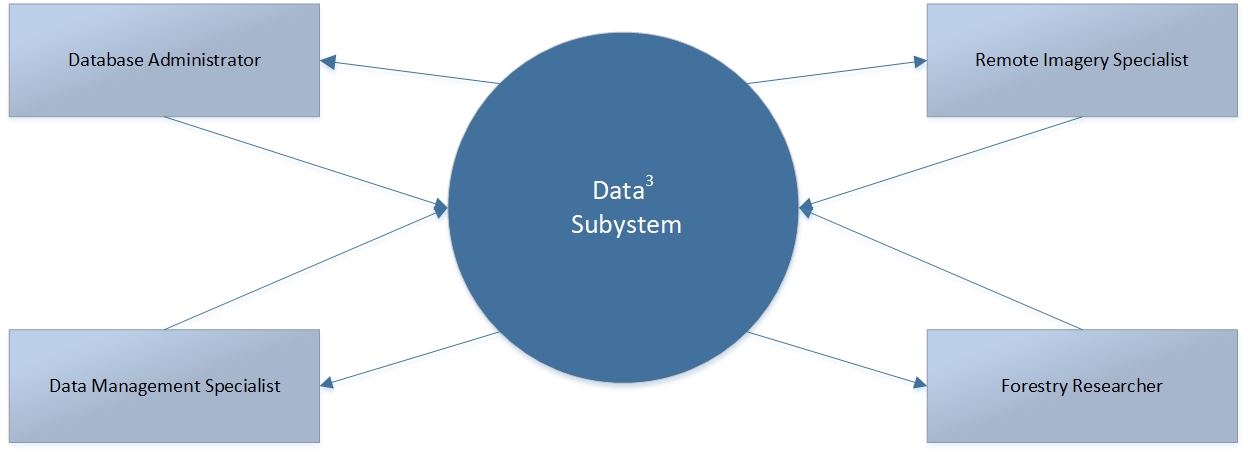


Figure Context Diagram

## **1.3 References**

This document compiles and reports on information gathered from many sources. As *Open* *Data Cube* is further developed, the online information sources will be updated with new information. The following sources represent the full breadth of technologies referenced in this project:

* <https://www.opendatacube.org/>
* <https://datacube-core.readthedocs.io/en/stable/index.html>
* <https://github.com/ceos-seo/django-datacube-wcs>
* <http://mapserver.org/index.html>
* <https://www.postgresql.org/>
* <https://qgis.org/en/site/>
* <https://openlayers.org/>
* [http://www.opengeospatial.org/standards/](http://www.opengeospatial.org/standards/wms)

## **1.4 Design Methodology**

The subsystem design makes use of standardized open-source software that provides consistent, re-usable, sustainable, and interoperable solutions to store raster data.

The Data3 Project team will follow Agile development methodology. Our approach will utilize iterative development, with the evolution of requirements and solutions encouraged by collaboration between self-organizing and cross-functional teams. Iterations will be planned to address the most essential use cases, and subsequent iterations will further develop and build upon previous work.

## **1.5 Quality Assurance**

## **1.6 System Background**

PFC requested a subsystem that will efficiently manage remotely-sensed raster data, and disseminate it over the National Forest Information System (NFIS). NFIS manages a collection of applications and databases that catalogue the state of Canada’s forests. NFIS allows forestry science researchers, forestry policy decision-makers, and other interested stakeholders to review the data resources. The current workflow used for raster data management uses local storage. This causes issues with discovery and analysis of remotely sensed raster data.

## **1.7 System Objectives**

## The subsystem seeks to resolve the issues surrounding storage and access to remotely sensed raster data. The high level objectives of the system include providing a networked database that provides indexing of metadata and product definitions. The software that accomplishes this is *Open Data Cube* paired with *Postgres*. The system needs to provide OGC compliant distributed access to the database and serve to data visualization clients, like *QGIS* or *OpenLayers*. We will use *MapServer* and *Django Data Cube* to satisfy this objective.

## **1.8 System Constraints**

This section describes assumptions that were made prior to or during the Business Requirements gathering and documentation phase.

* The project team will use the following technologies: *Data Cube*, *Postgres,* *Django Data Cube,* *MapServer,* *QGIS,* and *OpenLayers.*
* The sponsor will provide analysis-ready test data to the project team during development, and will include products from *Sentinel Data* and *National Forest Inventory*.
* The sponsor will provide the development team with a set of appropriate resources, including virtual machines which mirror the PFC ecosystem, for development and testing.
* PFC has a complex technology ecosystem within which the system must integrate. Their operation system is based on *CentOS 7* and *Postgres* database, with various security layers on all networked applications.

## **1.9 Guiding Principles**

## The subsystem

## **1.10 Revision of System Design to Match As-Built System**

## The subsystem

## **Business Goals & Objectives to Achieve**

The primary goal for this project is to implement a subsystem that will efficiently process, store, and manage remotely sensed data using a standardized construct, *Data Cube* paired with *Postgres*, and data interaction with *Python* APIs.

## **Benefits/Rationale**

The subsystem will use standardized open-source software that provides consistent, re-usable, sustainable, and interoperable solutions to store raster data. The system makes analysis-ready data accessible in time-series, and allows combined use of multiple data sets. The subsystem will minimize the time and expert knowledge required to access and analyze remotely-sensed data. These key benefits of the proposed system will enhance the ability to fully utilize the resource of remotely-sensed data.

## **Stakeholders**

Stakeholders are the individuals or groups who have a vested interest in this project and whose interests need to be considered throughout the project. The following individuals are stakeholders for which these business requirements are documented:

* Brian Low, Manager, NFIS
* Saryta Schaerer, CS Department Chair, Camosun College
* Jeannette Strand, Project Team Member, Camosun College
* Mingyang Wang, Project Team Member, Camosun College
* Tyler Hall, Project Team Member, Camosun College
* David Kuitunen, Project Team Member, Camosun College

## **Dependencies on Existing Systems**

This section describes the dependencies between the application for which these Business Requirements are written and the other existing applications/systems.

PFC has a complex technology ecosystem within which the system must integrate. Their operation system is based on *CentOS 7* and *Postgres* database, with various security layers on all networked applications. The subsystem is based on open-source software that has many dependencies which will be recorded in the Software Design Document.

## **References**

This section lists the references to external and previous documentation that are related to the project’s Business Requirements.

