Software Requirements Specification

NGDS

Version 2.7

*11/05/2012*

Version History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Date** | **Reason** |
| 0.1 | SCR | 08/24/2012 | Initial Draft Created |
| 1.0 | SCR | 09/15/2012 | Initial Requirements Specifications |
| 1.1 | SMR/USGIN | 9/18/2012 | Review and comment (track changes on) |
| 1.2 | SCR | 10/2/2012 | Address Feedback |
| 2.0 | SCR | 10/11/2012 | Incorporated Steve Richard’s changes |
| 2.1 | SCR | 10/16/2012 | Incorporated AnthroTech end-user research input |
| 2.2 | SCR | 10/18/2012 | Incorporate Matt MacKenzie’s revisions |
| 2.3 | SCR | 10/19/2012 | Incorporate Ryan Clark’s revisions |
| 2.4 | SCR | 10/19/2012 | Duane Nickull |
| 2.5.3 | SCR | 10/26/2012 | Restructured document to incorporate Steve Richard’s proposal. |
| 2.6 | SCR | 10/31/2012 | Incorporate Matt MacKenzie’s revisions |
| 2.7 | SCR | 11/05/2012 | Incorporate Duane Nickull |

**Executive Summary**

It is the U.S. Department of Energy (or DOE) vision to discover and exploit geothermal energy sources (s. Section 1.1). The DOE and other organizations are funding a variety of research activities around that vision. So far these research activities focus on either **collecting** geothermal data or **presenting and analyzing** such data.

**Existing Data Collection Activities:**

Data collection activities focused on the creation of various repositories of geothermal data. For example, DOE-GDR – Geothermal Data Repository ([https://gdr.openei.org](https://gdr.openei.org/)), SMU – Southern Methodist University (http://geothermal.smu.edu), EGI – Energy and Geosciences Institute (<http://egi.utah.edu/>), USGS – U.S. Geological Survey (<http://energy.usgs.gov/OtherEnergy/Geothermal.aspx>), and AASG – Association of American State Geologists (<http://www.stategeothermaldata.org/>). These repositories collect structured data (e.g. well headers or heat flow data expressed in well formed tables) as well as unstructured data (mainly publications as well as other documents – which have varied content as text, photos and had written text, that are usually not represented in well known tabular formats). Due to these initiatives, a large amount of data has been made digitally available: Structured data is aggregated in datasets which are exposed as Web Feature Services [WFS], Web Map Services [WMS] and Web Coverage Services [WCS] while unstructured data is made available for download. These repositories are all based on individual software systems which all comply with standardized protocols (WFS, WMS, and WCS).

Moreover each repository exposes a catalog that allows for searching within the data of the repository. The catalog has a metadata entry for each structured dataset (i.e. aggregated structured data). The metadata entry describes the content of the dataset, such as the type of data (e.g. heat flow, well log, etc.), its origin, and the geographic region covered by the dataset. The details of the metadata entry depend, at least partially, on the type of data. In the case of unstructured data, the metadata entry comprises the results of a keyword indexing service.

The catalog is made accessible via the Catalog Service for the Web (CSW) for all users without any password protection. The publication of datasets in NGDS, however, is restricted to authorized users.

**Existing Presentation and Analysis of Data Activities:**

Data analysis activities focused on the development of tools (such as the geothermal prospector [URL: <http://maps.nrel.gov/gt_prospector>]) that allow for analyzing and visualizing data. There are also commercial tools for data analysis available.

**How is NGDS going to contribute to this landscape of research projects?**

NGDS (National Geothermal Data System) shall fill the gap between data collection activities and data analysis activities in existing DOE funded projects.

The most important new feature will be its ability to harvest the catalogs of all existing geothermal repositories (specifically the SMU repository as well as the AASG repository) provided that they comply with the standardized CSW and make it freely available.

But NGDS shall provide more than an aggregating CSW catalog: It shall also provide the future default solution for geothermal repositories and help the DOE to build a grid of geothermal data repositories. The content of all repositories in this grid shall be made searchable via a federated search mechanism that will give the user the possibility to execute facetted searches across all repositories, evaluate the detected datasets and make them available for download[[1]](#footnote-2). The federation will make search transparent to end users/ data consumers that will have access to all data in the NGDS network of repositories through any node of the system.

The target audience of NGDS, here called End Users/Data Consumers is a variety of users including legislators, federal and state agencies, financial investors, researchers, educators and students, interested public in general and industry representatives.

In the following we list the five basic needs that need to be realized by NGDS:

1. NGDS shall enable data collectors to create and administrate a repository for geothermal data.
2. NGDS shall enable end users/data consumers to search geothermal data across a multitude of repositories
3. NGDS shall enable end users/data consumers to evaluate discovered data
4. NGDS shall enable end users/data consumers to acquire (i.e. download) selected data
5. NGDS shall enable the end users/data consumers to analyze selected data

These five basic needs will be further explained as follows.

**Need for a Standard Data Repository**

As outlined above, the currently funded activities lead to a variety of software systems for archiving geothermal data. However, the DOE requires a default software system to be used for future geothermal data collection projects. We call this software system a “node-in-a-box”.

This node-in-a-box must have a simple way for setting up such a repository. The repository must be simple to administrate, flexible with respect to configuration and adaptation by the data collector, and must rely on standard technologies. Most importantly, it must allow for federating its content thus allowing the DOE to realize a grid of repositories for geothermal data.

Besides enabling data collectors to store their geothermal data, it must also provide a minimum set of housekeeping features such as system monitoring, user management, logging of activities, support for backups, and basic security. The system must also realize a number of basic features to adjust access rights to data within the system, distinguishing between data consumers (readers that can access and search for data published in the system) and providers (writers that can provide new content and modify existing data and metadata).

Also, the repository must support some basic business process for uploading, evaluating and publishing datasets. This process may involve multiple users, e.g. one user responsible for uploading data, and a second user responsible for reviewing the uploaded data and making it publicly available (or rejecting it).

**Search across Multiple Repositories**

NGDS must allow for searching through the catalog of datasets within the grid of federated NGDS nodes. This catalog, which indexes the data across all nodes, must be accessible by each participating node.

End users/data consumers must be provided with a user interface that allows for executing faceted searches that combine many different search filters such as geographic region, type of data, data provider as well as keywords. The found results shall be visualized in an appropriate user interface. The most important aspect of this user interface is that it shall visualize the found datasets in a map in an appropriate way (if found results can be geo-located and displayed appropriately). The map must allow for the usual map features such as panning and zooming. Also, the user interface must display metadata for the found search results in an appropriate way. It must be possible to sort the tables and to export them in an appropriate file format (typically CSVs).

**Data Evaluation**

Data analysts need to have the possibility to quickly review and evaluate found data sets. They may do this, for example, by relying on peer ratings of these datasets (ratings given by other users). Hence, data analysts need to be enabled to rate a dataset as well. They may also want to give ratings to the origin of data. Analysts may also want to be able to triangulate regions of interest by overlaying information from multiple datasets, utilizing the geo-location associated to each dataset.

**Data Acquisition**

Metadata must contain a URL for accessing the complete datasets. In case of unstructured data, the URL may point to resources such as a PDF, TIF, JPEG or other type of file. In case of structured data, links for the supported services (WFS, WMS, WCS) must be provided and it must be possible to download the complete dataset from the originating server. The NGDS portal must provide a user interface that allows for downloading datasets in the appropriate form.

**Data Analysis**

Data Analysis is the strength of existing commercial and open source tools. The NGDS user interface may provide some very basic features for analyzing data. However, it is not the goal to develop a full fledged data analysis tool. Therefore, data analysis will be limited to some very basic features such as display data in tables, and layering of different types of data as a way to improve data discovery. All other data analysis will be performed by third party tools that will download the dataset for their particular use.

Table of Contents

[Software Requirements Specification 1](#_Toc339446643)

[NGDS 1](#_Toc339446644)

[Version 2.6 1](#_Toc339446645)

[Version History 2](#_Toc339446646)

[Table of Contents 6](#_Toc339446647)

[List of Figures 8](#_Toc339446648)

[List of Tables 8](#_Toc339446649)

[1 Introduction 9](#_Toc339446650)

[1.1 Vision & Goal 9](#_Toc339446651)

[1.2 Landscape of Geothermal Data Tools 9](#_Toc339446652)

[1.3 End User Survey Results 11](#_Toc339446653)

[2 User Communities and Roles 13](#_Toc339446654)

[Figure 2 Main user roles and their relationships 13](#_Toc339446655)

[2.1 Data Provider Community 14](#_Toc339446656)

[2.1.1 Data Submitter Role 14](#_Toc339446657)

[2.1.2 Data Steward Role 15](#_Toc339446658)

[2.1.3 Node Administrator Role 15](#_Toc339446659)

[2.2 End User Community 15](#_Toc339446660)

[2.3 System Administrators 16](#_Toc339446661)

[2.4 Software Developer Community 16](#_Toc339446662)

[3 Use Case Models 16](#_Toc339446663)

[3.1 High-level Use Cases View 16](#_Toc339446664)

[Figure 3: High-level Use Case View 17](#_Toc339446665)

[3.2 Data Provider Use Cases 17](#_Toc339446666)

[Figure 4 : Data Entry and Submission Use Case View 19](#_Toc339446667)

[3.2.1 Data Submitter 20](#_Toc339446668)

[3.2.2 Data Steward 30](#_Toc339446669)

[Figure 5 Quality Assurance Use Case View 33](#_Toc339446670)

[3.2.3 Node Administrator 42](#_Toc339446671)

[Figure 6 Administration Use Case View 43](#_Toc339446672)

[3.3 Software Developer Use Cases 53](#_Toc339446673)

[Figure 7 Software developer use cases 53](#_Toc339446674)

[3.3.1 Develop Applications 53](#_Toc339446675)

[3.3.2 Register New Application with NGDS 54](#_Toc339446676)

[3.4 End-User Use Cases 55](#_Toc339446677)

[Figure 8 End-user use cases overview 57](#_Toc339446678)

[3.4.1 Discover and Gather Data 58](#_Toc339446679)

[Figure 9 Data gathering supporting use cases 58](#_Toc339446680)

[3.4.2 Validate and Evaluate Data 65](#_Toc339446681)

[Figure 10 Data validation supporting use cases 65](#_Toc339446682)

[3.4.3 Analyze and Visualize Data 75](#_Toc339446683)

[3.5 System Administrator Use Cases 76](#_Toc339446684)

[Figure 12 NGDS Administration use cases 77](#_Toc339446685)

[3.6 Use cases common to all users 81](#_Toc339446686)

[4 Overall system quality attributes 82](#_Toc339446687)

[4.1 Maintainability 82](#_Toc339446688)

[4.2 Usability 83](#_Toc339446689)

[4.3 Performance and Scalability 84](#_Toc339446690)

[4.4 Security 86](#_Toc339446691)

[Here follows the System’s requirements for security and accessibility. 86](#_Toc339446692)

[4.5 Supportability 88](#_Toc339446693)

[4.6 Data Requirements 88](#_Toc339446694)

[4.7 Design Constraints 90](#_Toc339446695)

[4.8 Licensing Requirements 91](#_Toc339446696)

[4.9 Applicable Standards 91](#_Toc339446697)

[4.10 Installation and Deployment 91](#_Toc339446698)

[**Appendix** 92](#_Toc339446699)

[5 Node-in-a-box Software Package 92](#_Toc339446700)

[Figure 13 NGDS Data Provider Software Package 93](#_Toc339446701)

[6 End-User/Data Consumer Software 94](#_Toc339446702)

[6.1 NGDS WebSite 94](#_Toc339446703)

[6.2 NGDS WebApp 94](#_Toc339446704)

[7 References 94](#_Toc339446705)

[7.1 Project References 94](#_Toc339446706)

[7.2 External References 95](#_Toc339446707)

[8 Acronyms, and Abbreviations 95](#_Toc339446708)

[9 Glossary of Geological Terms 96](#_Toc339446709)

[10 NGDS Data Access Protocols 97](#_Toc339446710)

[11 Data Model 98](#_Toc339446711)

List of Figures

[Figure 1 Geothermal Landscape of Tools and Data Repositories. Red-framed elements will be developed within NGDS. 12](#_Toc339282979)

[Figure 2 Main user roles and their relationships 14](#_Toc339282980)

[Figure 3: High-level Use Case View 18](#_Toc339282981)

[Figure 4 : Data Entry and Submission Use Case View 20](#_Toc339282982)

[Figure 5 Quality Assurance Use Case View 33](#_Toc339282983)

[Figure 6 Administration Use Case View 43](#_Toc339282984)

[Figure 7 Software developer use cases 53](#_Toc339282985)

[Figure 8 End-user use cases overview 56](#_Toc339282986)

[Figure 9 Data gathering supporting use cases 57](#_Toc339282987)

[Figure 10 Data validation supporting use cases 65](#_Toc339282988)

[Figure 11 Data analysis 75](#_Toc339282989)

[Figure 12 NGDS Administration use cases 78](#_Toc339282990)

[Figure 13 NGDS Data Provider Software Package 95](#_Toc339282991)

List of Tables

[Table I types of data users are interested in searching for 12](#_Toc339282992)

[Table 2: Project Reference Documents 97](#_Toc339282993)

[Table 3: External Reference Documents 97](#_Toc339282994)

[Table 4: Abbreviations 98](#_Toc339282995)

[Table 5: Terms 99](#_Toc339282996)

# Introduction

This Software Requirements Specification (SRS) collects, organizes and describes requirements for the NGDS software system captured through use-case models, and from natural language requirements statements from a sample of system users. These include functional requirements, non-functional requirements (NFR), design constraints, and other factors needed to provide a comprehensive picture of the software’s operation.

Expected readers of this document are:

* Project management
* Project partners
* Software architects and designers
* User interface designers
* Software developers
* Software testers

This document was prepared by Siemens Corporate Research (SCR) to document requirements for the National Geothermal Data System (NGDS).

## Vision & Goal

The ultimate goal of the National Geothermal Data System (NGDS) is to support the discovery of geothermal sources of energy. The NGDS will provide online access to important geothermal-related data from a network of data providers in order to:

1. Increase the efficiency of exploration, development and usage of geothermal energy by providing a basis for financial risk analysis of potential sites
2. Assist state and federal agencies in making land and resource management assessments
3. Foster the discovery of new geothermal resources by supporting ongoing and future geothermal-related research
4. Increase public awareness of geothermal energy

## Landscape of Geothermal Data Tools

The NGDS will enable the discovery of geothermal data covering a wide range of topics, from well logs and drilling data to temperature, geochemical, and geophysical measurements. Standardized data access to important datasets will facilitate utilization of these information resources.

A key component of the system is the catalog service through which data providers will register the availability of geothermal data resources, and through which users will discover, evaluate and access these data resources. A data resource will be considered part of the system when it is published. i.e. its metadata is available in the NGDS catalog, allowing it to be discovered. The metadata record shall describe, among other things, the geo-location of the data resource and how the data resource can be accessed. Data providers will maintain nodes in the network, connected through the use of predetermined web-service protocols and standardized data and metadata representations for exchanging information. These standards will be developed in conjunction with the US Geoscience Information Network (USGIN), thereby providing interoperability with a wider range of geo-scientific information.

Figure 1 illustrates the overarching NGDS vision, showing its role in the landscape of Geothermal Tools and Data Repositories: NGDS will facilitate the integration of geothermal applications (front-ends) with a diverse set of data repositories (“back ends”). A core feature of NGDS is a **catalog service** that allows users to locate data repositories based on their metadata information (the catalog will include geographic location information thus allowing the representation of data on a map). Once the data is located, using the catalog services, user applications can directly access and import that data from a registered NGDS or third party repository, through the use of standard protocols (such as WCS and WFS).

Note that NGDS is not only going to provide the NGDS catalog service. It is in addition going to provide the **NGDS repository** that will provide simple-to-use means for data providers for uploading, validating and publishing their data to NGDS.

Note also that NGDS is going to provide a user application called **NGDS WebApp**. This Web-based frontend shall serve as a user interface for end users who are consumers of geothermal data. The WebApp shall allow for efficient search mechanisms, powerful means to represent discovered data, validate and access it. The WebApp shall also include simple mechanisms for analyzing the data, however its focus Is placed on searching and representing on a map-centric user interface. Specifically the WebApp shall be able to overlay various datasets on a map thus allowing for locating regions that are of potential interest for the end user (e.g. regions that have a high potential to be used for geothermal power exploitation.

For a deeper statistical analysis of individual datasets the user shall be enabled to download the data from the originating data repository.

The combination of the WebApp, with the NGDS repository, and a local node of NGDS catalog represent a node-in-a-box distribution, an application that, once installed, will allow users to participate in the NGDS network. Hence, there may be many instances of node-in-a-box components. i.e. there may be many NGDS repositories, many NGDS WebApps and many NGDS catalogs (that are federated into a distributed catalog services layer).



Figure 1 Geothermal Landscape of Tools and Data Repositories of NGDS.   
Note that red-framed elements represent sub-components of NGDS node-in-a-box.

