Horizon 2020
European Union Funding
for Research & Innovation

OpenGovIntelligence

Fostering Innovation and Creativity in Europe through Public Administration Modernization towards Supplying and Exploiting Linked Open Statistical Data

Deliverable 3.4

Report on OpenGovIntelligence ICT tools – second release

Leading partner:	NUIG
Participating partners:	CERTH, TUT, ProXML, SWIRRL, NUIG
Version-Status:	1.0
Dissemination level:	PU

This project has been funded with the support of the **H2020 Programme** of the European Union

Deliverable factsheet

Project Number:	693849
Project Acronym:	OpenGovIntelligence
Project Title:	Fostering Innovation and Creativity in Europe through Public Administration Modernization towards Supplying and Exploiting Linked Open Statistical Data

Deliverable title:	Report on OpenGovIntelligence ICT tools – second release
Deliverable number:	D3.4
Official submission date:	31 October 2017
Actual submission date:	31 October 2017

Editor(s):	Arkadiusz Stasiewicz (NUIG) Mohamed Adel (NUIG)
Author(s):	Dimitrios Zeginis (CERTH) Areti Karamanou (CERTH) Evangelos Kalampokis (CERTH) Konstantinos Tarabanis (CERTH) Arkadiusz Stasiewicz (NUIG) Mohamed Adel (NUIG) Paul Hermans (ProXML) Bill Roberts (SWIRRL) Rick Moynihan (SWIRRL)
Reviewer(s)	Bill Roberts (SWIRRL)

Abstract:	This deliverable provides details on the prototypes of software components delivered as a result of the second development stage of OpenGovIntelligence project.

Effort of Participating Partners

	Name	*Short Name*	*Role*	*Person Months*
1.	Centre for Research & Technology - Hellas	CERTH	Participant	1
2.	Delft University of Technology	TUDelft	None	0
3.	National University of Ireland, Galway	NUIG	Leader	1.5
4.	Tallinn University of Technology	TUT	Participant	0.2
5.	ProXML bvba	ProXML	Participant	0.1
6.	Swirrl IT Limited	SWIRRL	Participant	0.5
7.	Trafford council	TRAF	None	0
8.	Flemish Government	VLO	None	0
9.	Ministry of Interior and Administrative Reconstruction	MAREG	None	0
10.	Ministry of Economic Affairs and Communication	MKM	None	0
11.	Marine Institute	MI	None	0
12.	Public Institution Enterprise Lithuania	EL	None	0

Revision History

Version	*Date*	*Revised by*	*Reason*
0.1	18-Sep-2017	A. Stasiewicz (NUIG)	First Draft
0.2	11-Oct-2017	A. Stasiewicz (NUIG)	Tools list update
0.3	19-Oct-2017	A. Stasiewicz (NUIG)	General updates
0.4	26-Oct-2017	A. Stasiewicz (NUIG)	General updates
0.5	31-Oct-2017	A. Stasiewicz (NUIG)	Addressing internal review comments
1.0	31-Oct-2017	CERTH	Submission to EC

Table of Contents

List of Figures

List of Tables

List of Abbreviations

The following table presents the acronyms used in the deliverable in alphabetical order.

Abbreviation	*Description*
API	Application Programming Interface
CMS	Content Management System
ICT	Information and Communication Technologies
LOSD	Linked Open Statistical Data
OLAP	OnLine Analytical Processing
RDF	Resource Description Framework
UI	User Interface
URI	Uniform Resource Identifier
WP	Work Package

Executive Summary

This document is the deliverable "D3.4 – Report on OpenGovIntelligence ICT tools – second release" (referred to as D3.4). It provides detailed information about the result of the second stage of the OpenGovIntelligence development in WP3.

The goal of WP3 ("ICT tools development") is to develop the OpenGovIntelligence ICT tools as a suite of open source and commercial tools. The final version of the developed ICT tools will support the OpenGovIntelligence framework created in WP2 ("Framework creation") and will enable:

a) the creation of Linked Open Statistical Data (LOSD) from various sources,
b) the expansion of LOSD with datasets from existing sources,
c) the exploitation of LOSD for the co-production of public services.