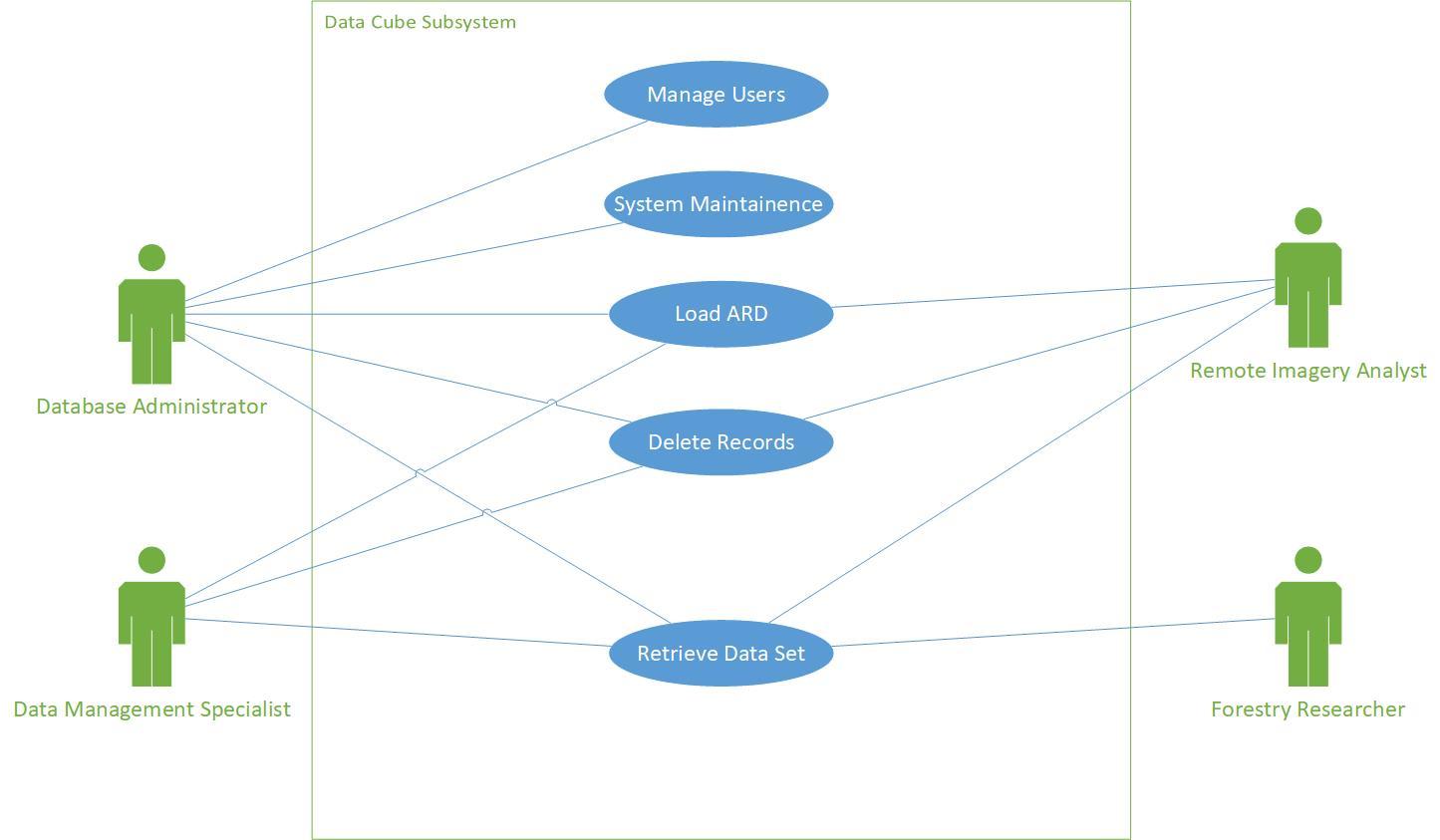
This document compiles and reports on information gathered from online sources. As *Data Cube* is further developed, the online information sources will be updated with new information. The following sources represent the full breadth of technologies referenced in this project:

* <https://www.opendatacube.org/>
* <https://datacube-core.readthedocs.io/en/stable/index.html>
* <https://github.com/ceos-seo/django-datacube-wcs>
* <http://mapserver.org/index.html>
* <https://www.postgresql.org/>
* <https://qgis.org/en/site/>
* <https://openlayers.org/>
* [http://www.opengeospatial.org/standards/](http://www.opengeospatial.org/standards/wms)

## **Assumptions**

# **1.0 Requirements Scope**

This section details the scope for business functionality, and defines scope in terms of use cases. Scope analysis involves determining and documenting a list of specific project goals, deliverables, features, functions, tasks, deadlines, and ultimately costs. In other words, scope is what needs to be achieved for the system to satisfy user’s needs. We have undertaken multiple user interviews, which are given in Appendix C.

Fig. 1. In Scope Use Cases

## **1.1 In Scope**

* Administrators and Managers shall load analysis ready data sets to *Data Cube*.
* Administrators and Managers shall delete data sets using *Data Cube*.
* All users shall retrieve data using client software.

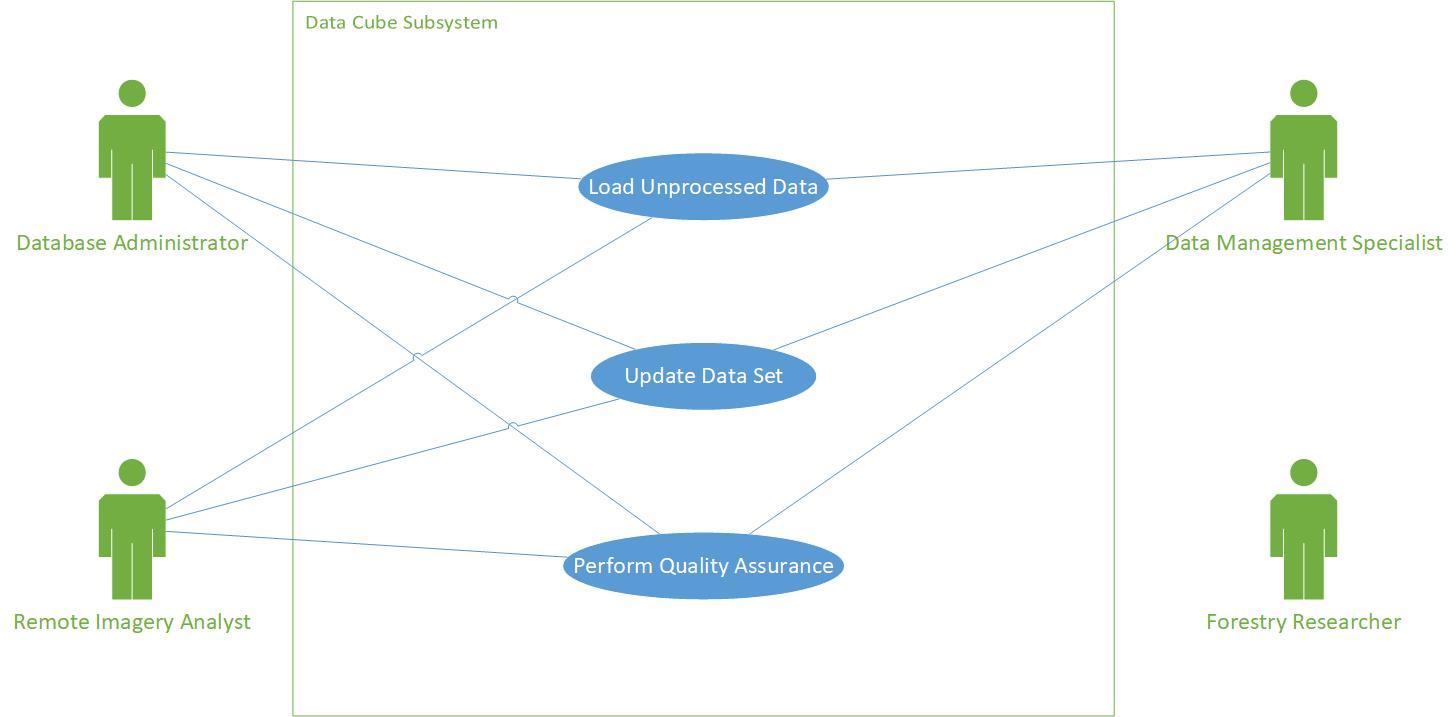


Fig. 2. Out of Scope Use Cases

## **1.2 Out of Scope**

* Administrators and Managers shall load unprocessed data sets using *Data Cube.*
* Administrators and Managers shall use *Data Cube* to perform quality analysis.
* Administrators and Managers shall update data sets using *Data Cube.*

# **2.0 Functional Requirements**

This section describes the functional requirements of the Business Requirements. A functional requirement describes what the system should accomplish. We consider actors, use cases, and data requirements in this section.

## **2.1 Actor Profile Specification**

Th Actor Profile Specification describes actors and their profiles within the context of Business Requirements being documented. An actor is a person, organization, or a software entity that interacts with the subsystem. Actors are external to the system, provide triggering events for a use case, and have goals to achieve by completing a use case. Our actor profile includes PFC users who produce, consume, and manage remotely-sensed raster and gridded data for NFI.