## End User Survey Results

Anthro-tech (<http://www.anthro-tech.com/> ) has executed a survey in order to collect information about end expectations to NGDS (**see /P05 document in References section**). Among other information Anthro-tech collected data about the types of data end users are interested in comparing to geothermal data stored in NGDS. This survey made it clear that users are not only interested in strictly geo-thermal data. They also need access to data that is indirectly related to the geothermal research field, such as location and type of power lines, roads, or locations of national preserves. Table I lists the results some of these terms from the survey:

Table types of data users are interested in comparing to geothermal data

|  |  |
| --- | --- |
| Active faults  BLM boundaries  Brown fields  Case studies  Commercial volume  County borders  Drill hole temperature  Electrical resistivity  Enhanced geothermal systems  Financial investments in geothermal exploration  Fluid composition of spring  Fluid composition of well water  Fluid inclusion data  Fractures  Fumeroles  Gamma ray  Gas sample  Geothermal permit data  Geothermometers  Gravity  Heat capacity of rock  Heat flow  Hot spring locations  Hydrothermal alteration  Hydrothermal eruptions  Leases  Magnetic survey  Micro earthquakes/ seismicity  Mineral exploration wells  Mineral rights  MT surveys  Mud pots  National parks  Permits issued  Permits requested  Power plants | Powerlines  Production data  Proppants  Proximity to transmission lines  Radioactivity  Remote sensing data of soil chemistry  Roads  Rock density  Rock type  Sage grouse habitat  Satellite images  Scaling  Seismic activity  Seismogenic folds  Soil gas chemistry  Soil samples  State borders  Surface expressions  Surface geological map  Surface manifestations  Temperatue of surface manifestation  Temperature gradient map  Thin-section analyses  Tracers  Transmission lines  Vegetation  Volcanic activity  Volcanoes  Water availability  Water rights  Water wells  Well casing  Well logs  Who owns the land  Wilderness area  Zoning information |

The survey made it also clear that “Overlaying” is a crucial feature for the NGDS user interface: It must be possible to overlay datasets on top of a geographic map thus allowing the end user to “triangulate” regions that are of specific interest. In the following we list data that the NGDS WebApp must be able to overlay during map-based searches:

* **Geographical –** coordinates for the area.
* **Geologica**l – rock type, mineral, seismic activity, surface expressions, heat capacity of rock, rock density
* **Geochemical** – soil samples,
* **Geophysical** – gravity, mineral mapping, electrical resistivity, geophysical surveys.
* **Well / Drilling data** – wells, well log temperature, lithology, resistivity, radio activity, gamma ray, who drilled, when, how deep, results, current status, logs, samples, related documents, publications
* **Land status** – who owns the land, zoning, mineral and water rights, permitting, near transmission lines proximity, water availability

# User Communities and Roles

A variety of user groups can potentially interact and use the NGDS system. In addition to the three main target user groups (defined in the System Vision document, see /P01/), we also include Administrators who are responsible for maintenance of the searchable catalog and entry-point web-application, as well coordinating management of system-wide standards and protocols. Various user groups or roles and their relationships are illustrated in Figure 2.

In Figure 2 we also outline the three main target user communities – Data Provider, Software Developer, and End User. These communities are discussed in general terms, with more detailed descriptions of their respective use cases outlined in the next section.



Figure Main user roles and their relationships

The National Geothermal Data System will be a network consisting of four linked communities:

* **Data providers** who will expose information to the system through standardized, internet-accessible interfaces and interchange formats
* **Data consumers** who will utilize the software and information provided by the system in order to understand and develop geothermal resources.
* **Administrators** who are responsible for administrating and monitoring the system. Typical tasks are installation and configuration, user management, node monitoring, or system backup.
* **Software developers** who will build applications that utilize the data in the system, and make it easier for end-users to interact with the system.

## Data Provider Community

Data providers represent the collection of users who will work together to publicize information to NGDS through standardized, internet-accessible interfaces using one of the supported interchange formats. The publication consists on making data available through a NGDS compatible repository, possibly the NGDS repository from the node-in-the-box distribution, and the publication of the metadata in the NGDS catalog, either via a WebApp component, or programmatically, using NGDS protocols.

The data provider community will generally consist of groups of individuals, often representing a single organization, who work together to maintain a repository of geothermal data that is accessible to NGDS users. This community of users can be further broken down into three distinct roles.

* **Data Submitter**: the user who uses the NGDS protocols and services to publish a piece of data
* **Data Steward**: the user who maintains the quality of a piece of published data
* **Data Originator:** the person that created a piece of data, e.g. a publication. The data originator is not an active actor in any of the use cases involved in the NGDS: as soon as the originator wishes to contribute to NGDS, they take on the role of Data Submitter
* **Node-in-a-Box (or simply Node) Administrator**: the user who is responsible for operating and maintaining an organization’s data repository and insures that the data is provided according to NGDS standards and protocols

In fact, data providers play a fundamental role in the success of the NGDS because only when a critical mass of information has been published into the system will it become a useful tool for end-users / data consumers.

### Data Submitter Role

Data submitters will publish geothermal related data to NGDS through one of several methods described below.

* Providing metadata to individual resource hosted in an existing repository, e.g. a URL to a file in a website, and registering it to the NGDS catalog service (through one of the federated NGDS catalog components).
* Performing a bulk registration of a collection of resources that are hosted in a Web repository. This is performed by uploading metadata in a template table in which each record describes a single resource (e.g. a URL to a file or spreadsheet) to the NGDS catalog service (again, using one of the federated nodes of the catalog).
* Registering a harvesting endpoint with the NGDS catalog that supports system protocols (CSW, WFS) for sharing metadata according to **NGDS standards**
* Uploading and processing data files in **supported formats** to a NGDS repository component, in order to deploy data services according to system protocols and interchange formats supported by NGDS, as well as constructing standard metadata for the new data and publishing it to the NGDS catalog service.

The data submitter user role will also be able to:

* Validate input files to verify that they conform to supported formats for automated service deployment
* View and browse services deployed from data files
* Validate metadata records
* Verify that access information in metadata records successfully locate and retrieve resources
* View logs created during data publication, processing and maintenance
* Replace existing metadata records or datasets with newer versions.

### Data Steward Role

The data steward user role will verify and maintain the quality of published data. The data steward will have write access to data under his or her responsibility.

The data steward will perform quality assurance tasks, such as:

* Viewing quality assurance reports
* Flagging a piece of data according to its quality, e.g. giving it a 1-5 star rating
* Performing error correction and data updates

### Node Administrator Role

The node administrator is responsible for the administration of one of the data repositories, or nodes, in the system. This user will perform maintenance tasks including:

* Add and delete users
* Assign user roles (data submitter, data steward)
* Administer logs
* Monitor system usage
* Backing up resources

If the node is implemented in a proprietary fashion, i.e. the node-in-a-box components are not utilized to host the data, this user is also responsible for insuring that the data in their repository or node is available to the NGDS in supported data formats and through appropriate system-wide protocols. If, otherwise, the NGDS node-in-a-box application is utilized, this will not be a concern since the repository, catalog and web app will comply with NGDS standards.

## Data Consumer Community

Data consumer will utilize the software and information provided by the system in order to understand and develop geothermal resources.

Specifically, Data Consumers will:

* Search for the entire NGDS system utilizing the federated NGDS catalog service either via the NGDS WebApp application, or potentially any other catalog search application that operates with NGDS search protocols and metadata interchange formats.
* Visualize and explore data in map, text, or other graphic presentations (as prioritized by user research findings).
* Select and acquire data via service interfaces using NGDS interchange formats
* Access files in NGDS data repositories
* Save and re-load search queries
* Set filters to be notified as new data of interest is available via the system

In general Data Consumers can access the NGDS data without authentication (i.e. anonymously). Any metadata in the catalog is freely available. Authenticated Data Consumers, however, gain the right to provide ratings for the datasets they are accessing, saving and re-loading searches, and posting subscriptions to new data. They, however, do not have the right to modify the data itself. Only Data Stewards and Data Providers can modify data they are directly responsible for.

## System Administrators

The system administrator is responsible for the administration of system components. Node-in-a-box administrators are responsible for their own node (repository, WebApp and catalog), while the NGDS administrator is responsible for the whole NGDS set of nodes, and the federated catalog service. Administrator can perform different tasks including monitoring and registering nodes in the system as well as maintenance of the NGDS catalog and entry-point web-application. This role includes tasks such as:

* Supervise the addition and deletion of nodes to NGDS system
* Monitor usage and system performance
* Respond to inquiries from end-users, active and potential data-providers
* Manage user accounts related to the entry-point web-application
* Security audits, system backup, setting policies, etc.

The system administrator is also responsible for coordinating review and adoption of system protocols and interchange formats and for the registration of new interchange formats and associated schema.

## Software Developer Community

Software developers may build applications that access NGDS resources using the protocols and standards outlined as part of the system architecture. These applications may serve a variety of purposes, from assisting in search and discovery of data available in the system, to visualization of datasets, through more complex analysis of the data. By utilizing existing, community-driven web-service protocols, the NGDS sets itself up to function using a number of already existing pieces of software, such as Data.gov, Geothermal Prospector, ArcGIS, OpenLayers and many others.

# Use Case Models

This section provides an overview of use case models representing interaction with the system by various user groups as defined above. These use case models are used to identify system requirements related to user interaction. Each of the use case models is represented by a use case table, which in turn delineates concrete functional requirements of the system. Use Case tables are identified by blue headers and UC\_XXX identifiers, while functional requirements are listed as bullet points in the functions row of the table. Whenever applicable, we also present derived non-functional requirements (or quality attributes) that are labeled with SRSXXX identifiers.

The use cases will later be linked to system components, explicitly verifying that the system design satisfies all requirements. See the **Software Design Description (SDD)** [**/P04/**](#P04_GTDA_Software_Design_Description), for system design details and the linkage between components and requirements. The following sections describe use-case views of the System with a brief description of each use case.

## High-level Use Cases View

Figure 3 shows a view of the NGDS system from the users’ perspective, illustrating different types of users that may interact with the NGDS system by performing different tasks. For instance, end-user/data consumers are using NGDS to gather data, validate data, and analyze the data after exporting it. Data submitters are as important as data consumers. The two major use cases for Data submitter are Batch Import of Dataset Files, and Create Metadata Record through a Form.



Figure 3: High-level Use Case View

## Data Provider Use Cases

Data provider community’s use cases are inherently correlated to the data repository in use by that subsection of the community. As such, these use cases and requirements are scoped to individual nodes or data repositories that make up the NGDS. The focus of these use cases is to create and publish a “resource”, which is defined here as a single, coherent dataset which may exist in a number of different representations (e.g. uploaded .xls or .csv files, as well as scanned images, web-services or other content already online and hosted elsewhere). Each resource has a corresponding metadata record that describes the work as a whole, and clearly indicates the access mechanisms available to view that work in any of its available representations.

Use Cases here are primarily broken down into categories according the user role responsible for performing them. However, data submitters, stewards and administrators will have to first log into their node to perform data publishing and administration tasks.

1. **Tier1: Unstructured data** - Manually creating database records through a forms interface, uploading files to a local file repository if they are not already online.
2. **Tier2: structured, but not standardized data** - Bulk loading metadata to the catalog from a metadata input CSV text file for resources that are already accessible online.
3. **Tier3: Structured standardized** data supported natively by NGDS.

In cases 1 and 3, if NGDS node-in-a-box repository is used to host the data resource, a metadata record will need to be created using a form interface in the NGDS WebApp UI. In Case 3, the NGDS WebApp UI quality control application will parse uploaded files, according to NGDS supported format, to determine if they conform to a known Content Model Template. After that, the data will be imported into the NGDS Data repository, having its metadata publicized to the NGDS catalog.

Alternatively, data providers can utilize command-line scripts and programs to automatically import data into the system. In those situations, no UI is provided, but users will still have to publicize the metadata to the NGDS catalog service utilizing NGDS protocols.

In Cases 1 and 3, uploaded files will be stored in a file repository (either the ones provided by NGDS nodes-in-boxes, or in a third-party repository supporting NGDS protocols, as previously discussed). The files are public, however, they can only be modified by their data stewards, submitters and administrators. Creation of metadata will be assisted by node-in-a-box components that parse and extract category and attribute information from uploaded files when possible (based on file type and content). Third-party repositories that utilize alternative software are also required to provide standard NGDS metadata including such category and attribute information.

For Tier 3 access, data will be mapped into prioritized content models determined by AZGS (Arizona State University) and encoded in the adopted interchange formats for publication via NGDS services. Tier 2 data is incompatible with NGDS data model and cannot be automatically indexed by NGDS catalog, however, they can be handled as Tier1 data. Tier 1 data (an image or PDF file for example) will be referenced by a URI pointing to its data location, will have a geo-location tag, and will be indexed based on metadata (manually) provided by the publisher. **See content model documentation /P02/**, for details of these categories. The standard content models will be used to structure a browse tree interface and data browsing capabilities with filtering and display functions. These will be posted in the NGDS Website.

Once the catalog entries have been made, it will be possible to modify them manually, by adding data items as necessary. This will enable the update and addition of metadata to the catalog entries, and will support third party data repositories to publish metadata about their site.

Additionally, end users can annotate metadata records with comments, and can provide star ratings (from 1 to 5). These ratings and comments will potentially help other users in the classification discovery, filtering and gathering of data.

Only Data Submitters, Stewards and System Administrators will have Read/Write access to the metadata posted in the catalog. All Data Consumers (Geothermal Analysts/End User) will have Read only access to the data. The NGDS repository will keep a record of all import operations performed. This will be provided as an import log when requested by Administrators, Submitters and Stewards.



Figure 4 : Data Entry and Submission Use Case View

Figure 4 above illustrates the functionality of the data provider node application that facilitates the submission of data to the NGDS. Once a piece of data is imported into the repository by the Data Submitter, it is handed over to the Data Steward who reviews it. If the Data Steward is satisfied with the content and quality of a dataset and its metadata, he or she can publish the data, making it available throughout the NGDS network by publishing its metadata to the NGDS catalog.

If the published data conforms to NGDS content model, the NGDS node-in-a-box provides functionality to upload a template file and deploy NGDS services based on the file.

Note that, from the point of view of End Users / Data Consumers, all data and metadata publicized through NGDS catalog & NGDS data repository will be freely available, with read only access. Data submitters, Administrators and Data Stewards will have write access to the data and metadata they are responsible for.

Here follows a task-based description of each of the use-cases.

As one of NGDS’s main purpose is to support the location of publicly available data, End Users/Data Consumers may freely access public catalog information and data. In these situations, they will be automatically assigned ‘guest’ access permissions. Data Submitters, data stewards and Administrators, however, will have to first log into the system to perform data publishing and administration tasks. It is assumed that Login and Logout use cases are included in each of the functional areas that follow.

In general, the use cases for NGDS system are organized by user category as follows.

* Data Provider Use Cases
* End-user/Data Consumer Use cases
* System administration Use Cases
* Software Developer Use Cases

When certain tasks or workflows require multiple users to collaborate, they will be represented as workflow use cases, such as Quality Assurance Workflow.

### Data Submitter

Data submitters publish geothermal data at their local node repository. Their primary objectives are to make files available online, set up data-services where appropriate, and to generate metadata describing a resource.

#### File Uploads

The first step in making a data resource available in the NGDS network is to store it in a web repository. This process is captured by the file upload use case. The data submitter should also be able to upload multiple files in one operation, if those multiple files pertain to a single resource. Generally such files are not expected to exceed 2GB in size.

