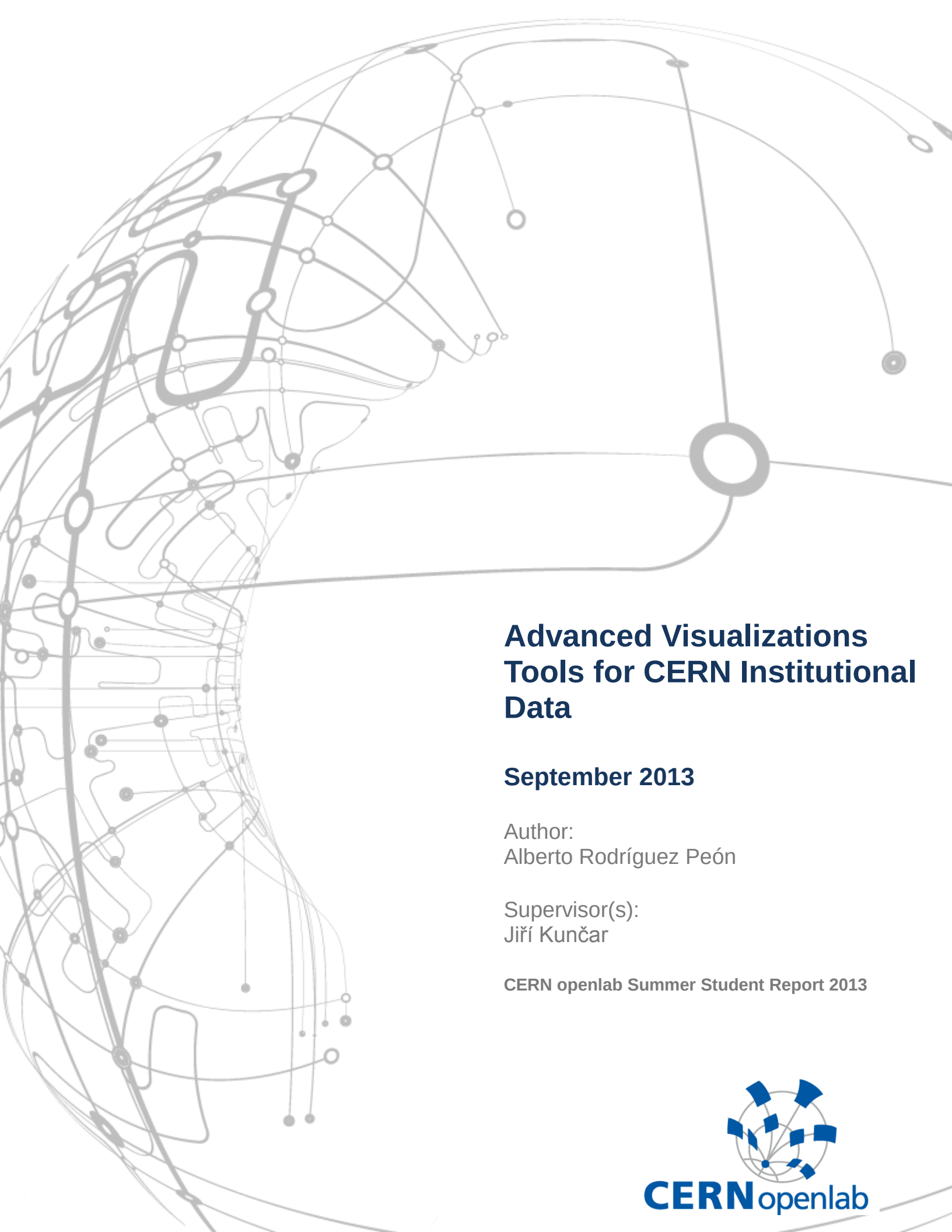

Advanced Visualizations Tools for CERN Institutional Data

September 2013

Author:
Alberto Rodríguez Peón

Supervisor(s):
Jiří Kunčar

CERN openlab Summer Student Report 2013

CERN openlab

Project Specification

The aim of this openlab summer student project is to provide intuitive and powerful visualisation tools for key institutional data about CERN, including budgets and contracts.

The project will be done in collaboration with the Open Knowledge Foundation under the framework of CERN's open data policy regarding scientific results from LHC. The student will use the model-view-controller web development framework with Flask/HTML5/jQuery/TwitterBootstrap technologies for the user interface and SQLAlchemy ORM for database persistence.

Abstract

CERN's Open Access Policy says that "*all results of its experimental and theoretical work shall be published or otherwise made generally available*". Following that, CERN has reached a collaboration agreement with the Open Knowledge Foundation in order for CERN to publish and visualize institutional data.

As part of this collaboration, we will develop a module for showing this data in a graphical way in the CERN side and a tool in the Open Knowledge Foundation site for automatizing the input of data.

Table of Contents

1 Introduction

1.1 Invenio

Invenio software is a web-based digital library able to handle several millions of records. It was originally developed at CERN to run the CERN Document Server. There are several running installations like Zenodo, INSPIRE and Arxiv. It provides features to organize and manage document repositories of large size, like storing, classification, indexing and publication.

The kind of documents that can be stored in Invenio instances vary widely. For example, one can find research papers and books, but also photos, videos, datasets, audio files and more, along the different instances of Invenio.

Because of this, we believe that providing advanced visualization tools will allow us to represent data in a more intuitive and meaningful way.

1.2 Open Knowledge Foundation

The Open Knowledge Foundation (OKFN) is a non-profit organization committed to promoting open data and open content, like government data, publicly funded research and public domain cultural content.