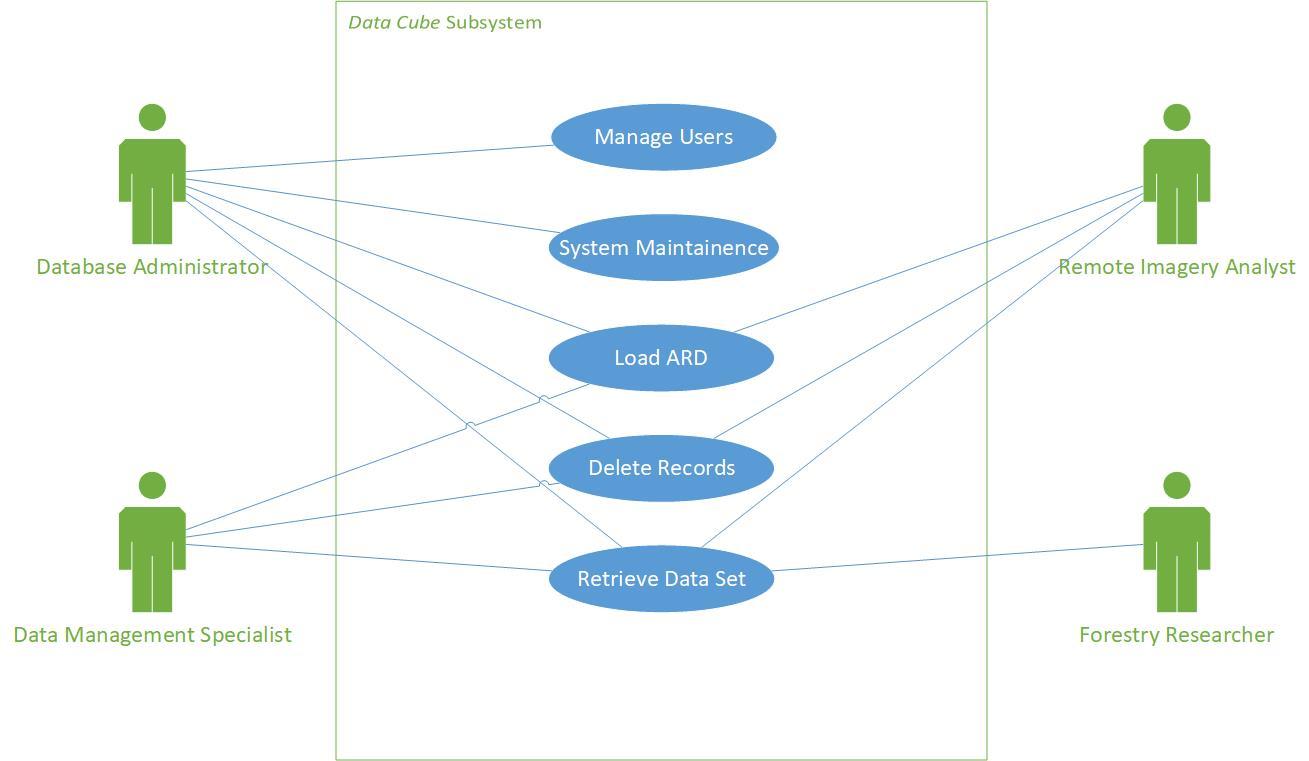
TABLE I

Actor Profile Specification Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor Name** | **Actor Type** | **Access Type needed** | |
| **Database Administrator** | * Stakeholder * Primary Actor * Supporting Actor | * Create * Read | * Update * Delete |
| **Database Management Specialist** | * Stakeholder * Primary Actor * Supporting Actor | * Create * Read | * Update * Delete |
| **Remote Imagery Analyst** | * Stakeholder * Primary Actor * Supporting Actor | * Create * Read | * Update * Delete |
| **Forestry Researcher** | * Stakeholder * Primary Actor * Supporting Actor | * Create * Read | * Update * Delete |

## **2.2 Essential Use Case Diagram**

An Essential Use Case is a generalized use case that captures user goals in a technology and implementation independent manner. We write use cases in order to consider system functionality from the users’ point of view. They are used as the starting point for system design and development.

Fig. 3. Essential Use Cases

## **2.3 Essential Use Case Specifications**

A fully documented essential use case is a structured narrative that describes a single well-defined task that a user carries out. A use case typically has one basic course of action and one or more alternate courses of actions. The basic course of action is the main start-to-finish path that the use case will follow. The alternate courses represent the paths used in situations of errors or exceptions. Essential Use Case Reports are given in the Appendix A.

## **2.4 Business Rules**

This section lists and describes the business rules applicable to the proposed system.

TABLE II

Business Rules Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| **Business Rule Id** | **Rule Name** | **Rule Description** | **Rule Source** |
| BR01 | Users must have access to PFC network | Only employees who have security clearance for access to the PFC network may use the subsystem | Policy manual  Strategic decisions |

# **3.0 Data Requirements**

This section describes the Data requirements part of the Business Requirements.

## **3.1 Data Architecture**

This section describes the Data Architecture requirements part of the Business Requirements.

### 3.1.1 Entity Relationship Diagram

This section depicts the Data Architecture in the form of Entity Relationship Diagram (ERD) [1].

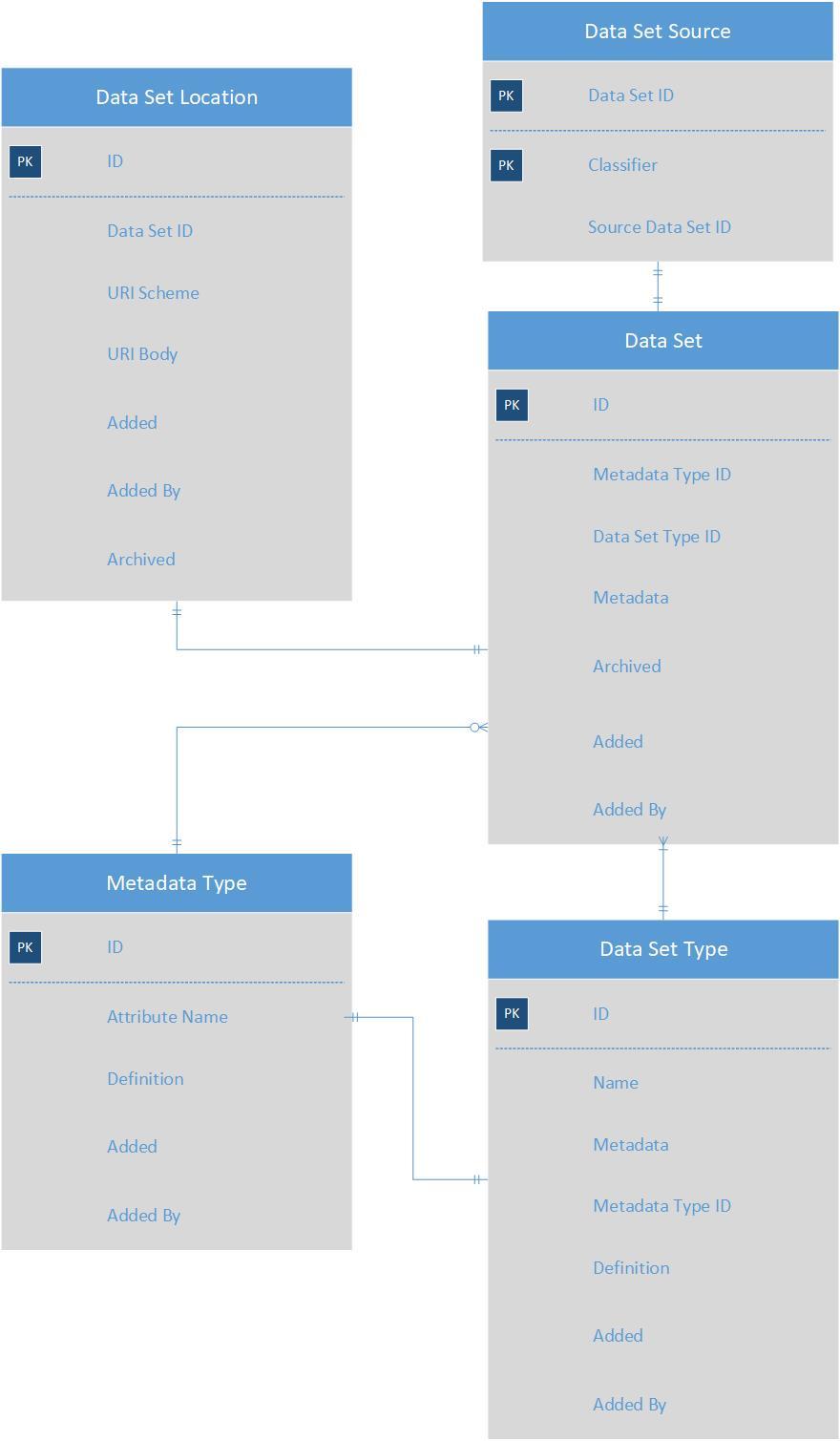


Fig. 4. Entity Relationship Diagram

## **3.2 Data Volumes**

This section describes the expected approximate data volumes for data ingested into *Data Cube*.