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_001a** |
| Use Case Name | | **Upload new files** |
| **Short Description** | | The goal of this use case is to allow data submitters to upload one or more files to be stored in the NGDS data repository. After the upload, the submitter will also update the metadata record of that file, thus allowing it to be cataloged.  We assume the file is opaque, i.e. stored as is, with no further content parsing. |
| **Actors** | | Data submitter |
| Pre-Conditions | | Data submitter is properly authenticated; |
| Success End Conditions | | The files are successfully uploaded and stored in the NGDS repository  The metadata record for the provided file is successfully created  The metadata remains “private”, waiting to be made public by a data steward |
| Data | | Metadata attributes for the specific data type as input to the form  Files to be uploaded  Geographical location of files |
| Functions | | * Upload files * Form-based metadata input for specific data type * Auto-complete of user contact information * metadata validation * URI creation * Metadata duplicate detection * Tagging of metadata with geographical information * Converting non-standard location coordinates into latitude/longitude and shapes. * Log changes to metadata log file |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Data Submitter | Navigates to the files upload screen from NGDS System |
| 2 | NGDS System | Presents upload files screen to Data Submitter |
| 3 | Data Submitter | Selects the list of files to be uploaded  Selects the metadata record type to be created for the files  Inputs geographical location |
| 4 | NGDS System | Presents the appropriate metadata import form to the metadata type the data submitter selected.  Presents a form for selecting and uploading files |
| 5 | NGDS System | Uploads selected files  Associates uploaded files to the data submitter account |
| 6 | Data Submitter | Fills out the form with metadata and finalized data input |
| 7a | NGDS System | Performs validation of metadata form based on content completeness  Creates a data location URI and updates metadata form  Performs duplicate detection  Validates, normalizes, converts geo-location attached to the data  Log changes to metadata log file  Provides a file upload success message. |
| 8 | Data Submitter | Acknowledges the success/failure completion of the operation |
| 9 | NGDS System | Makes metadata “private” for discovery, waiting to be made public by a Data Steward  Send data steward an e-mail notification about the new data  or  If the user is both a Data Submitter and a Data Steward, makes the data “public”, notifying the user of the successful operation. |
| Variants | | |
| Step | Actor | Description |
| 7b | NGDS System | In case of duplicates, or incomplete information in the form, the system will provide a failure notification message, indicating the type of error.  The user-provided metadata form will be presented to the Data Submitter for correction |
| 7c | Data Submitter | Will correct the form data and resubmit for validation or will quit the import procedure. |
| Exceptions | | |
| Step | Actor | Description |
|  | NGDS System | In case of internal file upload/metadata record creation failure, the system will roll back all existing transactions, returning to its previous state. |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | Can't it be assumed that the submitter is working in cahoots with the data steward? Can the approval step be short circuited by assigning both submitter and steward roles to an individual? Yes, it is possible. | |
| 2 | IN step 9, perhaps the data steward should be notified that there is a new record to be approved? This could be done by the NGDS system once the data submitter confirms the upload is completed? | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_001b** |
| Use Case Name | | **Update uploaded file** |
| **Short Description** | | The goal of this use case is to allow data submitters to update a file that has been uploaded. This process includes the update of the metadata record of the file.  We assume the file is opaque, i.e. stored as is, with no further content parsing. |
| **Actors** | | Data submitter |
| Pre-Conditions | | Data submitter is properly authenticated; |
| Success End Conditions | | The file is successfully updated and new content stored in the NGDS repository  The metadata record for the provided file is successfully updated  The entry for the new file is also propagated & updated in the NGDS catalog. |
| Data | | Possible new metadata record changes  Files to be uploaded |
| Functions | | * Upload files * Auto-complete of user contact information * metadata validation * Metadata duplicate detection * Tagging of metadata with geographical information * Converting non-standard location coordinates into latitude/longitude and shapes. * Log changes to metadata log file |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Data Submitter | Searches for a file in the database includes <<gather data>> use cases  Selects an option to update file record |
| 2 | NGDS System | Presents a update file metadata record screen |
| 3 | Data Submitter | Selects new file to be uploaded  Updates metadata record for the file  Selects update command |
| 4a | NGDS System | Updates file and/or updates metadata record  Performs validation of metadata form based on content completeness  Updates a data location URI and updates metadata form  Performs duplicate detection  Provides a file upload success message.  Validates, normalizes, converts geo-location attached to the data  Log changes to metadata log file |
| 5 | Data Submitter | Acknowledges the success/failure completion of the operation |
| 6 | NGDS System | Makes metadata “private” for discovery, waiting to be made public by a Data Steward  Send data steward an e-mail notification about the new data  or  If the user is both a Data Submitter and a Data Steward, makes the data “public”, notifying the user of the successful operation. |
| Variants | | |
| Step | Actor | Description |
| 4b | NGDS System | In case of duplicates, or incomplete information in the form, the system will provide a failure notification message, indicating the type of error.  The user-provided metadata form will be presented to the Data Submitter for correction |
| 4c | Data Submitter | Will correct the form data and resubmit for validation or will quit the import procedure. |
| Exceptions | | |
| Step | Actor | Description |
|  | NGDS System | In case of internal file upload/metadata record update failure, the system will roll back all existing transactions, returning to a valid state. |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | What happens to the old file? Is it deleted in the database or DOEs it remain there? Is a history of all metadata changes kept?  DN: This is a good question and one that probably has no right answer for everyone. Does the new item get a new UUID or is it a revision (version X++) of the existing item? Earlier in the document there is a statement that says “all” node interactions are logged. This might require that the old version stay there. Probably best to ask domain experts. | |

#### Data Processing

In some cases, the file to be uploaded may conform to an NGDS content model and be provided in a valid file-format, allowing the information from the file to be processed in order to create a compliant NGDS web-service. Processing will include validation of the data’s schema, loading of data from the uploaded file into the repository. The file will then be available for publication by the data steward.

Publication consists in making the metadata of the file available in the NGDS catalog and having its contents publicized via supported protocols (i.e. WMS and WFS).

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_006** |
| Use Case Name | | **Process data file in NGDS content model template** |
| **Short Description** | | This allows data submitter to process (upload/parse) a file to NGDS data provider nodes.  The difference between the upload/update file use cases is that the file here is formatted according to an existing template, and therefore, can be parsed and checked for correctness. Processing will include validation of data schema, loading data into a data store on the provider node. |
| **Actors** | | Data Submitter |
| Pre-Conditions | | Requires authentication, access permission to edit metadata records.  File must be formatted according to one of the NGDS content model templates (See /P02/ data specification from Arizona State University, for supported file formats and content models |
| Success End Conditions | | File is submitted to a repository, being accessible through a valid URI |
| Data | | Files properly formatted according to NGDS supported formats |
| Functions | | * import data files * Validate data file content & formats * Log changes to metadata log file |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Data Submitter | Navigates to the page that allows upload of files to NGDS data repository |
| 2 | Data Submitter | Provides file path & name to the system  Provides a data type from the NGDS supported content models and file formats  Send file for upload |
| 3 | NGDS System | Validates file data format based on its provided type  Informs the user about possible errors in the format  Perform file content validation  Performs duplicate detection  Informs the user about errors and fails, without importing the file;  Or provides a success message  Makes metadata “private” for discovery, waiting to be made public by a Data Steward  Send data steward an e-mail notification about the new data  Or  If the user is both a Data Submitter and a Data Steward, makes the data “public”, notifying the user of the successful operation. |
| 4 | Data Submitter | Verifies the operation status. If a file has errors, the user will have to correct them without the help of the system. |
| Variants | | |
| Step | Actor | Description |
| 3b | Data Submitter | Is notified of suboptimal data content  Chooses to submit the data anyways |
| 3c | NGDS system | Accepts the file, flags the problems with content or metadata  Hand file over to the data steward |
| Exceptions | | |
| Step | Actor | Description |
|  | NGDS System | In case of internal import failure, the system will roll back the existing transaction, returning to a valid previous state. |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 |  | |
|  |  | |

#### Metadata Generation

The creation of metadata follows three distinct use cases. First is to generate a metadata for a single resource through a form-based interface. The purpose of such an interface is to make it as easy as possible for the data submitter to create high-quality, NGDS standards-compliant metadata describing a single resource.

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_003** |
| Use Case Name | | **Create metadata record through a form** |
| **Short Description** | | The goal of this use case is to allow data submitters to create a metadata record describing a resource (tier1, tier2 data) by input of information manually through a form interface for inclusion in the NGDS catalog. |
| **Actors** | | Data submitter |
| Pre-Conditions | | Data submitter is properly authenticated;  Data is available through one of the NGDS data repositories, and is identified through a valid URI.  Metadata includes the geological location of the geological feature associated to the data. |
| Success End Conditions | | The meta-data for the provided geological feature is successfully imported into the NGDS catalog  The data remains “private”, waiting to be made public by a data steward |
| Data | | Metadata attributes for the specific data type as input to the form |
| Functions | | * Form-based metadata input for specific data type * metadata validation * URI validation * Metadata duplicate detection * Tagging of metadata with geographical information * Converting non-standard location coordinates into latitude/longitude and shapes. * Log changes to metadata log file |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Data Submitter | Navigates to the metadata input screen from NGDS System |
| 2 | NGDS System | Presents metadata input screen to Data Submitter |
| 3 | Data Submitter | Selects the meta-data type to be imported |
| 4 | NGDS System | Presents the appropriate metadata import form to the metadata type the data submitter selected. |
| 5 | NGDS System | Automatically fills out form with data submitter contact information |
| 6 | Data Submitter | Fills out the form with metadata and finalized data input |
| 7a | NGDS System | Performs validation of metadata form based on content completeness  Validates data location URI  Performs duplicate detection  Provides an import success message.  Validates, normalizes, converts geo-location attached to the data  Log changes to metadata log file |
| 8 | Data Submitter | Acknowledges the success/failure completion of the operation |
| 9 | NGDS System | Makes metadata “private” for discovery, waiting to be made public by a Data Steward  Send data steward an e-mail notification about the new data  or  If the user is both a Data Submitter and a Data Steward, makes the data “public”, notifying the user of the successful operation. |
| Variants | | |
| Step | Actor | Description |
| 7b | NGDS System | In case of duplicates, invalid URIs or incomplete information in the form, the system will provide a failure notification message, indicating the type of error.  The original metadata form will be presented to the Data Submitter for correction |
| 7c | Data Submitter | Will correct the form data and resubmit for validation or will quit the import procedure. |
| Exceptions | | |
| Step | Actor | Description |
|  | NGDS System | In case of internal import failure, the system will roll back all existing transactions, returning to a valid state. |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | Can you define content completeness? What’s the criteria to be used here? Are there required fields and optional fields for example?  ND: Believe this will be set forth in the metadata cardinality rules (example 0..1) | |
| 2 | In case of duplicates, which one is the most important the new instance of the data or the existing version of it? How to resolve these conflicts?  DN: The domain experts said if records are overlapping, they would want both. If they are truly identical, it doesn’t really matter does it (from a pure logic perspective anyways). | |
| 3 | What if the user DOEs not provide a geo-location to the data?  DN: Given the UCD feedback, a geospatial reference is mandatory for every record for map based searching. | |
| 4 | The existence of a make public use case performed by a Data Stewart implies that after an import, the data is made private, waiting for being publicized by the steward. Is it a correct assumption?  DN: I believe so. This is reflected elsewhere in the document. | |

Where a data submitter wishes to create metadata for a larger collection of resources for which files are already available online, such metadata information can be uploaded to the node through the use of a metadata template table. In such a table, each row represents a single resource, and the columns are translated into appropriate parts of a standard NGDS metadata record.

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_004a** |
| Use Case Name | | **Bulk upload metadata from metadata content template table** |
| **Short Description** | | The goal of this use case is to allow data submitters to import a CSV file containing metadata into the NGDS catalog.  New records will be marked ‘submitted’; Data Steward will then have option to review metadata through forms interface; records will become visible to public when marked ‘published’. Metadata will be validated for content completeness, URLs checked for http 200 responses, and new metadata record will be run through a duplicate-detection process to identify existing metadata that may already describe the resource. |
| **Actors** | | Data Submitter |
| Pre-Conditions | | Data submitter is properly authenticated;  Individual data items, referenced to in the CSV file are available through one of the NGDS data repositories, and is identified through a URI.  Metadata includes the geological location of the feature associated to the data.  CSV file containing metadata in table form according to the NGDS Compilation template Metadata Excel workbook |
| Success End Conditions | | The meta-data for the provided geological feature is successfully imported into the NGDS catalog.  The imported data remains private, waiting for the Data Steward to make it public. |
| Data | | CSV file containing resources metadata, formatted according to the NGDS compilation template metadata excel workbook. |
| Functions | | * Form-based metadata input for specific data type * metadata validation * URI validation * Metadata duplicate detection * Tagging of metadata with geographical information * Converting non-standard location coordinates into latitude/longitude and shapes. * Log changes to metadata change log file |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Data Submitter | Navigates to the bulk metadata input screen from NGDS System |
| 2 | NGDS System | Presents bulk metadata input screen to Data Submitter |
| 3 | Data Submitter | Selects the metadata CSV file type to be bulk imported |
| 5 | NGDS System | Automatically fills out form with data submitter contact information |
| 4 | Data Submitter | Inputs CSV file name and path into the metadata input screen |
| 6 | Data Submitter | Starts import process |
| 7 | NGDS System | Validate metadata for content completeness  Validates URI  Detects metadata duplication  Marks problematic entries for revision  Marks individual entries as “submitted”  Normalizes geo-location coordinates to the internal system representation (UML or lat/long)  Log changes to metadata change log file |
| 8 | NGDS System | Presents import data report to user in a form |
| 9 | Data Steward | Reviews “submitted” data for errors. The forms allow users to modify/correct the individual metadata records |
| 10 | Data Steward | After reviewing the data and correcting errors, send metadata for publication |
| 11 | NGDS System | Repeats step 7 for a final check |
| 12 | NGDS System | Makes metadata “private” for discovery, waiting to be made public by a Data Steward  Send data steward an e-mail notification about the new data  OR  If the user is both a Data Submitter and a Data Steward, makes the data “public”, notifying the user of the successful operation. |
| Variants | | |
| Step | Actor | Description |
| 11b | NGDS System | Rejects data based on invalid records |
| Exceptions | | |
| Step | Actor | Description |
|  | NGDS System | In case of internal import failure, the system will roll back the transaction, returning to a previous valid state. |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | See: “**Create metadata record through a form”** use case. The same issues are present here | |
| 2 | Only CSV files are supported? Prescriptive? | |

Finally, a data submitter may wish to update metadata records for existing resources, perhaps one at a time, or in bulk. In either case, the data submitter must first be able to locate the resources that s/he wishes to see updated.

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_008** |
| Use Case Name | | **Browse and manage resource directory** |
| **Short Description** | | Allows the User to view & manage all of the resources (or datasets) under their stewardship (data steward) or that they have submitted (Data submitter), based on metadata describing the resources. The resource listing will be presented to the User in a tree view directory structure.  Users can define collections (folders, subdirectories) to organize the listing according their needs. A resource may be assigned to multiple collections. Access control may be assigned at the collection level. The display should indicate clearly any resources that have quality issue flags attached |
| **Actors** | | Data submitter, Data Steward |
| Pre-Conditions | | Requires authentication, access permission to view & edit individual user metadata records & datasets. |
| Success End Conditions | | Users can adequately manage (create, rename, delete) resources  Users can assign resources to collections (updating metadata)  Users can delete existing resources (and their associated metadata)  Metadata is kept in synchrony with changes in the collection |
| Data | | Resources: metadata records or files |
| Functions | | * Visualize flagged resources with quality issues * Create, delete, reorganize resource collections * delete resources * Visualize resources & collections |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Users | Navigate to the resources management page |
| 2 | NGDS System | Provides a tree view of existing resources and their collection structure  Allow users to navigate through the structure  Allow users to perform management operations (move, delete, add, rename) |
| 3 | Users | Perform collection management operations (add, delete, rename, move)  Perform resource operations (delete, move, rename) |
| 4 | NGDS System | Responds to user operations, providing an up-to-date view of the resources available for the user.  DOEs not allow the users to see resources that are not under their responsibility.  Apply selected management operations on the resources |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | It seems to me that, in this view, the only operation that users can do with resources is delete. (Importing and modification of resources must be done in other views). Is it correct? | |
| 2 | Missing requirements: tree-view portrayal of resource hierarchy. More clarity required as to who can create collections and sub-collections at what levels in the hierarchy. [Data steward assigns permissions to create public collections; perhaps allow authenticate users to define personal collections that they persist in a user workspace to use between sessions]. How do permissions work if metadata can belong to more than one collection? [Metadata record has only one owner; do we need a use case for reassigning ownership of a metadata record?] If a record is part of one collection and I have edit privileges to that collection, but it is also included in another collection to which I do not have privilege, what are my options? [Looks like assigning access control at collection level won’t work; work access control through users and group membership, with permissions at group level to share edit capabilities] | |
| 3 | DN: I found this to be very prescriptive (are all resources always hierarchic?) Is it possible they have nodes that are traversed in other ways? | |

Once the data submitter has located the record or records that need to be updated, they are either adjusted using the form-based interface described above for single resources, or through a bulk updating workflow for the case of multiple resources.

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_005** |
| Use Case Name | | **Bulk update metadata records** |
| **Short Description** | | Metadata records may be selected based on a content-based filter query based on fields as contact information or linkage URIs. Once selected, those records can be replaced using regular expressions. User selects metadata content item to update, value to replace, and new value to use. |
| **Actors** | | Data submitter, Data steward, System administrator |
| Pre-Conditions | | Requires authentication, access permission to edit metadata records. |
| Success End Conditions | | All relevant metadata record attributes that match the search criteria are replaced with the new value defined by the user |
| Data | | Specific fields of metadata records: contact information and URIs |
| Functions | | * Search & replace of metadata record attributes based on simple regular expressions * Log changes to metadata log file |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Users | Navigates to the search & replace metadata record feature page |
| 2 | Users | Input search criteria using simple regular expression  Input replace data  Select metadata record attributes to be replaced |
| 3 | NGDS System | Performs search & replace routine using user input parameters on the specified metadata record attributes  Presents a report of affected records and their suggested changes  Asks users for confirmation  Logs changes in metadata change log file |
| 4 | Users | Accept or reject bulk change based on report produced by the system |
| 5 | NGDS System | Commit metadata record changes;  or roll back if users reject changes |
| 6 |  |  |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  | NGDS System | In case of internal import failure, the system will roll back all existing transactions, returning to a valid state. |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | How interactive should be the process of search & replace? | |
|  |  | |

Once the data submitter has completed the process of creating a resource, that resource will be marked as submitted. As a submitted resource, access to the metadata, data and files associated with the resources is restricted to users with permissions appropriate to edit, and the metadata is not included in any public harvesting endpoints (i.e. CSW interface).

### Data Steward

It is the job of the data steward to verify and maintain the quality of published data. Users in this role will perform quality assurance tasks, as well as maintain appropriate permissions over a subset of resources available on a node. These permissions are likely to be applied to groups of users on distinct collections of resources.

#### Activity Logs

In order to assist the data steward in the analysis of resource quality, activity logs will be generated during any data or metadata creation, update or QA procedure. These logs will include information such as:

* time of activity
* actions taken
* user responsible
* user comments

These activity logs will be accessible to data submitters and data stewards with appropriate permissions for a particular resource.

