DEVELOPING THE NATIONAL GEOTHERMAL DATA SYSTEM
ADOPTION OF CKAN FOR DOMESTIC & INTERNATIONAL DATA DEPLOYMENT

Ryan J. Clark[1], Christoph Kuhmuench[2], Stephen M. Richard[1]

[1]Arizona Geological Survey,
416 W. Congress St. Tucson, Arizona, 85701
e-mail: ryan.clark@azgs.com, steve.richard@azgs.az.gov
[2]Siemens Corporate Research
755 College Road East
Princeton, NJ, 08540, USA
e-mail: christoph.kuhmuench@siemens.com

KEYWORDS: GEOTHERMAL DATA SYSTEM, CKAN, DATABASE, OPEN SOURCE

ABSTRACT

The National Geothermal Data System (NGDS) Design and Testing Team is developing NGDS software currently referred to as the "NGDS Node-In-A-Box". The software targets organizations or individuals who wish to host at least one of the following:

- an online repository containing resources for the NGDS;
- an online site for creating metadata to register resources with the NGDS
- NDGS-conformant Web APIs that enable access to NGDS data (e.g., WMS, WFS, WCS);
- NDGS-conformant Web APIs that support discovery of NGDS resources via catalog service (e.g. CSW)
- a web site that supports discovery and understanding of NGDS resources

A number of different frameworks for development of this online application were reviewed.. The NGDS Design and Testing Team determined to use CKAN (http://ckan.org/), because it provides the closest match between out of the box functionality and NGDS node-in-a-box requirements.

The goal of this software development is to provide NGDS data consumers with a highly functional interface to access the system, and to ease the burden on data providers who wish to publish data in the system. It is important to note that this software package constitutes a reference implementation. Because the NGDS is based on open standards, other server software can make resources available, and other client applications can utilize NGDS data.

INTRODUCTION

Significant growth in the geothermal energy contribution to the national energy portfolio requires reducing the risk and cost of defining resources, characterizing new classes of larger energy resources, and optimizing management and expansion of exploited geothermal fields. Achievement of these objectives depends in large part on access to accurate geoscientific information. Much of

the existing information is inaccessible or difficult to locate and access. A DOE report by Deloitte (2008, pg. 27) concluded that "A study conducted in 2000 for NREL (Entingh, D., 2000) revealed that over a 25-year period, numerous geothermal research efforts were conducted with state and federal funding. Despite these efforts, the analysis and information contained in those research documents is difficult to access without significant research efforts. That same study cited that much geothermal resource attribute data also exists but is distributed among numerous locations and often stored in boxes, without any data index or organization."

Most of this data is unstructured and consists of documents such as publications, notes, images, or figures. There is also a rapidly growing body of structured data available that allows for new understanding or industrial exploitation of geothermal systems. To facilitate utilization of existing data by the geothermal research and development community, the National Geothermal Data System is developing three types of functionality: (1) data presentation and analysis applications, (2) applications for efficient searching across data repositories, and (3) repositories for geothermal data. This system is a full scale deployment of the distributed geoscience information network concepts developed by the USGIN project [http://usgin.org].

Figure 1depicts the deployment and interaction of the three application types. Data repositories (3) enable storage of research data with associated metadata to support data discovery, evaluation, and utilization. Presentation and analysis applications (1) are specifically designed to make use of structured data. They allow for statistical analysis as well as for human-user friendly presentation of structured data. The Lookup service (2) acts as glue between the analysis applications and the repositories. It allows for searching across multiple repositories. The lookup service includes a Portal application that provides a user interface for searching for and reviewing data. It allows for simple data exploration, addition of annotation on data, and data downloading.

The NGDS is designed based on a service-oriented approach using open standards to support data access by a wide variety of software applications, promote novel approaches to data analysis, and foster the development of tools by third parties. The key protocols for current deployment are the Open Geospatial Consortium (OGC) geographic data protocols (WFS, WMS, and CSW) and the underlying standard Web protocols (HTTP, FTP). HTTP provides functionality for basic file access, while the OGC protocols are used to search across the repositories, access data and to display it in appropriate form.

