

## OpenBudgets.eu: Fighting Corruption with Fiscal Transparency

Project Number: 645833

Start Date of Project: 01.05.2015

Duration: 30 months

### Deliverable 4.4: OpenBudgets.eu Platform Documentation

Dissemination Level	Public
Due Date of Deliverable	Month 24, 30.04.2017
Actual Submission Date	Month 28, 10.08.2017
Work Package	WP 4, OpenBudgets.eu Requirements, Platform Architecture Integration and Development
Task	T4.2
Type	Report
Approval Status	Final
Version	0.4
Number of Pages	2626
Filename	D4.4 - OpenBudgets.eu Platform Documentation

#### **Abstract:**

This deliverable lists documentations of OBEU components from users' perspective. In total ten different components are documented. For each component, we provide a general description, the installation guide, some screenshots of a running component, and a web page for the source code and more detailed tutorials.

The information in this document reflects only the author's views and the European Community is not liable for any use that may be made of the information contained therein. The information in this document is provided "as is" without guarantee or warranty of any kind, express or implied, including but not limited to the fitness of the information for a particular purpose. The user thereof uses the information at his/ her sole risk and liability.



## History

Version	Date	Reason	Revised by
0.1	23.04.2017	Initialization	Lazaros Ioannidis
0.2	26.06.2017	First version	Tiansi Dong, Lazaros Ioannidis
0.3	25.07.2017	Second version	Lazaros Ioannidis, Tiansi Dong
0.4	04.08.2017	Final version	Fabrizio Orlandi

## Author List

Organisation	Name	Contact Information
OKGR	Lazaros Ioannidis	<a href="mailto:larjohn@gmail.com">larjohn@gmail.com</a>
OKGR	Sotirios Karampatakis	<a href="mailto:karampatakis@okfn.gr">karampatakis@okfn.gr</a>
UBonn	Tiansi Dong	<a href="mailto:tdong@uni-bonn.de">tdong@uni-bonn.de</a>
UEP	Jakub Klímek	<a href="mailto:klimek@ksi.mff.cuni.cz">klimek@ksi.mff.cuni.cz</a>
FhG	Fabrizio Orlandi	<a href="mailto:orlandi@iai.uni-bonn.de">orlandi@iai.uni-bonn.de</a>

# Executive Summary

This deliverable provides a documentation of the OBEU platform and its components from a user perspective. In total, ten different components are documented: (1) Indigo, (2) OBEU Viewer, (3) Data-mining tools, (4) LinkedPipes ETL, (5) Microsite, (6) Participatory Budgeting, (7) Virtuoso triplestore, (8) Rudolf, (9) RDF Browser, (10) RDF alignment . For each component, we provide a general description, the installation guide, some screenshots of a running component, and a web page including the source code and more detailed tutorials. More technical details about the final architecture of the OBEU platform and the functionalities of its components/tools are provided in the report of deliverable D4.5.

## Abbreviations and Acronyms

LOD	Linked Open Data
LP-ETL	LinkedPipes ETL
OS	OpenSpending
OBEU	OpenBudgets.eu
FhG	Fraunhofer IAIS
OKI	Open Knowledge International
OKGR	Open Knowledge Greece
UEP	University of Economics, Prague
RDF	Resource Description Framework

## Table of contents

1	Introduction .....	7
2	Components.....	8
2.1.	Indigo for exploring, visualizing datasets .....	8
2.1.1.	Installation.....	9
2.1.2.	Two backends of Indigo .....	9
2.1.3.	Indigo as the front-end for data analysis and mining.....	10
2.2.	OBEU Viewer .....	11
2.3.	Data-mining tools .....	12
2.3.1.	Installation.....	13
2.3.2.	Data-mining services.....	14
2.4.	LinkedPipes ETL .....	15
2.5.	Microsite.....	17
2.6.	Participatory Tool .....	19
2.7.	Triplestore Virtuoso .....	21
2.8.	RDF Browser.....	21
2.9.	RDF Alignment Tool .....	23
2.10.	Rudolf .....	25
3	Conclusion .....	26

# List of Figures

- Figure 1. The architecture of OBEU platform
- Figure 2. Indigo User Interface
- Figure 3. Indigo front-end for the data-mining task of rule-mining
- Figure 4. OB.eu Viewer for the Bonn Budget 2019
- Figure 5. The treemap of Bonn Budget 2019 dataset
- Figure 6. The architecture of Data Mining Module
- Figure 7. Graphic illustration of Frequency-based Outlier Detection service
- Figure 8. The description of LP-ETL at the Indigo's front-end
- Figure 9. An HTTP get component of Linkedpipes-ETL
- Figure 10. The configuration interface of HTTP get component
- Figure 11. Microsite for viewing and commenting budget data
- Figure 12. The view site of Microsite in the flip mode
- Figure 13. The comment site of Microsite in the flip mode
- Figure 14. The OpenBudgets Participatory Budgeting installed at [www.herokuapp.com](http://www.herokuapp.com)
- Figure 15. Citizens can vote for their favourite budgeting proposals
- Figure 16. A screenshot of the RDF Viewer developed by OKFGR
- Figure 17. The flowchart of the RDF alignment tool developed by OKFGR

# 1 Introduction

In this deliverable, we provide the detailed documentation of the final platform of the OBEU project, which incorporates components of the OpenSpending project<sup>1</sup>. This deliverable complements Deliverable 4.5, where more technical details are provided. The architecture of the OBEU platform is illustrated in Figure 1. We will describe the components from the user's perspective, including how to install them, and detailed descriptions of functionalities plus some screenshots.

