

OpenBudgets.eu: Fighting Corruption with Fiscal Transparency

Project Number: 645833

Start Date of Project: 01.05.2015

Duration: 30 months

Deliverable 4.3: Integrated OpenBudgets.eu platform - Working prototype

Dissemination Level	Public
Due Date of Deliverable	Month 16, 31.10.2016
Actual Submission Date	18.11.2016
Work Package	WP4, OpenBudgets.eu Requirements, Platform Architecture Integration and Development
Task	T 4.3
Type	Demonstrator
Approval Status	Final
Version	1.0
Number of Pages	17
Filename	Deliverable 4.3 - Integrated OBEU platform – Working prototype

Abstract:

This report describes the first working prototype of the integrated OpenBudgets.eu platform. The final version of the platform will be released, with deliverable D4.5, in April 2017 (M24). The open source code of the platform is available online and the platform itself is currently available on a demo Web server. Essentially, the platform consists of: the OpenSpending platform at its core, for fiscal data loading, management and exploration; an RDF data store with ETL pipelines for data transformations; modules for data mining algorithms and visualisations on budget and spending data.

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	23.08.2016	First contribution – structuring	Lazaros Ioannidis
0.2	01.09.2016	First draft	Maik Lukasch
0.3	28.09.2016	Improvements, updates and screenshots added	Maik Lukasch
0.4	10.10.2016	Improvements and minor edits	Fabrizio Orlandi
0.5	29.10.2016	Version ready for review	Maik Lukasch
0.6	10.11.2016	Reviewed version	Sotiris Karampatakis
0.7	15.11.2016	Final corrections	Maik Lukasch
1.0	17.11.2016	Final version	Fabrizio Orlandi

Author List

Organisation	Name	Contact Information
OKFGR	Lazaros Ioannidis	larjohn@okfn.gr
Fraunhofer IAIS	Maik Lukasch	maiklukasch@gmail.com
Fraunhofer IAIS	Fabrizio Orlandi	fabrizio.orlandi@iais.fraunhofer.de
OKFGR	Sotiris Karampatakis	karampatakis@okfn.gr

Executive Summary

This deliverable describes the first working prototype of the OpenBudgets.eu-platform. Further improvements and extensions of this version of the platform will still be performed during the next 6 months, when the final version of the platform will be released, together with deliverable D4.5 (due at M24). The proposed platform includes the OpenSpending platform at its core. It uses and extends the functionalities of OpenSpending by adding OpenBudgets.eu-specific applications. In particular, it offers additional features leveraging the RDF-format on budgetary and spending datasets.

Fiscal datasets in OpenSpending are modelled according to the FDP-format (a CSV-based data model including JSON metadata) and they can be transformed to RDF using the OBEU-platform. The transformation to RDF follows a specific OBEU data model (or ontology, described in D1.5) and the Linked Data Principles. Different applications on the OBEU-platform are designed to work with such RDF data and benefit of additional semantics and seamless integration with the Web of Data. Data mining algorithms and specific visualisation components have been developed on top of this data infrastructure. This document provides an overview of all these components and the overall architecture. In addition, a short user manual and details on the technical implementation are provided.

Abbreviations and Acronyms

WP	Work Package
OS	OpenSpending
OBEU	OpenBudgets.eu
LP-ETL	LinkedPipes ETL and LinkedPipes Visualisation
RDF	Resource Description Framework
FDP	Fiscal Data Package
SVG	Scalable Vector Graphics
OKI	Open Knowledge International
OKGR	Open Knowledge Greece
UEP	University of Economics, Prague
SPARQL	SPARQL Protocol And RDF Query Language
ETL	Extract, Transform, Load

Table of Contents

[Introduction](#)

[Requirements](#)

[Components of the OBEU-prototype](#)

[Plans for changes and additional components](#)

[User Manual](#)

[Uploading Data to OpenSpending and OpenBudgets](#)

[Exploring uploaded datasets](#)

[Visualize uploaded datasets](#)

[Data Mining](#)

[Implementation](#)

[Initial approach](#)

[Technical requirements](#)

[Integration](#)

[Conclusion and Future work](#)

1 Introduction

In this section we provide an overview of the main requirements for the OBEU platform (as in the Grant Agreement). Then, a sub-section describing the main components of the prototype is included, followed by a description of the planned improvements and future work. In Section 2, a user manual is provided to guide end users through the main stages and functions of the platform. Before concluding the deliverable, details about the technical implementation of the prototype are described (Section 3).