In this article we outline the design of a system that enables storing and annotating data with geographic reference, searching data across multiple data repositories and use of this data in simple data browsing tools. As a basis for this system we are using CKAN, an open, python-based framework application that brings many features required for the file-repository and search function, as well as extensions that implement some basic geographic data capabilities. An important aspect of our system is the implementation of an intuitive, functional user interface that supports search across the grid of data repositories.

The next section discusses example user scenarios. We then analyze the gap between NGDS requirements and the off-the-shelf features provided by CKAN, and finish with a high level view of the node-in-a-box implementation using CKAN.

USER SCENARIO: GEOTHERMAL DATA REPOSITORY

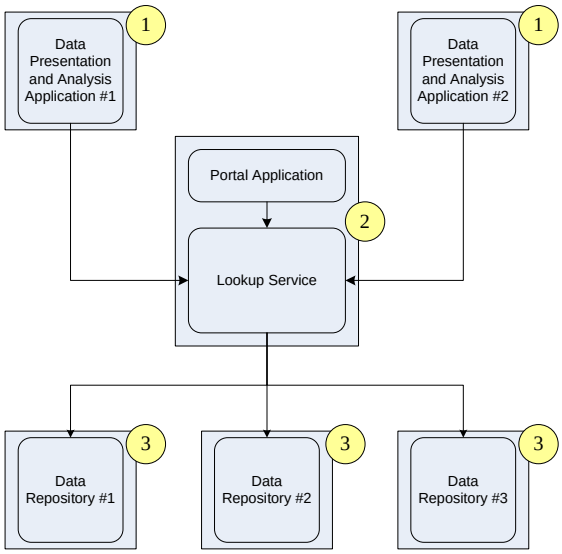

Figure 1. Geothermal Repositories, Lookup Services and Presentation Applications.

A user has a scanned report on reservoir modeling at a geothermal development prospect, based on a dataset of temperature, lithologic and geophysical logging of 200 boreholes. The report is to be registered as a document resource for the NGDS, the maximum temperatures measured for each well are to be published as a data service and the scanned logs are to be registered a resources indexed to the borehole from which they were obtained.

Figure 2 shows the services, features, data instances and metadata that would be involved in describing the data resources from this collection. Services include the metadata catalog service, and data services for well headers (one record for each well), well log observations (one record for each well log), and borehole temperature observations (one record for each temperature measurement, 0 to many measurements per well at different times or depths). Metadata would include a metadata record describing the scanned report document, with a link to the web location from which the report can be obtained. Ideally, the metadata for the report would also include links to related well header records for that report, and the well header records would include links to the related report that discusses results from the well. Each well header record also includes links to related well logs and temperature observations from the well.

The fields in the feature or record types are defined by NGDS content models. Users access data and metadata through service interfaces that provide responses in standard XML interchange formats that implement the NGDS content models. NGDS services are defined by OGC specifications (CSW 2.0.2, WMS 1.3.0 and WFS 1.1.1).

The described dataset could be incorporated into the NGDS at several levels. The simplest would be to create a metadata record for the scanned reservoir report and get the scanned report accessible online. The summarized temperature measurements would be included as tables in the paper. This is referred to as Level 1 data delivery.

Level 2 data delivery would consist of setting up any spatial data tables (well locations, temperature measurement points) that are included in the original data as feature services, using the data schema from the original data. This does not result in an interoperable data service, but still provides better access to the structured data. Metadata for the services would have to include a data dictionary.