---

<sup>1</sup> <https://next.openspending.org/>

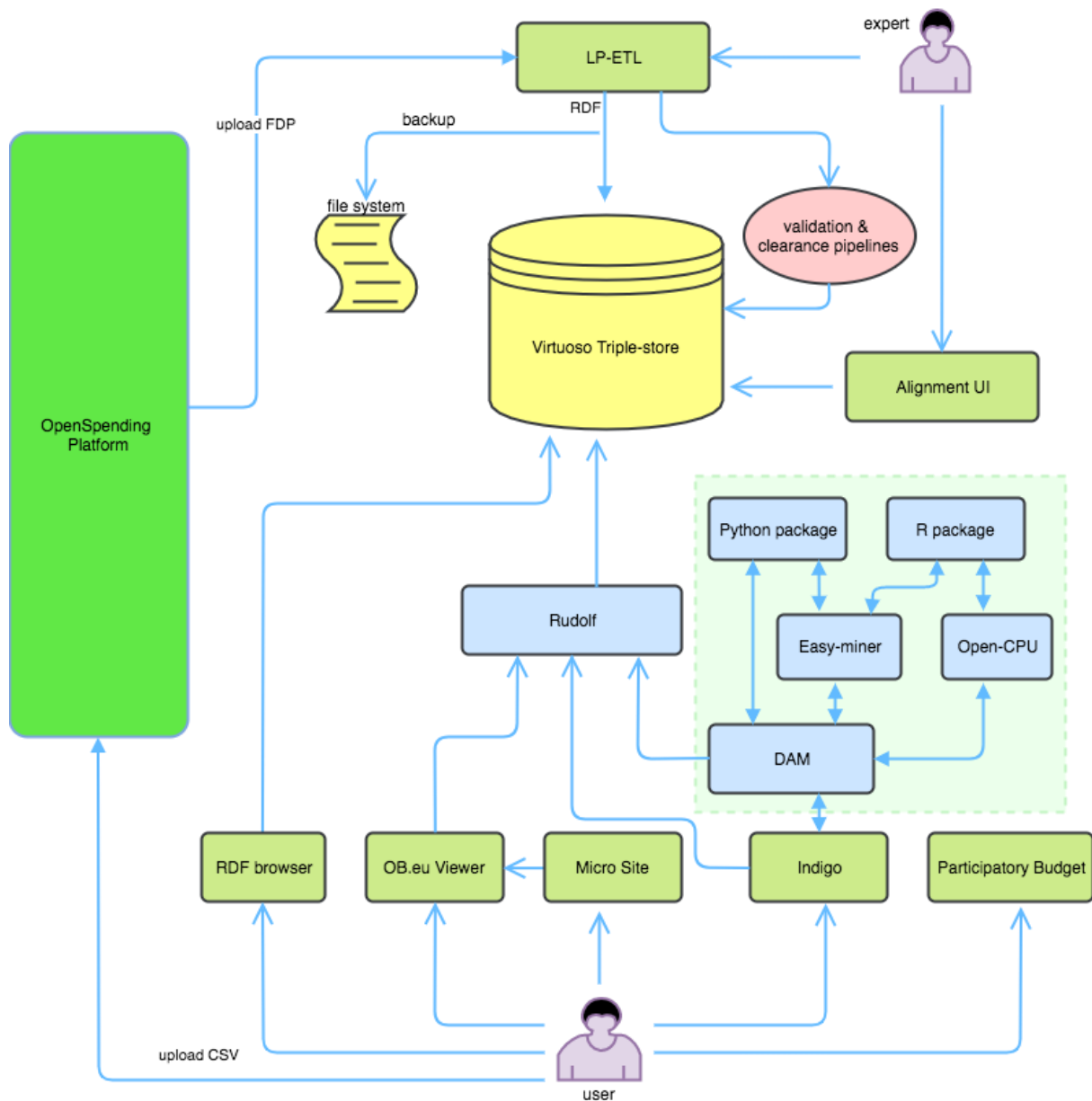


Figure 1. Architecture of the OBEU platform

## 2 Components

### 2.1. Indigo for exploring, visualizing datasets

Indigo is the front-end, browser application that can be used to configure and execute analysis and data mining functions on subsets of fiscal data. It is also used for exploring and



searching for datasets, as illustrated in Figure 2. Its main interface is currently deployed at <http://apps.openbudgets.eu>.

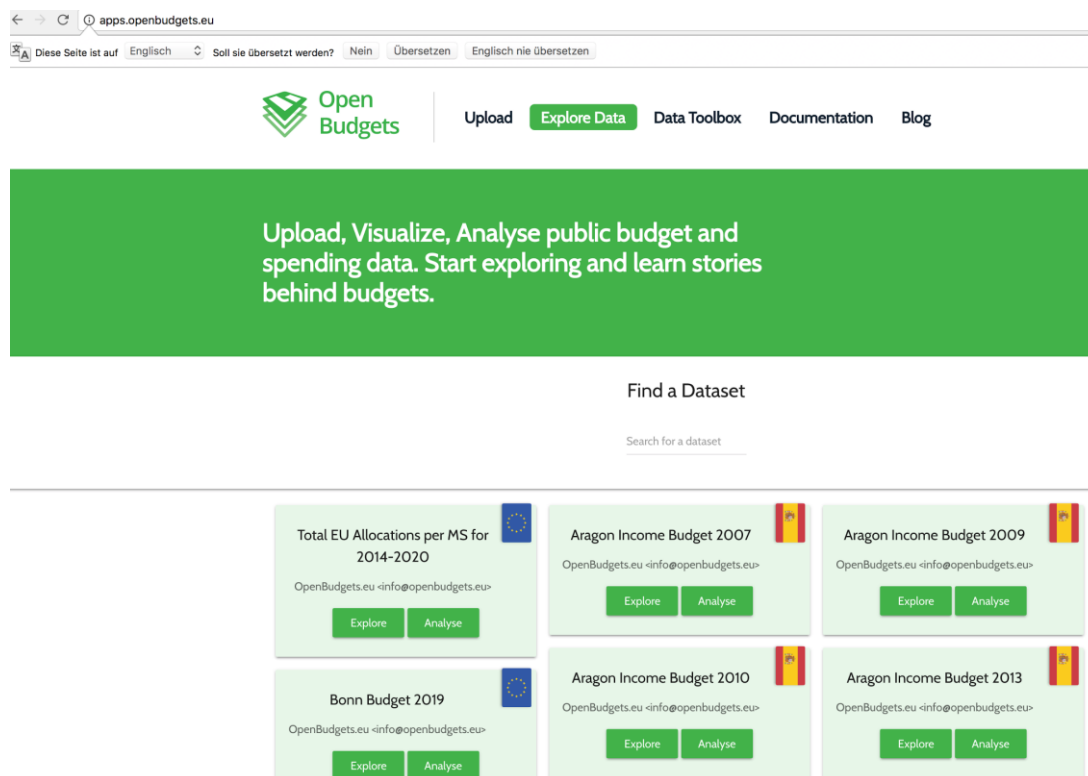


Figure 2. Indigo User Interface

### 2.1.1. Installation

Indigo comes as a part of the OpenBudgets.eu Docker package. To install it separately, one has to download the source and then execute the installation commands, as documented on the README file contained in the sources <https://github.com/okgreece/indigo>.

### 2.1.2. Two backends of Indigo

Indigo combines two backends, the OpenSpending-compatible Rudolf API for data acquisition, and the DAM backend for data analysis and mining. The users first select a dataset from a list that is fetched from the Rudolf API. They can then select an analysis task to execute, again from a list, which now comes from the DAM API. For each task, there can be multiple configuration options. For instance, the outlier detection task provides one configuration that uses the FQR algorithm, and two configurations that use the LOF algorithm, with the latter two being distinct because they are targeted to different data input.

### 2.1.3. Indigo as the front-end for data analysis and mining

In order to perform an expected analysis task on the selected dataset, the users have to specify the input of the algorithm. For example, in Figure 3, a user specifies the ‘preferred label’ as the consequent column for rule-mining. The rule-mining result is shown on the right side.

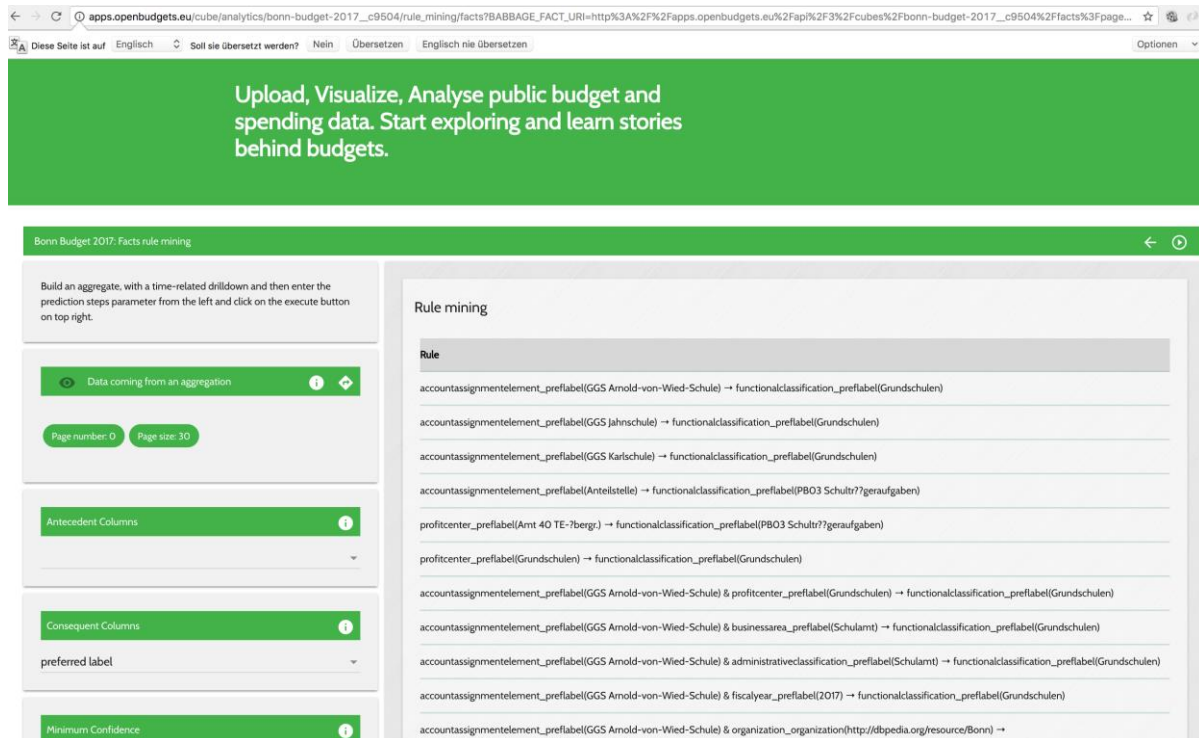


Figure 3. Indigo front-end for the data-mining task of rule-mining.