Requirements

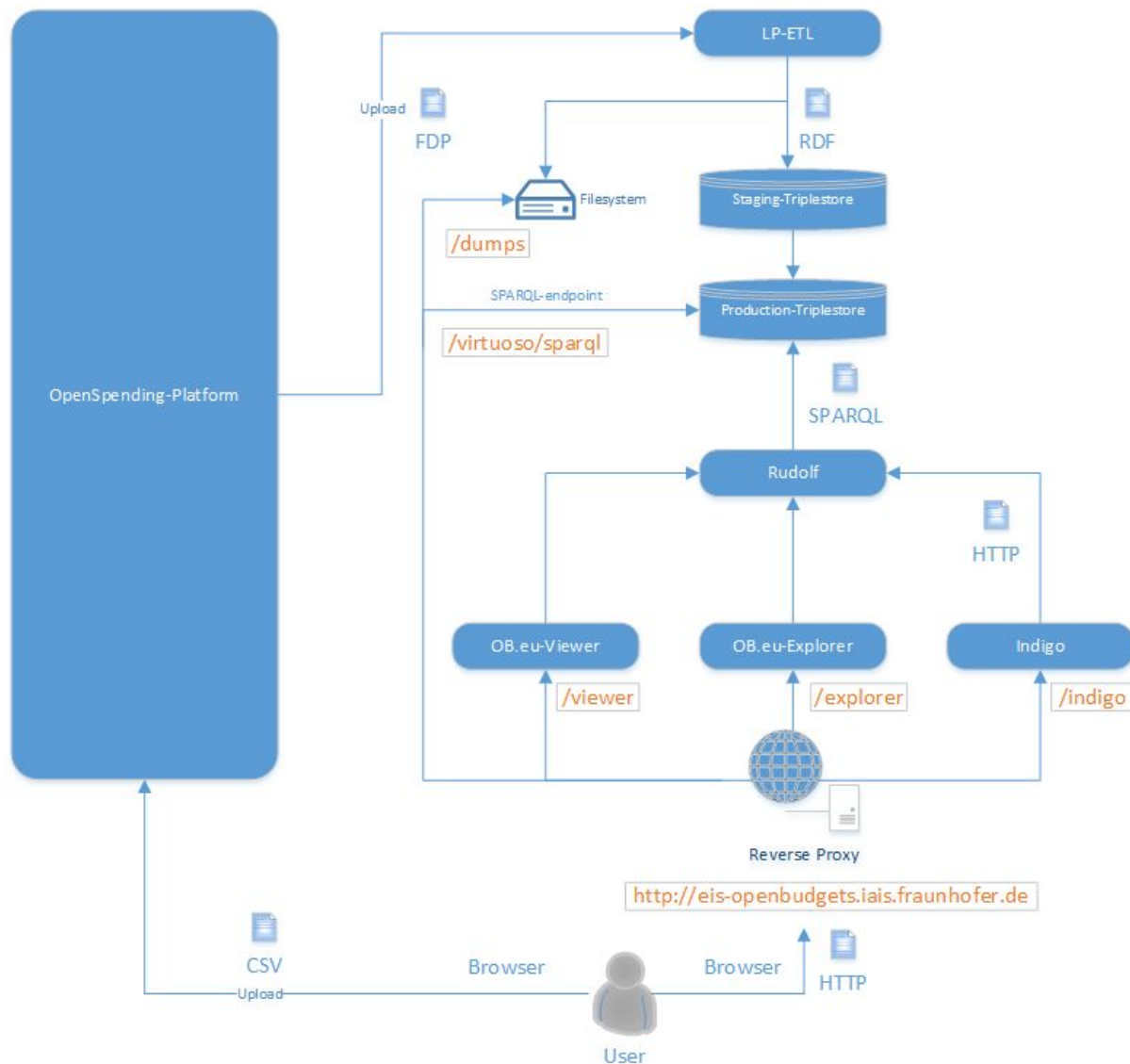
In this section we summarise the main requirements for the OBEU-prototype. These have been extracted from the OBEU Grant Agreement, and in particular from the description of WP4. More detailed technical requirements are listed in Section 3. These requirements guided our work for the development of an integrated OBEU platform:

- Making public budgetary and spending data more transparent and usable;
- Integration of the current results from Datamining & Comparative Analysis in WP2;
- Integration of the current results from Data Visualisation and Exploration Service in WP3;
- Integration of the current results from Feedback Service in WP7;
- Enabling of Upload & Download of data to & from the OBEU-platform;

- Transformation of Data from and to RDF on the OBEU-platform;
- Exposing a public SPARQL endpoint for querying imported datasets on the OBEU-platform.

Components of the OBEU-prototype

In order to summarise all the components of the current prototype and their interfaces / architecture, in the following figure the individual components of the OBEU-prototype are displayed.



1. **OS-Platform:** The OS-platform <http://next.openspending.org> is a software system being developed by OKI (in part) via the OBEU-project. The OS-platform has been connected to other components in the OBEU-platform, for example to LP-ETL. Furthermore some of the User-Interface-components of OpenSpending have been

reused in the OBEU-platform, specifically the OS-Viewer and the OS-Explorer. Users are able to upload datasets into the OS-platform and also to trigger an upload to the OBEU-platform additionally afterwards. The source code of the OS-platform is available under: <https://github.com/openspending>.