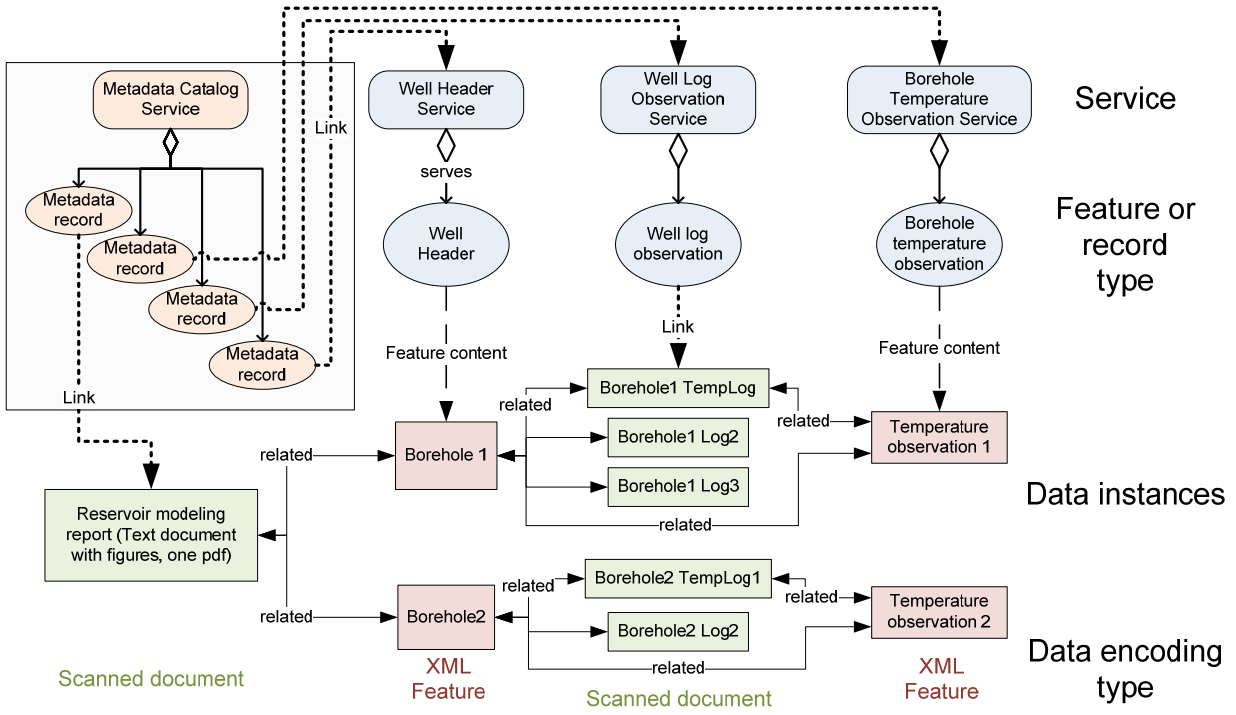

Figure 2. Services, features, and content associated with a reservoir modeling report based on a collection of well log data.

The preferred data publication approach is to load data for well locations into the NGDS well header content model, information about the well logs into NGDS well log observation content model, and the temperature measurement data into the NGDS borehole temperature content model, and publish the data using OGC WMS (for map images showing the locations of wells), and WFS (to provide structured data describing the wells, logs, and temperature measurements). This is Level 3 data delivery, and requires data integration (into the standard content model and interchange format) by the data provider, but makes data acquisition simpler for the data consumer.

User requirements for a data provider in this scenario include a metadata editor to create NGDS-compliant metadata, a publicly accessible web location to place the file for Level 1 access. For level 2 or level 3, a WMS and WFS server software installation must also be available; the user must also have the necessary permissions to copy files or data into the publicly accessible file system or database supporting the OGC service server.

EXAMPLE USER SCENARIO: DATA DISCOVERY

Metadata and data resources can be published from any node in the system, but an important system requirement is that a data consumer can search and access these various data resources from a single point of entry. This requirement will be met by providing a system node that aggregates metadata from the various publishing nodes, and provides a map-based search interface. Searching via this portal application will locate registered resources anywhere in the system; the metadata for the resources will include necessary information to access the resource without the user having to go to a new web location. This scenario implies two sets of requirements. One set encompasses the functions necessary to configure an aggregating node to harvest metadata from other nodes, and the functions necessary to execute the harvest—getting the records, avoiding duplication of metadata, and validating acquired records. The second set of requirements encom-

Table 1. Open source frameworks reviewed for project use.

Project	URL	Description
CKAN	http://ckan.org/	Open source project, maintained by Open Knowledge Foundation
GeoNode	http://geonode.org/	platform for the management and publication of geo-spatial data; mature open source project
DSpace	http://www.dspace.org/	turnkey institutional repository application; mature

pass the search application functions, including: maintaining search indexes; displaying search results as extents on a map, in summary lists, and the full metadata record formatted for reading on screen; simple data browsing capabilities for data evaluation; and data download of full or filtered datasets.

The two scenarios discussed here are fundamental to system operation, but do not include a variety of other use cases for system administration, monitoring, and maintenance. These are not considered here.