Generally, the input contains one or more of the following items:

- A data subset builder
- A dimension/measure reference
- A parameter

The data subset builder is a form that allows the user to build a query against the Rudolf API, either an aggregate or a facts table. The query can be optionally instantly executed to preview the fetched data, but it is submitted to the DAM backend as a URI, so that the data are not transferred between the two applications.

The dimension/measure reference is a selection menu that is used to indicate a dimension of the dataset that should be used during the algorithm execution, for instance as an aggregator or to extract amounts from.

After selecting the inputs, the user clicks the execution button and waits for the task to complete. When the task is completed, Indigo receives the algorithm execution results and builds the appropriate visualizations. Most of the visualizations can be shared as a link and be further embedded to other websites. For more details on the data mining algorithms see Deliverable 2.4.

OBEU Viewer is the main front-end for visualizations of the OpenBudgets.eu data store, as illustrated in Figure 4. The user is redirected to OBEU Viewer, after selecting the dataset from a dataset browser, which in the case of the OpenBudgets.eu platform is Indigo’s dataset browser. The OBEU Viewer is a customisation of the OpenSpending Viewer<sup>2</sup>, tailored for the visualisation of OBEU RDF datasets. The current deployment of the OBEU Viewer is available at <http://apps.openbudgets.eu/viewer/>. In the OBEU Viewer, the users can directly preview the facts contained in the dataset. The main functionality of the application is contained in visualizing aggregates.

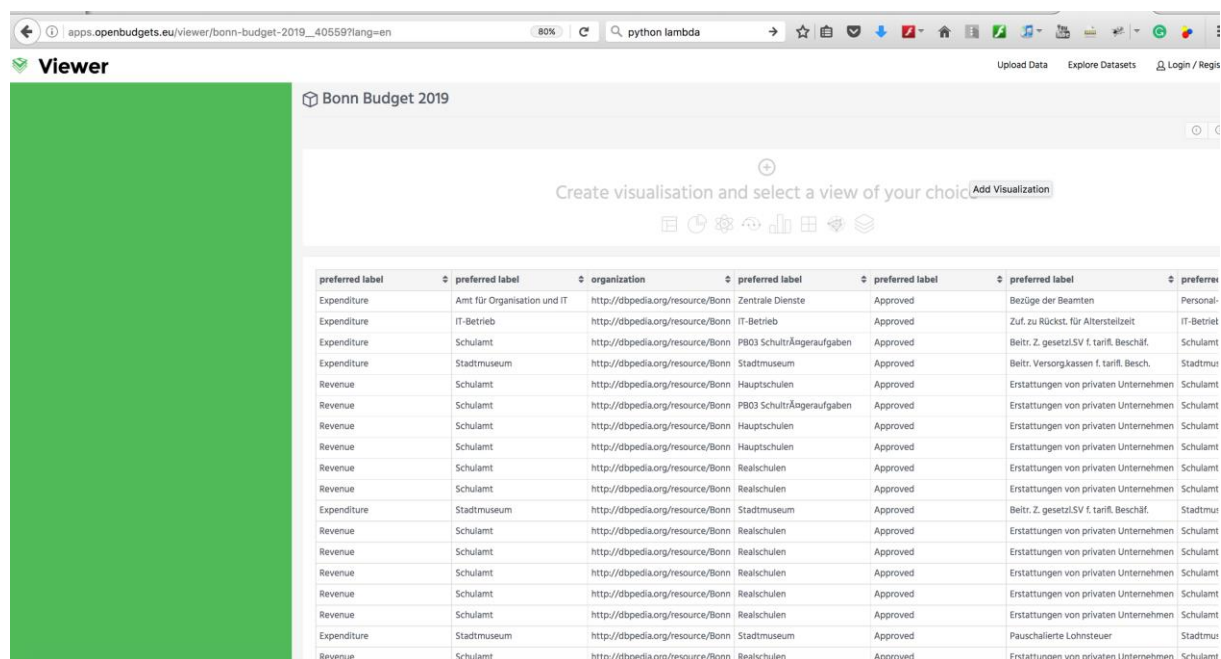


Figure 4. OB.eu Viewer for the Bonn Budget 2019

<sup>2</sup> <http://docs.openspending.org/en/latest/developers/viewer/>

Users select a visualization in the main pane of the application. A dimension is automatically selected to aggregate data. The aggregated data are visualized in the main pane. The user can select other aggregation dimension, filter the data based on a dimension, and sort data, which is useful on table-like visualizations. The available visualizations are: Bar chart, Line chart, Treemap, Geomap, Pie chart, Pivot table, Radar chart. Figure 5 illustrates a treemap of the Bonn Budget 2019.

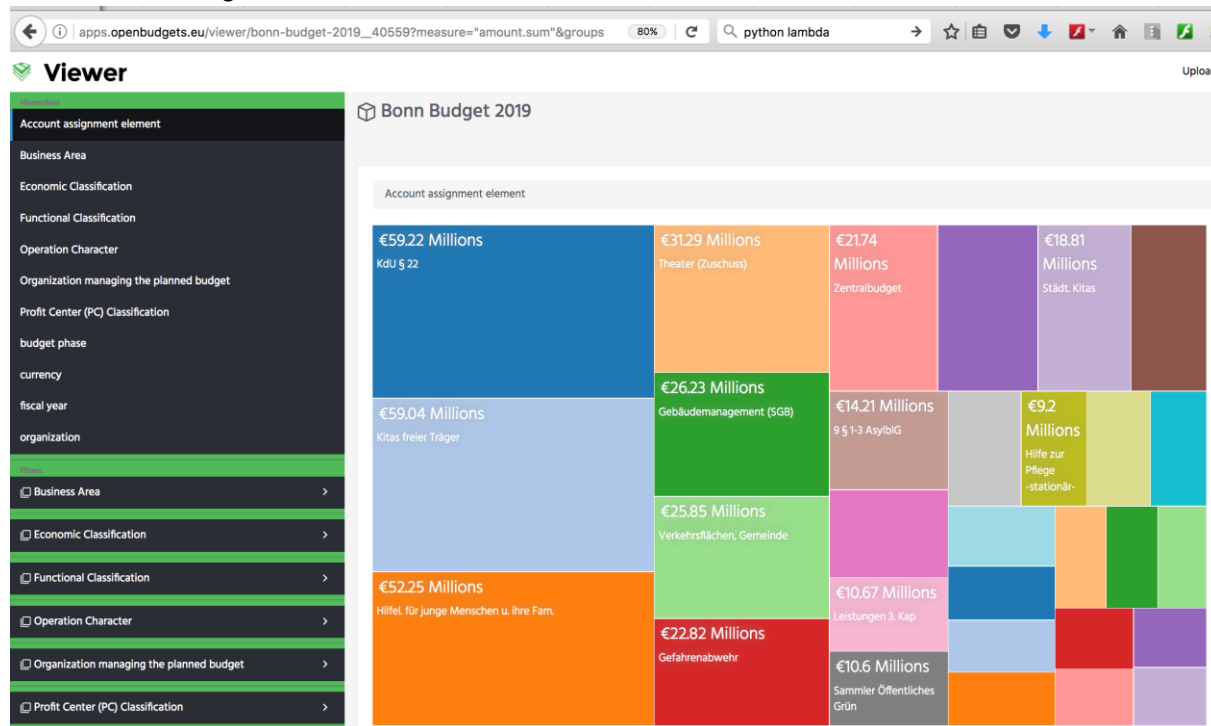


Figure 5. The treemap of Bonn Budget 2019 dataset

## 2.3. Data-mining tools

When users select a dataset at Indigo, and click the 'Analyse' button, Indigo will provide a data-mining interface, where users can specify data-mining function and set parameters. Indigo sends these information into DAM. DAM does the data pre-processing, creates concrete data-mining job, sends the job to particular data-mining server, and receives data-mining results. The architecture is illustrated in Figure 6. More information about the data mining tools architecture and functionality is provided in Deliverable D2.5.