2. **LP-ETL with Custom Transformation Pipelines and the FDP-to-RDF-pipeline:** LP-ETL includes 2 applications: LinkedPipes ETL (<http://etl.linkedpipes.com>) and LinkedPipes Visualization (<http://visualization.linkedpipes.com>) and they have been developed by the Charles University of Prague with the support of UEP for OBEU-specific use cases for transforming datasets from different formats into RDF-format via LP-ETL-pipelines. A custom FDP-to-RDF-pipeline has been developed by UEP for transforming datasets from FDP-format used in OS into the RDF-format used in OBEU. In the first step the user can upload CSV-files to the OS-platform. The data will be stored in FDP-format on the OS-platform. In the following step the user can trigger the transformation and import into the OBEU-platform. The FDP-to-RDF-pipeline handles the transformation to RDF-format and the import onto the OBEU-platform. The source code of LP-ETL is available under: <https://github.com/linkedpipes/etl.git>. The source code of the FDP-to-RDF-pipeline can be found here: <https://github.com/openbudgets/pipeline-fragments/tree/master/FDPtoRDF>

3. **Triplestores:** Two triplestores are used in the OBEU-platform: One for staging and one for production. Transformed RDF-datasets from LP-ETL are uploaded first to the staging triplestore. A specific LP-ETL-pipeline will be later developed and installed onto the OBEU-platform in order to clean, validate and move the datasets from the staging triplestore to the production triplestore. The SPARQL-endpoint of the production-triplestore is exposed as read-only to the public at: <http://eis-openbudgets.iais.fraunhofer.de/virtuoso/sparql>. Virtuoso has been chosen as triplestore for the OBEU-platform because it is licensed as free OpenSource software under GNU General Public License (GPL) Version 2. Formerly Fuseki (based on Apache Jena: <https://jena.apache.org/documentation/fuseki2/index.html>) had been used as triplestore in the OBEU-platform but was later replaced by Virtuoso because of the performance gain on the execution time of SPARQL-queries in Rudolf. In addition, according to several benchmarks (<https://www.w3.org/wiki/LargeTripleStores>) it shows better reliability and performance for large datasets. The source code of Virtuoso is available under: <https://github.com/openlink/virtuoso-opensource>

4. **Reverse Proxy:** Multiple applications are offered under the single domain <http://eis-openbudgets.iais.fraunhofer.de>. HTTP-requests from outside are delegated by a Reverse Proxy to the proper application according to their URL-Pattern such as:
 - <http://eis-openbudgets.iais.fraunhofer.de/viewer> (OBEU-Viewer)
 - <http://eis-openbudgets.iais.fraunhofer.de/explorer> (OBEU-Explorer)
 - <http://eis-openbudgets.iais.fraunhofer.de/indigo> (Indigo)
 - <http://eis-openbudgets.iais.fraunhofer.de/dumps> (RDF-Datasets-Files)
 - <http://eis-openbudgets.iais.fraunhofer.de/virtuoso/sparql> (SPARQL-Endpoint)

Nginx is used as a Reverse Proxy in the OBEU-platform.

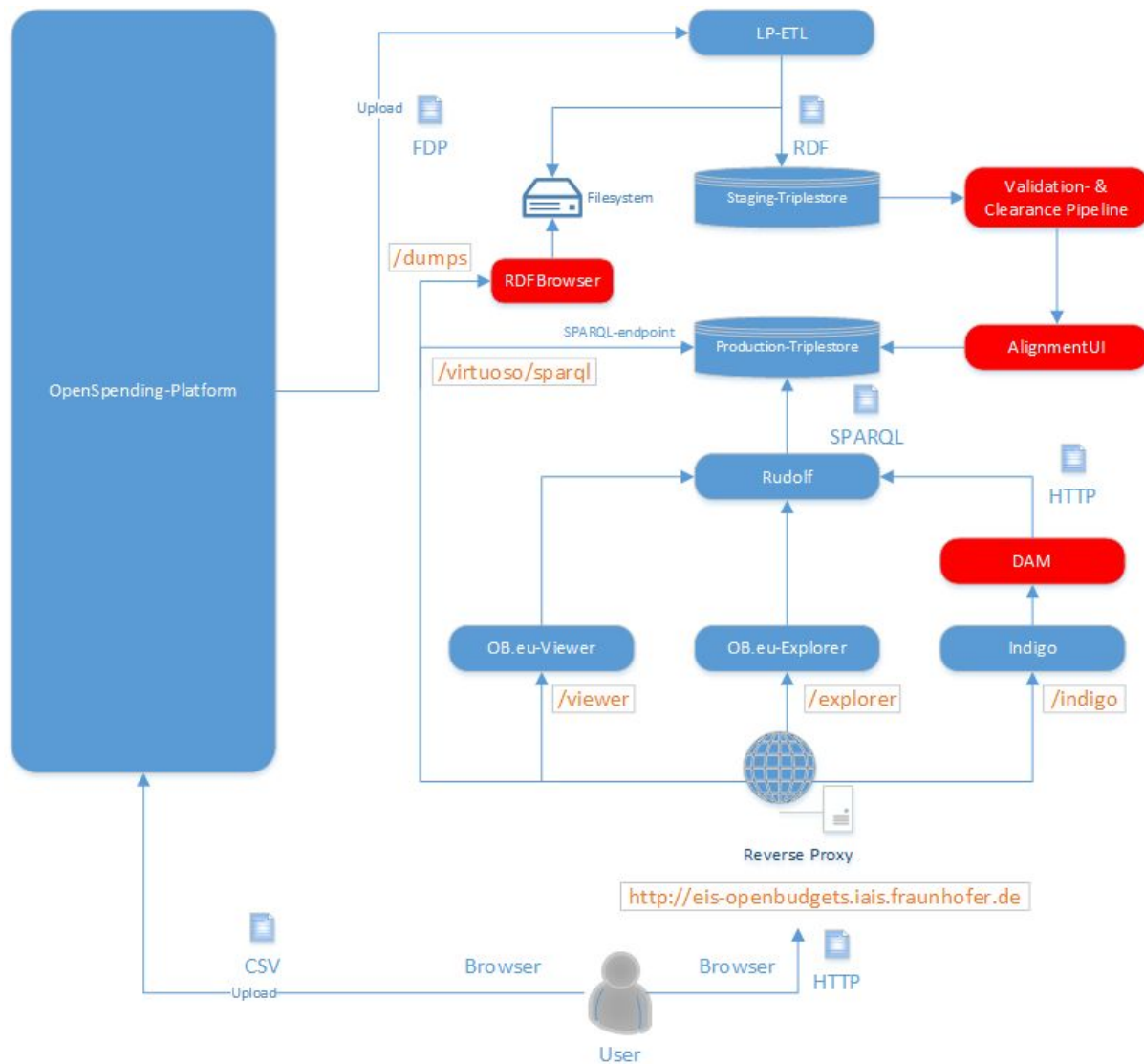
The source code of nginx is available under: <https://github.com/nginx/nginx>