FRAMEWORK CANDIDATES

The first major choice facing the development team was to select a development framework. AZGS developers had done significant work on components to support the AASG Geothermal Data project (AASG package), and Siemens developers had done significant work developing a software package to support the Southern Methodist University Heat Flow Database project (GTDA package). Several other software projects were identified that provided some of the capabilities required for the NGDS node-in-a-box concept; these are summarized in Table 1.

ARCHITECTURAL DRIVERS

A major consideration in the evaluation is the long-term viability of the node-in-a-box application developed for NGDS. Building on an existing, active and widely used open-source project is considered a major asset to assure long term viability of the application.

Other significant factors that were considered of high value in the development framework include: 1) adaptability of the user interface to be compatible with an independently developed user experience concept for NGDS users; 2) ease of extensibility, with a plug-in architecture that allows addition of functionality without having to modify the core codebase; 3) support for geographic data and map-based search and data browsing; 4) support for administrative activities like user management, access control, and activity logging.

GAP ANALYSIS

The five candidate packages (AASG, GTDA, DSpace, CKAN, and GeoNode) were evaluated in a gap analysis comparing out of the box capabilities against a compiled set of requirements for the NGDS node software. The analysis considered feature completeness, based on how many of the required features are supported by the framework, and the difficulty of implementing missing functionality.

The result of this analysis placed CKAN as the best candidate framework. Decision to utilize CKAN was reinforced by its adoption by various government open data initiatives, and the responsiveness of the CKAN community to our requests for information and assistance.

CKAN

CKAN is a sophisticated application framework that provides many functions helping us to implement the application functions as laid out in the introduction. CKAN is written in Python and makes use of a variety of open source frameworks. The base framework is Pylons [http://www.pylonsproject.org/about/pylons], which itself is a combination of various open source frameworks integrated to form the basis for Web-based Enterprise-level applications.

The primary CKAN user scenario is data storage and management. Its core functionality consists of three features:

1. File storage,
2. Metadata management, and
3. Management of structured data.

In addition it offers a plug-in mechanism enabling developers to rapidly extend CKAN's core functionality. There are now a number of powerful extensions available supporting the development of productive data management sites. Most importantly, it provides an extension (ckanext-spatial) that supports geographic features as well as exposing metadata according to the OGC standard catalog service (CSW). Also, it is possible to modify or override CKAN's rather simple default UI in various ways.

Finally, CKAN implements typical housekeeping features (user management, logging, etc.) that are tedious to implement but crucial for the site's usability.

In the following we discuss these features and how we intend to extend CKAN with the remaining missing features we need for our data storage application.