This project has been done in partnership with the OKFN, following CERN's open data policy regarding scientific results from LHC, so it is important that the data we use for our visualizations can be accessible and interpretable. For that, we will be following the standard specified by Open Data Protocols to share the metadata of each dataset in order to be easily harvested from outside of Invenio.

As part of this internship, a RESTful API for loading datasets has been also developed on the OKFN side (specifically, in the OpenSpending platform) in order to be able to harvest data dynamically and without filling a form.

1.3 Tools

For adding data visualization to the Invenio ecosystem, we have used two JavaScript libraries developed by the Open Knowledge Foundation.

1.3.1 Recline.js

Recline.js is a pure JavaScript library that allows creating different kinds of visualizations from several sources of data, like CSV, Google Docs spreadsheets and SOLR. Recline.js unifies different visualization libraries by adding a new layer on top of them. It provides the following visualizations:

- Grid (based on SlickGrid)
- Graph (based on Flot)

- Map (based on Leaflet)
- Timeline (based on Verite Timeline)

In this project, we have developed Grid, Graph and Map views using Recline.js.

1.3.2 BubbleTree

BubbleTree is a complement to the views offered by Recline.js, as it allows representing hierarchical data instead of tabular rows.

In a Bubble Tree, each element can be represented as a node. By clicking in the nodes, you can move up or down in the tree structure and the selected node's children are visible.

This library has been used in several OKFN projects, like OpenSpending and WhereDoesMyMoneyGo?

2 Visualizations

2.1 Grid

The Grid view is the simplest visualization. It represents tabular data in the form of a table that can be sorted by its columns and can be reordered.

Name	*Users by Nationality*	Users by Nation of Insitute	State
Italy	1760	1417	MB
Germany	1259	1318	MB
Russia	982	859	OB
USA	961	1749	OB
France	866	908	MB
United Kingdom	685	784	MB
Spain	380	363	MB
China	270	115	OTH
Poland	264	206	MB
Switzerland	255	397	MB
Japan	254	225	OB
India	215	134	OB
Czech Republic	202	202	MB
Greece	173	105	MB
Netherlands	169	186	MB

Figure 1 Grid view of CERN Users by country

Although it can be created as a standalone visualization, the Grid works better along other visualization like the Graph or the Map.

2.2 Graph

The Graph view is ideal for tabular data, especially if we combine it with the Grid view. This view supports different kind of charts and scales that can be changed on the fly.

For example, the chart in *Figure 2* is a column graph that represents the number of workers at CERN per year in two different categories: Staff and No-Staff (Fellows, Associates, Students, Users and Apprentices

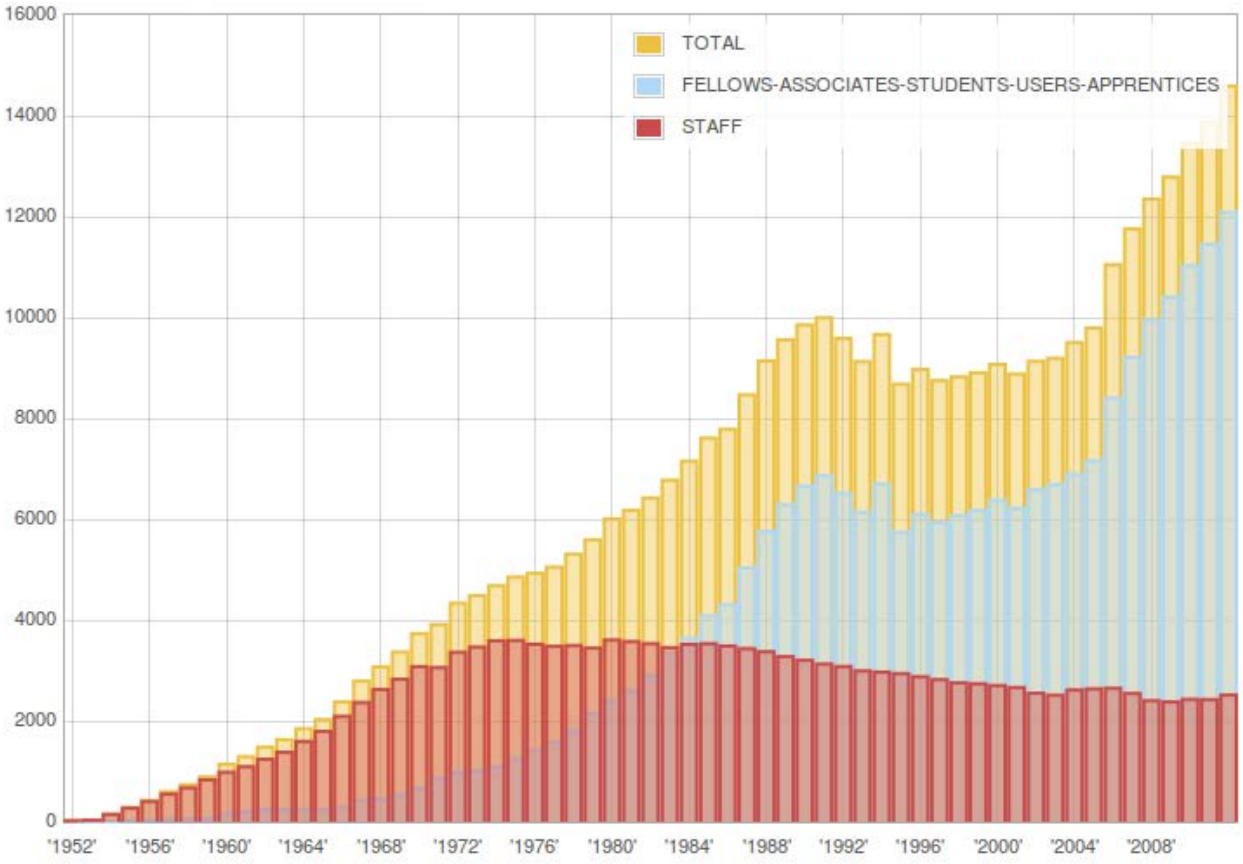

Figure 2 Column graph view of CERN Staff

This chart can also be represented as lines and points instead of columns, perhaps more appropriate for measuring how the data changes, as we can see in *Figure 3*.