5. **Rudolf:** The Rudolf application developed by OKGR basically transforms HTTP-requests into SPARQL-Queries and is currently used in the OBEU-prototype as a backend application for the OBEU-Viewer and the OBEU-explorer. But Rudolf can be used also by any other application as well since it is publicly exposed (for example the endpoint: <http://eis-openbudgets.iais.fraunhofer.de/api/3/cubes>). A specific API (see: <http://docs.openspending.org/en/latest/developers/api/>) is implemented by Rudolf which corresponds to the backend application of the OS-Viewer in the OS-platform. Therefore we have been able to integrate the OS-Viewer and the OS-Explorer as OBEU-Viewer and OBEU-explorer into the OBEU-platform. The source code of Rudolf is available under: <https://github.com/openbudgets/rudolf>
6. **Customized OBEU-Viewer & OBEU-Explorer:** The OBEU-Viewer and OBEU-explorer are OBEU-specific adaptations of the OS-Viewer and OS-explorer developed and used by OKI on their OS-platform. These applications have been made customizable by OKI in order to adapt the styling and design of those frontends to the needs of the OBEU project. The OBEU-Viewer <http://eis-openbudgets.iais.fraunhofer.de/viewer> visualizes the datasets uploaded to the OBEU-platform. The OS-explorer <http://eis-openbudgets.iais.fraunhofer.de/explorer> can be used to browse through all the datasets on the OBEU-platform. Both applications use Rudolf as backend application for accessing indirectly the triplestore.
7. **Indigo:** The Indigo application developed by OKGR is meant to serve as a Datamining frontend. Rudolf API is used to acquire the data in json format files which in turn are used as inputs for data mining algorithms. [OpenCPU](#) API, which is responsible for data mining process, includes data mining algorithms¹ that consumes the json input files, implements the selected algorithms and returns every needed parameter also in json format in order to get the proper visualizations. The results of the Datamining algorithms are shown by Indigo to the user. Currently the only Datamining algorithm being offered by Indigo is the Timeseries analysis developed by OKGR. The source code of Indigo is available under: <https://github.com/okgreece/indigo>

Plans for changes and additional components

In the following a diagram our plan for the final OBEU-platform is shown:

¹ <https://github.com/okgreece/DescriptiveStats.OBeu>
<https://github.com/okgreece/Cluster.OBeu>
<https://github.com/okgreece/TimeSeries.OBeu>



1. **AlignmentUI:** For interlinking RDF-datasets an additional component is planned to be integrated into the OBEU-platform. The AlignmentUI is a system-aided manual linking tool and offers a GUI presenting the source and the target ontology, thesauri or vocabulary as graphs. System can provide suggestions to the user, using predefined or custom provided, string matching algorithms. But it is up to the user if the suggested link is accepted. The AlignmentUI uses as a backend component the Silk Linked Data Integration Framework (<https://github.com/silk-framework/silk>). The user can decide to link two entities using his own experience/knowledge. This tool can be used for the interlinking and aligning the codelists in OBEU-RDF-mode, offering comparison capabilities between heterogeneous data.
2. **RDF-Browser:** In order to offer URI dereferencing of exposed RDF-dataset-files at <http://eis-openbudgets.iais.fraunhofer.de/dumps> we plan to integrate an additional application called RDF-Browser developed by OKGR into the OBEU-platform. The

source code of RDF-Browser is available under:
<https://github.com/okgreece/RDFBrowser>.