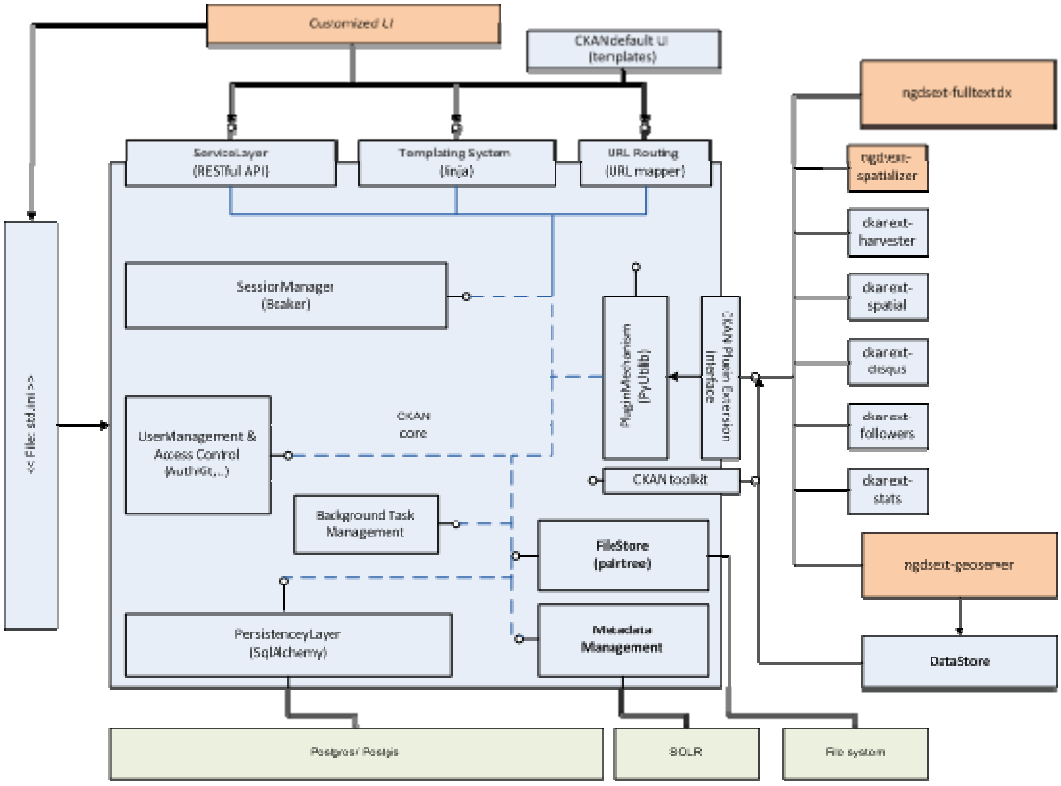

Figure 3. CKAN Architectural Components. Blue components are default components, orange-colored components need to be added for NGDS requirements.

Figure 3 shows the most important components of CKAN. The most elemental functions are grouped in the CKAN core component. A large number of these components provide infrastructure for the two core function blocks *CKAN file storage* and *CKAN Metadata management*. The remaining components inside the core provide basic essential services such as information persistence on the data management layer or session management on the service layer. Often these services are implemented using existing open source frameworks. All of these components expose a well-defined API, accessible in two ways: (1) a RESTful Web-service accessible to Web UI designers. (2) via an extension interface. Developers can use the second approach to add features to a component, or to "hook into" CKAN's core features. Extensions can inject callback functions that are called when certain activities are ongoing (e.g. upload of a new file). The following section describes some of the more important components depicted in Figure 3.

CKAN File Storage

The CKAN file storage feature provides the functionality to store, update, and delete files of arbitrary type. Stored files can be grouped into datasets and exposed under a single URL allowing external reference to them. Similar to a resource management system like Subversion (SVN), CKAN keeps track of the history of files. CKAN provides two file storage implementations: The first one maps the uploaded files to the file system of the underlying operating system. The second one allows for storing files on cloud drives. In both cases a Postgres database is used for persisting the state of the file store.

CKAN Metadata Management

The CKAN metadata feature provides the functionality to associate metadata entries to datasets. Out-of-the-box, CKAN implements a rather simple metadata concept that maintains very limited metadata content about the stored documents. However, CKAN is designed so that the default metadata content and UI can be extended with some Python programming.

The metadata can also be used to search for datasets. CKAN uses SOLR for indexing the metadata and provides an API for searching through that index. Its default UI provides a very simple search bar. Any dataset with metadata containing the entered key words will be returned as a list.

Structured Data

CKAN's file management feature allows for managing arbitrary files, which are handled as atomic blocks that cannot be searched internally or accessed in sub units. However for structured data, e.g. a table stored in a CSV file, it is useful to be able to return individual fields to the UI so that they can be viewed in an appropriate form (e.g. tables, or charts). CKAN's datastore extension takes care of this functionality. It allows for uploading structured documents into a database so that they can searched and search results be returned to the UI. In fact the datastore extension allows for returning such structured data as JSON objects that can be handled in an appropriate way by the UI.

Custom extensions are necessary to take full advantage of the datastore's features; these also require implementation of an appropriate UI. One required extension that must be developed is support for OGC services required by NGDS architecture.

Spatial data

Many data storage applications have to keep track of data associated with geospatial information, i.e. they are associated with areas on a map. CKAN comes with an extension that allows for as-

sociating geospatial information with datasets and provides a geospatial search feature, i.e. datasets can be searched on a map. The feature also works closely with map widgets that can be incorporated into the UI. However, there is no default UI for map-based search, and this will need to be developed by the project. The spatial extension also provides the capability of exposing metadata via the standard OGC catalog protocol (CSW) required by NGDS architecture.

Harvesting

Harvesting is an operation in which one node replicates metadata records from another node. The basic NGDS deployment plan is to harvest metadata from all NGDS nodes into a single aggregating node that will host the search application for the NGDS data access portal. CKAN provides an extension that enables harvest functionality. For our geothermal application this is a crucial feature. CKAN offers a UI for managing harvesting but the functionality can also be accessed via a command line tool.