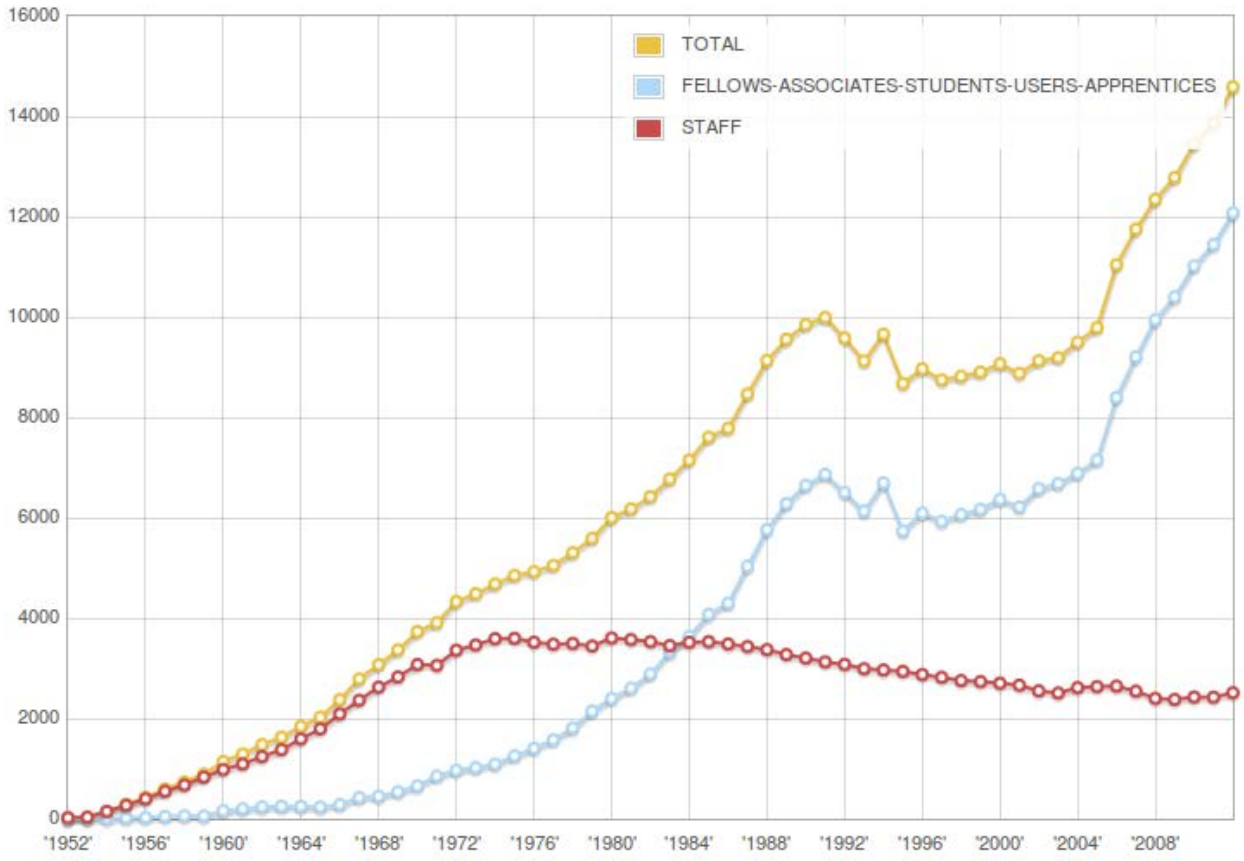

Figure 3 Lines and point graph view of CERN staff

In some cases, the values of the data can change a lot over the chart and using a linear scale is not very useful.

For example, *Figure 4* and *Figure 5* represent the same data, Invenio installations in several organizations. The blue columns represent the total number of records and the yellow ones the difference of number of records with last month.