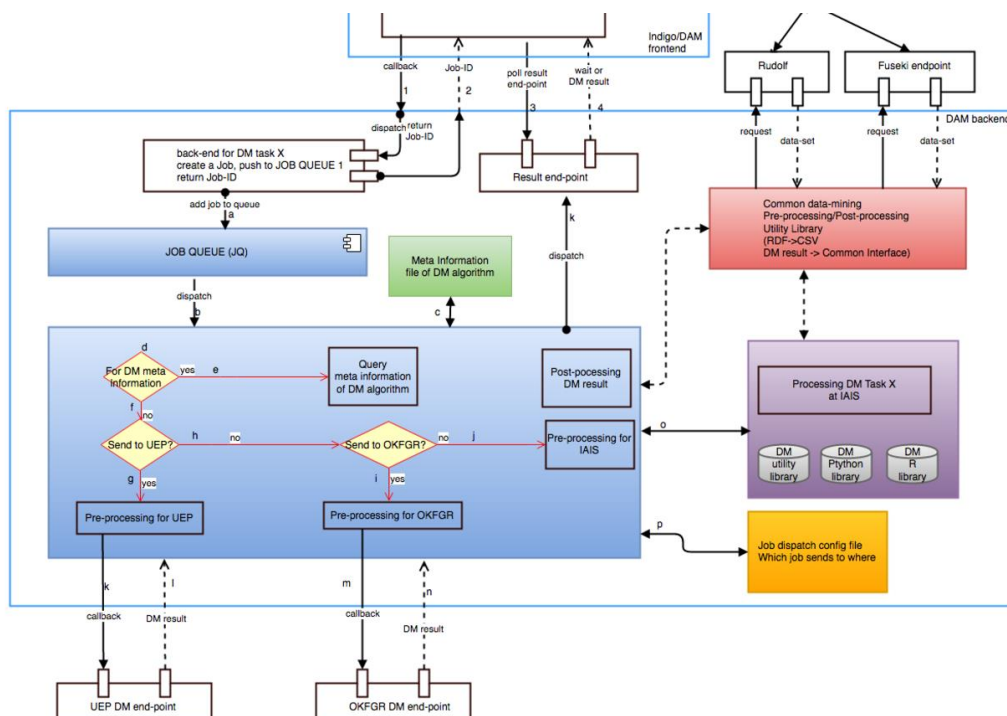


Figure 6. The architecture of DAM, the Data Mining Module

### 2.3.1. Installation

The installation of the data-mining base module on a local Ubuntu platform is described in the [README.md](#) file at [https://github.com/openbudgets/DAM/tree/staging\\_indigo](https://github.com/openbudgets/DAM/tree/staging_indigo).

Step 1. First clone it, by typing

```
$ git clone https://github.com/openbudgets/DAM
```

Step 2. check out the staging\_indigo branch, by typing

```
$ git checkout staging_indigo
```

Step 3. go to the DAM directory by typing

```
$ cd DAM
```

Step 4. install necessary libraries, such as git, redis-server, postgresql, python, and virtual environment. All these are realized by typing

```
$ make pre
```

Step 5. activate virtual environment by typing

```
$ source env/bin/activate
```

Step 6. install the basic DAM module, by typing

```
$ make dam
```

To start the data-mining server, we need to open three terminals, each in the `DAM` directory, and activate virtual environment. Type `$ python3 manage.py runserver` in the first

terminal. Then type `$ redis-server` in the second terminal. Type `$ python3 worker.py` in the third terminal.

Communicating with the UEP data-mining server is carried out by **uep\_dm** module<sup>3</sup>; Communicating with the OKFGR data-mining server is carried out by **okfgr\_dm** module<sup>4</sup>.

### 2.3.2. Data-mining services

Six data-mining services are provided: (1) Descriptive statistics, (2) Time series analysis and predictions, (3) Clustering and similarity learning, (4) Comparative analysis, (5) Rule and pattern mining, (6) Outlier detection. Figure 7 is an illustration performed with Indigo of the data-mining result of the Frequency-based Outlier Detection service provided by UEP server.

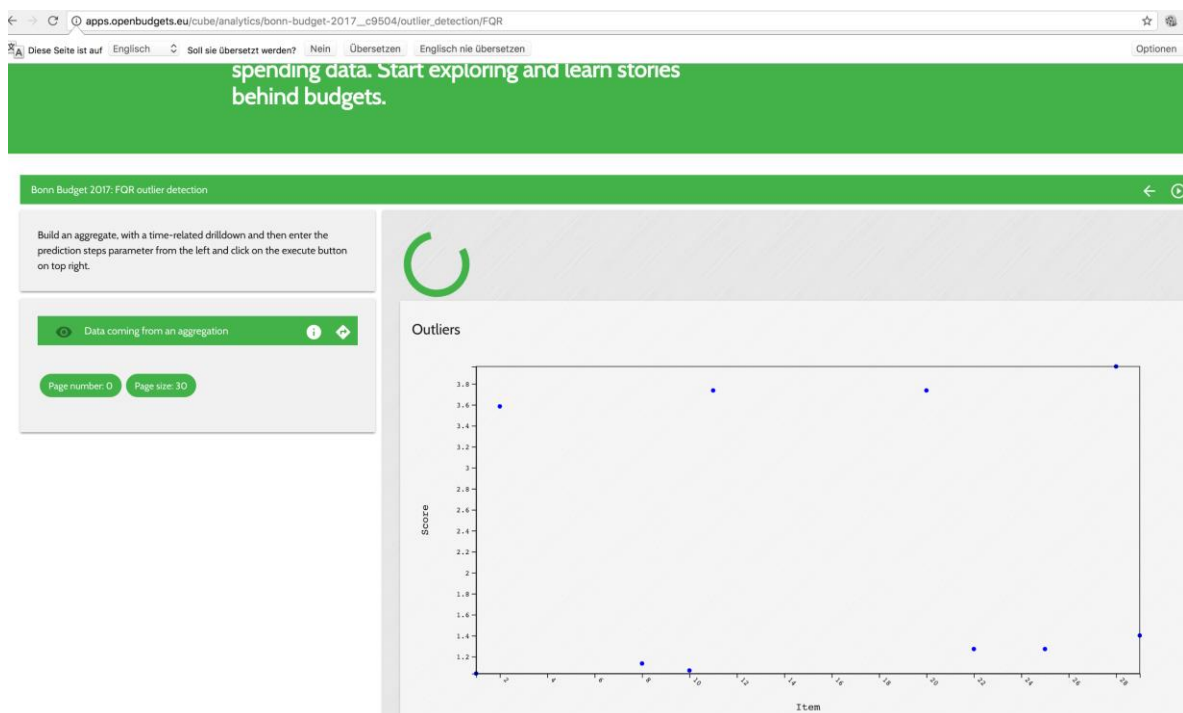


Figure 7. Graphic illustration of Frequency-based Outlier Detection service

<sup>3</sup> [https://github.com/openbudgets/uep\\_dm](https://github.com/openbudgets/uep_dm)

<sup>4</sup> [https://github.com/openbudgets/okfgr\\_dm](https://github.com/openbudgets/okfgr_dm)

## 2.4. LinkedPipes ETL

LinkedPipes ETL (LP-ETL) is a generic Linked Data based ETL tool for data transformation and publishing which is targeted at data scientists who are familiar with RDF and SPARQL. Mainly developed by UEP, its main website is located at <https://etl.linkedpipes.com/>.