Extensions via the Plugin Interface

CKAN provides an extension interface allowing developers to add functionality to the storage system. (see Figure 3). The extension interface provides for three features: (1) It allows new extensions to "hook" into CKAN's core components, (2) it allows new extensions to make use of the API of the system which is provided as a "toolkit", and (3) it allows new extensions to make new "hooks" available.

Hooks are places within the component's code at which extensions can register callback functions to be called when the relevant code segment is executed. This way an extension may add additional functionality, for example when a file is uploaded to the system. An extension can also provide hooks on its own so that other extensions can register callback functions. The functionality of the CKAN core features is aggregated in a toolkit that provides a well-defined, stable interface with a standardized obsolescence strategy. Using these three features developers can modify or extend all important steps in the data storage workflow.

Extension of the UI

CKAN provides a very basic out-of-the-box UI, mostly for educational purposes as an example to helps developers design UI's dedicated to their specific application. CKAN incorporates the Jinja 2 HTML templating system (http://jinja.pocoo.org/) for user interface construction. Web interface applications can access CKAN's core functionality via the CKAN RESTful API. CKAN's URL routing system also allows mapping URLs to functions within a user extension for additional customization options. UI developers can use any of these three mechanisms to override CKAN's default UI by providing a path to a UI overriding directory in CKANs configuration file. CKAN will search first in this directory for resources and only afterwards in the default directories.

ADAPTATION OF CKAN FOR NGDS

CKAN offers many of the features required for NGDS but there are several adaptations and extensions that must be implemented. The simple default CKAN UI needs to be replaced with a UI following the NGDS look and feel and supports the NGDS-specific features. This NGDS user interface design has been developed using wireframes and user feedback, independently of the decision on the implementation software framework. The interface needs to incorporate a very flexible map widget that will be used to search for resources as well as visualize search results. The UI shall also realize faceted search capabilities to facilitate seamless combination of multiple

search parameters to filter search results. The look and feel and functionality of the NGDS UI requires a ground-up UI implementation based on CKAN's template system.

Other features not provided by CKAN or any of its extensions, but required for NGDS include:

1. Creation of NGDS metadata, including spatial extent
2. Uploading structured data with geospatial information
3. Consistency check for well-known structured data files
4. Providing OGC services for uploaded structured files with geo-spatial information
5. Full-text indexing of documents
6. Role-based right for uploading and publishing data
7. User feedback and rating of uploaded data

The most important ones are discussed in the following sections.

Handle structured data that includes geo-spatial columns

One basic extension we require is the capability to handle structured data that contains geospatial information. Note that the CKAN's spatial extension can search spatially but it does not support uploading CSV files with geospatial information. It merely assumes that the data are already uploaded. In order to support this functionality we will have to add a new hook into the CKAN datastore and to implement a new extension. The following Figure 4 depicts how the components interact with each other.

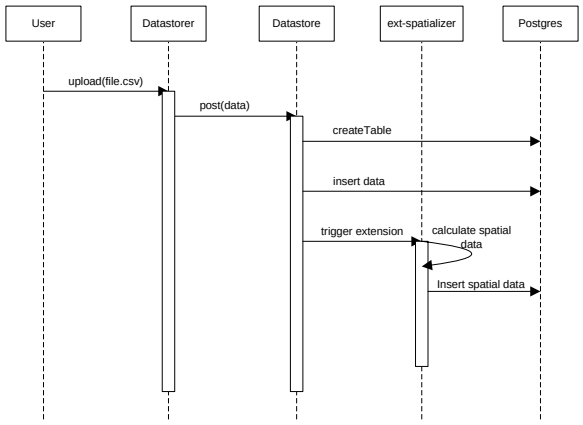

Figure 4: Message Sequence Chart handling structured data with spatial information. ext-spatializer is the custom component to be implemented by the project.

As before the CKAN datastorer and datastore will be used to upload structured data files and to create a table with the data in the Postgres database. The datastore will then trigger the new extension to create the column in the table containing the spatial information required by PostGIS (the FOSS geospatial extension for Postgres) for spatial operations.

Checking consistency of uploaded structured data files

When structured data that use a registered template format are uploaded, CKAN will validate the uploaded data for consistency with the template requirements. This operation can provide automatically calculated metadata such as minimum values, maximum values or averages in a dataset. A new extension is required to realize the validation function. Exactly where this extension will hook into the CKAN workflow remains to be determined.