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_007 & UC\_046** |
| Use Case Name | | **View resource submission and update logs** |
| **Short Description** | | Allows authorized users to view the logs for metadata record creation, file uploads to file repository, and processing of NGDS files that conform to valid content model templates.  These logs are created during any data submission or update activity. |
| **Actors** | | Data submitter, Data Steward, System administrator |
| Pre-Conditions | | Requires authentication, access permission view metadata records import logs.  These logs are created during any data submission or update activity, so at least one data submission must have occurred. |
| Success End Conditions | | Successful and correct visualization of data submission logs.  No missing logs |
| Data | | Data submission logs containing details such as time of activity, actions taken, data submitter, Data Steward, size of data, submitter comments, etc |
| Functions | | * Submission log capture * Submission log visualization |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Users | Navigate to the data submission log page  Select among existing logs by date or by name of data upload operation or metadata creation operation |
| 2 | NGDS System | Searches for the log given user search criteria  Succeeds presenting the log  Or fails, providing a error message if no data record is found |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | Missing requirements detailing the logs and their creation. What exactly is logged? Also, how is the log portrayed to the user, or is that too much in the realm of design? | |
|  |  | |

#### Validation and Quality Assurance Logs

When data or metadata are processed in NGDS content model templates, these logs will include a report on validation of the processed information and adherence to the content model (syntax, completeness, cardinality, data types, URL checking and de-duplication). A validation report will be created which will list any recognized quality issues.

The data steward will review these quality assurance reports, and perform appropriate actions based on their assessment of the report.

Figure 5 illustrates the functionality supported by the data provider node application to facilitate quality assurance workflow. Functionality includes automated quality assurance and reporting functionality where that is practical. When data or metadata are processed in NGDS content model templates, the application will validate adherence to the content model (syntax, cardinality, data types, other tests that may be automated), and will generate a report listing recognized quality issues. Part of the validation process will be a function to compare the new metadata to existing metadata in the local catalog to identify existing records that may describe the same resource. Data Submitter will need to manually inspect the flagged records in the submitted data or metadata to correct issues. They can modify the record, or indicate that the record is correct and add a note explaining their reasoning.

In addition, authorized End Users can flag a data or metadata record for quality issues if they consider that the content is not accurate. The System will handle these flags (annotations) by notifying the appropriate Data Submitter to take action to correct the issue. The data submitter than sends it back to the steward for approval and publication.



Figure Quality Assurance Use Case View

Here follows a brief description of each of the use-cases.

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_046** |
| Use Case Name | | **View catalog audit log** |
| **Short Description** | | At every change the system creates a log for a given catalog item. This use case allows authorized users to view the log of all changes made to a data or metadata record, changes made both manually and automatically. |
| **Actors** | | Administrator, Data Steward |
| Pre-Conditions | | An audit log of catalog changes has been created and actively updated by the system |
| Success End Conditions | | Users are able to visualize the change logs for a given record |
| Data | | Catalog audit log |
| Functions | | * View catalog audit log |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Users | Navigate to quality assurance page  Select a metadata record  Visualize the log for that record |
| 2 | NGDS System | Responds by loading and displaying the data record log. |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | Missing requirements detailing the validation and generation of the validation log. Here are two, but where are ones about URL checking? Syntax? Is the validation log attached to the activity log for a particular resource? | |
| 2 | DN: I am also confused if this is happening at the node or NGDS system level? If another node change must be recorded in every other node, this could get quite messy fast (math suggests that it would be equal to ((N \* (N2 – 1) \* R) \* MR) where N is the number of nodes, R is the number of records and MR is the number of metadata records and that is assuming a 1:1 relationship between instances or R and MR. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_042** |
| Use Case Name | | **View quality assurance reports** |
| **Short Description** | | Allows the Data submitter or Data Steward to view quality assurance reports for resources they submit or maintain.  Quality assurance can be automatic or manual. As an automated process, it is performed by NGDs during import, flagging possible typos and simple input errors in the metadata being managed by the system  As a manual process, it involves users that see possible issues and flag them in the data records. |
| **Actors** | | Data Submitter, Data Steward, End User/Data Consumer |
| Pre-Conditions | | The metadata record has been imported into NGDS catalog and a set of quality assurance functions were automatically ran in those records. As a result, the metadata records were flagged for possible errors.  Users have write access to the data i.e. they are either submitters or stewards of that data |
| Success End Conditions | | Users are able to identify the flagged errors in the metadata they provide or maintain |
| Data | | Metadata records, Quality Report |
| Functions | | * Visualize quality assurance report |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Users | Navigate to quality assurance page  Select view quality assurance reports |
| 2 | NGDS System | Responds by displaying a list of metadata records that were flagged as having possible quality assurance problems |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | Should consumers be able to see QA reports? If they are not addressed, it seems like it would serve as a good "warning" to analysts.  The Quality Report is for steward and submitter consumption, with lots of details that may not be relevant to End Users, e.g. wrong measures, duplicated fields. We assume the steward will fix them before making the data public. However, some stewards may want to make that data public anyway, in that case, consumers could benefit from this information. | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_043** |
| Use Case Name | | **Flag resource quality issue** |
| **Short Description** | | Allows authorized & authenticated End-User/Data Consumer to create a flag, indicating that some issue exists with the quality of a data or metadata record. The process creates an annotation record documenting the dataset, URI for the resource in question, identity of the user raising the flag, timestamp. The annotation record should also include notes on resolution process, who, when, what. Flagged resources should be clearly marked |
| **Actors** | | Data Steward. End User/Data Consumer |
| Pre-Conditions | | The metadata record has been imported into NGDS catalog.  Metadata QA records can be modified by any user in the system |
| Success End Conditions | | Users are able visualize metadata records, and to provide quality assurance feedback to these records as they see fit. |
| Data | | Metadata records (read only)  Metadata QA (quality attribute) records (read/write)  User information |
| Functions | | * Manually flag resource quality issues |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | End User/Data Consumer | Navigate to quality assurance page  Visualize metadata records  Edit record quality attributes  The system will automatically fill in user information based on her credentials |
| 2 | NGDS System | Responds by displaying a list of metadata records that were flagged as having possible quality assurance problems  Include use case <<Notify data submitter of data or metadata problem>> |
| Variants | | |
| Step | Actor | Description |
| 1b | End User/Data Consumer | End user can flag data records directly on the metadata visualization page, without going to the quality assurance page. |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 |  | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_046c** |
| Use Case Name | | **Browse flagged data catalog entries** |
| **Short Description** | | Allows Users to navigate through the catalog of resources that have had their quality flagged for correction. A special view of the resource browse list, filtered for records that have quality flags raised from those that do not have them |
| **Actors** | | Data Steward, Data Submitter |
| Pre-Conditions | | a list of metadata resources is available, allowing filtering by quality attribute flag |
| Success End Conditions | | Users are able to identify flagged entries in the metadata catalog |
| Data | | Metadata record |
| Functions | | * Browse data catalog entries * Filter catalog entries by flagged attribute |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Users | Navigate to quality assurance page  Visualize metadata record  Filter out un-flagged metadata records |
| 2 | NGDS System | Responds by displaying filtered metadata records |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 |  | |
|  |  | |

Finally, the data steward is capable of resolving quality issues with resources which they are authorized to edit. Updates to data and metadata are performed similarly to as described for data submitters above. As described above, activity logs are generated for all of these actions, including indication of the user who made the corrections and user-provided notes indicating what was adjusted.

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_044** |
| Use Case Name | | **Perform manual error correction** |
| **Short Description** | | Allows a Data submitter or Data Steward to manually address issues present in a data or metadata record that has been flagged for having quality issues. Note that this process may require resubmission of data files.  Quality check and error editing will be built into the data submission process, but will require the steward to access the data in the appropriate environment for quality issues recognized after a submission is complete. The process will require the reviewer to make a note that is recorded as annotation on the record, along with the identity of the reviewer and a time stamp. |
| **Actors** | | Data Submitter, Data Steward, Administrator |
| Pre-Conditions | | The metadata record has been imported into NGDS catalog. |
| Success End Conditions | | Users are able visualize metadata records, and to provide quality assurance feedback to these records as they see fit. |
| Data | | Metadata records (read/write)  Metadata quality attribute records (read/write)  User information |
| Functions | | * Edit metadata record * Edit data * Log metadata record change |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Users | Include use case <<browse flagged data catalog entries>>  Edit record quality attributes  Edit metadata record  Edit data  Include use case <<clear quality flag>>  The system will automatically fill in user information based on his/her credentials |
| 2 | NGDS System | Responds by displaying a list of metadata records, and by allowing users to edit metadata and data. Data will be typically edited by a third party too, e.g. excel spreadsheet, and will have to be re-imported by the user after it is modified. |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 |  | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_045** |
| Use Case Name | | **Clear quality flag** |
| **Short Description** | | Allows authorized User to clear a quality flag on a data or metadata record. This will occur when the User believes that the flagged issue is actually valid in the context of that record. The process will require the reviewer to make a note that is recorded with the flag, along with the identity of the reviewer and a time stamp. |
| **Actors** | | Administrator, Data Steward |
| Pre-Conditions | | Metadata records having quality flag attributes are available in the system  The users have corrected the quality attribute issues, especially those that are automatically detected. |
| Success End Conditions | | Users are able to remove data quality attribute flags |
| Data | | Metadata quality attribute records (read/write) |
| Functions | | * Remove quality issue flag * Log metadata record change |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Users | Include use case << browse flagged data catalog entries>>  Select data with quality issue flag  Turn off quality issue flag |
| 2 | NGDS System | Responds by :  Automatically validates data once more to check for possible remaining quality issues  turning off the quality attribute flag in case of no quality issue found  logging operations |
| Variants | | |
| Step | Actor | Description |
| 3 | NGDS System | If some quality attribute is found, the metadata record flag is turned back on and the user is notified of the issue. |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | DN: Who has ultimate authority to determine the true data in the event of a dispute? | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_004b Catalog cleanup** |
| Use Case Name | | **Catalog cleanup** |
| **Short Description** | | The NGDS catalog will periodically and automatically verify the liveliness of the data sources referenced in its metadata catalog. That information will be used to mark the “unreachable”, or “non-public” data items in the metadata catalog.  The data steward can use that information to delete metadata records that are invalid, or to fix those records with valid URIs. |
| **Actors** | | Data Steward |
| Pre-Conditions | | Requires authentication, access permission to edit metadata records.  The catalog has metadata records for which URIs that will be checked for liveliness  There is also a predetermined threshold time period above which the metadata record will be marked as having “broken links”  There is also a period of time parameter used by the NGDS catalog to periodically sweep the catalog, looking for those links |
| Success End Conditions | | The NGDS catalog is cleared up from metadata records that have invalid URIs or those records have their URIs fixed. |
| Data | | All metadata records of the NGDS catalog |
| Functions | | * Automatic verification of broken links * Automatic verification of “private links” from third party repositories * Automatic tagging of broken metadata records * Deletion of catalog metadata records based on broken link attributes |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | NGDS System | Runs period data catalog verification routine (There should be a very reasonable timeout period on this like 30 days where the record is marked private to allow data stewards ample time to fix things if on vacation or away from their premise.)  Mark records as “unreachable” when URIs are inaccessible for a period more than the threshold  Mark records as “private” in case authentication is required.  Keeps record of unreachable links  Keeps record of private links. |
| 2 | Data Steward | Request “unreachable” metadata records report/screen |
| 3 | Data Steward | Corrects URIs making them “alive” again, possibly by contacting the data submitter to ask rather than embark on a search unless there is some other data that would aid them. |
| 4 | Data Steward | Instructs the system to remove selected “unreachable” metadata records |
| 5 | Data Steward | Select remaining “unreachable” metadata for deletion |
| 6 | NGDS System | Deletes selected “unreachable” metadata records after |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  | NGDS System | In case of internal import failure, the system will roll back all existing transactions, returning to a valid state. |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | Is there any undo capability? All deletions are final?  DN: I would defer this question to the domain experts. IN some cases, I would imagine that knowing there once was a record might be useful if it can be tracked down via other means (or if they want to talk to the data submitter directly). Good question | |
| 2 | Before any data can be deleted or removed (potentially via the duplicate detection process) the node must ensure that there are no metadata records existing for it. | |

The data steward may choose to delegate the responsibility of correcting flagged resources to the data submitter responsible for the record. In these cases, the data steward can send notifications to data submitters indicating resources flagged for quality that need to be addressed.

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_004c** |
| Use Case Name | | **Notify data submitter of data or metadata problem** |
| **Short Description** | | The goal of this use case is to allow the system to notify data submitters whenever other users other than the submitter herself, detect problems in the data or metadata. |
| **Actors** | | Data Submitter, Data Steward, End User |
| Pre-Conditions | | The metadata record has been imported into NGDS catalog. |
| Success End Conditions | | The data submitter receives e-mail notifications with quality issues detected by other users. |
| Data | | Metadata records (read/write)  Metadata QA (quality attribute) records (read only)  Data submitter information: e-mail |
| Functions | | * Send e-mail to data submitter with metadata and data issues |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Users | Inspect data or metadata record  Find issues with the data or data record, for example, typos, wrong information  Choose to create an e-mail describing the issue  Provide e-mail for further contact  Sends e-mail |
| 2 | NGDS System | Responds by automatically filling in the e-mail of the data submitter and the reviewer and producing an e-mail with the collected comments from other users. |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  | NGDS System | Bounding of e-mail may occur if the data submitter e-mail is invalid |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 |  | |
|  |  | |

#### Resource Publication

Once the data steward is confident that a resource’s quality is acceptable, the user can mark the resource metadata as “published”. This allows un-restricted, read-only access to the resource’s metadata, and includes the resource’s metadata in any public-facing harvest interface (i.e. CSW).

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_009** |
| Use Case Name | | **Make resource public** |
| **Short Description** | | This allows Data Steward to indicate that a resource is available for public discovery and access once they are satisfied with the data quality. |
| **Actors** | | Data Steward |
| Pre-Conditions | | Metadata record is properly created in the catalog, waiting to be made public by a data steward  Data is properly uploaded to a data repository, waiting to me made public. |
| Success End Conditions | | The metadata record is publicized  AND the data is made available through a public URI. |
| Data | | Metadata records in the catalog or files in the repository |
| Functions | | * Make data public though a URI * Make metadata record public |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Data Steward | Include use case <<Browse and manage resource directory>> |
| 2 | Data Steward | Select data or metadata for publication |
| 3 | NGDS system | Make data or metadata public (as a valid URI or in the catalog respectively) |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 |  | |
|  |  | |

### Node Administrator

The primary purpose of the node administrator is to perform administrative actions on the node, manage user accounts related to their node, to insure the node’s performance and that it communicates relevant information according to NGDS standards and protocols. Figure 6 illustrates the node administration use cases.



Figure Administration Use Case View

Here follows a brief description of each of the use-cases.

#### User Account Management

The node administrator must be able to manage user accounts. This includes the ability to add and remove users, assign permissions to users or groups, add and remove users from groups.