LP-ETL configures data-pipelines to do data transformations. Each data-pipeline represents individual data processing tasks, consisting of interconnected components, where each component represents a configurable data transformation, and the interconnection represents the data flow among the components. Indigo provides an introduction of LP-ETL, as illustrated in Figure 8. LP-ETL has an independent user interface.

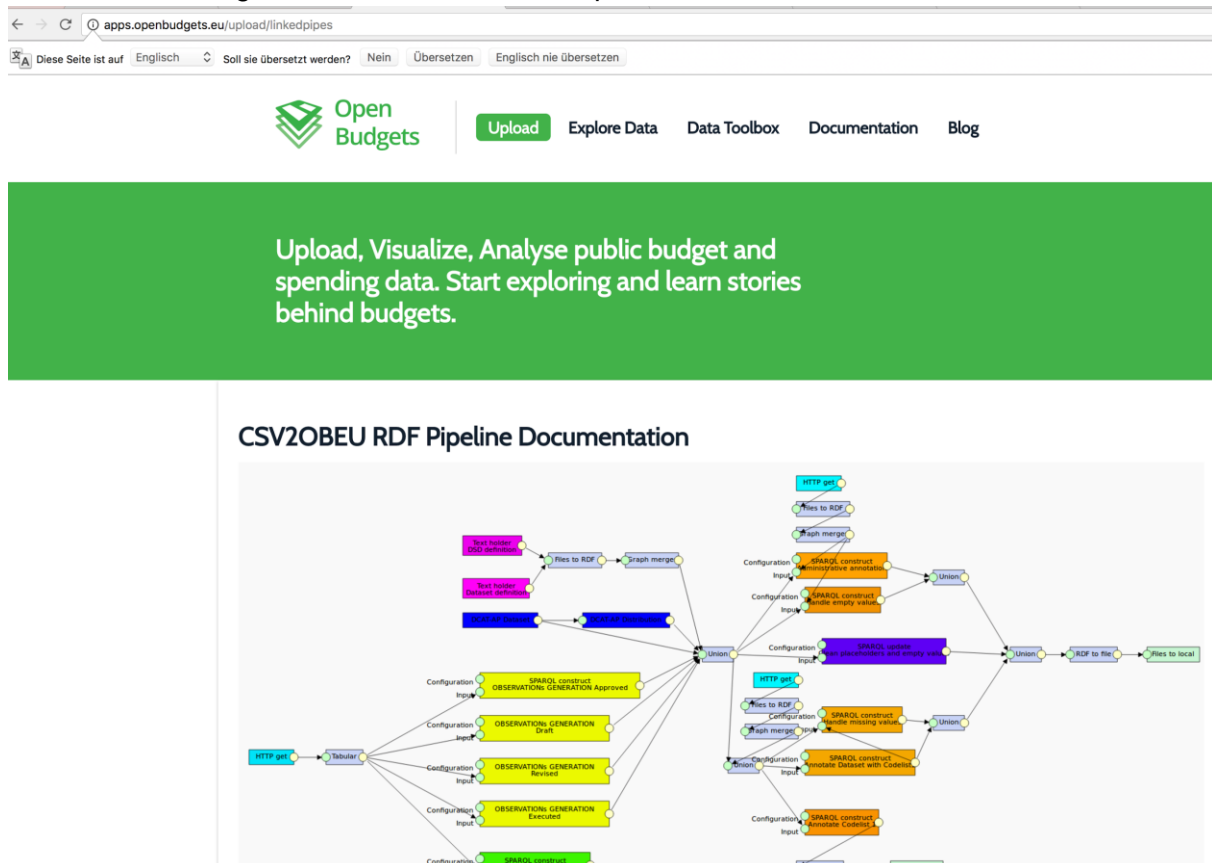


Figure 8. The description of LP-ETL on the OBEU site with additional documentation for the CSV2OBEU pipeline

LP-ETL requires Java 8, Git, Maven and Node.js. Then it can be installed in 4 simple steps:

1. Clone the source files from the GitHub repository at <https://github.com/linkedinpipes/etl>
2. Run Maven's `mvn install` command
3. Optionally adjust the `configuration.properties` file in the `deploy` subfolder
4. Run the four main components of LP-ETL using scripts in the `deploy` subfolder:  
`executor.sh`, `storage.sh`, `executor-monitor.sh` and `frontend.sh`.

Then, LP-ETL is available at <http://localhost:8080>.

The user can start with creating his first pipeline using the “plus” button and then by adding, e.g. the HTTP get component, as shown in Figure 9

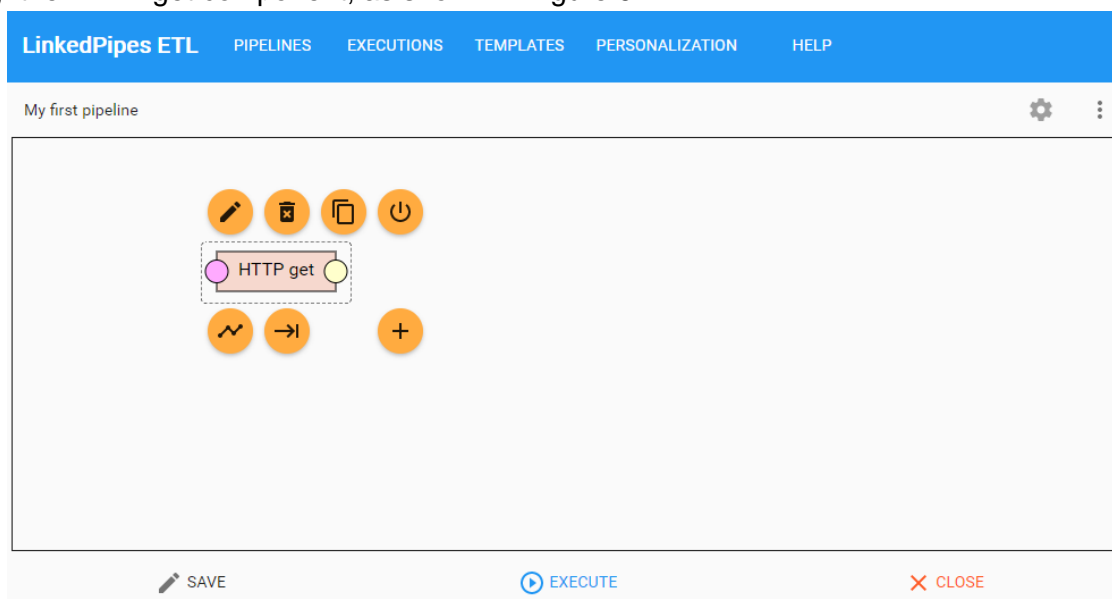


Figure 9. An HTTP get component of Linkedpipes-ETL

The component configuration dialog can be opened by double-clicking the component or by using the “Edit” button, as shown in Figure 10.



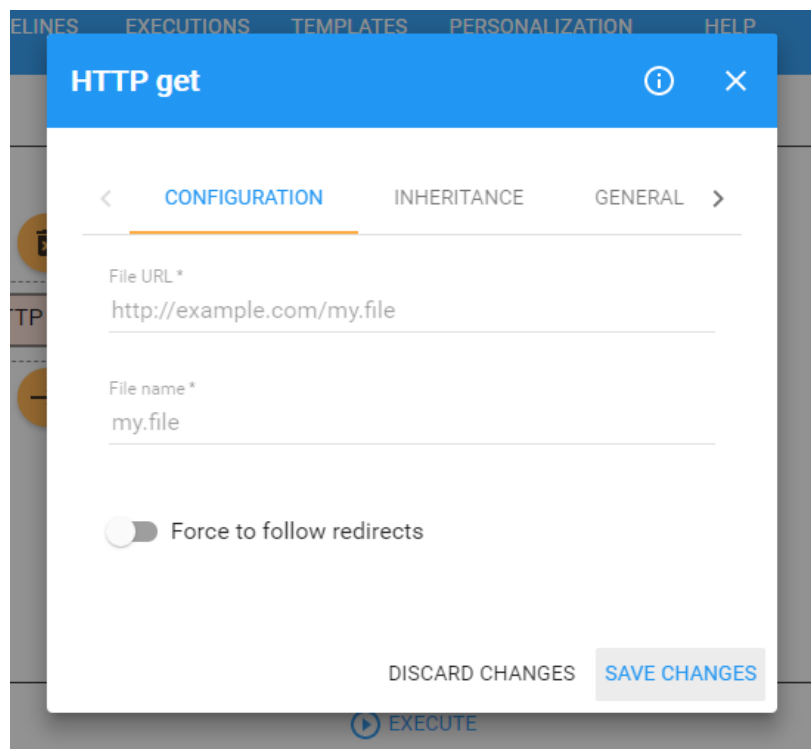


Figure 10. The configuration interface of HTTP get component.