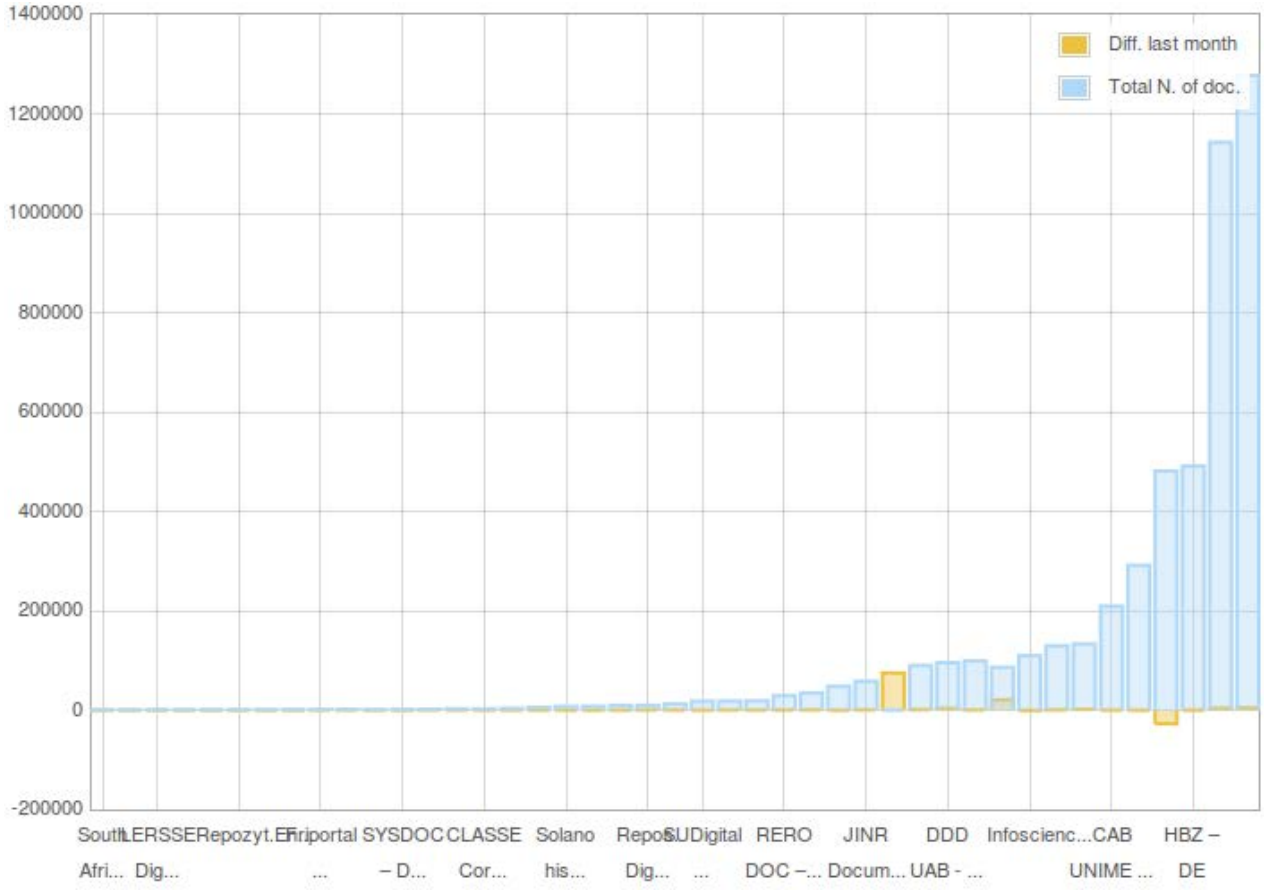

Figure 4 Invenio installations graph with linear scale

As we can see in *Figure 4*, about 50% of the columns are so small that it is impossible to see which values they have. In this case, using a logarithmic scale will fit better.

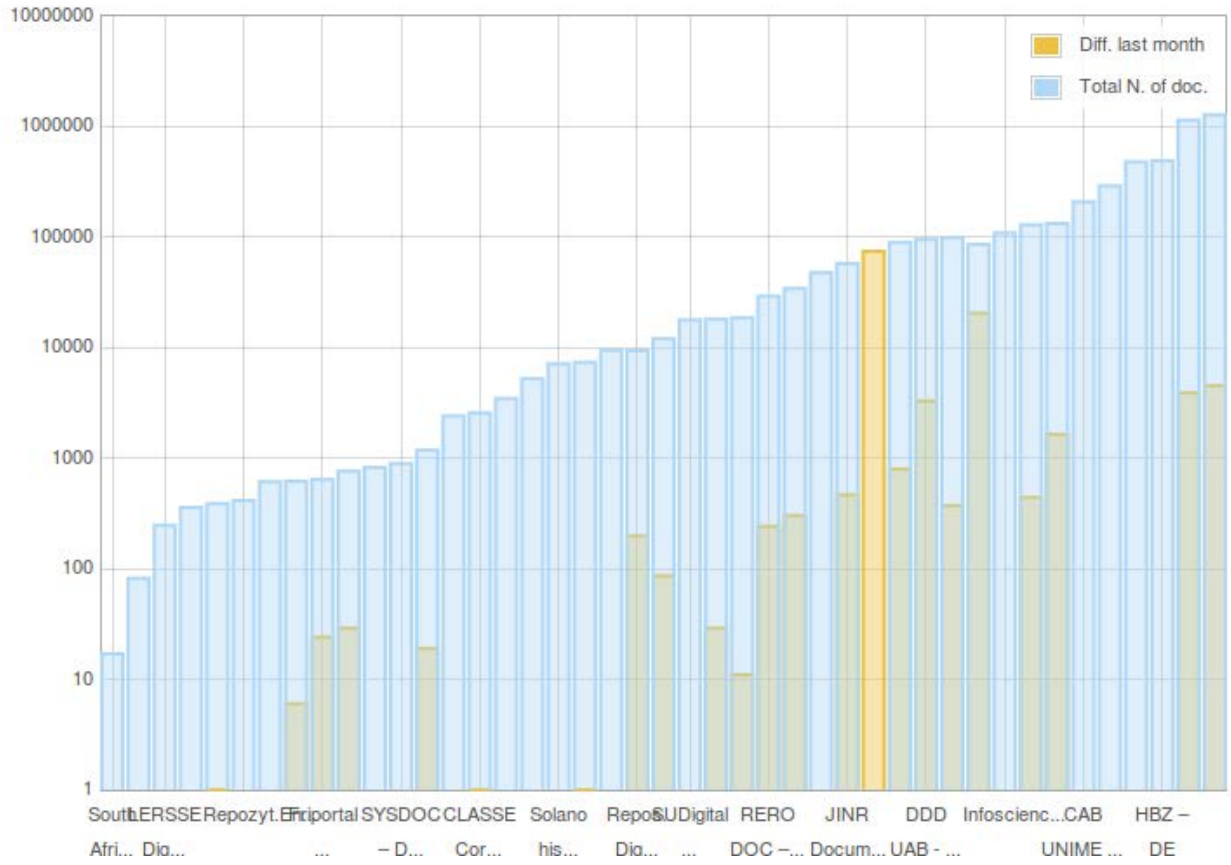

Figure 5 Invenio installations graph with logarithmic scale

The best part of these visualizations is that the graph is not just a static image, but a representation that the user can interact with, as it is possible to sort the columns by any of the values and highlight different results.