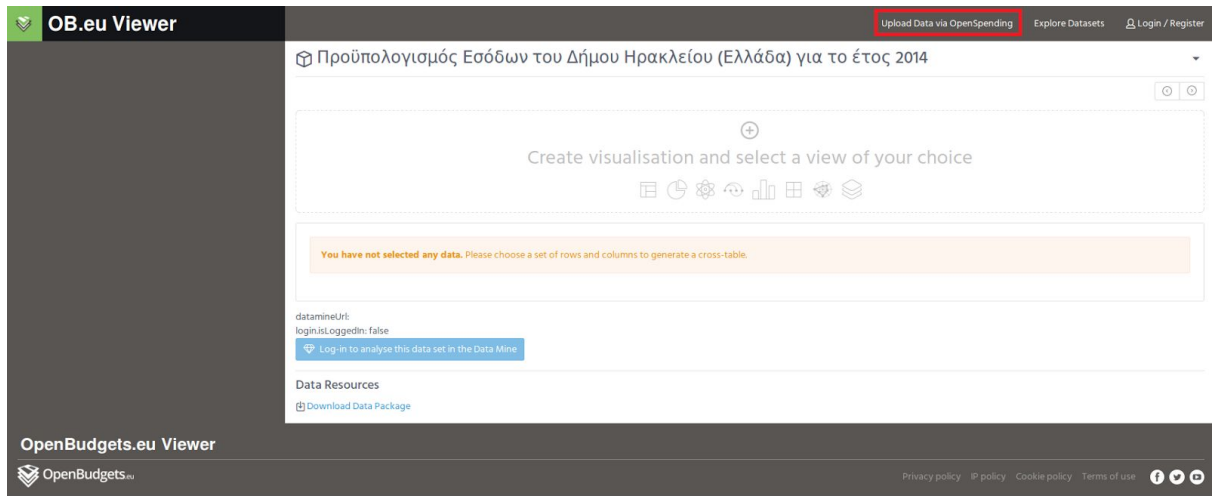
3. **Validation- and Clearance-Pipeline:** This LP-ETL-pipeline will be installed in the LP-ETL application on the OBEU-platform and will handle the validation and clearing of datasets in the staging triplestore. After validation and clearing the datasets will be imported into the production triplestore.
4. **DAM:** DAM is a backend component under development and will be used later as middleware between the several Datamining algorithms and the Datamining frontend (Indigo). Each Datamining algorithm may expect another input format and return a different output format. In order to abstract from these differences between the individual Datamining algorithms DAM will offer a single interface to the Datamining frontend (Indigo).
5. **More Datamining Algorithms:** For the OBEU-prototype only the Timeseries-analysis has been integrated so far. When more Datamining algorithms (such as Outlier/Anomaly detection, Pattern matching, Comparative Analysis etc.) are completed in WP2, they will be integrated as well into the OBEU-platform.
6. **More RDF-datasets:** For the OBEU-prototype only the datasets of several greek municipalities <https://github.com/openbudgets/datasets/tree/master/greek-municipalities> and Aragon <https://github.com/openbudgets/datasets/tree/master/Aragon> have been imported into the OBEU-platform. But for the future more datasets will be added including datasets from the OS-platform, since now a way has been established to import and transform datasets from the OS-platform.

2 User Manual

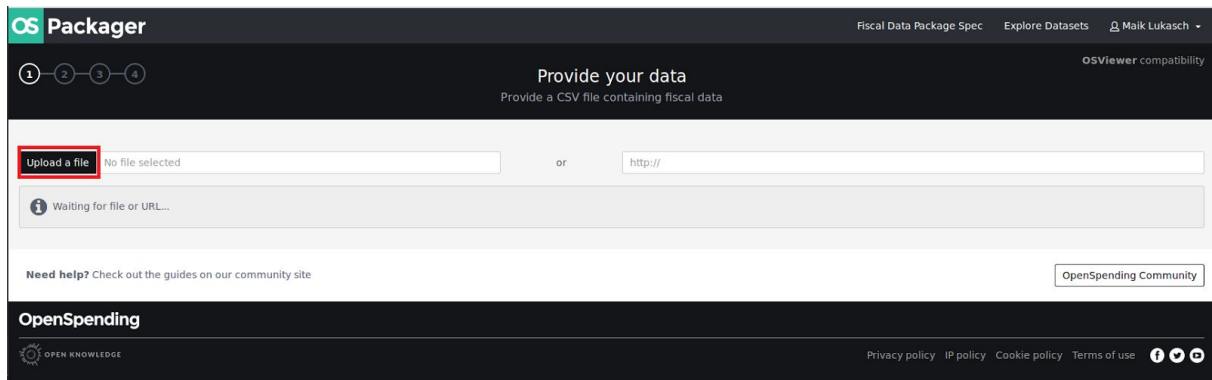
In order to provide users with a guide for testing the current version of the platform, here we describe the main activities users can perform at the current stage of the prototype. The main documented activities allow uploading, exploration, visualisation and mining of fiscal data.

