iridium project CKAN use case

iridium project output

Project Information			
Project Title	*iridium - Institutional Research Data Management at Newcastle University*		
Project Hashtag	*#iridiummrd*		
Start Date	01 October 2011	End Date	30 June 2013
Lead Institution	Newcastle University		
Project Director	Janet Wheeler		
Project Manager	Lindsay Wood		
Contact email	*lindsay.wood@ncl.ac.uk*		
Partner Institutions	Not applicable		
Project Web URL	*http://research.ncl.ac.uk/iridium/*		
Programme Name	*JISCMRD02*		
Programme Manager	Simon Hodson		

Document Information			
Author(s)	Dr Ben Allen		
Project Role(s)	Research Computing Analyst, ISS		
Date	12 June 2012	Filename	*iridium*_CKAN_case_study_12_6_2013_v1_BA.docx
URL	*http://research.ncl.ac.uk/research/outputs/*		
Access	CKAN case study for external review.		

Document History		
Version	**Date**	**Comments**
v1	12 June 2012	*iridium*_CKAN_case_study_12_6_2013_v1_BA.docx (coversheet added by LW)

Introduction

CKAN (ckan.org) is an open source data portal and repository developed and maintained by the Open Knowledge Foundation, a non-profit organization "dedicated to promoting open data and open content". It is used to deliver a number high profile data portal sites including data.gov and data.gov.uk. There has also been considerable interest from the UK HE sector, looking to adopt CKAN as a Research Data Management tool.

Several other JISC RDM projects are testing CKAN, and the Orbital project arranged the CKAN4RDM workshop as a requirements gathering exercise. The discussion below relates to our own experiences of CKAN during the iridium project, and should be considered in conjunction with outputs from other JISC MRD2 projects.

Installation of CKAN

The university's linux systems are standardized around a RedHat / CentOS deployment, which is unfortunately not a platform supported by CKAN. However, CentOS installation instructions were available (https://github.com/okfn/ckan/wiki/How-to-Install-CKAN-2.0-on-CentOS-6.3), and were complete enough to allow the installation of a development version of the code for testing. The choice to use an unstable version of CKAN was made with a view to testing new features. As discussed below, the decision was also driven by a desire to implement Shibboleth authentication for which (at the time) only a single, version specific extension existed.

While the installation was "relatively" easy, much of the documentation on the main ckan.org site can be confusing and in some cases misleading. At the time of installation, a common problem was the mixing up of instructions specific to a certain code version. While no doubt exacerbated by our choice to run a development version, it seems other early adopters experienced the same frustrations, as evidenced both by messages to the CKAN developer mailing list, and the outputs from CKAN4RDM workshop in London.

A final issue experienced during the installation and setup was that some key CKAN features, namely the preview and analysis of numeric data sets, didn't function as expected. Although not made explicit in any general documentation, CKAN makes use of external data processing web services for some of its functionality. An initial decision to allow access to CKAN solely from within the university's private network meant that these external services could not retrieve the required data sets for preview. There were two solutions, the simplest of which was to allow access to CKAN from the whole internet. Alternatively, and preferably, work has already begun on developing alternative, locally hosted service to provide the preview functionality.

Shibboleth integration

We have successfully integrated Shibboleth authentication into our CKAN instance, so users not only authenticate with their institutional credentials, but are automatically registered the first time they log in.

Since user authentication within CKAN is performed using the repoze.who python framework, in theory the integration of different authentication providers should be straightforward.

In our case, we were able to adapt an existing Shibboleth extension developed for the Finnish Science Data Catalog (https://github.com/kata-csc/ckanext-shibboleth), and targeted at CKAN version 2.0a.

Several local modifications to the extension were required, both to remove FSDC specific code, and update the extension for compatibility with our locally installed version.

There were only a few weeks worth of changes between code versions, but the differences were sufficient to render the extension inoperable in its original form. However, the development work needed to fix this was relatively minor, and integration of our version of CKAN and Shibboleth was completed in under a week.

The rapidly changing CKAN codebase has caused subsequent problems though, and some of our attempts to upgrade the running system have failed. It is assumed that these issues are a consequence of using an unstable development version of CKAN, and would not have occurred had we deployed a stable release. As has been mentioned elsewhere, such problems may be indicative of the relatively immature status of the code, and a lack of clear direction.

Finally, it is worth noting that recent months have seen the development of a second Shibboleth extension, ckanext-saml2. With better integration into core CKAN functionality, it is being developed by the OKFN and should be considered the preferred method for Shibboleth integration in the future. We have however not yet tested it on our deployment, since it is incompatible with our implementation.

NUE-Crops data harvesting extension

The NUE-Crops project at Newcastle University creates reasonably sized data sets containing detailed records of crop yield, weather conditions, harvest dates etc at their research farm. While this data was originally stored in a disparate set of excel spreadsheets, notebooks, and scraps of paper, a database and web front-end has now been developed internally within AFRD (Agriculture, Farming and Rural Development). Features such that powerful querying and reporting are now available to authorized users. We undertook a small pilot project to investigate the possibility of integrating the prototype institutional CKAN repository with such a bespoke system.

One of the attractive features of CKAN is its powerful data harvesting facilities. Provided by the ckanext-harvester family of extensions, and originally designed to retrieve data sets (and associated metadata) from other CKAN services, the extension is easily modified for the retrieval of data from other sources. A useful collection of customized harvesters is available at https://github.com/okfn/ckanext-pdeu, which were used as the starting point for our own NUECrops harvester.

It took just two days to implement the new harvesting extension, with most of that time spent getting up to speed on the mechanism itself rather than writing code. It is expected that a skilled developer familiar with CKAN's internals could customize and deploy a new harvester in just a few hours. Automatically attaching meta-data to the harvested objects is a similarly simple task, and is performed during the retrieval and import process. For the pilot, the imported data was simply tagged with a retrieval time, file name, source url and project name. However, it would be perfectly possible to include other metadata during the import process, perhaps derived from either the data itself, or another source. For instance, the custom harvester examples in the ckanext-pdeu repository all parse the initial data source list to extract pertinent information.

This work demonstrated the ease with which CKAN can be customized to automatically retrieve data from a “non-standard” source. Further investigation will be required to determine how scalable this approach would be. Maintenance of several hundred harvesters, even if individually simple, is likely to be a non-trivial task.

General observations

- At the start of the project, members of the team had only a passing familiarity with Python web development. It is testament to the design of both CKAN and the Python language in general the development work undertaken has been completed successfully. However, any larger, long term deployment of CKAN would necessitate the development of such skills within the support team.

- The decision to deploy a development version of CKAN brought with it a number of challenges, although in hindsight it was probably the correct one. It is unlikely we would have successfully integrated Shibboleth authentication with the release version available at the time. CKAN 2.0 has recently been released as the current "release" version, and will be installed as soon as possible.

- The work undertaken so far has only scratched the surface of what might be possible with CKAN. Of potential interest would be using CKAN's "Datastore" backend to provide useful research data processing, extraction and visualization, rather than simply acting as a repository. This might be relevant to situations like the NUE-Crops database, where CKAN might provide the advanced functionality already developed within AFRD. We are aware that other JISC projects, primarily the Orbital project at Lincoln are investigating this functionality.

Ben Allen
June 2013