With regards to the creation of LOSD, OpenGovIntelligence designed and developed ICT tools that enable the transformation of public sector data to standard machine readable forms (specifically to RDF data cubes) and the validation of the generated RDF. With regards to the exploitation of LOSD, OpenGovIntelligence designed and developed ICT tools that enable visualisation and analysis of statistical data.

While the first release of the OpenGovIntelligence ICT tools was focused on the tools supporting the creation and exploitation of LOSD, the next release focuses on exploitation of LOSD by providing new tools and improvements (functional and performance) to tools arleardy developed and evaluated.

The tools presented in this deliverable will be evaluated during the pilot implementation stage. Based on the outcomes, the existing tools will be improved. Moreover new tools will be developed in order to cover remaining features of the OpenGovIntelligence Framework.

The second release of the OpenGovIntelligence ICT tools was guided by challenges and needs identified in WP1 (D1.1 *OpenGovIntelligence challenges and needs*), the OpenGovIntelligence framework (D2.2 *OpenGovIntelligence framework*) as well as the pilots operation and initial OpenGovIntelligence evaluation (D4.2 *Evaluation results - First round*).

Results presented in this document are the outputs of Task 3.2 OpenGovIntelligence ICT tools – second release. This deliverable will be updated together with the ongoing dvelopment of the OpenGovIntelligence tools as D3.6 *Report on OpenGovIntelligence ICT tools*. The final release is scheduled for October 2018.

1 Introduction

This section introduces the background of the work carried out in WP3 "ICT tools development". Sub-section 1.1 presents the scope and the objectives of the current document, sub-section 1.2 describes the intended audience for this document, sub-section 1.3 outlines the structure of the document, while sub-section 1.4 addresses the comments received from the 1st Project Review.

1.1 Scope

This report documents the ICT tools developed during the second phase of WP3. To guide readers in understanding the context of these tools, Section 2 presents the overview of the OpenGovIntelligence tools. The Second release provided tools that enables creation and exploitation of LOSD. Final release of the tools will aim to cover expansion of existing datasets and exploitation of LOSD for co-production of public services.

1.2 Audience

The intended audience for this document is the OpenGovIntelligence consortium, in particular partner organisations responsible for the development of pilot trials, the European Commission (EC) and those who are interested in challenges and needs for opening-up and exploiting LOSD for the co-production of innovative data-driven services on governments.

1.3 Structure

The structure of the document is as follows:

- Section 2 presents the OpenGovIntelligence ICT tools overview
- Section 3 provides a detailed description of the individual OpenGovIntelligence ICT tools developed during the second development stage of OpenGovIntelligence project;
- Finally, Section 4 concludes the report and outlines the future development plan.

1.4 Addressing reviewer's comments

This sub-section address the general comments from the 1st Consolidated Expert Review Report. The Review Report did not include specific issues concerning D3.2. Therefore, in Table 1 Recommendations concerning future work we present recommendations over the future work.

Table 1 Recommendations concerning future work

No.	Comment	Addressed in
1	It is recommended that future iterations of D3.2 should make a clear statement about the technical starting point of the project and what has transferred from the DaPaaS and OpenCube projects.	Details are provided in Section 2.
2	"Figure that gives OGI architecture is shown in this document, where the same figure is used in the deliverable D.2.1 (Figure 4)."	While D2.1 provides overview of the architecture, D3.2 explain in detail each of the elements.

2 OpenGovIntelligence ICT tools – overview

This section aims to provide background information related to LOSD and OpenGovIntelligence tools.