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_029a** |
| Use Case Name | | **Administer users** |
| **Short Description** | | Allows the system administrator to manage data provider node users. This will allow the system administrator to add and remove users on the administered node, and assign user roles and group membership. |
| **Actors** | | Node-in-box administrator |
| Pre-Conditions | | Node-in-the box is properly installed and configured |
| Success End Conditions | | The administrator is able to perform the main administration operations |
| Data | | User records |
| Functions | | * Add user * Delete user * Modify user permissions and roles |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | User | Include use cases <<add user>>, <<administer user roles>>, <<delete user>> |
| 2 | NGDS System | Responds to administration operations, enforcing role and users policies. |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | DN: Within this requirement, it will be necessary to perform a check to ensure that a user being deleted does not have custody of any records otherwise we may end up with orphaned records (unless they default to the node administrator). Just a though that came to mind. | |
| 2 | DN: A user should only be deleted once all their records (data and metadata) is transferred to another user. Otherwise there will be orphaned data and metadata. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_029b** |
| Use Case Name | | **Enrolment** |
| **Short Description** | | Allow users to self enroll. i.e. to create their account in the NGDS system for the purpose of supporting NGDS data import/export and exploration activities. |
| **Actors** | | End User/Data Consumer, Data Submitter |
| Pre-Conditions | | Node-in-the box is properly installed and configured |
| Success End Conditions | | The users have their accounts created, and their user data and credentials accessible throughout the system |
| Data | | User records |
| Functions | | * Enroll user * Un-enroll user |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | User | Navigates to the enrollment screen of the system  Inputs enrolment data including: password, login, e-mail, and some preferences  Includes <<add user use case>> |
| 2 | NGDS System | Responds by:  Validating enrollment data, for example, checking for repeated logins  Creating a user account |
| Variants | | |
| Step | Actor | Description |
| 1b | User | Navigates to the enrollment screen of the system  Requests deletion of her profile -> un-enrolment |
| 2b | NGDS System | Responds by:  Deleting the user record and its credentials  Includes <<delete user>> use case |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | Do we really want to allow users to self enroll as data providers?  DN: My answer would be yes since the system has been designed with the data steward as a failsafe against publishing bad data. The less human involvement the better. | |
| 2 | DN: Same caveat here WRT orphaned data and metadata. Do not allow a user to remove themselves if they have data submitted until the ownership of the data is re-assigned or the data is removed. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_030** |
| Use Case Name | | **Add users** |
| **Short Description** | | Creates a new user account in the system, allowing the storage of important user information such as e-mail, name, login, password, address, and enabling features such as subscription and search saving, and auto complete. |
| **Actors** | | Node-in-box administrator, Users |
| Pre-Conditions | | Node-in-the box is properly installed and configured |
| Success End Conditions | | A new user account is created |
| Data | | User records |
| Functions | | * Add user * Add users via invitation |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Administrator | Navigates to user management screen  Starts new user creation  Inputs user information |
| 2 | NGDS System | Responds by checking for repeated user credentials and if positive, creating a new account for a user |
| Variants | | |
| Step | Actor | Description |
| 1b | Administrator | Navigates to user management screen  Starts new user creation  Inputs user information  Send invitation to user |
| 2b | NGDS System | Responds by checking for repeated user credentials and if positive, send invitation to user |
| 3 | User | Responds by accepting or rejecting invitation |
| 4 | NGDS System | Creates user account or does nothing if the invitation was rejected |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 |  | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_031** |
| Use Case Name | | **Delete users** |
| **Short Description** | | Removes a user, its credentials and associated information from the system |
| **Actors** | | Node-in-box administrator |
| Pre-Conditions | | Node-in-the box is properly installed and configured |
| Success End Conditions | | A new user account is deleted, together with its data including subscriptions, saved searches, history, etc. |
| Data | | User records |
| Functions | | * remove user |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Administrator | Navigates to user management screen  Selects the user to be removed  Selects user removal option |
| 2 | NGDS System | Responds by removing user profile with its associated information including subscriptions, saved searches and history.  Notify administrator of unpublished records by a data steward, and change permissions of those data records to the administrator. |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | What happens if a user that is a data steward is deleted? Are her metadata records unpublished, assigned to a system pseudo user? | |
| 2 | DN: QUESTION: Should the system remove the associated information or should certain info persist (logs, metadata etc)? | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_032** |
| Use Case Name | | **Administer user roles** |
| **Short Description** | | The goal of this use case is to allow the administrator to assign different roles to users. These roles control the users’ abilities to publish data in the system or to administer system functions. |
| **Actors** | | Node-in-box administrator |
| Pre-Conditions | | Node-in-the box is properly installed and configured |
| Success End Conditions | | User role assignment is updated according to administrator needs |
| Data | | User records |
| Functions | | * assign role to user * remove use role |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Administrator | Navigates to user management screen  Selects the user to be managed  Selects user roles |
| 2 | NGDS System | Responds by assigning selected roles to user. |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | Are users federated across nodes?  RSSF: Stewards and submitters are local users, end user/consumers are global users. | |
| 2 | DN: DERIVED REQUIREMENT: Every Node must have at least one administrator, one steward correct? | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_032b** |
| Use Case Name | | **Backup** |
| **Short Description** | | The goal of this use case is to allow node administrators to backup the data and meta-data being stored in the system. |
| **Actors** | | Node-in-box administrator |
| Pre-Conditions | | Node-in-the box is properly installed and configured |
| Success End Conditions | | The data, metadata and indexes of a node-in-a-box is successfully backed up |
| Data | | User records  Metadata records  Data  indexes |
| Functions | | * manual backup NGDS node * automatic backup of NGDS node |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Administrator | Navigates to user management screen  Selects backup option  Provides destination folder |
| 2 | NGDS System | Responds by backing up node data to the assigned repository |
| Variants | | |
| Step | Actor | Description |
| 1 | Administrator | Navigates to user management screen  Selects backup option  Set ups automatic backup option by providing a destination and a day of week/time |
| 2 | NGDS System | Responds by backing up node data to the assigned repository at the given day of week/time |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 |  | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_032b** |
| Use Case Name | | **Restore** |
| **Short Description** | | The goal of this use case is to allow node administrators to restore the backed up data and meta-data being stored in the system. |
| **Actors** | | Node-in-box administrator |
| Pre-Conditions | | Node-in-the box is properly installed and configured |
| Success End Conditions | | The data, metadata and indexes of a node-in-a-box is successfully backed up |
| Data | | Backed up blob with important node data |
| Functions | | * restore NGDS node |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Administrator | Navigates to user management screen  Selects restore option  Provides source folder and file |
| 2 | NGDS System | Responds by restoring up node data to the assigned repository |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 |  | |
|  |  | |

#### System Management (Meet NGDS System Requirements)

If a third party repository is used to store data, it is the responsibility of the node administrator to ensure that their node provides information in accordance with NGDS standards and protocols. This set of requirements must be filled by any system that wishes to play the role of a node or data repository within the NGDS. Alternatively, the NGDS node-in-a-box, described in this document, will simplify the node administrator’s workload by providing an easily installed and configured software package that satisfies these requirements.

The following are requirements that the node administrator must insure are satisfied. They are outlined as requirements since they are targeted functionality for software development.

• The node shall provide access to tier 1 data and file-based representations of tier 2 and 3 data via standard HTTP protocols

• The node shall provide access to tier 2 and 3 data through OGC web-services (WFS, WMS, WCS) where appropriate

• The node shall provide metadata that conforms to the USGIN metadata profile (http://repository.usgin.org/sites/default/files/dlio/files/2011/u11/usgin\_iso\_metadata\_1.1.3.pdf)

• The node shall provide access to metadata through a CSW 2.0.2 discovery service

• The node shall create de-referenceable URIs for all applicable resources, including data sets, metadata records, files, and features within data services

Once the node administrator is satisfied that their node meets the demands of standards and protocols required to participate in the NGDS, the node administrator may request registration of their node with the system’s aggregating catalog, thus publishing their node’s data into the system.

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_034** |
| Use Case Name | | **Register node** |
| **Short Description** | | The goal of this use case is to allow new nodes to be added to the NGDS grid. These nodes can be of different types included “node-in-a-box” installations, or third party data provider repositories. |
| **Actors** | | Node-in-a-box Administrator, NGDS Administrator |
| Pre-Conditions | | Node-in-the box properly installed as a NGDS node management hub  Client node-in the box properly installed but not yet registered |
| Success End Conditions | | A new node is registered in the NGDS network, and the data it provides becomes available to the other nodes, and searchable through the by the NGDS catalog |
| Data | | e-mails, node credentials |
| Functions | | * add new node * index new node |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Node administrator | Get NGDS administrator information  Include <<e-mail node-in-a-box administrator>> to NGDS administrator |
| 2 | NGDS Administrator | Include << register new node into NGDS>> |
| 3 | NGDS System | Responds by:  Locating the NGDS administrator user/node  automatically filling in the user e-mail address;  sending an e-mail to the NGDS administrator |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | Who manages the grid?  Is the grid configured centrally?  NGDS Administrator in a special user in a node elected to be the main node. | |
| 2 | DN: DESIGN QUESTION: Will the NGDS system use a one master – many slaves model or a truly federated model where there are only peers?  RSSF: We will decide this after a trade study. | |

#### Routine Maintenance

The node administrator must also perform routine maintenance tasks pertaining to their node. These include upgrading software and responding to user questions and requests.

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_033** |
| Use Case Name | | **e-mail (NGDS or node-in-a-box) administrator** |
| **Short Description** | | The goal of this use case is to allow users to send e-mail to administrators of the system to handle matters such as granting of especial access rights, or to register new nodes in the NGDS network, or other issues.  There are two types of administrators: node-in-a-box administrators, and NGDS administrators. |
| **Actors** | | All users, Administrators |
| Pre-Conditions | | Node-in-the box is properly installed and configured  Administrator has registered her e-mail information |
| Success End Conditions | | Users can communicate their needs/issues with the node-in-a-box administrator |
| Data | | e-mails |
| Functions | | * send email to administrator |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | All users | Navigates to e-mail node-in-a-box administrator page  Select send e-mail to administrator  Type –mail message  Selects send e-mail, confirms operation |
| 2 | NGDS System | Responds by: automatically filling in the user e-mail address;  sending an e-mail to the administrator |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 |  | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_035** |
| Use Case Name | | **Upgrade node-in-the-box application** |
| **Short Description** | | The goal of this use case is to upgrade the software that implements the node-in-the-box application. This can potentially include the data repository, the catalog service, and the WebApp application |
| **Actors** | | Node-in-a-box administrator |
| Pre-Conditions | | Node-in-the box is properly installed |
| Success End Conditions | | The software components(s) of the node are updated without data loss and will minimum impact on other nodes of the NGDS network |
| Data | | Possibly all the data stored in the data repository and index |
| Functions | | * update software components * shutdown node * restart node |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Node Administrator | Gracefully disconnects the node from the network  Shuts down the server component (s)  Update software  Restart service(s) |
| 2 | NGDS System | Shuts down gracefully & restarts without data loss |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 |  | |
|  |  | |

## Software Developer Use Cases

Software developers are users who build applications that access NGDS resources using the system’s public, service-oriented APIs. These applications may be built in order to satisfy any number of functions, including data discovery, accessibility, visualization and analysis.

Figure 7illustrates use cases for software developers**,** whowill build applications that access NGDS resources using the public web services API of the system for their own applications, and the public API of the system will mainly support the standard web services WMS, WFS, and CSW. The current system will allow users to publish links to their application websites, but will not provide any advanced catalog feature to manage those applications such as what is provided by app stores.



Figure Software developer use cases

### Develop Applications

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_040** |
| Use Case Name | | **Develop apps using NGDS standard protocols** |
| **Short Description** | | Through the use of web protocols, in particular: CSW, WFS and HTTP, software developers can build applications that utilize the data and meta-data stored in NGDS. They can also use NGDS to locate referenced data (data that is not stored in NGDS components but is referenced by the NGDS catalog). They can, for example, develop applications that display that information on maps, perform data analysis and discover geological information of geothermal sites within US |
| **Actors** | | Software developer |
| Pre-Conditions | | NGDS data repositories and catalogs are available and accessible through standard internet protocols.  Software developers may need an account to interact with the system if their application involves the updating of information in repositories and catalog, or if they utilize services such as posting of reviews. |
| Success End Conditions | | Software developers can successfully build applications that utilize the NGDS system resources |
| Data | | Data models, metadata schemas, key system URIs, e.g. catalog service main URI, protocol and data model documentation |
| Functions | | * Standard protocols * Ability to interact with NGDS via APIs * Logging of API usage statistics and accesses |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Software Developer | Build applications that utilize standard protocols and data models from NGDS |
| 2 | NGDS System | Responds to these protocols interactions via well known APIs |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | DN: It might be a good idea to issue API keys to control the number of queries and use of data or at least have a metric to understand who is using it and for what.  DN: Discussion about developer API keys. Not critical but something to keep in mind. Given there are not concrete plans on how to pay for maintenance after the system is up and running, providing unlimited API access might not be something that can be afforded. | |
|  |  | |

### Register New Application with NGDS

Furthermore, the software developer will wish to register their new application with the NGDS in order to promote its use throughout the system.

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_041** |
| Use Case Name | | **Contact NGDS admin to provide link to application** |
| **Short Description** | | The goal of this use case is to capture the need for a way to software developers request the inclusion of application links in the NGDS web site. They will do so via a public e-mail address or via a form that allow them to submit this information. |
| **Actors** | | Software developer, NGDS administrator |
| Pre-Conditions | | NGDS website is available and its URI is know by the software developer |
| Success End Conditions | | NGDS software developers can provide a link to their applications and this information is incorporated in the list of applications hosted in the NGDS website |
| Data | | URIs from software developers  e-mail or URI to website location where developers can post those requests |
| Functions | | * Publish new application links via the NGDS system |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Software Developer | Post links to software via the NGDS website |
| 2 | NGDS System | Captures the request and routes it to the NGDS administrator for approval |
| 3 | NGDS Administrator | Responds by approving or rejecting a new website link and description |
| 4 | NGDS System | If approved, automatically updates the website. |
|  |  |  |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 |  | |
|  |  | |

## End-User Use Cases

Once datasets are published into the NGDS from a node, it will be possible for end-users to discover and access those datasets through a variety of different applications.

End-user use cases are organized around the basic workflows that the NGDS must support. These are: discover and gather, validate and evaluate, and analyze and visualize data. As shown in Figure 8 in each one of these steps, users can perform different activities.

Note that the use cases discussed in this section summarize the results described in the End-User Research Summary provided by Anthro-Tech (**see document /P05**)

Once datasets have been published to the system by registering them in the NGDS catalog and making the resource accessible through standard web service protocols and data formats, it will be possible for End Users to discover and browse the dataset using the NGDS WebApp application. They can browse resource descriptions returned by searching the NGDS catalog, and view the resource described by a particular catalog entry. Browsing capabilities will depend on the type of resource. Documents can be accessed through standard web linking and browser display for file types like PDF, TIFF, TXT. Tabular data in represented in NGDS supported formats will be displayed in a standard table type grid display. Geo-referenced resource locations will be visualized through a map interface that provides data exploration capability. This map-based data exploration utilizes the information provided in the metadata hosted in the catalog, as well as third party data layers to help users triangulate the data with that external information. Map-based search can be combined with other types of searches such as text-based, and can be further narrowed down by data type and metadata content filters.

For file-based resources, End Users will be able to download the associated files using links from catalog entries. The WebApp application will provide data filtering and download capabilities for data sets that accessible through NGDS services in standard NGDS interchange formats. End users will be able to enter filter criteria, and subset datasets to meet user-defined criteria, and last, end users will be able to export these results to their local file system.



Figure End-user use cases overview

In the following sections, we analyze these 3 different steps individually.

### Discover and Gather Data

The first and most important use of the NGDS to end-users is the ability to discover and gather geothermal data. Gathering data involves searching the NGDS metadata catalog for different types of geothermal information, casually browsing the search results, and further filtering those results based on the user’s own criteria of relevance. The complete tree of Use Cases is illustrated in Figure 9. We will describe only the most important of those use cases in the following.



Figure Data gathering supporting use cases

The “gather data” use case, shown in Figure 9, can be broken down into sub use cases. It consists of searching the metadata catalog for different types of geological information, doing a casual browsing, and further filtering that information based on the user’s own criteria of relevance. Users can also save existing searches, load them in a later occasion, or use those searches as subscriptions. This publish/subscribe model allow users to be notified whenever new data matching a certain criteria is input into the system catalog.

The NGDS catalog can also be searched by data geo-location. The geo-location search can be based on landmarks, on areas in the map, or can include more precise, coordinate-based searches.

Once a search is performed, it is usually the case that information must be further prioritized and filtered. The filtering of information allows users to select a relevant subset from the totality of data returned by a general search, and to rank the information according to the user’s needs. Filtering can be performed by data type, e.g. by selecting or discarding certain data types, including well drilling, geographical, geochemical, geophysical, geological and land data, as well as publications; by geo-location, i.e. filtering results that fit within an area in the map, or by other metadata attribute, including the document content if this information can be extracted from the dataset (e.g. Tier3 dataset types). These filtering capabilities are not mutually exclusive, and are done in a combined and interactive way.

Here follows a brief description of each of the use-cases.

#### Map-Based Search

In searching for data, geographic location provides the first mechanism by which end-users may use to filter results. To that end, users expect a map-based search interface in which they can quickly filter results by panning and zooming the map, or moving directly to an area-of-interest by specifying a landmark name or coordinates.

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_014** |
| Use Case Name | | **Map-based search** |
| **Short Description** | | The goal of this use case is to support users discovery of metadata by using maps. Map-based search consist in zooming, panning and selecting a region in a map. This search method can be used together with other search methods, to narrow down the data in a search. |
| **Actors** | | End User/Data Consumer |
| Pre-Conditions | | System is correctly installed and operational |
| Success End Conditions | | The user can visualize a result of a search as layers in a map. |
| Data | | Metadata features shown as layers (WFS protocol), metadata summary of selected data items on map (when users click on a data point in the map) |
| Functions | | * Map navigation capability: panning, zooming, selecting sub-regions * Retrieve metadata record for selected elements in a map |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | User | Navigates to the map screen  Utilizes the map functions (panning, zooming) to navigate to a geographical location  Visualizes data points in a map  Include <<filter metadata on map>> use case |
| 2 | NGDS System | Responds to user commands by updating maps and data layers |
| Variants | | |
| Step | Actor | Description |
| 1a |  | Include other types of search as landmark-based, coordinate-based, content-based, etc. |
| Exceptions | | |
| Step | Actor | Description |
|  | NGDS System | In case of failure to display layers due to network errors, for example, the system should notify user of the situation |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 |  | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_013** |
| Use Case Name | | **Landmark-based search** |
| **Short Description** | | The goal of this use case is to allow users to utilize landmarks (state, city, county, district, known geothermal sites) to narrow down the search in the catalog. This search method can be used together with other search methods, to narrow down the data in a search. |
| **Actors** | | End User/ Data Consumer |
| Pre-Conditions | | A database of landmarks and their geo-locations must be available for search |
| Success End Conditions | | The user finds information based on landmarks |
| Data | | Landmark name |
| Functions | | * Find landmark * Retrieve landmark geo-location |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | User | Uses landmark name as a search criteria |
| 2 | NGDS System | Consider a list of metadata items around a landmark as a filter in the current search |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | As noted by one of the monitors this needs to be clarified with Antro-tech. what is a landmark? What are examples of landmarks? How difficult would it be to support landmark-based search in our approach? Can we reuse functionality from search engines as Google map services to do this?  DN: Agree. I think pushing that off to the map vendor is a good approach as funding is limited within NGDS. Still, a request from Anthro-tech for clarification would be good. | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_015** |
| Use Case Name | | **Coordinate-based search** |
| **Short Description** | | The goal of this use case is to allow users to utilize geographical coordinates to narrow down the search in the catalog. |
| **Actors** | | End User/Data Consumer |
| Pre-Conditions | |  |
| Success End Conditions | | The user finds information based on geographical coordinates |
| Data | | Geographical coordinates |
| Functions | | * Validate coordinates * Retrieve data from the system based on proximity or containment within geographical coordinates |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | User | Uses coordinates as a search criteria by possibly typing them in |
| 2 | NGDS System | Validates input from the user  Consider a list of metadata items around or within the provided coordinates as an additional filter in the current search |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | As noted by one of the monitors this needs to be clarified with Antro-tech. do the users need to type coordinates? Is there a UI-based way to support this search without requiring users to type those coordinates, for example, by drawing a box in a map?  I think the onus should be on the end user to figure out where the landmark is and locate it on the map. It is very unclear what the landmark means. | |
|  |  | |

#### Keyword-Based Search

End-users wish to be able to specify a keyword or set of keywords that restrict results to resources which utilize those keywords. This should work in tandem with map-based search, allowing users to filter simultaneously based on keyword and location.