The storage required for the system is commensurate with the amount of data the system is given to manage. This includes historical, current, and future data sets. As the quantity, complexity, and measurements grows, *Data Cube* will require more space to manage the volume.

The initial volume estimates for NFI data holdings to load in *Data Cube* are:

* 100 terabytes of historical data.
* 700 acquisitions per year, maximum.

*Data Cube* provides the following formula to estimate growth with acquisition volume [2]:

*Estimate = (number of path/rows)*

*\* (365/revisit time in days)*

*\* (years of storage)*

*\* (data size)*

*\* 1.5 (margin of safety)*

## **3.3 Data Conversion and Storage**

Data stored in *Data Cube* must be analysis ready. Analysis Ready Data (ARD) has the following corrections applied:

* Orthorectification
* Cloud Masking
* Atmospheric Correction
* Instrument Calibration
* Geolocation
* Radiometry

A *Data Cube* instance may contain records about 3 different data sets, as summarized in Table III.

TABLE III

*Data Cube* records summary [3]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type** | **Indexed** | **Available** | **Comments** | **Example** |
| Referenced | Yes | No | The existence and metadata of these data sets is known, but the data itself is not accessible to *Data Cube*. | Raw Landsat Telemetry |
| Indexed | Yes | Maybe | Data has a file location or URI with associated metadata available in a format understood by *Data Cube*, and is created externally of *Data Cube*. | USGS Landsat Scenes with prepared  agdc-metadata.yaml  GA Landsat Scenes |
| Ingested | Yes | Yes | Data is created and/or is managed by *Data Cube*. The data has typically been copied, compressed, tiled, possibly re-projected, and stored in NetCDF4 files. | Tiled GA Landsat Data, ingested into Australian Albers Equal Area Projection (EPSG:3577) and stored in 100km tiles in NetCDF4 |

During the data indexing process, the existence, location, and metadata of a data set is recorded in *Data Cube* index. None of the data set itself is copied, moved or transformed. This is, therefore, a relatively safe and fast process [4]. The following section describes files which are required for the indexing process.

Product Definition

*Data Cube* can handle many types of data and requires information to know how to process them. The product definition file provides a description of the type of data that will be contained in data sets of its type [5]. One product definition file is required for each product type kept in *Data Cube*. The common metadata includes the following items:

* Short name
* Description
* Basic source metadata
* List of measurements:
  + Ordered list of data
  + Name, aliases
  + Data type
  + Unit types
  + Null data
  + Bit level descriptions
  + Spectral response

Data Set Metadata Document

One data set metadata document is required for each individual data set. The document is stored in the database index for *Data Cube* to use for data access. *Data Cube* will not recognize a data set if it does not have a Data Set Metadata document [6].

The document describes what the data represents, where the data came from, and what format it is in. The minimum information to include are fields which shall be queried. It defines critical metadata including:

* Measurements
* Platform and sensor names
* Geospatial extents
* Projection
* Acquisition time

Metadata Type Definition

This document defines the fields that shall be searchable in the product or data set metadata [7]. The default metadata type ‘eo’ is generated by *Data Cube*, and includes fields platform, instrument, latitude, longitude and time. Custom fields for product searches are created with metadata type definitions.

The ingestion process maps data into NetCDF4 format files, and tile it up into a faster storage format or projection system [8]. The following document is required in addition to the previous three documents for the ingestion process:

Ingestion Configuration File

The Ingestion Configuration File defines how *Data Cube* should ingest data for high performance access [8]. The preparations include slicing data into regular chunks, re-projecting data, and compressing data. The file contains:

* Source product name
* Output product name
* Output file location and file name template
* Global metadata attributes
* Storage format:
  + Driver
  + CRS
  + Resolution
  + Tile Size
  + Tile Origin
* Measurement details:
  + Output measurement name
  + Source measurement name
  + Re-sampling method
  + Data type
  + Compression options

## **3.4 Data Retention and Archiving**

Because *Data Cube* is a storage mechanism for important scientific facts, data should be retained indefinitely. We will not incorporate purge or archive cycles into our configuration. Instead, our client will use a combination of reports, statistics, and ranking of importance to manage data retention.

## **3.5 Freedom of Information / Privacy Impact Assessment (FOIPA)**

This section describes the sensitivity levels of each class of data. Each database entity must have sensitivity levels defined in order to protect the data and provide appropriate access.

TABLE IV

Information Sensitivity Matrix

|  |  |
| --- | --- |
| **Entity Name** | **Data Sensitivity Level** |
| Data Set | Non-Sensitive\* |
| Metadata Type | Non-Sensitive\* |
| Data Set Type | Non-Sensitive\* |
| Data Set Location | Non-Sensitive\* |
| Data Set Source | Non-Sensitive\* |

\*Non-sensitive information is information that would not reasonably be expected to cause injury (harm) if released to the public.

## **3.6 Data Definition Report**

This section describes the Data Architecture and definition in a report format.

### 3.6.1 Entity Definition Report

This section is applicable only to the structured approach. This section describes Data Architecture in narrative text form.

The report matrices are given in the Appendix B.

# **4.0 Non-Functional Requirements**

In the Non-Functional Requirements section, we define criteria to measure the qualitative attributes of the system. Defining these criteria will ensure the client and the development team are on the same page. Typically, non-functional requirements include areas that address how the system will deliver security, usability, performance, and user help.

## **4.1 Security Requirements**

### 4.1.1 Authentication

Authentication is out of project scope. All users must have federal government security clearance in order to access the subsystem, users will be inside the PFC network, and securing the authentication system is handled by PFC security experts. The subsystem will exist under a security layer implemented in the PFC system, and will initially be accessible only by local network users.

### 4.1.2 Authorization and Access Control

The authorization process determines if an authenticated user is permitted to access or take action on a requested resource. Roles are a set of access privileges a user class is permitted. Roles are created in the database software. Administrators assign users with a role and the system will check the user’s access privileges for resources with each request.

TABLE V

Authorization and Access Control Matrix

|  |  |
| --- | --- |
| **Actor Name** | **Type of Access Control Needed on the Role:**  **C Create, R Read, U Update, D Delete** |
| Forestry Researcher | R |
| Remote Imagery Analyst | CRD |
| Data Management Specialist | CRD |
| Database Administrator | CRD |

## **4.2 Availability Requirements**

Availability Requirements describe the required up-time of a system. Users should be able to access the system when they need to, and the system must deliver availability to support business functions. Since our subsystem is not primary, the impact posed by availability is low. We expect that since *Data Cube* software is in development, and as it matures, there will be downtime due to system updates and fixes. The following table summarizes availability requirements.

TABLE VI

Availability Requirement Matrix

|  |  |
| --- | --- |
| **Use Case** | **Availability Requirements** |
| Load ARD | Regular Work Hours |
| Retrieve Data Set | Regular Work Hours |
| Delete Records | Regular Work Hours |
| System Maintenance | Regular Work Hours |
| Manage Users | Regular Work Hours |

**4.3 Usability Requirements**

The subsystem is built with open-source software components that provide visualization tools, and standardized *Python* APIs provide data interactions between clients and the database. Because of this, many usability specifications are external to our system. The subsystem’s users have technical expertise in using and managing GIS data software and databases. The subsystem must deliver user experience in its functionality, data conversion and storage, and the component’s integration. The following items summarize Usability Requirements:

* Administrators and Managers must be able to load data for storage.
* Users must be able to connect to the system through compatible GIS applications.
* Users must be able to retrieve data held in the system using GIS software.