Uploading Data to OpenSpending and OpenBudgets

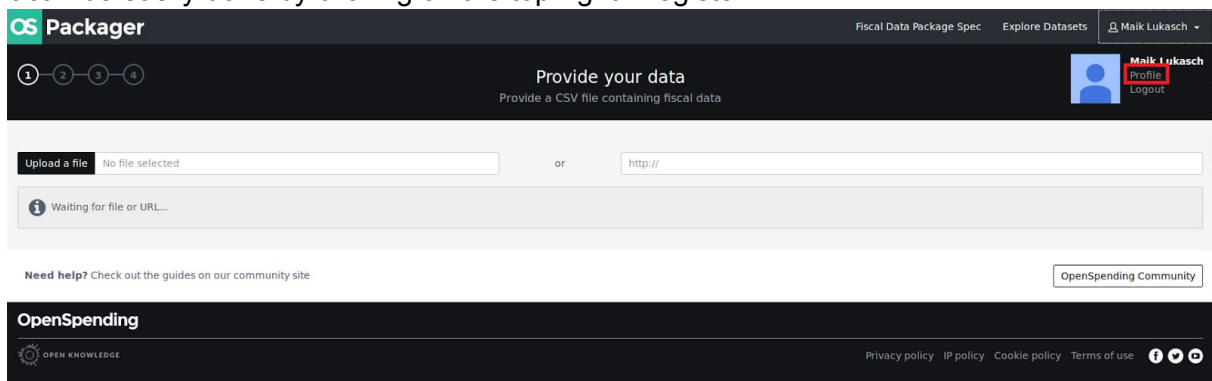
Visit <http://eis-openbudgets.iais.fraunhofer.de/viewer> and click on “**Upload Data via OpenSpending**” as shown in the screenshot:



You will be directed to the OpenSpending-Platform where you can upload your data given in CSV-format:



After your data has been successfully uploaded to OpenSpending, visit your profile page (as shown below). Here we assume a user is already registered on the OpenSpending site, if not it can be easily done by clicking on the top-right "Register" link.



Here you can select and trigger the upload of the FDP-datasets from OpenSpending to OpenBudgets (click on "Run External Hooks" for the selected dataset):

Title	Publication status	Edit	View	Download	Hooks
test	Publish				
test fdp 1	Publish				
test fdp 2	Publish				
test fdp 3	Publish				
test fdp 5	Publish				
test fdp tordf	Publish				
test1	Publish				

The selected dataset is now uploaded both on the OpenSpending site and the OBEU data store.

Exploring uploaded datasets

After the datasets have been transformed and uploaded to OpenBudgets, we can explore them using the Explorer: <http://eis-openbudget.iais.fraunhofer.de/explorer>.

OB.eu Explorer

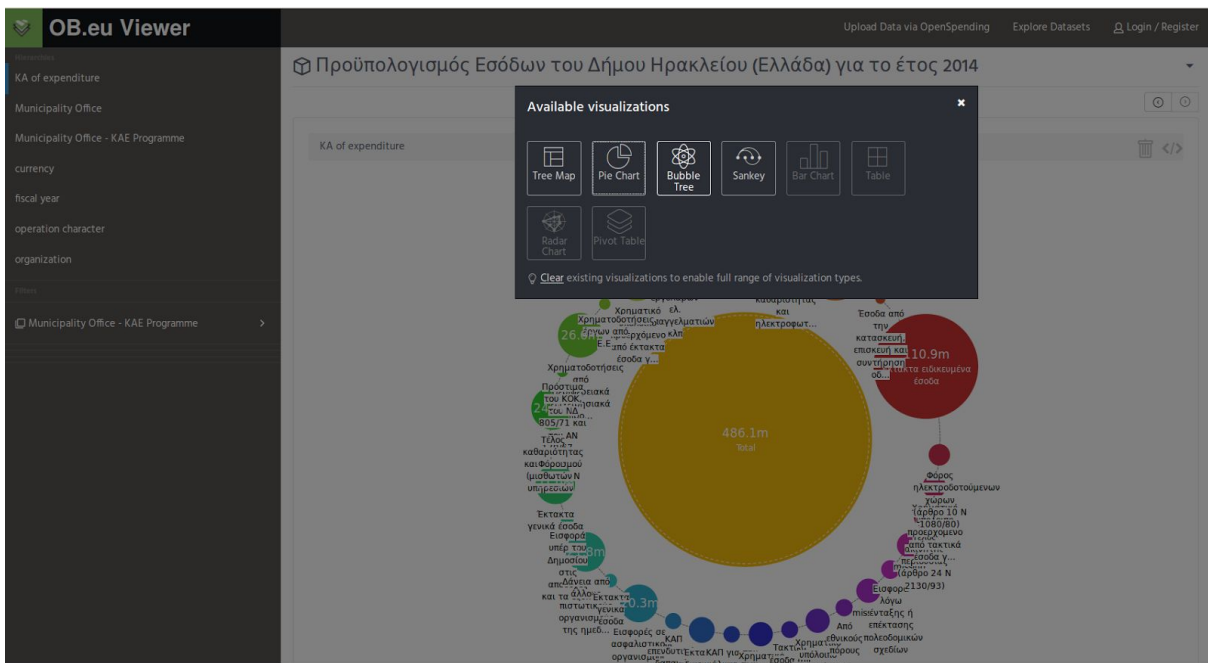
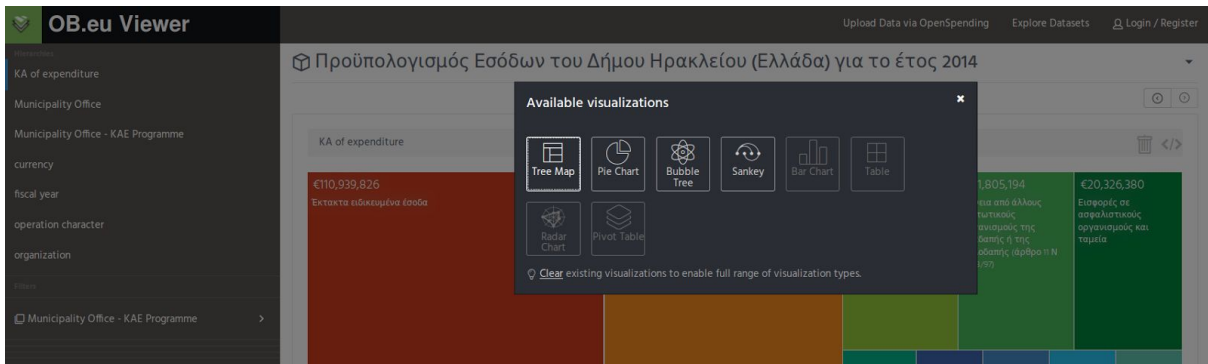
Searchable, shareable access to OpenBudgets.eu datasets.