In general, the developed tools cover parts of the OpenGovIntelligence Architecture and are part of a loosely connected ecosystem of tools. The OpenGovIntelligence Architecture (presented in D3.2) for LOSD and data-driven public services enables stakeholders to collaborate towards the production of innovative data-driven public services by exploiting Linked Open Data technologies and statistical datasets. The initial version of the architecture was documented in detail in D3.2 and was tested and refined during the first round of pilot evaluations. The architecture is organised as follows:

- The architecture is divided into five layers: (i) Data Provision, (ii) Data Platform, (iii) Process Layer, (iv) Service Design, and (v) Service Provision.
- Key management responsibilities are shared across all layers.
- Each layer has a set of components that performs tasks specific to that layer.

The architecture guides the pilot implementations, as well as the other future implementations of the OGI software in other projects together with OpenGovIntelligence framework (documented in D2.2).

2.1 Tools developed during the second development stage

In the first implementation stage, the developed tools were mainly supporting data conversion and exploitation of the data. The second release of OpenGovIntelligence includes additional tools and supports a larger part of the OpenGovIntelligence framework.

While the development of majority of the tools were initialted during the OpenGovIntelligence project, some of them were using outputs of the DaPaaS[1] and OpenCube[2] projects as the technical starting point, and subsequently improved and extended within OpenGovIntelligence. Details are presented in Table 2. OGI ICT Tools technical starting point and visualised in Figure 1. LOSD Tool Ecosystem. Also, Table 2 shows which tools are new components, started in year 2, and which tools were further developed in year 2 building on work started earlier.

Table 2. OGI ICT Tools technical starting point

No.	Name	Started in Year 2	Foundation
1	JSON API For Data Cube Specification	No	OpenGovIntelligence
2	JSON API For Data Cube Implementation	No	OpenGovIntelligence
3	Table2QB And Grafter	No	DaPaaS

[1] https://project.dapaas.eu
[2] http://opencube-project.eu

4	Assisted Cube Schema Creator	Yes	OpenGovIntelligence
5	Data Cube Builder	No	OpenGovIntelligence
6	Data Cube Explorer	No	OpenCube
7	Data Cube Aggregator	No	OpenCube
8	LOSD Machine Learning Component	Yes	OpenGovIntelligence
9	OLAP Browser	No	OpenCube
10	QB Multi-Dimensional Charting	Yes	OpenGovIntelligence
11	RDF Data Cube Geo Data Dashboard	Yes	OpenGovIntelligence
12	SPARQL Connector for Exploratory	Yes	OpenGovIntelligence

Section 3 documents the tools, which were delivered during the second development stage as well as links by which they can be accessed. In general, tools developed during the project lifecycle can be accessed at the GitHub repository available at: http://github.com/OpenGovIntelligence