## **4.4 System Help Requirements**

We will prepare thorough written documentation to ensure long-term usability and maintainability of the subsystem. The documentation will include standard operating procedures for essential use cases, system rebuilds, and maintenance. The advantage of building the subsystem with off-the-shelf open-source components is that online documentation will be maintained and updated by the organizations creating the software. The primary goal for our documentation is to provide details on the specific implementation details and customizations that are critical for maintaining the system in the PFC ecosystem.

TABLE VII

System Help Requirement Matrix

|  |  |
| --- | --- |
| **Use Case** | **Help Requirements** |
| Load ARD | Offline Operations Manual / Online Documentation |
| Retrieve Data Set | Offline Operations Manual / Online Documentation |
| Analyze Data Set | Offline Operations Manual / Online Documentation |
| Delete Records | Offline Operations Manual / Online Documentation |
| System Maintenance | Offline Operations Manual / Online Documentation |
| Manage Users | Offline Operations Manual / Online Documentation |
| System Rebuild | Offline Operations Manual |

## **4.5 Performance Requirements**

*Data Cube* software uses high-performance architecture and algorithms for raster data management and retrieval, and is an industry leader in response times for raster data management. The biggest impact on the subsystem’s response time is network latency and PFC network security protocols. Estimating performance time is a low priority item, and we will provide more details on response times as the project moves forward.

TABLE VIII

Performance Requirement Matrix

|  |  |
| --- | --- |
| **Use Case Name** | **Performance Requirements** |
| Load ARD | TBA |
| Retrieve Data Set | TBA |
| Analyze Data Set | TBA |
| Delete Records | TBA |
| System Maintenance | TBA |
| Manage Users | TBA |
| System Rebuild | TBA |

## **4.6 Scalability Requirements**

Scalability Requirements describe the potential of the subsystem to handle expansions, add-ons, interoperability, or to process increasing workloads. Because the system is composed of loosely-coupled open-source software that complies with OGC standards, it can be adapted to future needs.

## **4.7 User Scalability**

The subsystem can integrate different GIS visualization software. As long as the selected software complies with OGC standards, it should be interoperable. Possible future subsystem enhancements include:

* A script to generate documents for the indexing and ingesting process.
* An interface to make APIs open to users outside of the PFC network.

## **4.8 Application Scalability**

The subsystem is composed of modular open-source components. Components can be swapped in and out, provided the API interfaces of the selected applications are interoperable, and they conform to OGC standards.

## **4.9 Database Scalability**

Database Scalability is supported by *Postgres* partitioning and load balancing. *Postgres* is a robust database system that supports standard optimizations, and provides many tools to implement the changes.

Partitioning is a process that first divides very large database tables into multiple smaller parts, then redistributes data between the smaller tables. By reorganizing the data in this manner, the database knows what section of memory the requested data is stored at, and has less records to scan in each section to find the data.

Load balancing can also be used to increase retrieval times. This process improves the distribution of workloads across multiple computing resources. *PgBouncer* is a tool provided with *Postgres* that implements load balancing.

# **5.0 Interface Requirements**

This section describes User and System Interface requirements for the proposed system.

## **5.1 User Interface Requirements**

The subsystem will make use of off-the-shelf open-source GIS clients, and other open-source servers, databases, and software. Because of this, designing user interfaces is out of scope for the project. Database Administrator, Data Management Specialists, and Remote Imagery Specialists will use a command line interface to manage data in *Data Cube*. We have suggested enhancing this interface with a streamlined process in the User Scalability section.

## **5.2 System Interface Requirements**

The subsystem will exist in a context that does not require an interface with any automated external data producers or consumers. The data is manually loaded to *Data Cube*, and the system outputs data in response to user queries via a GIS application.

### 5.2.1 Input and Output Requirements

The NFI has a well-defined process to manage data procurement; it is not automated. A Remote Sensing Specialist manages procurement of required data sets from aerial photography and satellite imagery. The Specialist ensures the data is contracted out for transformation to analysis-ready data.

Other Remote Sensing Specialists manage acquisitions of different measurements for NFI, including ground plot measurements. The Specialist ensures the data undergoes a quality assurance testing and transformation process to ensure it conforms to data rules.

The output of these processes is data in a format that *Data Cube* can accept. The remotely sensed imagery data is stored on local drives, while ground plot data is stored in a *Postgres* database. *Data Cube* requires documents that describe the product, the data set type, and the metadata type to either reference, index, or ingest data [9]. More information is given in the Data Conversion and Storage section.

The subsystem will output data in response to requests from GIS clients, and the interface between clients and the servers is standardized. Therefor, the output data requirements of the subsystem are limited to ensuring the clients and servers communicate using the same standard, and that clients can access the data from the servers.

The following points summarize Input and Output requirements.

Input Requirements:

* Raster and sensor data must be analysis-ready.
* Data for referencing and indexing requires definition documents.
* Data for ingestion requires ingestion documents.
* Data for ingestion must be accessible.

Output Requirements:

* Data must be accessible from GIS client.

# **6.0 Throw Away Prototypes**

Throw Away Prototypes involve building a small part of the system to experiment, explore, and test implementation. The advantage of prototyping is the ability of the client to evaluate and provide feedback that can be quickly incorporated into further development. Essentially, the prototyping technique reduces risk by validating or deriving final system requirements.

Our subsystem will make extensive use of prototypes to determine system requirements, *Data Cube* processes, and to develop our understanding of the system. Our prototypes will be created on virtual machines using test data; the environment we create will mimic the real PFC ecosystem. Using virtual machines allows the implementation to be deleted easily and rebuilt. This will allow us to determine dependencies, configurations and other requirements for successful operation. We will thoroughly document each build in order to track progress and include with our system documentation.

# **7.0 Glossary**

|  |  |
| --- | --- |
| **Django Data Cube** | An OGC compliant WCS server for a Data Cube instance. |
| **Geographic Information System** | An information system that integrates, stores, edits, analyzes, shares, and displays geographic information. |
| **GridSpec** | Specifies the geometry of the grid that a subplot will be placed. |
| **MapServer** | MapServer is an Open Source platform for publishing spatial data and interactive mapping applications to the web. |
| **Network Common Data Form** | A set of software libraries and self-describing, machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data. |
| **National Forest Inventory** | A database of measurements taken from specific plots in Canada's Forests. |
| **National Forest Information System** | Provides web tools to access information on the state of Canada's Forests. Uses the NFI. |
| **Open Geospatial Consortium** | An international voluntary consensus standards organization, whose primary function is to collaboratively reach consensus for development and implementation of open standards for geospatial content and services. |
| **OpenLayers** | OpenLayers is an open source JavaScript library for displaying map data in web browsers as raster tile maps. |
| **PgBouncer** | Lightweight connection pooler for PostgreSQL. |
| **PostGres** | An object-relational database management system with an emphasis on extensibility and standards compliance. Free and open-source. |
| **QGIS** | A desktop geographic information system (GIS) application that supports viewing, editing, and analysis of geospatial data. |
| **Web Coverage Service** | OGC standard that defines web-based retrieval of coverages. |
| **Web Map Service** | An OGC standard protocol for internet serving of georeferenced map images which a map server generates using data from a GIS database. |