For example, in the CERN Staff graph, we can sort descending by the number of Staff members and highlight the first value, the one with the highest value.

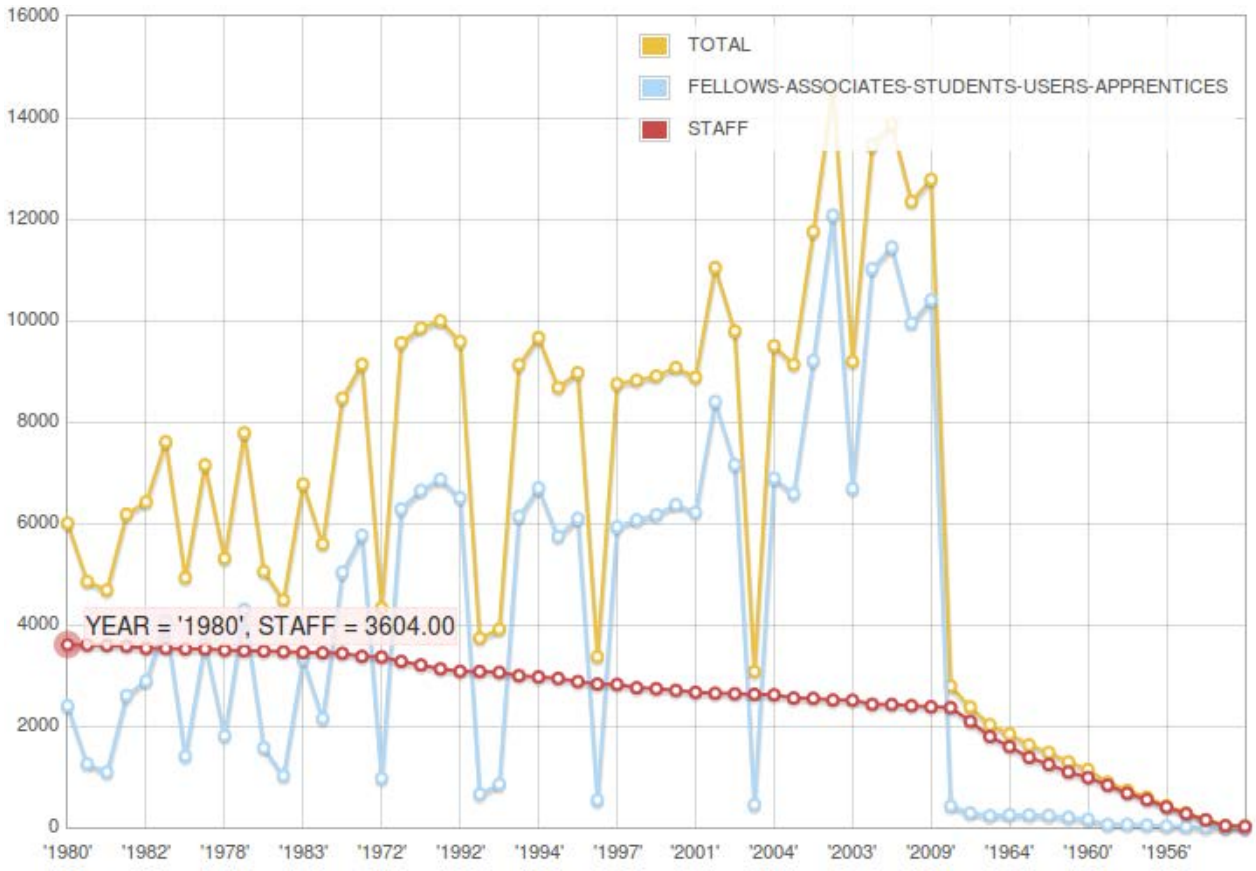

Figure 6 Column graph view of CERN Staff sorted by number of Staff

2.3 Map

The map view can be used with data that has geopoints or the coordinates in Latitude and Longitude.

It is very customizable as it is possible to add custom markers and bubbles to them.

For instance, the following map represents the Meyrin and Prevessin sites at CERN with two custom markers.