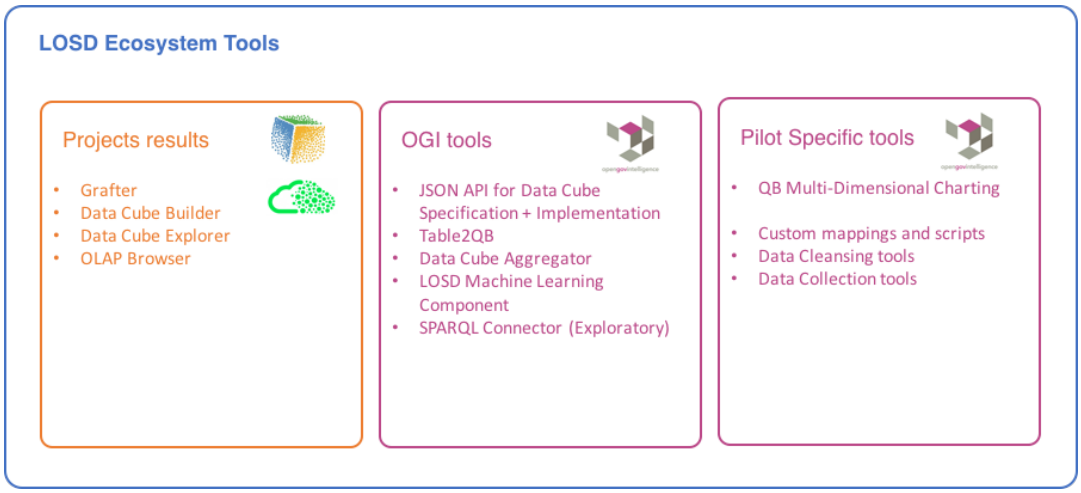


Figure 1. LOSD Tool Ecosystem

Note: The OpenGovIntelligence project will not develop software tools for every component of the architecture or a single integrated system that matches this architecture as a whole. Implementations of the OpenGovIntelligence approach will combine a selection of tools and processes relevant to the use cases they address, following the principles set out in the architecture. Moreover, some parts of the architecture might end up being human processes or use appropriate pre-existing tools from outside of the OpenGovIntelligence project. Existing tools available for the purpose will re-used where ever possible and new tools will be developed when required. We have already identified a number of tools that can be re-used for example in collaboration space. This list of tools is included in deliverable D2.1 – OpenGovIntelligence framework – first release.

3 OpenGovIntelligence ICT tools - second release

This section provides a detailed description of the individual OpenGovIntelligence ICT tools developed during the second development stage of the OpenGovIntelligence project and details of the major updates to the tools developed it the first stage of the OpenGovIntelligence project. Previously developed tools descriptions are available in deliverable "D3.2 – Report on OpenGovIntelligence ICT tools – first release". The tools developed during the project lifecycle can be accessed at the GitHub repository available at: http://github.com/OpenGovIntelligence .

3.1 Assisted Cube Schema Creator

LOSD publishing process always starts with the process of mapping of the desired data set. This process is required to conform with RDF Data Cube Vocabulary defined concepts and properties in order to enable interoperability.

In the first stages of LOSD tools experimentation and development schema mapping process was accomplished by analysis of the available datasets and manual processing of the mapping. Results were used as e.g. for inputs for RDF Data Cube Builder tool. Any changes of the source data required repetition of the mapping process. For sustainability reasons, we could not depend on this methodology for our LOSD publishing process.

The analysis of the other LOSD creation methodologies [3] [4] [5] [6] [7] [8] lead to development of Assisted Cube Schema Creator. To ensure sustainability, the tool was designed to support variety of data sets and different use cases. Moreover, tool had to be simple to use by different user types.

3.1.1 Functionality Description

This publishing pipeline is created to ease the mapping process of government statistical datasets into RDF according to the RDF Data Cube vocabulary. This tool integrates spreadsheets, Open Refine [9], RDF Refine [10] and RDF Data Cube [11] to produce a generic LOSD mapping tool that fits any pilot or use case.

3.1.2 Implementation Description

The following steps are included in the publishing pipeline:

[3] https://github.com/GMDSP-Linked-Data/GenericStatsCube
[4] http://www-etis.ensea.fr/WOD2013/wp-content/uploads/2013/06/Publishing-Census-as-Linked-Open-Data..pdf
[5] http://wifo5-03.informatik.uni-mannheim.de/bizer/r2r/
[6] https://www.w3.org/2001/sw/rdb2rdf/wiki/Implementations
[7] http://wifo5-03.informatik.uni-mannheim.de/latc/toollibrary/screencast.html
[8] https://www.w3.org/TR/r2rml/#definitions
[9] http://openrefine.org
[10] http://refine.deri.ie/publications
[11] https://www.w3.org/TR/vocab-data-cube