**8.0 References**

|  |  |
| --- | --- |
| [1] | Open Data Cube, “Data Model” *Open Data Cube*, 2018. [Online]. Available: <https://datacube-core.readthedocs.io/en/stable/dev/data_model.html>. [Accessed April 4, 2018]. |
| [2] | Open Data Cube, “System Requirements” *Open Data Cube*, 2018. [Online]. Available: <https://www.opendatacube.org/system-requirements>. [Accessed April 4, 2018]. |
| [3] | Open Data Cube, “Overview,” *Open Data Cube*, 2018. [Online]. Available: <https://datacube-core.readthedocs.io/en/stable/user/intro.html>. [Accessed April 4, 2018]. |
| [4] | Open Data Cube, “Indexing Data,” *Open Data Cube*, 2018. [Online]. Available: <https://datacube-core.readthedocs.io/en/stable/ops/indexing.html>. [Accessed April 4, 2018]. |
| [5] | Open Data Cube, “Indexing Data: Product Definition,” *Open Data Cube*, 2018. [Online]. Available: <https://datacube-core.readthedocs.io/en/stable/ops/indexing.html#product-definition>. [Accessed April 4, 2018]. |
| [6] | Open Data Cube, “Configuration: Dataset Metadata Document,” *Open Data Cube*, 2018. [Online]. Available: <https://datacube-core.readthedocs.io/en/stable/ops/config.html#dataset-metadata-doc>. [Accessed April 4, 2018]. |
| [7] | Open Data Cube, “Configuration: Ingestion Configuration,” *Open Data Cube*, 2018. [Online]. Available: <https://datacube-core.readthedocs.io/en/stable/ops/config.html#ingestion-config>. [Accessed April 4, 2018]. |
| [8] | Open Data Cube, “Configuration” *Open Data Cube*, 2018. [Online]. Available: <https://datacube-core.readthedocs.io/en/stable/ops/config.html#id1>. [Accessed April 4, 2018]. |
| [9] | Open Data Cube, “Configuration” *Open Data Cube*, 2018. [Online]. Available: <https://datacube-core.readthedocs.io/en/stable/ops/ingest.html>. [Accessed April 4, 2018]. |

# **9.0 Revision Log**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Date*** | ***Version*** | ***Change Reference*** | ***Reviewed by*** |
|  |  |  |  |
| 04/23/18 | 1.0 | - Add Section Numbers | Jeannette |
|  |  |  |  |
|  |  |  |  |

# 

# **Appendices**

## **Appendix A – Essential Use Case Report**

TABLE IX

Essential Use Case Report DC01

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: | Delete Data | UniqueID: | DC01 |
| Area: | Data3 Functionality | | |
| Actor(s): | Database Administrator | | |
| Stakeholder(s): | Forestry Researcher, Remote Imagery Specialist,  Data Management Specialist, Database Administrator | | |
| Level: | Blue | | |
| Description: | User must delete data set from the system. | | |
| Triggering Event: | User enters command. | | |
| Steps Performed (Main Path) | | | |
| 1. User enters command in CLI. 2. System retrieves and deletes the data. 3. System displays successful delete message. | | | |
| Extensions or Alternate Scenarios | | | |
| 1. Delete fails. 2. System displays unsuccessful delete message. 3. No action taken on database. | | | |
| Pre-Conditions: | User is logged in, data is referenced, indexed, or ingested. | | |
| Post-Conditions: | User successfully deletes data references from system. | | |
| Assumptions: | User has Database Administrator role. | | |
| Priority: | High | | |

TABLE X

Essential Use Case Report DC02

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: | Load Data | UniqueID: | DC02 |
| Area: | System Functionality | | |
| Actor(s): | Remote Imagery Specialist, Data Management Specialist, Database Administrator | | |
| Stakeholder(s): | Forestry Researcher, Remote Imagery Specialist,  Data Management Specialist, Database Administrator | | |
| Level: | Blue | | |
| Description: | User needs to reference a data set in the system. | | |
| Triggering Event: | User enters command. | | |
| Steps Performed (Main Path) | | | |
| 1. User prepares metadata document. 2. User prepares product definition document. 3. User determines either to index or ingest data set. 4. User ensures data is accessible. 5. User enters appropriate command into CLI. 6. System processes data appropriately. 7. System displays a successful index or ingest message. | | | |
| Extensions or Alternate Scenarios | | | |
| 1. Load action fails. 2. System display an unsuccessful index or ingest message. 3. No action taken on database. | | | |
| Pre-Conditions: | User is logged in. | | |
| Post-Conditions: | User receives result containing requested data. | | |
| Assumptions: | User has authorization to access PFC network  User has role assigned in the database | | |
| Priority: | High | | |

TABLE XI

Essential Use Case Report DC03

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: | Retrieve Data | UniqueID: | DC03 |
| Area: | Functionality | | |
| Actor(s): | Forestry Researcher, Remote Imagery Specialist,  Data Management Specialist, Database Administrator | | |
| Stakeholder(s): | Forestry Researcher, Remote Imagery Specialist,  Data Management Specialist, Database Administrator | | |
| Level: | Blue | | |
| Description: | User wants to retrieve data | | |
| Triggering Event: | User enters a command to CLI | | |
| Steps Performed (Main Path) | | | |
| 1. User enters values required for query in GUI client. 2. System processes request and sends requested data set. 3. User receives data set. 4. User manipulates data using GUI tools. | | | |
| Extensions or Alternate Scenarios | | | |
| 1. Read action fails. 2. System returns read failure message. 3. No action taken on database. | | | |
| Pre-Conditions: | User is logged in, and data must be indexed or ingested in the database | | |
| Post-Conditions: | User successfully retrieves the requested data set | | |
| Assumptions: | User has authorization to access PFC network  User has role assigned in the database | | |
| Priority: | High | | |

## **Appendix B – Entity Definition Matrix**

TABLE XII

Entity Definition Matrix for Data Set

|  |  |  |
| --- | --- | --- |
| **Entity Name** | Data Set | |
| **Entity Description** | Describes information from a specific data set | |
| **Initial Data Volume** | N/A | |
| **Annual Data Growth** | N/A | |
| **Annual Data Growth** | **Name** | **Description** |
| ID | Primary Key |
| Metadata Type ID | Foreign Key to Metadata Type entity |
| Data Set Type ID | Foreign Key to Data Set Type entity |
| Metadata | Describes how the measurement was taken |
| Archived | Date the record is set to archive |
| Added | Date the record is created |
| Added By | Name of user who created the record |

TABLE XIII

Entity Definition Matrix for Metadata Type

|  |  |  |
| --- | --- | --- |
| **Entity Name** | Metadata Type | |
| **Entity Description** | Defines the fields that should be searchable in the product or data set metadata. | |
| **Initial Data Volume** | N/A | |
| **Annual Data Growth** | N/A | |
| **Entity Attributes** | **Name** | **Description** |
| Attribute Name | Name given to a metadata attribute |
| Definition | Information about metadata definition |
| Added | Date the record is created |
| Added By | Name of user who created the record |

TABLE XIV

Entity Definition Matrix for Data Set Type

|  |  |  |
| --- | --- | --- |
| **Entity Name** | Data Set Type | |
| **Entity Description** | Describes what the data represents, where the data came from, and what format it is in. Also provides a description of the type of data that will be contained in data sets of it’s type | |
| **Initial Data Volume** | N/A | |
| **Annual Data Growth** | N/A | |
| **Entity Attributes** | **Name** | **Description** |
| ID | Primary key for entity |
| Name | Name given to a data set/product type. |
| Metadata | Information about metadata type |
| Metadata Type ID | Foreign key from Metadata Type entity |
| Definition | Product definition type |
| Added | Date the record created |
| Added By | User who created the record |

TABLE XV

Entity Definition Matrix for Data Set Location

|  |  |  |
| --- | --- | --- |
| **Entity Name** | Data Set Location | |
| **Entity Description** | Defines the location of data indexed in the database using a URI. | |
| **Initial Data Volume** | N/A | |
| **Annual Data Growth** | N/A | |
| **Entity Attributes** | **Name** | **Description** |
| ID | Primary key for entity |
| Data Set ID | Foreign key from data set entity |
| URI Scheme | Scheme information portion of URI |
| URI Body | Pathway information portion of URI |
| Added | Date record was created |
| Added By | User who created the record |
| Archived | Date set is archived |

TABLE XVI

Entity Definition Matrix for Data Set Source

|  |  |  |
| --- | --- | --- |
| **Entity Name** | Data Set Source | |
| **Entity Description** | Originator of data set | |
| **Initial Data Volume** | N/A | |
| **Annual Data Growth** | N/A | |
| **Entity Attributes** | **Name** | **Description** |
| Data Set ID | Foreign key from data set entity |
| Classifier | Classifier Information |
| Source Data Set ID | Foreign Key link to parent data set |

## **Appendix C – User Interview**

Morgan Cranny , Remote Sensing Analyst

March 19, 2018

Interviewers: Jeannette Strand, David Kuitunen

Report Prepared by: Jeannette Strand

Morgan is a Geospatial Technician AKA a Remote Sensing Analyst for Canada’s National Forest Inventory (NFI). While the level of analysis he performs is low, remotely sensed data is critical to his position.