In the component, the URL of a file to be downloaded and its name in the file system can be configured. Then, the pipeline can be executed, downloading the specified file. Then, the user can proceed with the pipeline creation by adding additional data processing components and executing the already finished parts of the pipeline for debugging purposes. For further documentation, tutorials and the list of available components see the LP-ETL website at <https://etl.linkedpipes.com/>. For an extensive documentation on the specific CSV2OBEU pipeline see the web page at <http://apps.openbudgets.eu/upload/linkedpipes>. This pipeline is a template pipeline that can be reused for inserting CSV fiscal data into the OBEU platform.

## 2.5. Microsite

The OBEU Microsite is an independent software which can be typically embedded within a web-page of a municipality, so that users can view and comment on budget data, as illustrated in Figure 11. Data and visualization software does not need to be hosted by the municipality but it is displayed on the fly by using a remote OBEU server instance.

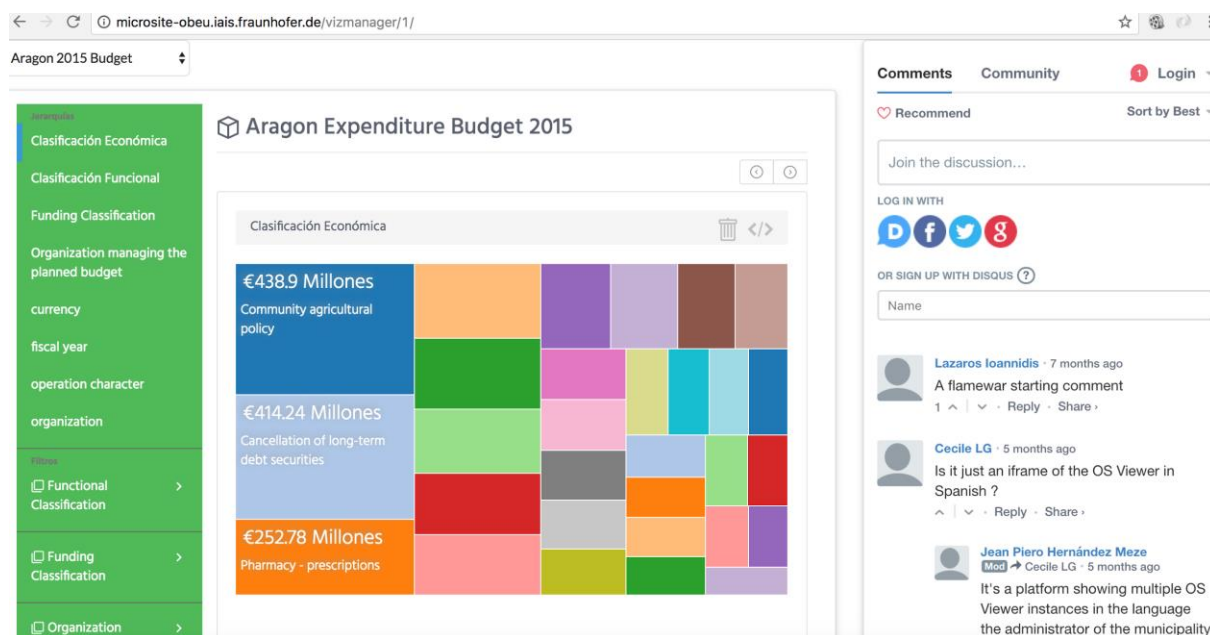


Figure 11. The OBEU Microsite for viewing and commenting budget data

The Microsite is designed to be configurable in terms of colors, language, datasets, arrangement of the views, etc. Using an admin account, we can change the location of the comment area into the flip-style. The view site is illustrated in Figure 12, and the comment site is illustrated in Figure 13.