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_016** |
| Use Case Name | | **Keyword content-based search** |
| **Short Description** | | The goal of this use case is to allow users to search data by its metadata content registered in the catalog.  If data comes in tier 3 format, it includes its indexed content; if it comes in Tier1 and Tier2 formats, the search is based on whatever could be extracted/converted/indexed into data or meta-data. |
| **Actors** | | End User/Data Consumer |
| Pre-Conditions | | There is metadata in the catalog |
| Success End Conditions | | The user finds information based on metadata content |
| Data | | All metadata stored in the NGDS catalog |
| Functions | | * Content-based search of metadata records * Content-based search of data content |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | User | Navigates to a content-based search panel  Types in search criteria |
| 2 | NGDS System | Validates input from the user  Retrieves metadata records based on user typed content  Shows results as text and, when possible, as layers in the map |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | DN: There may be a need for a basic thesaurus but hard to make with a limited budget. | |
|  |  | |

#### Refining Results and Faceted Search

Once the end-user has generated a set of search results, usually through a combination of the methods outlined above, that user will wish to continue to refine their search results. This can occur through further map-based and keyword-based search, or may be accomplished by selecting from a set of search facets, or fields.

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_019** |
| Use Case Name | | **Filter results by type** |
| **Short Description** | | The user can also narrow down its search results by specifying certain data types of interest, thus ruling out all other data that do not belong to these types from the returned list of search results. |
| **Actors** | | End User/Data Consumer |
| Pre-Conditions | | A search was performed and a subset of the metadata from the catalog was retrieved by the NGDS catalog.  The search result is displayed as a list of metadata records |
| Success End Conditions | | The user can narrow down the search results |
| Data | | A subset of metadata obtained by a search  A list of metadata types present in the subset of metadata under consideration |
| Functions | | * Filter metadata set by type |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | User | Selects one or more data types from a list of types displayed for a search result |
| 2 | NGDS System | Responds by displaying only the metadata list for the types the user selected  The map may also update in response to this search |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 |  | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_020** |
| Use Case Name | | **Filter results by metadata attributes** |
| **Short Description** | | Different content models prescribe different attributes to different types of data, these attributes can be used to further refine the search result, for example, excluding data points for which their metadata record do not have certain attribute content values. |
| **Actors** | | End User/Data Consumer |
| Pre-Conditions | | A search was performed and a subset of the metadata from the catalog was retrieved by the NGDS catalog.  The search result is displayed as points in a map and as a list of metadata records |
| Success End Conditions | | The user can narrow down the search results |
| Data | | A subset of metadata obtained by a search  A list of metadata types present in the subset of metadata under consideration |
| Functions | | * Filter metadata set by attribute content |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | User | Types in a filtering criteria based on supported metadata attributes |
| 2 | NGDS System | Responds by displaying only the metadata list matching user filter criteria  The map may also update in response to this search |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | Missing Filters: by location terms, geothermal thematic terms, source organization, publication date, popularity and user ratings. Requirements need to clearly indicate that these vocabularies will exist on which such faceted filtering can be performed.  ND: This should be a requirement for the metadata team | |
|  |  | |

End-users also expect to be able to draw a specific area of interest on the map, and have the search results only reflect data from that specific area.

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_018** |
| Use Case Name | | **Filter results on map** |
| **Short Description** | | Once a search is made and search results is displayed as points on a map, the user can narrow down its search by selecting a sub-area in the map, thus filtering out all data points that are outside that geographical region. |
| **Actors** | | End User/Data Consumer |
| Pre-Conditions | | A search was performed and a subset of the metadata from the catalog was retrieved by the NGDS catalog.  The search result is displayed as points in a map |
| Success End Conditions | | The user can narrow down the search results |
| Data | | A subset of metadata obtained by a search |
| Functions | | * Select region in a map * Filter search results based on a bounding box on a map |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | User | Select a subset of elements in a map by defining a region. i.e. creating a bounding box in a map |
| 2 | NGDS System | Respond to user input by zooming the map to the selected region and filtering out results that are outside the bounding box. The list of metadata result displayed in a textual form may need to be updated. |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 |  | |
|  |  | |

### Validate and Evaluate Data

Once an end-user has located resources of interest, they next investigate to learn what they can about the dataset or file content to determine if it is valid for their intended purpose. The first step in the process is adequate visualization of the search results themselves. The complete tree of use cases is illustrated in Figure 10.



Figure Data validation supporting use cases

In the following we list the most important use cases of this category.

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_017** |
| Use Case Name | | **Browse/view metadata search results** |
| **Short Description** | | The goal of this use case is to allow users to visualize the results of a search and inspect its content.  This visualization is supported by metadata lists and maps. |
| **Actors** | | End User/Data Consumer |
| Pre-Conditions | | A search was performed and a subset of the metadata from the catalog was retrieved by the NGDS catalog. |
| Success End Conditions | | The user can find what he/she is looking for |
| Data | | A subset of metadata obtained by a search |
| Functions | | * Browse search results * Inspect elements in a map |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | User | Include <<search metadata catalog>>, <<filter metadata set>> |
| 2 | NGDS System | Shows results as text and, when possible, as layers in the map  Shows metadata record when user clicks on a map or on metadata lists |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 |  | |
|  |  | |

#### Metadata Evaluation

The user first wishes to assess information available in a resource’s metadata such as the originator’s description of the dataset, the resource’s accessibility, its provenance and peer-reviews of the resource. In order to do so requires that the user be able to access the resource’s metadata in its entirety.

The evaluation itself is outside the scope of the system. The system just provides the metadata (and data as per the above use case). The end user’s evaluation will probably happen once they get the data they are seeking.

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_026** |
| Use Case Name | | **View metadata record** |
| **Short Description** | | After locating a piece of metadata in the catalog, the user inspects the metadata record in more detail, for example, to decide if it refers to the data she is looking for. |
| **Actors** | | End User/Data Consumer |
| Pre-Conditions | | A search was performed and a metadata record was selected for further inspection |
| Success End Conditions | | The user is able to access and view the contents of the metadata |
| Data | | Documents and structured records stored in GTDA repository or third party repositories. The data is located through a URI |
| Functions | | * Retrieve metadata record * Visualize metadata record |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | User | Select metadata record  Opens metadata record in a visualization mode |
| 2 | NGDS System | Responds by showing the contents of all the content of the metadata record |
| 3 | User | Views the metadata record content |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  | User | In case the URI pointed by the metadata record becomes unavailable during the execution of this procedure, the system must provide an error message. The metadata record may be marked as invalid. |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | Missing requirements for other “views” of metadata, for example as human-readable HTML, or as an ATOM entry. These are just low-hanging fruit. | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_024** |
| Use Case Name | | **Provide peer ratings** |
| **Short Description** | | By inspecting data reviews posted by other users, a user can gauge the accuracy and validity of data. Peer ratings can include textual description, star ratings or both.  This UC captures the fact that the system must provide support for peer ratings.  As part of this use case, users can also post peer ratings. There is no restriction of who can post those ratings. The user log-in information is used to identify the peer rating poster |
| **Actors** | | End User/Data Consumer |
| Pre-Conditions | | The existence of ratings posted by other peers for the case of review  None if the user will be the first to post a rate |
| Success End Conditions | | The user is able to view peer ratings text and star ratings in the metadata posted in the NGDS catalog. |
| Data | | Peer ratings and their textual description that are attached to metadata in features of the map and the search results list |
| Functions | | * Provide peer ratings star rating and text for a given metadata record * Post a peer rating * Store peer reviews |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | User | Include family of use cases <<gather data>> |
| 2 | NGDS System | Shows user a result of a data gathering activity  Displays peer ratings together with returned data |
| 3 | User | Opens peer ratings of selected items  read peer ratings star rate and their textual description |
| Variants | | |
| Step | Actor | Description |
| 2b | User | Include use case <<view metadata content>>, <<view document content>>  Post a peer rating |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | This feature leads me to suggest that optional user profile fields indicating who a reviewer is (e.g Steve Richard of AZGS rates this 3 stars) | |
|  |  | |

#### Data Comparison

Often end-users wish to look more carefully at the data’s values and compare it to other datasets to which they are familiar. This may involve visualization and data analysis within the search interface, or in an external environment. See the discussion of data visualization and analysis below for more information.

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_022** |
| Use Case Name | | **Triangulate with other sources** |
| **Short Description** | | Users may compare the metadata returned by the system with external data layers, provided by external WFS sources such as demographics, topological, weather, and so on, thus helping them to make inference on the quality of the information. These use cases can also be performed by third party applications that consume the data published in NGDS |
| **Actors** | | End User/Data Consumer |
| Pre-Conditions | | A search was performed and a subset of the metadata from the catalog was retrieved by the NGDS catalog. |
| Success End Conditions | | The user compares the returned metadata with third party data sources and comes to a conclusion |
| Data | | External data sources as WFS providers for map layers  Metadata from the system |
| Functions | | * Download data * Export data via standard protocols to third party applications |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | User | Include family of use cases <<gather data>> |
| 2 | NGDS System | Shows user a result of a data gathering activity |
| 3 | User | Selects download report |
| 4 | NGDS System | Creates a metadata report |
| 5 | User | Compares the data manually with their own, or with the help of a CAD tool. |
| Variants | | |
| Step | Actor | Description |
| 3b | User | Uses CAD tool, via WFS to read NGDS repository data |
| 4b | NGDS System | Exports data via WFS for third party CAD tool. |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | It is not clear if the scope of the system includes the mashing up of information with external data sources. This seems to be an advanced feature that may be better achieved by utilizing third party CAD tools.  DN: Agree! | |
|  | RC: The general idea is that you compare the data in a particular dataset to some other known data. This should be fleshed out as requirements for accessibility and download of data. | |
|  | MM: There is an opportunity here to enable crowd sourced information about resources via this use case. | |

#### Storing and Sharing Search Results

The process of data evaluation can take place over a long period of time, and may involve multiple end-users. In order to facilitate this, end-user search interfaces must support the capability of a user to save and share a set of filtered search results with other users.

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_025** |
| Use Case Name | | **View data content** |
| **Short Description** | | After locating a piece of data, the user inspects the data content by URI element that the metadata refers to. |
| **Actors** | | End User/Data Consumer |
| Pre-Conditions | | A search was performed and a metadata record was selected for further inspection |
| Success End Conditions | | The user is able to access the data pointed by the metadata record |
| Data | | Documents and structured records stored in NGDS repository or third party repositories. The data is located through a URI |
| Functions | | * Retrieve URI document * Open and display document content to user |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | User | Select document view |
| 2 | NGDS System | Responds by retrieving the document or data record, and opening it with an appropriate viewer, that will allow user to inspect the document or data record content  A tabular viewer for a structured data record would be useful. |
| 3 | User | Views the data content |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  | User | In case the URI pointed by the metadata record becomes unavailable during the execution of this procedure, the system must provide an error message. The metadata record may be marked as invalid. |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | DN: Long URLs can be aliased via a URL shortener. This may be better as it does not require the system to preserve the state of a specific search yet allows the search to be shared. A discussion to have. Rather than save the search criteria, maybe it should just provide a URL encoded string that can be used to represent the same state. This is less expensive from a systems perspective. Example - https://www.google.ca/search?q=NGDS&oq=ngds&sugexp=chrome,mod=0&sourceid=chrome&ie=UTF-8 | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_010** |
| Use Case Name | | **Save selected search criteria** |
| **Short Description** | | The goal of this use case is to allow users to save searches, to be reused in a later time, and for setting up subscriptions to content changes.  In this use case, after the user performs a search, she saves that search parameters for further use. This search then can be used to subscribe to new data, and to continue a previous discovery activity. Searches are saved on the end-user accounts, for their private use. In the future they may be shared among other users.  When saving a search, users can opt to make search public so others can reuse it. |
| **Actors** | | End User/Data Consumer |
| Pre-Conditions | | In order to allow saving and retrieval, the user must be identifiable; hence, there is a need for users (in particular the end user) to be logged in using their unique account. |
| Success End Conditions | | The search criteria is properly validated and saved into the system under a given name. |
| Data | | Search criteria |
| Functions | | * Save search criteria * Validate search criteria * Record search parameters by monitoring user input * Make search public to other users |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Users | Include use cases <<Search meta-data catalog >>, <<filter meta-data set>> |
| 2 | NGDS System | Performs search according to user discovery workflow  Record save parameters by monitoring user input |
| 3 | Users | Select save search criteria option from the UI  Give the search a name  Select option to make search public/private |
| 4 | NGDS System | Saves search criteria  Make search criteria public if user selected so. |
| Variants | | |
| Step | Actor | Description |
| 1b | Users | Input search criteria in a separate content-based search form, instead of using the included use cases |
| 2b | NGDS System | Uses form-based search as criteria |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | Should the system support form-based search only or should it record data as shown above? Which option is better?  DN: I would adhere to the architectural principles known as REST. A URI represents the state of a specific resource etc. | |
| 2 | Should the saved searches from one user be visible to other users as Antro-tech indicated that searching is a collaborative process?  Maybe searches are not saved under a use id but treated as a resource. They could then be saved under the resource name (URI) and the (anonymous) user could create a bookmark to access the search later. This bookmark/URL can also be shared with other users.  DN: Agree. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_012** |
| Use Case Name | | **Load previous search criteria** |
| **Short Description** | | The goal of this use case is to support users in loading previously saved search criteria. They do so by browsing through their list of saved searches. |
| **Actors** | | End User/Data Consumer |
| Pre-Conditions | | In order to allow saving and retrieval, the user must be identifiable; hence, there is a need for users (in particular the end user) to be logged in using their unique account. |
| Success End Conditions | | The loaded search criteria is loaded and executed, displaying results in the system |
| Data | | Saved search criteria |
| Functions | | * Load saved search criteria |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | User | Navigates to the saved search screen  Browses through existing saved searches (if any)  Selects an existing saved search  Instructs the system to load the search |
| 2 | NGDS System | Loads a saved search  Uses the search criteria to search the catalog  Display search results to user |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | DN: Again, about the need to authentication in order to save a search, this is possibly not true. A saved search can be represented by a URI. Who made it is not relevant. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_023** |
| Use Case Name | | **e-mail metadata record URI to third party users** |
| **Short Description** | | After a search, users can also choose to e-mail the metadata set URI to other users in order to collect opinions on the quality of the metadata and possibly the data also. |
| **Actors** | | End User/Data Consumer |
| Pre-Conditions | | A search was performed and a subset of the metadata from the catalog was retrieved by the NGDS catalog. |
| Success End Conditions | | An e-mail is sent out to a recipient with a URI to the report on the current metadata set obtained through the system |
| Data | | URI to Metadata from the system  Search information |
| Functions | | * E-mail metadata report * Build metadata report |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | User | Include family of use cases <<gather data>> |
| 2 | NGDS System | Shows user a result of a data gathering activity |
| 3 | User | Selects e-mail metadata results functionality  Provides e-mail recipient information |
| 4 | NGDS System | Creates a report with the metadata  Sends e-mail to the address provided by user |
| Variants | | |
| Step | Actor | Description |
| 1b | NGDS System | Runs periodic subscription query  Goes to step 4 |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | This ought to include not only email, but RSS, Facebook, Twitter, Google+ and maybe Reddit and/or Yammer? | |
| 2 | DN: DERIVED REQUIREMENT: If you do this, you must also add in mechanisms to prevent this system from being used to spam people or abuse it in other ways. | |
|  |  | |

The end-user should also be notified if new data is published to the NGDS that satisfies a particular saved search criteria that they have elected to follows.

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_011** |
| Use Case Name | | **Subscribe to new data** |
| **Short Description** | | The goal of this use case is to allow users to utilize saved search criteria as subscriptions to new content published in the catalog that matches specific criteria.  Users will be notified via e-mail, when new data that has been input in the system, matching that subscription criteria was published |
| **Actors** | | End User/Data Consumer |
| Pre-Conditions | | In order to allow subscriptions and notifications to occur, the user must be identifiable; hence, there is a need for users (in particular the end user) to be logged in using their unique account.  The search criteria used in the subscription is properly validated and saved into the system under a given name |
| Success End Conditions | | A subscription is successfully performed |
| Data | | saved search/subscription criteria,  user e-mail obtained from user profile  subscription name |
| Functions | | * Load search criteria * Subscribe to search criteria * System notification service that periodically notifies users of matched searches |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Users | Include use cases <<Load previous search criteria >>  Select use search criteria as a subscription option |
| 2 | NGDS System | Fills in user e-mail information using her profile information or  Saves search criteria as subscription |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | DN: Is it possible this can be done without using saved search criteria? | |
|  |  | |

### Analyze and Visualize Data

Finally, once the data set is ready, it can be exported for further analysis. As shown in Figure 11, the user can export metadata to CSV file, or can further gather the data by following the URIs provided by the catalog.

