Revision History			
Date	Version	Description	Author
11/10/15	1.0	Draft Version	Vitezslav Kriz
12/10/15	1.2	Conclusion	Vitezslav Kriz

CKAN

1 Open data

CKAN main aim is in publishing data, which should be open. It is important start with definition of Open Data and key information.

“Open data is data that can be freely used, shared and built-on by anyone, anywhere, for any purpose”[1]

There are 2 important elements to openness:

- Legal – Data is published under Open License
- Technical – data should be machine readable, **PDF** (or printed papers) makes information extremely difficult to work with.

2 Architecture

To understand how CKAN works is really important to know basic structures.

2.1 Organization

Organization is structure responsible for publishing Datasets. Organization can have members with different roles: Admin, Editor, Member. Member can view private datasets, Editor can edit and publish Datasets, and Admin can manage Members, and Add, Edit and Delete Datasets.

2.2 User

User have to be logged in for editing, viewing is normally allowed to everyone (By CKAN terminology user named visitor). User also can have rights to Creating new Datasets, Organisation, Groups, … depending on server settings. User can be member of organization.

2.3 Dataset

Every data in CKAN is published as Dataset. Dataset contains metadata: title, publisher, format of data, license, tags … and dataset than contains resources. Dataset is unit used for searching. User can search only for whole dataset not for resources in it. Old name for Dataset in CKAN version less than 1.4 was Data Package. This naming is also used in some API call.

1 http://blog.okfn.org/2013/10/03/defining-open-data/

Dataset is normally owned by an Organization, but is also possible to change server settings and Datasets could be unowned. User can make Dataset after this change, but Dataset will not be owned by User, and in the system will not be any visible connection between User and Dataset. Unowned dataset can be edited by any logged user.

2.3.1 Resource

Resource is actual data in dataset. CKAN doesn't care about format of data. Of course user can specify it, and there is also option that CKAN can guess it, but CKAN doesn't provide searching in Resources. Document can be stored internally in CKAN server or it could be only a URI to actual data (web page). Dataset can contains more than one Resource.

2.4 Group

Group is collection of Datasets. Each Dataset can belong to any numbers of groups. Group can have also Members and Admin, but no Editors, because groups are not responsible for publsihing document that responsibility belong to Organization.

3 REST API

Representational state transfer (REST) is software architecture used for world wide web. It communicates over HTTP protocol and used same commands (GET, POST, PUT, DELETE, …). CKAN API used REST and for CKAN and this technology connection works well.

3.1 Authorization

Call to API have to be authorized with API key. API key is unique for every user. Only calls which are allowed also for visitors can be made without API key.

3.2 Integration to Python

Connecting to CKAN API can be realized through basic HTTP (REST) request or with some specialized library. Library ckanclient is now deprecated and new suitable library is ckanapi. It provides only a little encapsulation of REST.

4 Installing

Installing of CKAN is not easy task. Documentation describe three ways, installing from source, installing from package and use docker image. Although docker image should be easiest way, official distributed docker file doesn't work since May 2015. Another docker solution is project datacats[2]. The safest way is to install it from package, but this only works on Ubuntu 12.04 LTS.

5 Usage in our system Σ

CKAN is most useful in publishing data with metadata and searching in them. Some extension of CKAN also includes possibility to discuss about Datasets, but this is not really usable for our

2 https://www.datacats.com/

purpose. We can take the best of CKAN publishing, and make our application client to CKAN database.

5.1 Briefly description

Our system has 5 roles: Student, Recruiter, Owner, Moderator, Administrator. Students publish their work with some Keywords about their skills, knowledge and experiences. Recruiter publish Job Offer also with same kind of Keywords. Recruiter can contact Student and Student can reply. Owner is creator of Company (In other our document is also Organization, but in this document I use Company to keep difference between Organization with CKAN terminology). He is responsible for manage Recruiters.

5.2 Possible mappings to CKAN

Skills, Knowledge and Experiences is reduced into one group Tags. Tags can be assigned only to Dataset.

Company (Organization) can be mapped directly to Organization in CKAN meaning. Owner is Admin of Organization and Editors are Recruiters.

Student profile should have possibility to adding Tags. Student can be only one Dataset with a lot of Tags and a lot of Resources for prove these Tags. Resource can be School Work, Thesis, Link to Course Sylabus, etc. This approach mix information together in one datasets. Recruiter cannot distinguish which work is prove for which Tag. Better solution is to publish Student Work in separated Datasets. These Dataset should be published under Organization, which contains only one member – Student.

Every Company is Organization and also every Student is Organization, but we need to distinguish between them.

Storing contact details of users should be outside CKAN, we don't want to make this information public. There is also possibility in CKAN to make Datasets private, but we storing contact information in Datasets is not good idea, because need this information structured. And also communication between students and company should be outside CKAN.

5.3 Possible Problems

5.3.1 User have to register twice.

We build two separate system, one is CKAN and second is our application. Forcing User to register in both Systems is not good idea, and really decreasing usability. Sysadmin of CKAN can create new CKAN user also via CKAN API. CKAN can be also configured to disallow registration via website.

5.3.2 How to get user's API KEY

Every user has own API key, which can be obtained (according official documentation) only in CKAN website. But sysadmin has also rights to get all users in one list via API, he can also search for user. This user information also contains API key. And also creating user via API returns user

information via API.

5.3.3 Distinguishing between Students and Job Offers

Everything is Dataset, and we need to distinguish between them. We can use two public groups to sort this two different types of information.

5.3.4 CKAN can change structure

If new version of CKAN change structured of basic elements or API, we need to change our application.

5.3.5 Installation problems

Installation was describe into chapter 4. It can be difficult to install CKAN to some systems. In our project we are not responsible for deploying CKAN. And if administrator of Server cannot install CKAN, we can use some hosted application.

6 Benefits of using CKAN instead of SQL

Instead of CKAN we can also use some DBMS with SQL language(MySQL, PostgreSQL, Oracle). Important question is what can we do better or easier with CKAN?

CKAN provide storage mechanism, but this can be easily be managed with links to other public storage or with running our own storage. With dataset CKAN store Tags, which is really important for our purpose, and without CKAN we need to implement this. And also CKAN have search engine for Datasets, which we can use. Implementing our search engine take a lot of time. But there is also question if CKAN search engine will be suitable for our purpose, because we have no resources to contributing into CKAN.

7 Conclusion

CKAN can be used in our project as storage for school works and for Job Offers. Using CKAN in our project can increase risks. If something went really wrong in the development phase, we can cover CKAN functionality with SQL database. We require 'sysadmin' rights for CKAN on cbuserver, and also we need separate instance of CKAN for our team, in order to not mix the data. There is also possibility to pay for our own hosted version.