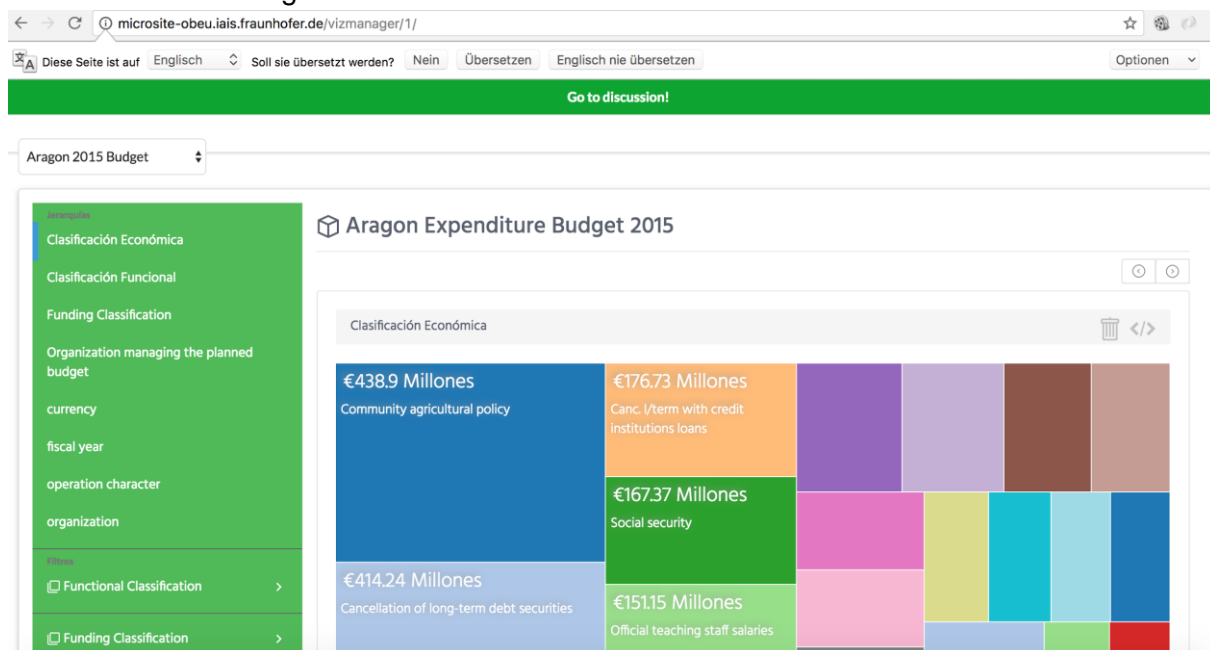


Figure 12. The view site of Microsite in the flip mode

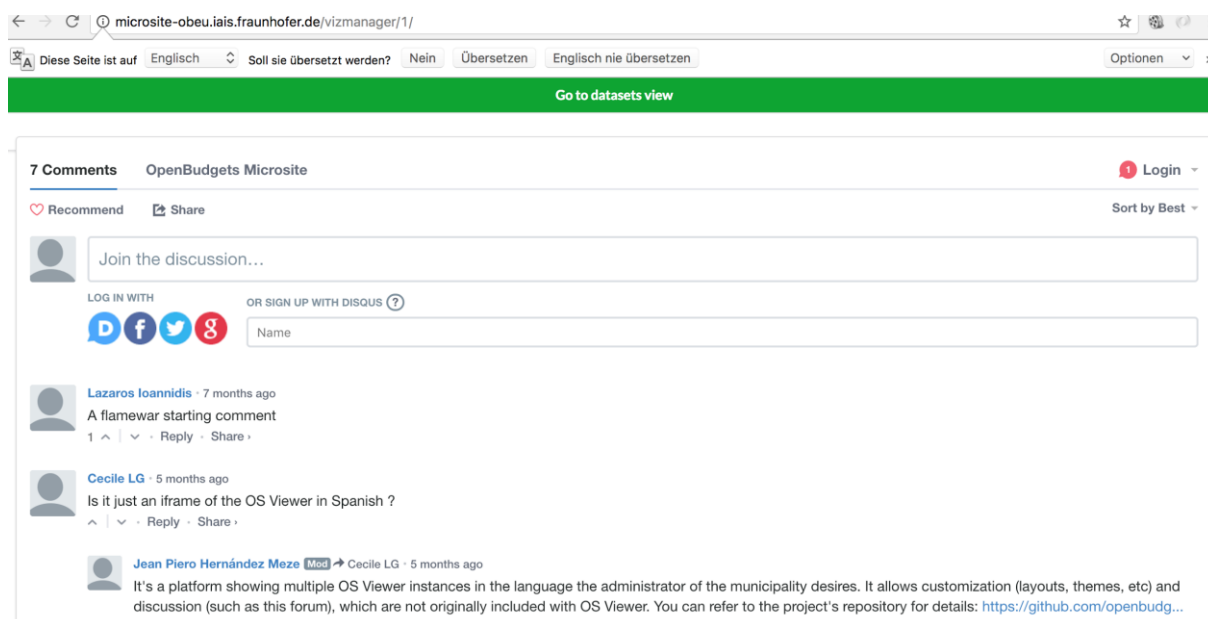


Figure 13. The comment site of Microsite in the flip mode

The source code and the installation instruction is located at <https://github.com/openbudgets/microsite>. The admin-site of Microsite is at <http://microsite-obeu.iais.fraunhofer.de/admin>; A demo of microsite is at <http://microsite-obeu.iais.fraunhofer.de/vizmanager/1/>.

## 2.6. Participatory Tool

OpenBudgets Participatory Budgeting (OBEU-PB) is an independent open-source software <https://github.com/openbudgets/participatory-budgeting> developed by OBEU team (CIVIO). It allows citizens to actively participate on the budget allocation and decision phase performed by municipalities. OBEU-PB is a Rails 5.0 application. The prerequisites are Git, Ruby 2.4.1, Bundler gem, Node 6.10.2 and PostgreSQL 9.4+

To prepare the environment, run the following shell commands.

```
$ bundle install
$ bin/rails db:create
$ bin/rails db:migrate
$ bin/rails db:seed
$ bin/rails db:test:prepare
```

To create d3.js custom bundle ([using npm & Rollup](#)), run `$ npm install`.

To rest, run: `$ bin/rails t`

There's also an included Guardfile to automate the test execution whenever file or directories are modified.

Run Guard with: `$ bin/guard`

Run the app with: `$ bin/rails s`

To inspect the contents of the emails sent by the app in development environment, use the following URL: `http://localhost:3000/rails/mailers/`

A demo is available at <http://openbudgets-monitoring-sample.herokuapp.com>, as illustrated in Figure 14.

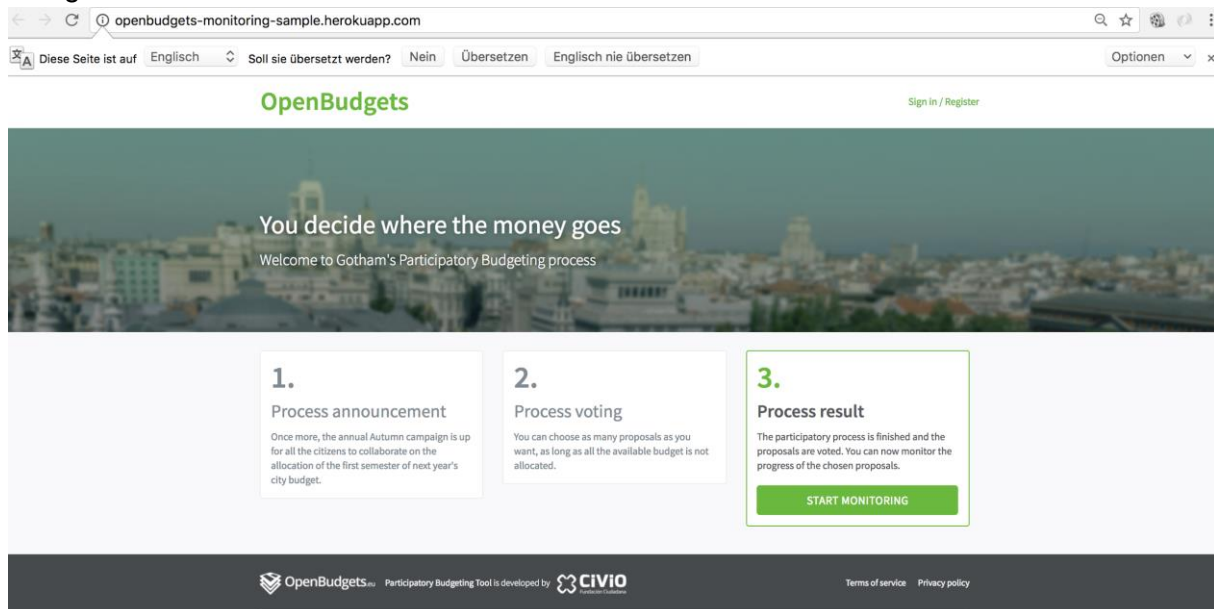


Figure 14. The OpenBudgets Participatory Budgeting demo installed at [www.herokuapp.com](http://www.herokuapp.com)

The main function of participatory budgeting tools is to let citizen participate into all decision stages of budgeting. Figure 15 shows the interface where citizen can vote for their favorite budgeting proposals.

## Vote your favourite proposals

The screenshot shows the OpenBudgets voting interface. At the top, there is a green 'FINISH' button and a 'TOTAL AMOUNT' bar showing '0 € (0 %)'.

On the left, a 'FILTER BY' sidebar includes:

- Budget:** A slider from 0 € to 1,000,000 €.
- District:** A dropdown menu set to 'All districts'.
- Area:** Checkboxes for 'Environmental Protection' and 'Cultural Affairs'.
- Tags:** Checkboxes for 'Garbage Disposal', 'Bicycles', and 'Parks and Recreation'.

Two proposal cards are displayed:

- Playground for the Docks Quarter:** Shows a description, a 'MORE INFO' button, the district 'Riverland', and a 'SIGN IN FOR VOTE' button. The amount is 102,344.32 €.
- Bonn Environmental Friendly:** Includes an image, a description, a 'MORE INFO' button, tags for 'Downtown', 'Environmental Protection', 'Garbage Disposal', 'Bicycles', and 'Parks and Recreation', and a 'SIGN IN FOR VOTE' button. The amount is 60,000.00 €.

Figure 15. Citizens can vote for their favourite budgeting proposals

## 2.7. Triplestore Virtuoso

Virtuoso is a data storage solution for managing and storing data in RDF format. It is a popular implementation of a triplestore (or RDFstore) and it offers a data management system together with a SPARQL endpoint and a UI for data administration. It is a third-party open source software, which is not developed by the OBEU team. For information about download, installation, and tutorial of Virtuoso, please refer to <https://virtuoso.openlinksw.com/rdf>.

## 2.8. RDF Browser

One of the main challenges of Linked Data deployment, is publishing them on the Web. The reason we give HTTP IRIs as names to resources is that both humans and machines should be able to lookup on these identifiers and get proper descriptions of the resources. Where proper means that it should be on the desired format. While there exist feature rich solutions, handling HTML representation and content

negotiation, partially or fully, there are situations where deployment is triple store or environment dependant. RDFBrowser is a customizable, easy to deploy, crossdomain and feature rich Linked Data publishing platform, developed by OBEU partner OKGR. RDFBrowser is a PHP web application, able to be deployed in most environments out of the box with minimum effort, lowering the barrier for publishing Linked Data on the Web. The feature rich HTML generator offers multilingual support, language detection, picture galleries, geo-information features and content filtering. All aspects of the application can be configured through an administrator panel on the fly, reducing administrative effort. <https://github.com/okgreece/RDFBrowser>. The HTML generator can be easily themed using a modern template engine.