Providing OGC services

A major gap in CKAN functionality is missing support for the OGC services WFS and WMS. Implementing these services as extension to CKAN would require significant effort, and is unnecessary because the services are implemented by GeoServer [http://geoserver.org/], another FOSS component we will use. An extension is required to connect the CKAN datastore with GeoServer. This extension will use the GeoServer API to dynamically configure and deploy new OGC services on the node GeoServer whenever a structured file is uploaded that contains geospatial data. In its simplest form this extension will provide GeoServer with an SQL view to the newly inserted data. More complex functionality in the future could create more complex OGC services that combine data from various tables or execute complex calculations before deploying a service.

Fulltext Indexing

Although we are putting significant effort into handling structured data the majority of uploaded data will consist of unstructured information such as publications. For such data it is important that we are able to have a full-text index available for searching. We will make use of SOLR in order to realize the full text indexing feature. SOLR will be triggered to index uploaded documents via an extension hooked into the upload process.

Role-based Uploading

In NGDS upload and publishing rights shall be restricted to authorized users. Moreover, it must be possible to separate the right to upload data from the right to publish data. We envision a process with selected users with upload permission (so-called data submitters) and selected users with publishing rights (so-called data stewards). Data stewards may review uploaded resources and decide if they are of sufficient quality for publishing.

It will not be possible to provide this functionality without modifying CKAN core features:

First of all we need to introduce the new role “data steward” which does not yet exist.

In a second step we have to provide a way for the system to control access to resources. We will do this on a per resource granularity. Hence, we introduce an attribute called “published” which can only be modified by users in the data steward role. We then have to integrate this new attribute into the dataset objects behavior so that they do not list resources that are not yet made public.

User feedback

We intend to enable the user to provide feedback to published datasets. This will be realized via an extension that keeps track of user ratings and annotations. Exactly where this extension will hook into the CKAN workflow remains to be determined.

NGDS Implementation

Figure 5 depicts how we will realize these features on top of CKAN and GeoServer.

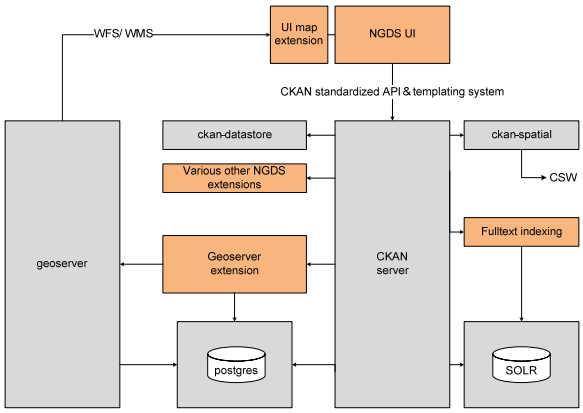

Figure 5: NGDS Component Architecture.

As can be seen GeoServer and CKAN access a single Postgres database, and an NGDS extension manages the GeoServer instance according to the data that is uploaded to CKAN. Other NGDS extensions described above are represented in a single box labeled "Various other NGDS extensions". The full-text indexing extension is represented separately because it will use SOLR as an additional external component.

CONCLUSIONS AND NEXT STEPS

Implementation and testing of the NGDS node-in-a-box will facilitate participation of new data providers in the system. Deployment of an aggregating node and NGDS portal will promote use of the information accessible through the system. An active community of providers and data consumers is essential to the long term viability of the NGDS. We have outlined an ambitious development program. Use of existing components and participation in active FOSS developer communities will facilitate progress and provides a path for long term utility of the software.

REFERENCES

Deloitte Consulting, LLP, 2008, Geothermal Risk Management Strategies: Report for Department of Energy - Office of Energy Efficiency and Renewable Energy, Geothermal Program, 44p. www1.eere.energy.gov/geothermal/pdfs/geothermal_risk_mitigation.pdf

Entingh, D., 2000, Status of DOE Geothermal Technical Report Collections: Princeton Energy Resources International, LLC, Geothermal Studies and Analyses: Report 6A, for Department of Energy - National Renewable Energy Laboratory, 36p. www.perihq.com/documents/Report%206A.pdf