He acquires remote sensing data through a contract arrangement with several satellite providers whereby the providers collect data based on acquisition targets provided to them.  Images that meet the needs of the NFI are ordered and catalogued in the NFI. He also works with provinces and territories to request access to their aerial photography and remotely sensed data. In addition, he manages contracts with companies which delineate and attribute the images to make the data analysis ready. Last year he managed the acquisition of 702 images and the interpretation of 247 images.

The NFI compiles the data to provide information on the status of Canada’s forests, and monitor any changes occurring in them. The changes are caused by many things including fires, floods, invasive species, and human activity.

Morgan manages procurement of primarily aerial photography and satellite imagery. The data format he specifies in the order is TIFF, some contractors send TIFF data and jpeg2000 format. The satellite image providers he deals with are Airbus, Digital Globe and KOMPSAT through several data resellers.

Metadata about how the image was captured is also included with images. This includes instrument type, height of capture, sun position, and other information. Using this information, researchers can extract a lot of information about forest condition.

Typically, the resolution is 50cm x 50cm. From this fine-grained resolution, researchers can identify forest type, tree height, and other information. Stereo photography provides the richest resolution; the overlap between images in this capture technique allows researchers to combine stereoscopic viewing and image metadata to determine spatial dimensions of objects.

Morgan first finds plot images which are clear of cloud, haze, and shadow; in other words, images that are 100% image! He then arranges contracts to have specialists apply corrections to make unprocessed data analysis ready.

Depending on how many plots are included in a contract, the process can take months for return of analysis ready data. He estimates the processing can take a day per plot. The process is time and resource intensive and that is why it is contracted out.

NFI grids Canada into numbered plots. Each plot is 2 x 2 km, and each plot occurs every 20 km x 20 km. There are between 10 – 20 thousand plots actively managed by NFIS.

The inventory system was established in 2000, to take advantage of advances in satellite imagery and create a store of remotely sensed data. Some of the data in the collection dates back 20 years.

Each plot is analysed and subdivided into polygons using line-work that draws around areas of similar forest types and ground cover. The makeup of each polygon is attributed and stored with the polygon (species types, ages, heights, etc.). The line-work and attribution persists across time until something happens to change the ground cover.

Measurement monitoring uses a 10-year visitation cycle, meaning every 10 years a plot is revisited for a new measurement. His software categorizes plots in several statuses depending on the stage of procurement they are currently in.

He provides order information to satellite providers for capture cycles that occur between June, July, and August. Summer months are best for captures as the ground is mostly snow free.

Morgan primarily uses ARCGIS. He uses the software to assist in determining what images to acquire for each capture cycle. He determines the priorities based on a few factors, including what images have not been captured, and plots that haven’t been captured within a certain time frame.

PCI Geomatics, ERDAS, and GRASS software are used for making data analysis ready; this process, however, does not happen on site.

Morgan uses the NFI plot number to manage data locally. The data is then stored on hard-drives, and is shared by loaning hard-drives. A PostgreSQL database is also used, which stores attribution data.

Data is shared by loaning hard-drives and copying information to other storage media like USB drives. FTP is also used for transferring images. Provinces will often transport data on a hard-drive to be included in NFIS.

Data can be shared amongst researchers at PFC, universities, and research centers. Sharing is greatly limited by permission and rights restrictions.

Morgan orders products in the Universal Transverse Mercator (UTM) projection. This is a common projection used in small scale within UTM zones.  It preserves both area and distance within a zone, and provides great accuracy within a zone. UTM salvages as much area and distance accuracy as possible.

Another format is the Canadian Albers Projection. This system preserves area well, but does not preserve distances. Ultimately, the project system used will reflect the information that is most important to retain.

The biggest problem is no network storage access. It would be best to have centralized network storage for user access and collaboration. This would eliminate local storage of data and lead to increased data discovery.  Morgan uses a terabyte hard-drive each year in storing new acquisitions.

There is a framework called “Neo Def” in Ottawa that stores remotely sensed data. Using it presents issues in the form of network latency and the way access to the system is organized.

Alex Song, Data Management Specialist

March 21, 2018

Interviewers: Jeannette Strand, David Kuitunen

Report Prepared by: Jeannette Strand

Alex works for the NFI program as a Data Management Specialist. He does not work with raster data at all. Under his management are data from ground plot measurements, and he must see to quality assurance and data integrity verification. He also performs statistical analysis on the data.

His role is highly collaborative. The NFI group currently has 8-9 individuals, who are from various backgrounds including forestry and program management. Alex also works with federal and provincial governments to gather and receive data for the NFI.

Data must undergo a quality assurance test before it is included in the database. Alex uses QA software developed in-house solely for use at NFI. It is a JAVA RCP software tool. The QA tool performs a series of tests on the data to ensure it conforms to rules defined in the NFI data dictionary.

The verification process inspects the incoming data against many rules and constraints, some of which are format, precision, and data types. There are also more complex rules including data attribute dependency testing and relational quality assurance.

The QA process is complex and this features only some of the checks performed. Once data passes the QA test, it is loaded into the database. At this point integrity becomes a primary focus, and Alex uses constraints to enforce integrity.

Alex works with a PostgreSQL database system, and uses PostGIS and PG Admin. Data visualization is not critical for his work, but he uses QGIS and ARC GIS if he needs that type of tool.

All of the data manipulation is done with SQL scripts executed against the data base. There isn’t an IDE or GUI used for those interactions. He uses a SASS software package for statistical analysis of data.

Some of the data analysis variables taken during data gathering include land coverage. This cluster of entities describes qualitative measurements taken from a forest sample. Some of the properties measured include tree width taken at a certain height, tree species, condition of trees, and damage to trees. Soil samples are also taken, which are sent to a laboratory for analysis including composition, wood debris, and PH testing. The aim here is to quantitatively describe the forest state at a specific location.

The sharing process for NFI is well defined. Data is collected collaboratively with the province and territories. There is an online data request system, agreements are sent to collaborators for rights release. There are authorization access controls. The data is not open to the public due to the need for rights management control.

Storage is always an issue. The NFIS group manages the db servers and data management, and because of data size, storage is always an issue for photo plot data. The ground plot data is not of a significant size, so storage and access isn’t as much of an issue.

Kristian Arndt, National Programs Database Administrator

March 26, 2018

Interviewers: Tyler Hall, David Kuitunen

Report Prepared by: David Kuitunen

Kristian is a National Programs Database Administrator. His role is to keep the databases running and meeting with end-users to ensure data integrity, and that constraints are put in place. Kristian will be providing a sandbox environment for us to implement ODC.

Kristian is mostly concerned about load and space on the database. Questions have been raised about metadata being stored on the database, but processing is off the database and that’s where most of the processing load will be. Do results from API calls get cached, or does it convert data to [netCDF4](https://en.wikipedia.org/wiki/NetCDF)? How does redis cache? We will need to answer these questions while experimenting in our sandbox environment that will be set up.

[PostGRES](https://en.wikipedia.org/wiki/PostgreSQL) 9.6 and [pgAdmin](https://www.pgadmin.org/) 3 are used.

He advises that the following rolescan be useful: Admins, Read-Write, Read-Only, Editors, Owners.

He advises that for issues of scaling, Once the database gets large enough it may be necessary to partition the DB (break up into different tables). Load balancing may be necessary at that point as well. [PgBouncer](https://wiki.postgresql.org/wiki/PgBouncer) is a a lightweight connection pooler for PostgreSQL.

Hao Chen, Remote Sensing Research Associate

David Hill, Forest Landscape Analyst, Remote Sensing

April 4, 2018

Interviewers: Jeannette Strand, Tyler Hall

Report Prepared by: Jeannette Strand

David Hill’s background is in remote sensing research and GIS. He has been in radar research since 1989 and has collected over 800 radar scenes for the Petawawa Research Forest. Some of the Earth Observation data he analyzes are from a variety of Radar (X,C,L Band) satellites  and numerous optical satellites (panachromatic to multispectral, low to high resolutions from 1Km. to 0.25cm.) , info-metrics product, LIDAR data, digital aerial photography, drones, and multiple terrestrial sensors including hyper-spectral sensors.