A. First step in the mapping process is implemented by collecting data set to RDF Data Cube mapping details from the user. To collect information about the mapping details from users a spreadsheet template with built in guidelines was prepared. This spreadsheet template ease the information collection process and make it simpler for all types of users. Users are able to:
    a. Start defining existing data set headers and their new names if wanted.
    b. Defining the RDF Data Cube types of those headers i.e. dimensions, measures, or attributes.
    c. Linking those headers with exiting linked data URI that matches the header concept and properties.
    d. Defining labels, comments, related concepts and datatypes for those elements.
    e. Defining the RDF data Cube's Data Structure Definition (DSD) element by linking the dimensions, measures, and attributes with DSD.
    f. Finally, user define the RDF Data Cube's Data Set (DS) element that include Linked Data annotation abilities i.e. data set publisher URI definitions, related subject(s) URI definitions, related location(s) and organization(s) definitions. Moreover, user defines the label, description and comments for the DS.
B. The second step in the mapping process is carried out using Open Refine with the RDF Refine extension. The RDF Data Cube Vocabulary structure was predefined in JSON format that can be consumed by Open Refine with assistance from the RDF Refine extension. This JSON RDF Data Cube Vocabulary structure is coupled with the spreadsheet template. The process is designed to be easy to use for all types of users. It simplifies and minimises the data set to RDF Data Cube mapping and conversion steps. As demonstrated in Figure 2 OGI LOSD Creation and Usage Flowchart – RDF Data Cube Schema Mapping Stage, the user needs to load, apply and extract the RDF schema by following the following steps:
    a. Loading mapping spreadsheet to Open Refine + RDF refine extension: the user starts step 2 of the schema creation process by loading the spreadsheet containing the data set to RDF Data Cube Vocabulary mapping details into the running Open Refine server as an Open Refine project.
    b. Applying the predefined RDF Data Cube Schema to the input spreadsheet: the user simply loads the provided JSON file containing the RDF Data Cube Vocabulary Structure and couples this with the spreadsheet template. This step is done using the standard 'apply operations' procedures available on Open Refine.
    c. Extracting the schema mapping of the dataset in RDF/XML or TTL formats: finally, the user uses the RDF refine extension to extract the RDF Data Cube schema file in RDF/XML or TTL formats. This file is produced by applying the JSON file instruction to the spreadsheet.

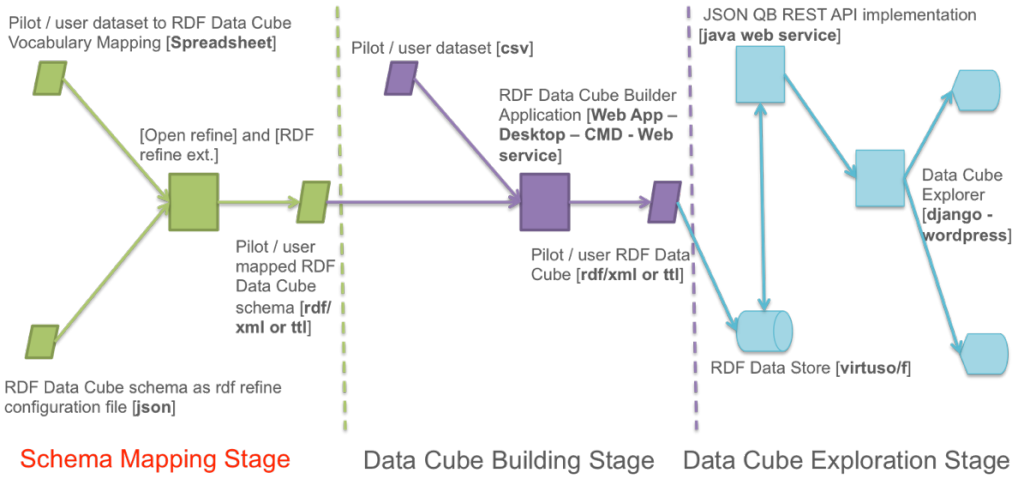


Figure 2 OGI LOSD Creation and Usage Flowchart – RDF Data Cube Schema Mapping Stage

3.1.3 Availability

Source code is available at GitHub:
https://github.com/OpenGovIntelligence/qb-assisted-schema-creator

3.1.4 License

The software is available as open source under the MIT License.

3.1.5 Pilots involved

The Irish Marine Institute, and Enterprise Lithuania are the initial users of the tool. This tool is generic and can be used across the OpenGovIntelligence Consortium.