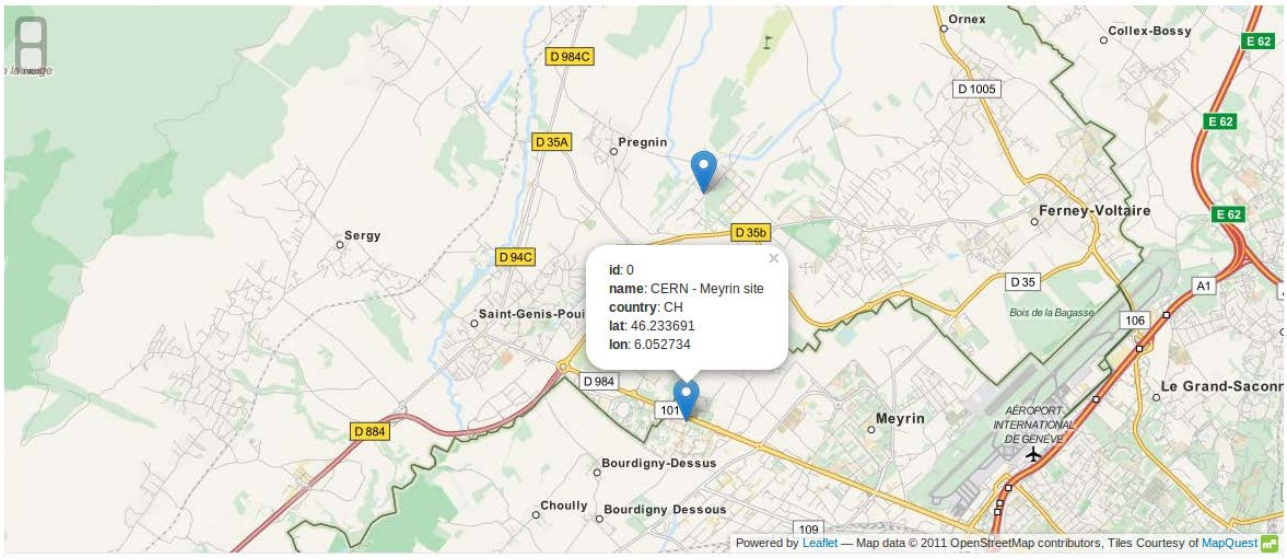

Figure 7 Map view of CERN

2.4 Tree

The Tree view provides a powerful representation for hierarchical and non-tabular data using the Bubble Tree library.

One example of data that can be represented as a Bubble Tree is the collection tree of all documents. The Collection tree aggregates records in different categories and subcategories.

In Invenio's main page, the collection tree is represented in the following way:

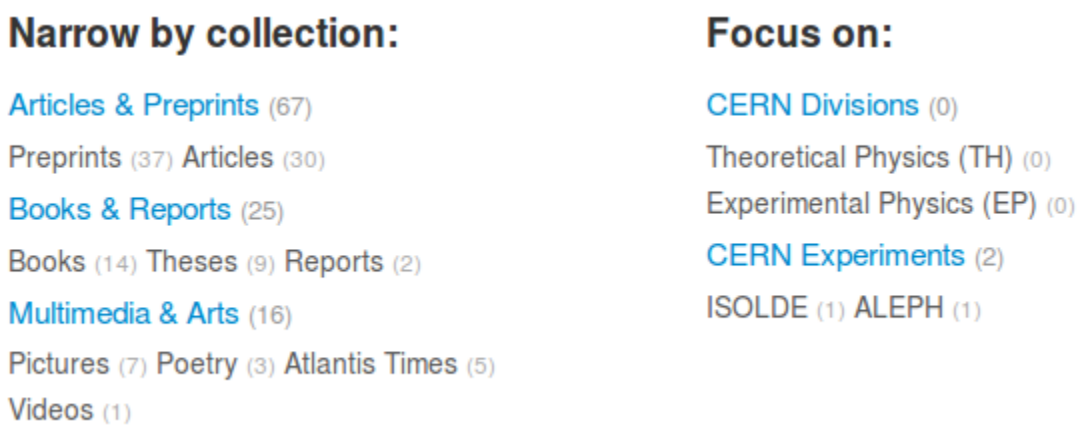

Figure 8 Invenio Collection Tree

We can see that each collection contains children collections (e.g. "Articles & Preprints has 67 records distributed in 37 "Preprints" and 30 "Articles". This kind of data fits perfectly the Bubble Tree visualization.

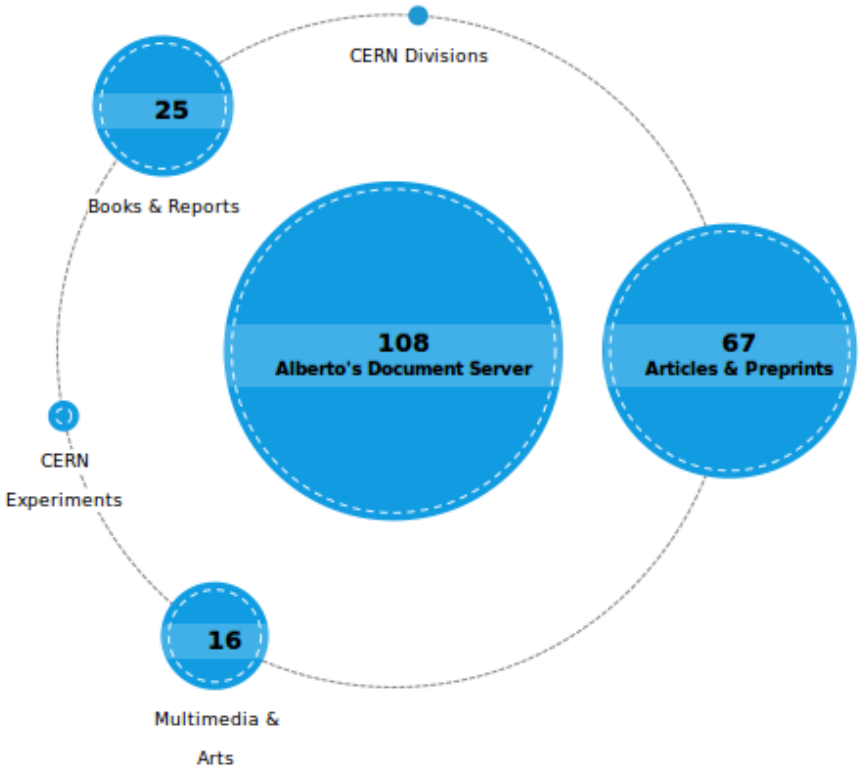

Figure 9 BubbleTree graph of the Collection Tree

By clicking on one node, the focus changes to it and its children are now visible and clickable.