Hao Chan’s background is in remote sensing data, GIS data processing and applications, and other data sources. He has worked with remote satellite and remote ground data as well.

One of the long-term research forests under management is the Petawawa Research Forest, located in Ontario. Research has accumulated over 100 years of GIS data and ground plot measurements from this forest. This forest is a region where a lot of data has been gathered and is an area of interest; because of this it is a good candidate for inclusion in the Data Cube test data group. They can access USGS satellite imagery, Radar satellite imagery, and other data sources of differing layer types.

They are very interested in the procedures the Data Cube uses, documentation on the system we implement, and to be able to adapt procedures to different approaches. They can provide us with same data sets of varying sizes that are not too time consuming to process. They want to know how we build the Data Cube, put it online and make it accessible to users. They are interested in finding out how the Data Cube handles projections and spatial resolutions. The project is essentially a proof of concept and can provide links to existing repositories of data.

They have an internal network storage that their group can access. They download a scene, process it, and store the product locally. Users can contact them for access to the product.

This leads to access of USGS data. USGS data can be accessed in a variety of ways; you can put in an order and set up a batch process to download the data.  Access through WMS would allow you to pull data in real time or use a service like Google Earth Engine. In Google earth Engine you can access the data, develop models, and download the results from your model.  Limitation is that you can only access data hosted on the Google server. You can upload your own data but that requires a subscription (cost) and may be prohibited by Government of Canada data information protocols as data would be held outside of Canada.

The data is analyzed to find measurements that can be used to derive further information. They look at deriving applications for forest products. For example, they would need to determine what scenes in an area would provide the greatest separability between two samples of forest types, then use that information to derive a product to apply an algorithm that identifies these forest types throughout the rest of Canada. They develop a hypothesis, test it using samples of analysis-ready data, and answer a bigger question. Multiple layers from data products could be utilized to derive a single layer that represents the forest canopy height. Remote sensing data is used as a tool to produce maps, via geographic information systems or image analysis software, and is the primary data from which they derive information.

Licensed software like ArcGIS presents limitations due to restrictions on use. They would like to use open-source software, like QGIS, Snap, and GXL, and want to determine if this type of software can solve problems at a certain level. In addition, there is interest in exploring Fusion Mapping connectivity with the Data Cube.

NFIS hosts a system that uses a best pixel approach. The software does a change mapping to produce a national map of Canada showing changes in the landscape and has some tools that use spectral analysis to estimate the cause of the change.  For example, the spectral response from forest harvesting is different from that of an insect defoliation event.

One issue is the accuracy of measurements. There are errors in measurement, such as instrument calibration and georeferencing, which can compound through the data as it is used in other studies and analyses. For example, when comparing one satellite image to another said to be of the same extent, often a pixel will represent two different locations. Transformations to make data analysis ready attempt to correct these errors, but often do not provide a history of information on the correction process that would give a user confidence in the accuracy of the data-product. Metadata is very important in achieving accuracy and confidence in a data set.

Establishing the accuracy of a data product is important. If there is not an in-depth paper describing the process used to derive a data product, there is no way to trust the accuracy of the product.

They see this project and the Data Cube system as an information building opportunity. They can produce a layer, use it to derive more information, and save the new product back to the Cube. Users at CFC across Canada could come to the cube to look for a product type before processing it themselves. They also would like to use it to build a spatially explicit model.

In regards to their use of vector and raster data, vector data typically maps borders and roads but can also be used for temperature measurements as well as other sensor readings. Raster data is large and cumbersome to process and store, that often vector data is easier to handle and can be used for all scales from very large photo plots, to smaller soil pit surveys.

Jeff Dechka, National Programs Database Administrator

March 26, 2018

Interviewers: Tyler Hall, David Kuitunen

Report Prepared by: Tyler Hall

Jeff has a geography background working with both government, and private sector. He has had the opportunity to travel all over the world and work with many people with a variety of backgrounds in geographical technology. Jeff is interested in the application of the technology as opposed to engineering and design.

His role at PFC is to oversee research programs involving EO data, and oversee samples being taken on the ground. NFIS was built following the vision of the Canadian Council Forest Ministers around 2000. Development was complete around 2004, and then ministers lost interest in NFIS.

The goal of NFI is to provide information to meet legislative commitment for reporting ie. National Forest Report. Data used in science within departments and externally by people with science and analysis specialities. Many applications, including soil data for carbon readings, has role in understanding carbon balance in the forests.

Other goal is to meet international reporting commitments. Challenge is sustainability of forests. Canada has houses a very significant portion of sustainable forests in the world. This data can be used to defend Canada from criticisms from the international community (ie. able to respond to accusations of complete deforestation with data). Show the contrast of real deforestation vs. what is really happening using international definitions.

There are two different elements to forestry in Canada. Provinces are land managers, and they provide detailed economic inventories. These inventories are used to assess sustainable annual allowable cuts in any given year. NFI is a more strategic inventory to tell how the forests are doing over time. It is done in conjunction with provinces and territories.

Canadian Forest Service works internationally to resolve disputes. International inquiries go to federal government before provincial. When challenges happen, NFI works with provinces to keep messages consistent. NFI doesn’t take sides in disputes, but provides informational support for decision making.

The goal of NFIS is to improve ability to report on Canada’s forests. NFIS is building applications that can be used many times but only has to be built once. Improve ability to report on Canada’s forests by connecting all provinces systems without inconveniencing anyone. Numbers would then be submitted back to the provinces to make sure the numbers make sense. Reduces load on provincial agencies. There has been lots of development over last 10-12 years. In some respects NFIS has fallen behind in interoperability. When ministers and others lose interest (senior managers), funding tends to dry up. Much of the money that government assigns to different activities is based on government’s priorities.

Some of the Earth Observation research programs he is responsible for include about a half-dozen research scientists across Canada that look at what the EO data can be used for. Monitoring Earth’s forests, applications for detailed forest inventory, and

how to map and monitor deforestation (and reforestation) activities are a few of program objectives. Afforesting has been happening in northern BC; companies offsetting carbon credits. There are not as many reforestation activities as deforestation. Reforestation only counts what’s planted, not what naturally grows back. EO also helps with fire monitoring, and predicting fire movement.

LIDAR is used for tree height and various tree structural elements (and models these). Exploring how radar satellites can assess biomass and severity of forest fires. People looking at drone usage, ie. a forest company has clear cut area and lots of biomass left on the ground, drones to be used to assess how much biomass is left. Verify enough biomass is left for area to be replanted at a future date.

EO data being looked at to monitor temporal trends. Recently a scientist created 30 years of mosaics for the entire country. What has been forested and what has been burnt, attribute it to each year. Provinces and territories don’t always have that information. They may only track what was there prior to change not what was happening during the change. Time slices were every year.

Regarding challenges with system interoperability used, in general interoperability hasn’t been an issue. Formats that data is acquired with are fairly standardized. There are always challenges if there are other systems that you try to talk to and you can’t. Ability to use and extract data not a problem, its separate systems being able to communicate with each other that causes difficulty. Not just looking at Victoria, using global images for big data analytics. Tools being developed to do more complicated stuff to rationalize what’s of value and what’s not. With more and more data, not everyone has access to supercomputers. Some work can be done on a workstation depending on how long you can wait for a result. Mapping forest change on an annual basis, can take much longer.

# **Authorization**

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Sponsor** |  |  |  |
| Brian Low |  |  |  |
| *Manager, NFIS* | Signature |  | Date |
|  |  |  |  |
| **Team** |  |  |  |
| Jeannette Strand |  |  |  |
|  | Signature |  | Date |
| Tyler Hall |  |  |  |
|  | Signature |  | Date |
| Mingyang Wang |  |  |  |
|  | Signature |  | Date |
| David Kuitunen |  |  |  |
|  | Signature |  | Date |