Note that, data export may be the perimeter where the NGDS node stops and anything that happens beyond this boundary is out of scope of the system.



Figure Data analysis

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_027** |
| Use Case Name | | **Export metadata** |
| **Short Description** | | Users can export metadata records for different purposes, for example, to integrate them into their reports and spreadsheets, to further analyze these records with a CAD system, etc. |
| **Actors** | | End User/Data Consumer |
| Pre-Conditions | | Metadata was gathered and filtered |
| Success End Conditions | | The user is able to save the metadata records |
| Data | | Metadata records managed by GTDA |
| Functions | | * Retrieve metadata records * Visualize metadata records |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | User | Selects metadata set (Include use case <<gather data>>, Include use case <<filter data>>)  Indicates the type of format to export the records (we assume only CSV is supported)  Initiates report generation |
| 2 | NGDS System | Responds by generating a CSV file(s) with the metadata records |
| 3 | User | Download generated metadata record files |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | If there is more than one metadata record type, how the user would like to have the report? One CSV file per metadata type? | |
| 2 | It is beyond the scope of this project the development of tools or providing support for detailed analysis of data. Hence, the need for exporting the data for further analysis. | |
| 3 | DN: Is CSV the only option here? Would it be possible that some users want JSON or XML?  Would suggest yes. JSON is probably the most useful but it depends on the data models (content models). I would hate to try and represent complex binary data as CSV. | |

## System Administrator Use Cases

As shown in Figure 12, the NGDS administrator will supervise the operation of the whole NGDS system, including its node-in-the-box and third party repositories. The administrator will be able to register/unregister nodes in the NGDS system, manage accounts of participants, including their publication rights. They can also communicate with node-in-box administrators.



Figure NGDS Administration use cases

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_036** |
| Use Case Name | | **Register new nodes into NGDS** |
| **Short Description** | | The goal of this use case is to allow NGDS administrators to respond to new node requests. The administrator should evaluate the validity of the request, accepting, or rejecting it. |
| **Actors** | | NGDS administrator |
| Pre-Conditions | | Node-in-the box properly installed as a NGDS node management hub  Client node-in the box properly installed but not yet registered  The new node must have at least one unique data or metadata record not currently in the system. |
| Success End Conditions | | A new node is registered in the network, and the data it provides becomes searchable in the by the NGDS catalog |
| Data | | e-mails, NGDS nodes registry |
| Functions | | * add new node * index new node |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | Node administrator | Include <<e-mail node-in-a-box administrator>> to NGDS administrator |
| 2 | NGDS Administrator | Receives a request for new node addition  Evaluates the request for its validity  If valid request, update node registry |
| 3 | NGDS System | Responds by: automatically scanning the new node and creating a metadata index for that node. Utilizes the standard communication protocols CSW. |
| Variants | | |
| Step | Actor | Description |
| 2b | NGDS Administrator | Rejects request based on external criteria |
| Exceptions | | |
| Step | Actor | Description |
| 1 | NGDS System | Raise exception in case of incompatible/invalid protocols.  Unless they support the common standards and protocols that they cannot join. |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 |  | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_038** |
| Use Case Name | | **Delete nodes from NGDS network** |
| **Short Description** | | The goal of this use case is to allow NGDS administrators to respond to remove previously registered nodes from the system. |
| **Actors** | | NGDS administrator |
| Pre-Conditions | | Node-in-the box properly installed as a NGDS node management hub  The node to be removed is currently registered |
| Success End Conditions | | A currently registered node is removed from the network, and the metadata records referencing this node are removed from the NGDS catalog |
| Data | | NGDS nodes registry, NGDS catalog data |
| Functions | | * Remove existing node * Remove index for node |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | NGDS administrator | Navigates to NGDS node administration page  Selects node to be removed  Removes selected node |
| 2 | NGDS System | Responds by: automatically removing the metadata for the removed node from NGDS index, and by removing the node registration record. |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
| 1 |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 |  | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_039** |
| Use Case Name | | **Communicate with the Node-in-a-box admin** |
| **Short Description** | | The goal of this use case is to allow node-in-a-box admins and NGDS admins to communicate in the handling of administration issues such as request node removal, check node registration information, etc. |
| **Actors** | | NGDS administrator, node-in-a-box administrator |
| Pre-Conditions | | Node-in-the box is properly installed and configured  Administrator has registered her e-mail information |
| Success End Conditions | | Administrators can communicate with each other |
| Data | | e-mails |
| Functions | | * send email to administrator |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | All users | Navigates to e-mail node-in-a-box administrator page  Select send e-mail to administrator  Type –mail message  Selects send e-mail, confirms operation |
| 2 | NGDS System | Responds by: automatically filling in the user e-mail address;  sending an e-mail to the administrator |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | DN: I would rank the implementation of this use case as a low priority. This can be done by simply placing a mailto:xxx@xxx.com link on a page and letting the users own email client handle it. This seems a bit much to build into the system. | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_037** |
| Use Case Name | | **Manage NGDS user accounts** |
| **Short Description** | | Allows the system administrator to manage NGDS users. This will allow the system administrator to add and remove users on the administered node, and assign user roles and group membership. |
| **Actors** | | NGDS administrator |
| Pre-Conditions | | NGDS network is properly installed and configured |
| Success End Conditions | | The administrator is able to perform the main administration operations |
| Data | | User records |
| Functions | | * Add user * Delete user * Modify user permissions and roles |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | User | Include use cases <<add user>>, <<administer user roles>>, <<delete user>> |
| 2 | NGDS System | Responds to administration operations, enforcing role and users policies. |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | DN: Not sure if this is a use case I agree with. I think it would be better to delegate this to node administrators. If there is a user that the NGDS super admin wants removed, he or she can communicate that to the node admin perhaps? The worry is that the super NGDS admin would have to understand the metadata and data that the user is associated with before deleting to ensure no data or metadata is left orphaned. The decision is not mine but I encouraged discussion on this point. | |
|  |  | |

## Use cases common to all users

Some of the use cases of the system do not belong to any of the functional groups in particular, but represent functionality that is used throughout the system.

In particular, users must login and logout the system in different situations as described in the use cases. This allows the system to track the user actions, and to automatically fill in contact information as name, e-mail, etc. these use cases are described as follows.

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_001** |
| Use Case Name | | **Login** |
| **Priority [1-3] higher is more important** | | 3 |
| **Reason for priority** | | User authentication is a basic activity that is a pre-condition for many other use cases. Failing to implement it is a show-stopper. |
| **Short Description** | | The goal of this use case is to uniquely identify and authenticate a user, allowing the system to enforce access policies, and to use the user information to automatically fill in forms data, save searches and subscriptions, identify comments, etc. |
| **Actors** | | Data Submitter, End User, Data Steward, System Administrator |
| Pre-Conditions | | The user is logged out of the system |
| Success End Conditions | | The user is logged in and authenticated with the system |
| Data | | User login and password, or credentials collected in a third party authentication service |
| Functions | | * Authenticate user using system credentials * Authenticate user using third party services, for example: Facebook, Gmail, and others |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | User | Navigates to the system login screen  Types in login information  Or utilizes one of the existing third party authentication services, eg. Gmail, Facebook. |
| 2 | NGDS System | Responds by letting user login, granting access according to her role credentials  Or by denying access to the user |
| Variants | | |
| Step | Actor | Description |
|  |  | 1. User forgets password but remembers username 2. User forgets username but remembers password 3. User forgets both username and password 4. User forgets username/password and the email they used to register. |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | DN: Password recovery tools could be added. Depending on the level of sophistication, enforcing a inimum set of standards for password might also be prudent. | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | | **UC\_002** |
| Use Case Name | | **Logout** |
| **Priority [1-3] higher is more important** | | 3 |
| **Reason for priority** | | User authentication is a basic activity that is a pre-condition for many other use cases. Failing to implement it is a show-stopper. |
| **Short Description** | | This allows a logged-in user to gracefully end and close their session |
| **Actors** | | Data Submitter, End User, Data Steward, System Administrator |
| Pre-Conditions | | The user is logged in the system |
| Success End Conditions | | The user is logged out with no negative side effects to the system |
| Data | | Current user section and credentials |
| Functions | | * Logout user |
| Main Sequence | | |
| Step | Actor | Description |
| 1 | User | Indicates logout action to the system or  Leaves the website, forcing a timeout logout |
| 2 | NGDS System | Responds by finalizing current user session, and logging out the user |
| Variants | | |
| Step | Actor | Description |
|  |  |  |
| Exceptions | | |
| Step | Actor | Description |
|  |  |  |
| Open Issues (Please use this field to indicate questions/comments on the use case) | | |
| ID | Issue Description | |
| 1 | DN: This implies session management. If that is the case, it is not noted as a requirement. Otherwise the user would not need to end a session. They could just close the browser window and walk away. | |
|  |  | |

# Overall system quality attributes

We captured the system functional requirements in the form of use cases. For each use case, we listed a set of system functions exercised. In this section, we discuss overall system quality attributes that further describe the conditions on which the system will function.

## Maintenance

In this application maintenance plays the major role among the non-functional requirements. The application will be designed such that the software stack can be changed.

In order to improve maintainability the system shall be optimized for a lean code base. This ensures that developers quickly gain an overview. In order to minimize the code base the system shall make use of third party components wherever and whenever possible. In other words “buying” is better than “making”. “Buying” in our case also includes making use of open source/ freeware.

There is a trade-off between buying and making. Powerful frameworks may require complex configuration. It may be more costly to configure an off-the-shelf framework than to develop the required functionality. Well-maintained (active) framework projects undergo frequent updates that must be applied to a deployed system in order to stay up-to-date or maintain security. Documenting the configuration of the components – an important aspect with respect to maintainability – may be difficult. Therefore, for each component it must be decided if the maintainability is higher when taking an off-the-shelf product or using self-developed components that are tailored to the specific use case.

For those components that are designed within the project we require unit tests to be available that test at least the minimum functionality. In addition system integration tests are recommended but due to the high costs of designing a system test toolkit this might not be possible with the given time and resources.

**NFR001** All project-developed source code shall have comments at least on a per-class level.

Supported Use Cases: ALL

**NFR002** The System’s architecture shall be documented.

Supported Use Cases: ALL

**NFR003** The System’s configuration parameters shall be documented.

Supported Use Cases: ALL

**NFR004** The System’s source code shall be covered by unit tests to at least 50% of coverage. Regression tests will be run as part of the software process.

Supported Use Cases: ALL

## Usability & Accessibility

Here follows the System requirements that affect its usability.

**NFR005a** The System shall provide a reasonably simple to use installation tool: The tool shall install all required components (potentially with the exception for Java and/ or Python) and guide the administrator through the initial configuration steps.

**NFR005** The system shall be cloud-ready: It must be available as an archived Virtual Machine (or VM) that is ready to use after a few configuration steps. Our main target is EX2(from Amazon) and other providers that are able to boot a VM image)

Supported Use Cases: ALL

**NFR006** The system shall include detailed instructions that guide the user through the process of installation of one node and joining a grid.

Supported Use Cases: ALL

**NFR008** The project-developed graphical user interfaces shall use a uniform look-and-feel for web applications, defined by the UX team. Minor customizations will be possible by adjusting Cascading Style Sheets, for example.

Supported Use Cases: ALL

**NFR009** The project-developed applications shall provide online help explaining how to perform user-related functions.

Supported Use Cases: ALL

**NFR010** The project-developed applications shall present the user with clear, understandable and accurate information explaining each task that can be performed using the software.

Supported Use Cases: ALL

**NFR011** The project-developed applications shall present the user with human understandable error messages explaining the errors that occur during user interactions.

Supported Use Cases: ALL

**NFR012** The key data import operations should be transactional. i.e.The user shall be able to abort operations before completion, without any negative consequences.

Supported Use Cases:

**NFR013** The project-developed applications shall provide a status indicator showing the progress towards completion of user triggered processing, search queries, exports and downloads.

Supported Use Cases:

**NFR013b** The system shall comply with the section 508 Amendment to the Rehabilitation Act of 1973 section related to Web-based Intranet and Internet Information and Applications.

Supported Use Cases:

**NFR013c** The system shall comply with the ISO/TS 16071, “Ergonomics of human-system interaction – Guidance on accessibility for human-computer interfaces.

Supported Use Cases:

## Performance and Scalability

Here follows the performance requirements of the System. We assume the system will be designed to support universities and data providers within USA, with a maximum of 1000 concurrent users.

Our estimates are based on the size of the Geothermal community. It is unlikely, for example, that Geothermal data will be of interest for the average citizen of the US. Besides, from the relatively small community of academic geothermal experts there will be a number of non-academic users, from either government or industry. However, the number of organizations able to exploit this data is limited.

Regarding hardware sizing guidelines: as the project goes along we will collect experience and then can also provide guiding guidelines. This will be addressed in our project plan.

**NFR014** Each data provider node must be capable to maintain a list of at least 100 other NGDS nodes for harvest or distributed search.

Supported Use Cases: ALL

**NFR015** Each data provider node shall indicate it has taken action in response to all user operations within 2 (two) seconds.

Supported Use Cases: ALL

**NFR016** Each data provider node shall be capable of supporting up to 50 simultaneous authenticated, logged-in users.

Supported Use Cases:

**NFR017** Each data provider node shall be capable of handling at least 50 (fifty) HTTP requests every 1 (one) minute.

Supported Use Cases:

**NFR018** Each data provider node shall respond to every request from the NGDS in no more than 10 (ten) seconds.

Supported Use Cases:

**NFR019** The System shall be able to handle the import of data files up to 2GB in size.

Supported Use Cases:

**NFR020** The System shall be able to handle the import of up to 1000 data files in any one import operation.

Supported Use Cases:

**NFR021** The System shall support the storing of up to 100000 data files in the import directory of each data provider.

Supported Use Cases:

**NFR022** The System shall support the storing of up to 500GBs of data files in the import directory of each data provider.

Supported Use Cases:

## Security

We assume that this system is mainly providing a portal to public information. Therefore, the security features of the system will focus on trustworthiness of the data being provided by through it.

We cannot discount, however, nefarious use of the email functions, spam comments being added if annotations are allowed and other typical bad behavior, for example.

The system will include the usual security aspects such as authorization, authentication, access and auditing (the 4 A’s of security). The two aspects “authentication”, and “access” can be easily integrated as there are very well-defined protocols and technologies available (namely HTTPSs and secure XML for access, and form-based or basic-authentication for authentication). Hence these two aspects have a binary character, i.e. they are either fulfilled or not fulfilled. Authorization and auditing can be implemented with more variability and hence they are not binary. In our case auditing is most likely of less importance. Access control must allow for distinguishing between Data Stewards, Submitters and basic users. Access controls must protect the system’s administration interface against unauthorized access. Data stewards need assurance of data integrity against arbitrary modification.

Data Stewards and submitters must have access only to appropriate record collections, requiring a business process for managing access rights for certain data.

In fact security can be one of the main obstacles of the architecture since it is a vertical aspect impacting the complete architecture stack. Due to its “open nature” we therefore intend to reduce security needs to the bare minimum knowing that this will make adding security enhancements in the future more difficult.

Security also impacts the deployment process as the system administrator has to set up some initial accounts, define some user groups and adjust access rights for those user groups. We also must prevent that “default passwords” (e.g. for accessing the database) linger beyond the installation phase.

Here follows the System’s requirements for security and accessibility.

**NFR024** The System shall embody a security plan and process to ensure that unauthorized users are denied access.

Supported Use Cases: Log

**NFR025**Valid login authentication is required for all data submitter, steward, and administrator functions.

Supported Use Cases: ALL that perform changes in data/metadata

**NFR026** The System shall only allow users access to write data they have permissions to write

Supported Use Cases: ALL with write accesses

**NFR027** The System shall only allow users access to download data files they have permissions to download

Supported Use Cases: ALL downloads

**NFR028** The System shall only allow the data steward for a resource permission to delete it.

Supported Use Cases: UC\_037

**NFR029** The communication between end-users and the services of the system will be encrypted using HTTPS protocol.

Supported Use Cases: ALL communication

**NFR030** Data communications between the External Systems and NGDS applications shall be secured by message authentication where applicable/necessary.

Supported Use Cases: ALL communication

**NFR031** A data provider node shall only transmit data to clients when the data is published by an authenticated data submitter or steward for access by that client’s permission group.

Supported Use Cases: ALL requiring authentication

**NFR032** The data provider node shall maintain the integrity and availability of all data stored in its local data store.

Supported Use Cases:

**NFR033** The data provider node shall maintain the integrity of all files stored in the node’s local file repository.

Supported Use Cases:

**NFR034** The data provider node shall maintain a log of activities for auditing purposes.

Supported Use Cases:

**NFR035** NGDS applications shall be developed considering good security coding practices, thus minimizing vulnerability to attacks. In particular, it should comply with FIPS (Federal Information Processing Standards)

Supported Use Cases:

## Supportability

Here follows the requirements that enhance the supportability and maintainability of NGDS project-developed software.

**NFR036** NGDS software shall be written using the standard coding style for the used programming languages. For example, Oracle Java Coding Styles, if Java turns out to be the selected language.

Supported Use Cases: ALL

**NFR038** NGDS software shall be designed utilizing the concept of encapsulation. Components shall be created that encapsulate related functionality within them, and nothing else.

Supported Use Cases: ALL

**NFR039** All software shall be modular to minimize the time and complexity involved in maintaining and extending the platform and application.