3.1.6 Future development plan

The plans for next year development include:

1. Updates to the spreadsheet filling guidelines
2. Evaluation and feedback collection
3. Updating the tool with user feedback, and tool testing results.

3.2 LOSD Machine Learning Component

Machine learning enables the creation of predictive models based on the analysis of high volumes of data. The wealth of the statistical data that is freely available online can contribute towards this direction. The multi-dimensional nature of statistical data enables the extraction of numerous features from a single dataset based on the value of the measured variable for the different values of the dimensions or the different aggregated functions (e.g. min, max, average etc.) that can be applied to the measure.

3.2.1 Functionality Description

The Machine Learning Component enables the automatic extraction of numerous features from LOSD based on the needs of the users and the predictive scenario that is implemented. It also enables the performance of dimension reduction based on relevant algorithms such as Forward Subset, Backward Subset, and Lasso in a user-friendly approach. In this way, users will be able to include big volumes of LOSD in machine learning scenarios and filter only the predictors that are relevant to the scenario at hand. These predictors can be later used in the creation of predictive models.

3.2.2 Implementation Description

The implementation of the Machine Learning Component is based on the JSON-QB API and R server.

3.2.3 Availability

Source code is available at GitHub:
https://github.com/OpenGovIntelligence/qb-machine-learning-component

3.2.4 License

The software is available as open source under the Apache License (v2.0).

3.2.5 Pilots involved

At this stage, the tools have not been actively used in the pilot use cases, thus the tool is generic and can be used across the OpenGovIntelligence Consortium.

3.2.6 Future development plan

The plans for next year development include:

1. Collecting users feedback on the tool.
2. Updating the tool based on the user feedback, and tool testing results.

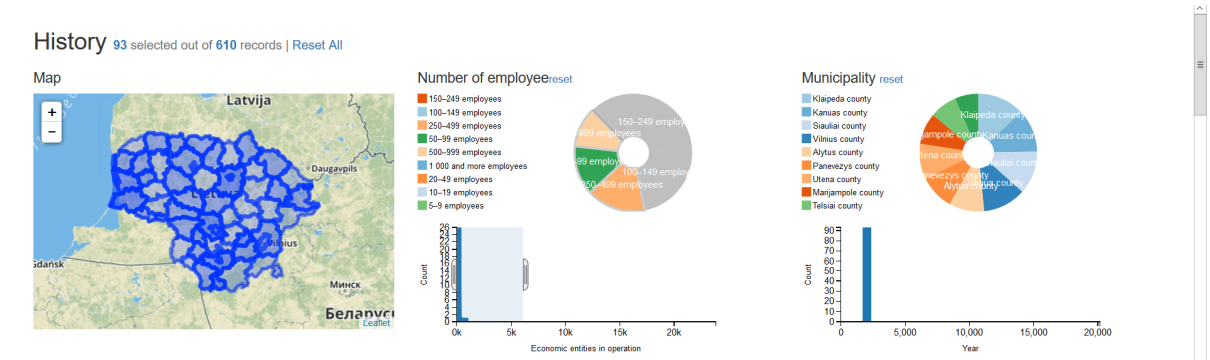
3.3 QB Multi-Dimensional Charting

QB Multi-Dimensional Charting tool is a multi-dimensional charting dashboard which is using a JavaScript charting library as a base. It supports cross filtering and allowing highly efficient exploration on large multi-dimensional. It leverages d3 engine to render charts in CSS friendly SVG format. Charts rendered using dc.js are naturally data driven and reactive therefore providing instant feedback on user's interaction.

The main objective of this tool is to provide an easy yet powerful JavaScript dashboard which can be utilized to perform data visualization for cube data and analysis in browser as well as on mobile device.

3.3.1 Functionality Description

This tool is providing an easy to use multi-dimensional charting dashboard which can be utilized to perform data visualization, data filtering, exploration and analysis for cube data in browser as well as on mobile device.