Search data

<p>Προϋπολογισμός Εσόδων του Δήμου Ηρακλείου (Ελλάδα) για το έτος 2014</p> <p>This datapackage has no description</p> <p>Place Holder <place.holder@not.shown></p>	<p>Προϋπολογισμός Εσόδων του Δήμου Θεσσαλονίκης (Ελλάδα) για το έτος 2012</p> <p>This datapackage has no description</p> <p>Place Holder <place.holder@not.shown></p>	<p>Revenues Budget of Municipality of Kalamaria for the fiscal year 2012</p> <p>This datapackage has no description</p> <p>Place Holder <place.holder@not.shown></p>	<p>Προϋπολογισμός Εξόδων του Δήμου Θεσσαλονίκης (Ελλάδα) για το έτος 2013</p> <p>This datapackage has no description</p> <p>Place Holder <place.holder@not.shown></p>
<p>Revenues Budget of Municipality of Karpensisi for the fiscal year 2016</p> <p>This datapackage has no description</p> <p>Place Holder <place.holder@not.shown></p>	<p>Expenditure Budget of Municipality of Kilkis for the fiscal year 2012</p> <p>This datapackage has no description</p> <p>Place Holder <place.holder@not.shown></p>	<p>Revenues Budget of Municipality of Sithonia for the fiscal year 2012</p> <p>This datapackage has no description</p> <p>Place Holder <place.holder@not.shown></p>	<p>Προϋπολογισμός Εσόδων του Δήμου Θεσσαλονίκης (Ελλάδα) για το έτος 2013</p> <p>This datapackage has no description</p> <p>Place Holder <place.holder@not.shown></p>

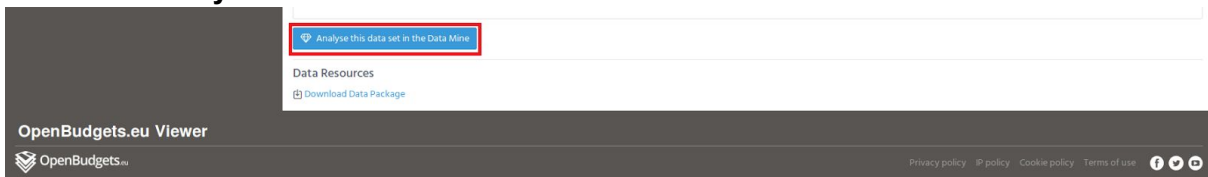
Visualize uploaded datasets

By selecting a dataset in the Explorer we can visualize it in the Viewer: <http://eis-openbudgets.iais.fraunhofer.de/viewer>. In the Viewer we can choose between different types of visualisations. Below, we included two examples depicting a “Tree Map” and a “Bubble Tree” visualisation.



Data Mining

In the Viewer, users can apply data mining algorithms to the selected dataset by clicking on the button **“Analyse this data set in the Data Mine”**.



This directs users to the Datamining-Frontend-application Indigo where users can configure and run one of the available data mining algorithms. For the time being, in this version of the prototype, only the Time Series Analysis algorithm (developed by OKGR) has been integrated.

Cube: Global dataset

C

Time Series

Input: Data coming from an aggregation

Aggregates

C Global amount

Global amount ≡

Filters

direct relation ≡

Drill down

C Global fiscal period--notation

notation ≡

Sorting

≡

Pagination

Input: parameter

Select an aggregate, a time-related drilldown and the prediction steps parameter from the left and click on the execute button on top right.