Supported Use Cases: ALL

**NFR040** NGDS software shall not contain any statically detectable dead code.

Supported Use Cases: ALL

## Data Requirements

The Data Assessment Team will inventory additional data to be submitted by project subcontractors, and project management will prioritize datasets for delivery based on recommendations from the Domain Steering Committee and User-Centered design team. A prioritized list of important geothermal data types will be ascertained and used as the basis for populating NGDS data assets. See the Data Requirements Specification [/P02/](#P02_GTDA_Data_Requirements_Spec), for a description of data categories and attributes that will be stored in the System’s database.

Specification of metadata for any data asset provides the information necessary to enable discovery and evaluation of that asset through various search mechanisms, as well as access to the asset by people or automated processes. For Geothermal and other Geosciences data assets, the NGDS will use the US Geosciences Information Network (USGIN)[[2]](#footnote-3) [Metadata Recommendations for Geoscience Resources](http://repository.usgin.org/sites/default/files/dlio/files/2011/u11/usgin_metadatarecommendationsgeoscienceresources_v1.04.pdf) for guidance on metadata content, and the [USGIN ISO metadata profile](http://lab.usgin.org/sites/default/files/profile/file/u4/USGIN_ISO_Metadata_1.1.4.pdf) and [implementation guidelines for encoding metadata content](http://repository.usgin.org/uri_gin/usgin/dlio/499) for interchange between catalog search and client applications. An [NGDS metadata compilation template](http://gdsdpwg.net/Lists/System%20Proposals/Flat.aspx?RootFolder=%2fLists%2fSystem%20Proposals%2fMetadata%20compilation%20template%20Excel%20workbook&FolderCTID=0x0120020036F7F455DF25CF4AA312970A719C86A3) is available for tabular metadata compilation or setting up existing metadata for bulk loading.

A number of content models are available defining data interchange content for important geothermal data. These content models are implemented in XML as Geography Markup Language (GML). Simple features for interchange using OGC Web Feature services to enable interoperability among computing systems. Content models include a description of data attributes for geothermal types such as:

* [Active Fault/Quaternary Fault](http://www.stategeothermaldata.org/data_delivery/content_models/active_faultquaternary_fault)
* [Aqueous Chemistry](http://www.stategeothermaldata.org/data_delivery/content_models/aqueous_chemistry)
* [Borehole Temperature Observation Feature](http://www.stategeothermaldata.org/data_delivery/content_models/borehole_temperature_observation_feature)
* [Direct Use Feature](http://www.stategeothermaldata.org/data_delivery/content_models/direct_use_feature)
* [Drill Stem Test Observations](http://www.stategeothermaldata.org/data_delivery/content_models/drill_stem_test_observations)
* [Earthquake Hypocenter](http://www.stategeothermaldata.org/data_delivery/content_models/seismic_event_hypocenter)
* [Fault Feature](http://www.stategeothermaldata.org/data_delivery/content_models/fault_feature)
* [Geologic Contact Feature](http://www.stategeothermaldata.org/data_delivery/content_models/geologic_contact_feature)
* [Geologic Unit Feature](http://www.stategeothermaldata.org/data_delivery/content_models/geologic_unit_feature)
* [Geothermal Area](http://www.stategeothermaldata.org/data_delivery/content_models/geothermal_area)
* [Geothermal Fluid Production](http://www.stategeothermaldata.org/data_delivery/content_models/geothermal_fluid_production)
* [Geothermal Power Plant](http://www.stategeothermaldata.org/data_delivery/content_models/geothermal_power_plant)
* [Heat Flow](http://www.stategeothermaldata.org/data_delivery/content_models/heat_flow)
* [Heat Pump Facility](http://www.stategeothermaldata.org/data_delivery/content_models/heat_pump_facility)
* [Lithology Interval Log Feature](http://www.stategeothermaldata.org/data_delivery/content_models/lithology_interval_log_feature)
* [Thermal/Hot Spring Feature](http://www.stategeothermaldata.org/data_delivery/content_models/hot_spring_feature_content)
* [Volcanic Vents](http://www.stategeothermaldata.org/data_delivery/content_models/volcanic_vents)
* [Well Header](http://www.stategeothermaldata.org/data_delivery/content_models/well_header)
* [Well Log Data Compilation](http://www.stategeothermaldata.org/data_delivery/content_models/well_log_data_compilation_workbook)

These content models are intended to specify interchange formats, not database tables for data management. The models are denormalized to facilitate queries via the OGC service interfaces, and to minimize joins and data lookups by data consumers. The objective is to make data access and utilization simple. The data provider node will support upload of data in a CSV encoded table structure implementing these content models. Such uploaded files will be validated for conformance with the model, then the data will be transferred to a GIS feature class and NGDS web services deployed using the data.

## Design Constraints

The follow requirements are derived from design decisions that represent constraints that are mandated and must be adhered to.

**NFR041** The NGDS components shall interface to NGDS data provider nodes via NGDS web services.

Supported Use Cases: ALL

**NFR042** The NGDS components shall use the API provided by NGDS web services for data functions, for example WCS, WFS.

Supported Use Cases:

**NFR043** The NGDS participants shall use a data abstraction layer for access to databases used for metadata management and management of data in NGDS content models.

Supported Use Cases:

**NFR044** The system components shall use web services for communication with NGDS client software and other NGDS nodes.

Supported Use Cases:

**NFR045**The NGDS components shall use web services to send data to NGDS client applications, including the NGDS portal application.

Supported Use Cases:

**NFR046** The NGDS components shall use web services for the querying of data from NGDS nodes.

Supported Use Cases:

**NFR047** The system shall provide requested metadata to the NGDS as XML files

Supported Use Cases:

**NFR048** NGDS metadata catalog services shall provide metadata search results using the USIGN ISO XML encoding profile in response to CSW 2.0.2 protocol.

Supported Use Cases:

**NFR049** The software shall be designed with Linux as its main target platform. However, as much as possible, it shall utilize portable technologies such as Java, that will facilitate its porting to other operating systems and platforms.

Supported Use Cases: ALL

**NFR050** Under no condition should the failure of one node be capable of crippling or rendering the entire NGDS system useless. The system must be capable of adding or removing nodes while maintaining normal operations.

Supported Use Cases: ALL

## Licensing Requirements

The software system being developed as part of the project shall have an open source license variant. The details of the license are being developed in coordination with DOE and project partners. The license will include provisions to allow users to copy, distribute and transmit the software, to adapt the software for other applications, and to make commercial use of the software, under the condition that the following attribution for the source of the software is included in any copy or derived work:

In order to preserve the original NGDS licensing terms, the use of third party libraries and application servers that violate these terms will be vetted.

The data provided by the data providers will not be made available to the public domain until it is released and published through the “publish” feature of the System. Other access control constraints may be applied by individual nodes at their discretion.

## Applicable Standards

See the WSS [**/P05/**](#P05_GTDA_Web_Services_Specification) for the details of the web services standards to be applied.

## Installation and Deployment

The NGDS website will be a web application hosted by a server to be identified by project management. See the Deployment Specifications Document (DSD) [**/P03/**](#P03_GTDA_Deployment_Specification) for more details. It contains details of hardware components, operating systems, licensing agreements, etc.

The SCR development team will deliver the NGDS node-in-a-box software stack (WepAPP, NGDS Repository and NGDS catalog) to BSU project management team for deployment, testing and acceptance on a server of their choice, in accordance to the project’s schedule. The NGDS node-in-a-box software will be delivered as an installable application to BSU project management for deployment, testing and acceptance on a server of their choice, in accordance to the project’s schedule.

The software package will include an installation program in the form of a shell script of an apt package for Ubuntu Linux.

**Appendix**

# Node-in-a-box Software Package

A redistributable, node-in-a-box, free/open-source software package will be developed (composed of an NGDS repository, an NGDS catalog and an NGDS Desktop as shown in Figure 13). This node-in-a-box software application will give data providers a simple way to register data sources, load data and expose those data as a node in the NGDS network. The software will support batch import and upload of shared datasets in supported formats adhering to standard content models. As part of the WebApp, a user interface will be provided to help users upload data to the system. The use of the NGDS data repository or the WebApp, however, is not required in order to participate as a node in the NGDS network; data providers may use whatever tools they wish to expose their data, as long as they utilize data interchange formats and web-service protocols conforming to NGDS specifications, and register themselves, as data providers, with the NDGS Catalog.

In this arrangement, different node-in-a-box instances can co-exist in the system. This requires their catalogs to be synchronized via a federation service or some sort of aggregating catalog. The mechanisms of federation, either centralized, hierarchical or peer-to-peer, for example, will be later decided, in the design stage of the project (see /P04/). A centralized aggregating catalog service is an option.



Figure NGDS Data Provider Software Package

Specifically, Figure 13 shows that the NGDS will facilitate publication, visualization and discovery of geothermal data using services and applications. The diagram depicts the various major interfaces for NGDS node. The NGDS will provide a catalog, a web top UI and a data repository application. End users/Data consumers participate in the network by browsing through metadata and consuming data. Data Providers can publish metadata to NGDS catalogue service, and can use NGDS repository to store their data. Optionally, they can provide their own data sources, as long as they publicize metadata to the system in the Catalog and make data available through standard protocols and data exchange formats. NGDS System Administrator can install and manage data providers, granting them the ability to publish data to the system. Software developers produce applications that consume the data and metadata published in the system. NGDS also relies on functionality from external Web services such as authentication, and maps (via WMS protocol).

# End-User/Data Consumer Software

End-users may interact with the system through a variety of entry points, but the project will implement two primary access points: a NGDS website and a NGDS WebApp application. As much as possible, these will be integrated in order to appear to the end-user as a single web-based experience. The primary access point to the system will be the NGDS WebApp application.

## NGDS WebSite

The website will be designed to provide information about all the NGDS participants. It will serve as gateway to the system, allowing users to discover data and applications that utilize NGDS resources. The site will include information on the project’s progress, NGDS specifications, the access to the map-centric search application, other software applications utilizing NGDS services, NGDS presentations, documentation and tutorials, a link to the catalog of NGDS nodes, and any other results as they become available. Note that Arizona State University is in charge of developing this Website.

## NGDS WebApp

A user-friendly, web-based application will be created in order to support finding, visualizing, mapping, and acquisition of data by end-users/data consumers. This application will provide a user interface that allows end-users/data consumers to discover and access resources made available across all NGDS nodes, and to search for data across the system based on topic, location, time or other criteria. Standardized metadata describing each dataset will provide the user with the information necessary to determine the utility of that dataset for their purposes. Geographic datasets will be visualized through a map interface that will also allow users to inspect the details of individual data points (e.g. wells, temperature measurements, etc.) from properly formatted datasets. Note that the map UI is not intended for analysis and comparison of different data layers. It works only as a graphical way for searching data in the map and inspecting individual elements metadata. The WebApp will also provide a user interface for node-in-a-box data providers to publish data to their NGDS data repositories.

# References

## Project References

The following table identifies all of the references to project documents applicable to the development of this SRS document.

|  |  |  |
| --- | --- | --- |
| **Reference** | **Status** | **Document Name** |
| /P01/ | Approved | System Vision |
| /P02/ | Planned | Data Specification (to be provided by AZGS) |
| /P03/ | Planned | Deployment Specification |
| /P04/ | Planned | Software Design Description |
| /P05/ | Planned | Web Services Specification |
| /P05/ | Reviewed | Anthro-Tech End User Research Summary Report |

Table : Project Reference Documents

Note that a number of the above referenced documents may be under development at the time of release of this document.

## External References

The following table identifies all of the references to documents external to the project that are applicable to the development of this SRS document.

| **Reference** | **Document Name** |
| --- | --- |
| /E01/ |  |

Table : External Reference Documents

# Acronyms, and Abbreviations

The following table lists the abbreviations used in this document, in order to promote their unique and unambiguous usage throughout the document and the Project.

|  |  |
| --- | --- |
| **Abbreviations** | **Definition** |
| DOE | Department of Energy |
| NGDS | National Geothermal Data System |
| SDD | Software Design Description |
| DIS | Data Import Schema |
| OGC | Open Geospatial Consortium |
| WSS | Web Services Specification |
| ORM | Object-Relational Mapping |
| WMS | Web Map Service |
| WFS | Web Feature Service |
| CSW | Catalogue Service for the Web |
| WCS | Web Coverage Service |
| NetCDF | Network Common Data Form |
| API | Application Programming Interface |
| CSV | Comma-Separated file format |
| URI | Uniform Resource Identifier |
| URL | Uniform Resource Locator |
| OAI-PMH | Open Archives Initiative – Protocol for Metadata Harvesting |
| UTM | Universal Transverse Mercator coordinate system |

Table : Abbreviations

# Glossary of Geological Terms

The following table lists the terms used in this document, in order to promote their unique and unambiguous usage throughout the document and the Project.

|  |  |
| --- | --- |
| **Terms** | **Definition** |
| Geological feature | Any type of geological phenomena such as:   * Faults * Fluid inclusions * Fluid rock interaction [Use for non-water fluids only, usually carbon dioxide] * Hydrothermal alteration [Same as rock-water interaction] * Fractures * Magmatism [Includes magma and magmatic processes; use more specific term “Volcanism” if applicable, although some articles address both; see also “Magma energy”] * Volcanism [Includes volcanoes and volcanic processes] * Permeability [Do not overuse; ability of fluid to flow through rock] * Pore pressure [Includes capillary pressure] * Porosity [Do not overuse; amount of pore space within rock] * Rock mechanics [Response of rocks to external forces, especially related to stress] * Seismic attenuation [Ability of rocks to absorb seismic waves] * Seismicity [Same as earthquakes; see also “Induced seismicity”] * Microseismicity [Same as microearthquakes] * Seismic velocity * Stress fields * Borehole breakouts [Bulging of wells, related to stress fields] * Tectonics [On a regional scale] * Any rock formations treated in detail   + Breccia   + Granite   + Greywacke [also spelled graywacke]   + Sedimentary basins |
| Data | Documents, Computer database, pictures, spreadsheets about geological features |
| Metadata | Summary of important characteristics of a piece of data used for the purpose of searching and discovery by users of a geological feature database. |
| Dataset | A group of geological data, typically related to one geological feature. |
| Geological survey | Gathering data using geological methods, or descriptions of regional geology; includes structure |
| Resources | A generic name to include data, metadata and other artifacts that are under the responsibility of a user. |
| Geophisical survey | Different types of descriptions of physical characteristics of regional characteristics including electromagnetic surveys, gravity surveys, magnetic, sysmic, shallow temperature surveys, etc. |
| Heat flow | Energy coming to the surface in a given area, often measured in watts/m2 and used in resource assessment; derived by multiplying temperature gradients and thermal conductivity; essentially equivalent to heat flux, which is an engineering term |
| Temperature gradients | Change in temperature with depth, e.g., °C/m; includes temperature depth profiles |
| Thermal conductivity | Ability to conduct heat; e.g., salt domes have high conductivity, while sedimentary basins have low conductivity |
| Well logging | Well measurements and descriptions, including borehole geology |
| Metadata template | A predetermined set of metadata attributes used to describe data about geological features. E.g. document meta-data include a document bibliographical citation, its geo-location, ownership, and URI (for the actual document) |
| Landmark | Locations and regions in the globe that are geo-referenced: city, district, state, county, known geothermal sites, etc. A land mark is implemented as a dictionary, a map from names into geographical coordinates. |
|  |  |

Table : Terms

# NGDS Data Access Protocols

Software developers and the applications they develop will utilize the system through various interfaces. The OGC Catalog Service for the Web (CSW 2.0.2) will be used to enable catalog search via a Web API. Data services will be implemented using OGC WMS, WFS, NetCDF services, as well as other services adopted by the technical and steering committee as the system evolves. File-based resources will be accessed using standard HTTP GET requests.

# Data Model

In order to be made available in the NGDS system, data must be provided in predetermined formats, and must be made available in the Web through standardized protocols. The publication of data in NGDS is supported by two types of data repositories: **NGDS Data Repositories**, and **External Data Sources.**

* **NGDS Data Repositories** (or node-in-a-box) are repositories which utilize software designed as part of this project to implement standard NGDS sharing protocols. These repositories are installed and maintained by individual data providers. NGDS data repositories support the WFS and WCS and CSW protocols, supporting the catalog and discovery of data.
* **External Data Sources** are repositories maintained by third party entities which utilize any software of their choosing. Integration with the NGDS is contingent on the publication of data in the supported protocols (CSW, WFS), the utilization of supported data representation, and the listing of their data in the NGDS Aggregating Catalog.

Users of the NGDS system may want to publish data in one of three major data formats:

* **Tier 1: unstructured** — represent file based resources, unstructured data in text, image, etc; requires human to extract data for analysis.
* **Tier 2: structured, but not standardized** — represent data structured in proprietary formats that are not compatible with standard NGDS content model. Data in this tier will need to be translated and adapted to NGDS standard representations, by their owners, before they can be published and accessible in the NGDS network.
* **Tier 3: structured, standardized** -- data published in the NGDS standardized protocols and interchange formats supported by NGDS content model.

Note that NGDS, and in particular, the NGDS data repository, will only support Tier1 and Tier 3 data, following Arizona State University data formats. Tier2 data will have to be first converted to standardized data models by their respective owners, before it can be made available via NGDS.

NGDS will provide links to those data templates, at Arizona state university, or DOE websites, and will provide forms to help users input that information in the system.

1. Datasets are made available for download if they are freely accessible. [↑](#footnote-ref-2)
2. <http://usgin.org> [↑](#footnote-ref-3)