For example, if we click in “Books & Reports”, we can see its three children.

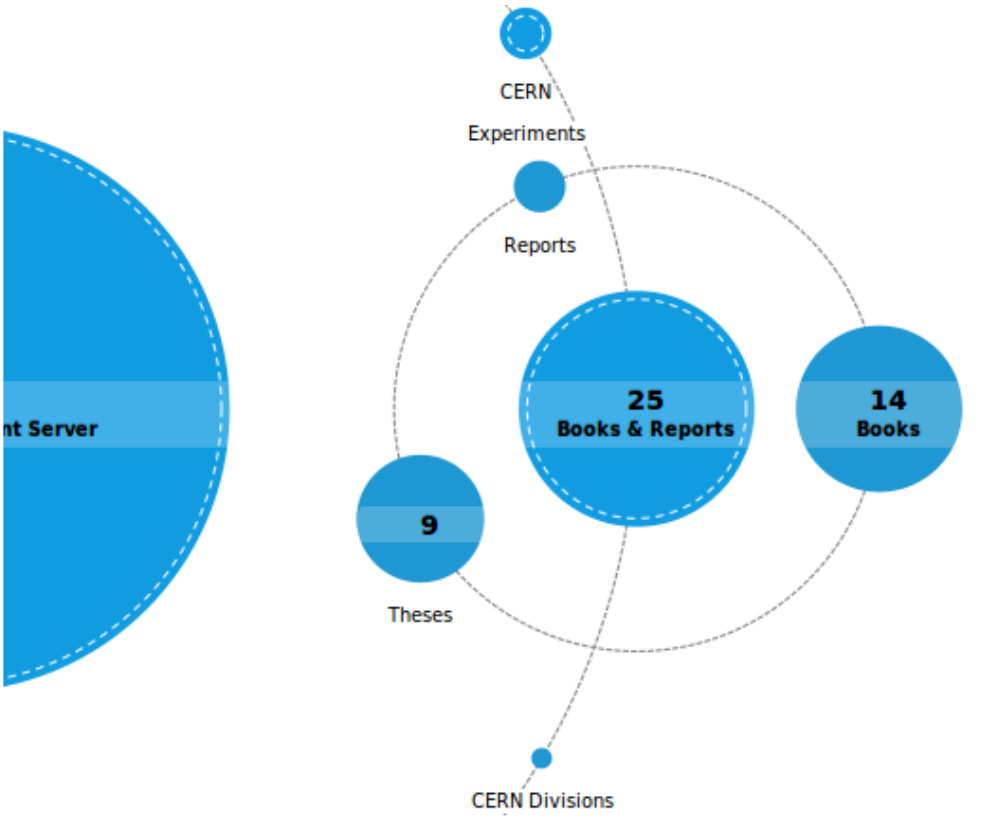

Figure 10 BubbleTree view of the Collection Tree

In addition to browsing through the visualization, we also load, through an AJAX request, some of the records which belong to the selected collection. Therefore, the user can decide if that is the collection he wants to inspect or not.