Auto-correlation

ACF

PACF

ACF residuals

PACF residuals

Decomposition

STL general information	
Degree	2
Degrees of freedom	5.288
Degrees of freedom (fitted)	4.9747

Trends

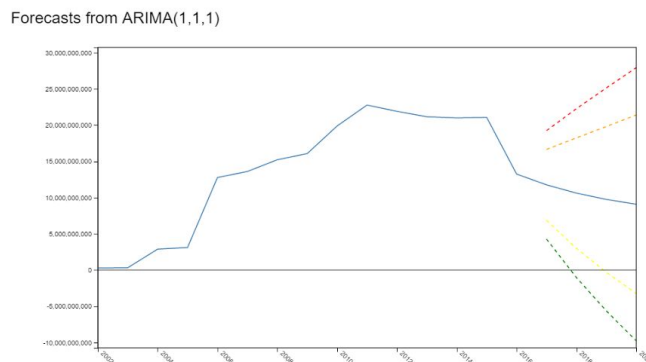
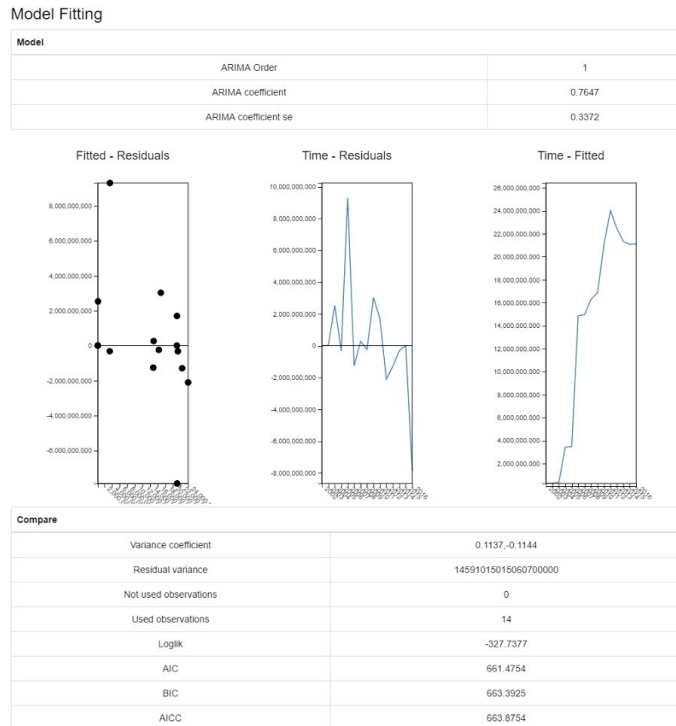
Remainders

Fitted - Residuals

Time - Residuals

Time - Fitted

Compare	
Residual variance	3006533985398310000
Used observations	2002
Loglik	-2104600000000000000
AIC	42091475795576400000
BIC	42091475795576400000
GCV	6693669992686870000



Additional data mining algorithms developed by the technical partners are currently being tested and integrated to the platform.

3 Implementation

Initial approach

Docker is one of the leading software containerization platforms. *“Docker containers wrap a piece of software in a complete filesystem that contains everything needed to run: code, runtime, system tools, system libraries – anything that can be installed on a server. This guarantees that the software will always run the same, regardless of its environment”* (<https://www.docker.com>). This technology supports OBEU in the integration of different modules and components. Moreover, due to the positive feedback given by OKI on using

Docker and Docker-Compose in production with their well-established OS-platform, we decided to use those technologies for the OBEU-prototype as well.

Technical requirements

The technical requirements for the OBEU-platform can be found in the Deliverable D4.2 Analysis of the required functionality of OpenBudgets.eu (<http://openbudgets.eu/assets/deliverables/D4.2.pdf>).

Integration

Multiple applications developed by different partners in the project have to be combined and work together. Therefore the partners had to agree on common interfaces between the applications. Furthermore all those different applications have to be installed on a server and to be connected to each other. The Docker-technology offers a way for running multiple applications in separated environments on a single server. Furthermore Docker-Compose offers a way to orchestrate multiple applications together and to make them communicate to each other.

The prototype is based on the Docker-technology: <https://www.docker.com>

Each application is contained within a Docker-Container. Docker containers are built from so-called Docker-Images which are either downloaded directly from DockerHub <https://hub.docker.com> or build from a Dockerfile during the deployment. Mostly only a single application of the OBEU-application-stack is contained within a Docker container.

In order to orchestrate the Docker-containers together an additional technology has been used in OBEU called Docker-compose <https://docs.docker.com/compose/install/>.

With Docker-compose the orchestration of the multiple Docker-containers is described in a single docker-compose.yml-file, which can be found here for the OBEU-prototype: <https://github.com/openbudgets/integration/blob/master/docker-config/docker-compose.yml>.

The source code for the OBEU-platform is available here: <https://github.com/openbudgets/integration.git>

As collaboration tool for technical issues and repository for the source code in the OBEU-project Github has been used: <https://github.com/openbudgets>.

4 Conclusion and Future work

At this stage of the OBEU development, a foundation has been established with the first OBEU-prototype which allows importing datasets into the OBEU-RDF-format. Especially users without any further knowledge of Data Transformation into RDF are able to upload datasets to the OBEU-platform via the OS-platform in a fully automated way. After the datasets have been imported to the OBEU-platform they can be visualized and inspected with different kind of diagrams provided by the OBEU-Viewer. Since now datasets can be transformed into RDF, Datamining algorithms using RDF can work on the datasets. So far only one Datamining algorithm: “Timeseries analysis” has been integrated into the OBEU-platform and the user can visualize the results of this algorithm via Indigo. As more Datamining algorithms will be finished in WP2 more of those algorithms will be integrated into the OBEU-platform. Further testing of the data transformations with additional datasets is still

ongoing, and additional output of WP3 with visualisations will be integrated in the platform. These activities will continue until the release of the final version of the platform (M24).