RDF Browser is developed with Laravel MVC PHP Framework. The installation steps are listed as follows.

```
$ git clone https://github.com/okgreece/RDFBrowser.git
$ cd RDFBrowser
# build dependencies
$ composer install
# copy or move folder on your public folder usually /var/www
$ cd ..
$ cp -r RDFBrowser/ /var/www/RDFBrowser
#...or create a symlink to it
# on the root folder execute
$ php artisan key:generate
# if you want to test with development server execute
$ php artisan serve
# on your browser go to http://localhost:8000/admin
```

Detailed installation is described on Github <https://github.com/okgreece/RDFBrowser>

A screenshot is shown in Figure 16

## Data structure definition for the expenditure part of the Aragonian budget (autonomous community in northeastern Spain).

A resource of type: `qb:DataStructureDefinition` , from Named Graph:

`http://data.openbudgets.eu/resource/dataset/aragon-2011-expenditure` , within Data Space: `http://data.openbudgets.eu/resource/dataset/aragon-2011-expenditure`

Property	Value
<code>rdfs:label</code>	<span style="color: red;">+</span> Data structure definition for the expenditure part of the Aragonian budget (autonomous community in northeastern Spain).
Showing 1 to 1 of 1 entries	
Previous <span style="border: 1px solid black; padding: 0 5px;">1</span> Next	
Property	Value
<code>rdfs:type</code>	<code>qb:DataStructureDefinition</code>

Figure 16. A screenshot of the RDF Viewer developed by OKFGR

### 2.9. RDF Alignment Tool

Ontology matching is a crucial problem in the world of Semantic Web and other distributed, open world applications. Diversity in tools, knowledge, habits, language, interests and usually level of detail may drive in heterogeneity. An open-source RDF alignment tool is developed within the OBEU project by OKFGR partner <https://github.com/okgreece/Alignment>. The alignment tool is a collaborative, system aided, user driven ontology/entity matching application.

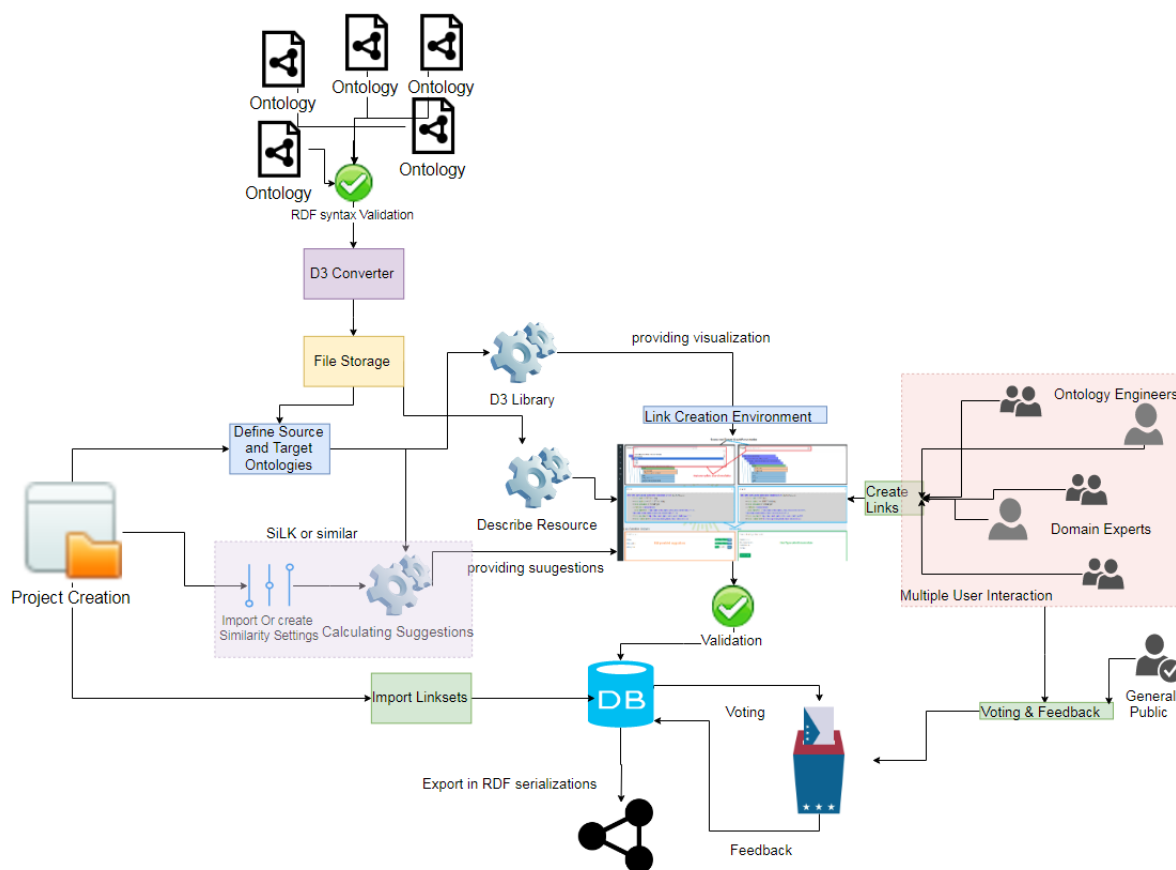


Figure 17. The flowchart of the RDF alignment tool developed by OKFGR

The installation steps are listed as follows.

```
$ git clone https://github.com/okgreece/Alignment.git
#run composer
$ composer install
#create an .env file from .env.example
$ cp .env.example .env
#change your database credentials using your favorite text editor
#run the migrations
$ php artisan migrate
#seed the database
$ php artisan db:seed
#run the Job Que. Change www-data accordingly, to reflect your server user name.
sudo -u www-data php artisan queue:listen --timeout=600 --sleep=30 --tries=5
```



Detailed instruction of the installation of the alignment tool is described at the Github <https://github.com/okgreece/Alignment>. The end-point at the Fraunhofer server is <http://apps.openbudgets.eu/alignment>.

## 2.10. Rudolf

Rudolf is an open source software <https://github.com/openbudgets/rudolf> developed by OBEU team (OKFGR). Rudolf is an HTTP API that wraps a SPARQL endpoint to expose [OpenBudgets.eu RDF-schema](#) - compatible datasets, according to OpenSpending's [babbage API](#) specification. Rudolf is based on Laravel 5.2. In order to install rudolf, you first have to setup a web server that supports the execution of PHP7 scripts. Rudolf may also work with PHP >5.5.9, but it has not been yet tested with versions of PHP other than version 7.

The PHP extension `php_mbstring` is also required. You can install this by running `sudo apt-get install php-mbstring` in your debian-based operating system.

To support friendly URL's please make sure you have enabled `mod_rewrite` in your PHP installation.

After setting up your environment, you can install rudolf:

1. Clone this repository:
2. git clone <https://github.com/larjohn/rudolf.git>
3. Get into the newly created directory:
4. `cd rudolf`
5. Run composer to install dependencies:
6. `composer install --no-scripts`

After installing rudolf, change the configuration file located at `config/sparql.php` and set the appropriate endpoint URI. Detailed information, including troubleshooting, is located at <https://github.com/openbudgets/rudolf> and <http://docs.openspending.org/en/latest/developers/platform/>

### 3 Conclusion

In this deliverable, we documented the OBEU platform, described its architecture, its components and their installation. Detailed descriptions and updates of these components are in general available at Github <https://github.com/openbudgets>. More technical details about the OBEU tools and the platform are included in Deliverable D4.5. For updates on the tools and the platform we suggest to consult the aforementioned Git repository and the OBEU website. The tools will still be updated and maintained by the community.