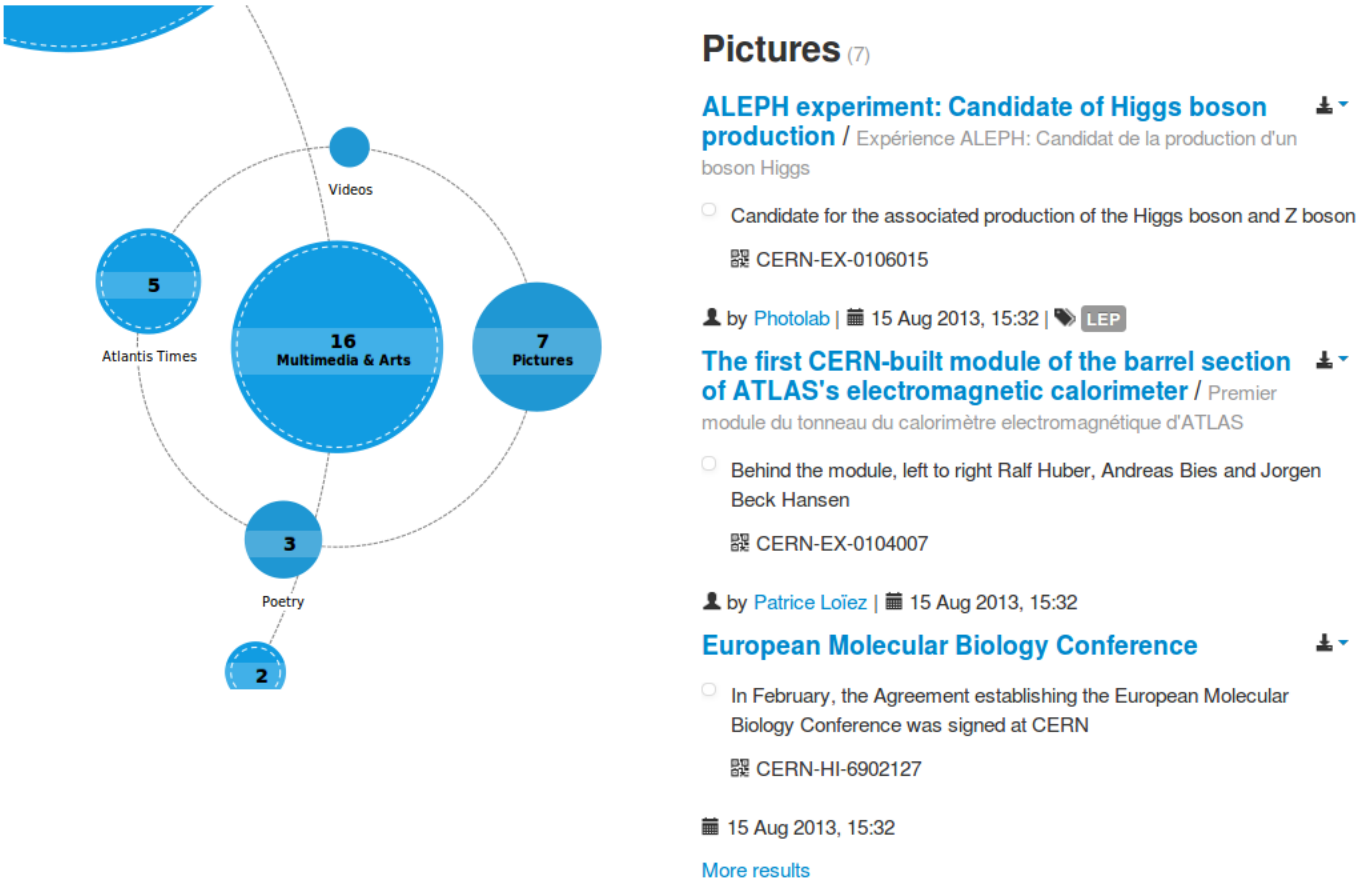

Figure 11 Browsing the collections through the Bubble Tree

2.5 FileManager

As we have seen, we can use the views provided by Recline.js to represent tabular data and the BubbleTree to represent hierarchical data. The problem is, in most cases, the data is not clean and contains fields and information that it is not useful to represent in a visualization.

To solve this, we have created a new module called FileManager which will provide several actions to transform data files.

These are:

- Join: Provides functionality to merge several CSV files which have the same header.
- Cut: Removes columns of a CSV file
- CSV to JSON: Transform a CSV file to JSON

The point of this module is that it is very easy to create a new action and extend the current module. This is very important as we cannot assume the format of a file will be good to the visualization we want.

For doing that, the only step needed is add a plugin and extend the class `FileManagerAction` and create a method `action(self, *args, **kwargs)` which should return the content of the new file.

These new files are cached using Redis so they don't need to be generated anytime and are accessible very fast.

3 OpenSpending's Loading Dataset API

3.1 Problem

The collaboration between CERN and the Open Knowledge Foundation implies the need of sending financial data to OpenSpending, the OKFN project for mapping the financial transactions of public governments.

The way OpenSpending manages the input of new datasets is through a CSV file with raw data and a form (containing information like name, country, language, currency, etc.) filled manually by the user.

The problem is that there is no way of adding automatically a dataset which is what CERN needs in order to push information in an automated way. The OpenSpending's API does not cover the process of introducing data, just searching and visualizing it.

To overpass this issue, we are developing this API in the OpenSpending site.

3.2 Solution

The idea consists of adding a new method to the existing API to replace the manual input of the metadata.

For that, we replace the form with a JSON file containing all the information that we have to provide in order to create a dataset. This works in the same way that the internal tool "ostool" is used in OpenSpending for the installation and setup.

So, technically, the API request should be something like this:

```
POST
/api/new?csv_file=<csv_file_url>&metadata=<json_file_url>
```

This information would be enough to process a dataset and add it to OpenSpending except that there is no way to know which user has made the request and therefore we do not know the creator of the dataset.

Each OpenSpending user has an API key which can be used to identify himself in an API request.

```
POST
/api/new?csv_file=<csv_file_url>&metadata=<json_file_url>&apik
ey=<user_api_key>
```

The problem is that we cannot just put the API key in the request as anybody can intercept it and use it as if it was its own.

3.2.1 Authentication via API key

To solve this issue, we propose a solution using symmetric key as the authorization method (in an Amazon-like way).

For each user, instead of having only a public API key, we create a "secret" one as well. The idea is put the public one in the request and adding a signature, calculated using the 'secret' API key and a cryptographic hash algorithm (in our case MD5)

```
POST
/api/new?csv_file=<csv_file_url>&metadata=<json_file_url>&a
pikey=<user_api_key>&signature=<computed_signature>
```

The signature is calculated concatenating all the params in the request, sorted alphabetically and starting with the 'secret' key.

```
<user_secret_key>api_key<user_api_key>csv_file<csv_file_url
>metadata<json_file_url>
```

The generated string is 'hashed' with MD5 to obtain the signature.

Therefore, in order to validate the user, the server will calculate the signature from the params and compare it against the provided one. If both are equal, the user is authenticated.

The developed API will provide different governments and organizations (including CERN) publish data in OpenSpending without the heavy task of adding it manually.

4 Conclusions

In this project, we have developed the following modules in collaboration between CERN and the Open Knowledge Foundation.

- A new Invenio module for representing data in the form of powerful visualizations, featuring graphs, maps and trees.
- Another Invenio module for manipulating files using custom made plugins, including some actions like cutting, joining and transforming CSV files.

- In the OKFN side, we have design and constructed an API for loading datasets into OpenSpending, the financial data site of the OKFN. This contribution will allow users and organization to publish their data into OpenSpending dynamically and without manual input.