Country	Number of employee	County	Municipality	Economic entities in operation	Sex	Year
Republic of Lithuania	500–999 employees	Vilnius county	Vilnius d. mun.	1		2016
Republic of Lithuania	500–999 employees	Vilnius county	Trakai d. mun.	1		2016
Republic of Lithuania	500–999 employees	Vilnius county	Elektrėnai mun.	1		2016
Republic of Lithuania	500–999 employees	Utena county	Visaginas mun.	2		2016
Republic of Lithuania	500–999 employees	Utena county	Utena d. mun.	3		2016
Republic of Lithuania	500–999 employees	Telsiai county	Plungė d. mun.	1		2016

Figure 3 QB Multi-Dimensional Charting dashboard

3.3.2 Implementation Description

QB Multi-Dimensional Charting dashboard is written in JavaScript on top of another JavaScript libraries like dc.js, Leaflet.js and crossfilter.js; dc.js is a JavaScript library that is used to create interactive charts and a corresponding table. Moreover, it supports an interactive map.

3.3.3 Availability

Source code is available at GitHub:
https://github.com/OpenGovIntelligence/qb-multi-dimensional-charting

3.3.4 License

The software is available as Open Source under the Apache 2.0 License.

3.3.5 Pilots involved

The Irish Marine Institute, and Enterprise Lithuania are the initial users of the tool.

3.3.6 Future development plan

The plans for next year development includes

1. Further integration of the tools

2. Releasing the tools as a WordPress plugin

3.4 RDF Data Cube Geo Data Dashboard

The RDF Data Cube Geo Data Dashboard allows visualisation and discovery of geo annotated data. RDF Data Cube Geo Data Dashboard uses the JSON QB API to access the geo annotated RDF Data stored in a RDF database. The retrieved data are visualized as map based dashboard to enable ease of information understanding to all types of users.

3.4.1 Functionality Description

This dashboard is designed and implemented to visualise geo annotated data and support pilot specific use cases.

3.4.2 Implementation Description

This tool was written in JavaScript using several libraries and tools as follow: Leaflet.js, GeoJSON, OpenStreetMap; Leaflet js is a lightweight open-source JavaScript library for building mobile-friendly interactive maps thanks to its strong mapping features; GeoJSON is a format for encoding a variety of geographic data structures,it supports the different geometry types like Point, Polygon, MultiPoint, and MultiPolygon; OpenStreetMap (OSM) is a collaborative project to create a free editable map of the world. Figure 4 and Figure 5 presents User Interface: information from the selected Data Cube are visualised as a tagged map.

3.4.3 Availability

Source code is available at GitHub: https://github.com/OpenGovIntelligence/qb-geo-ui

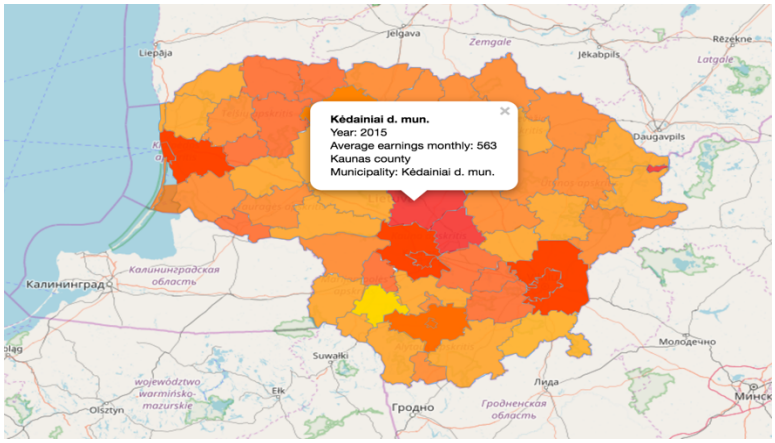


Figure 4 Lithuanian Data Cubes Geo based Visualization screen samples – A

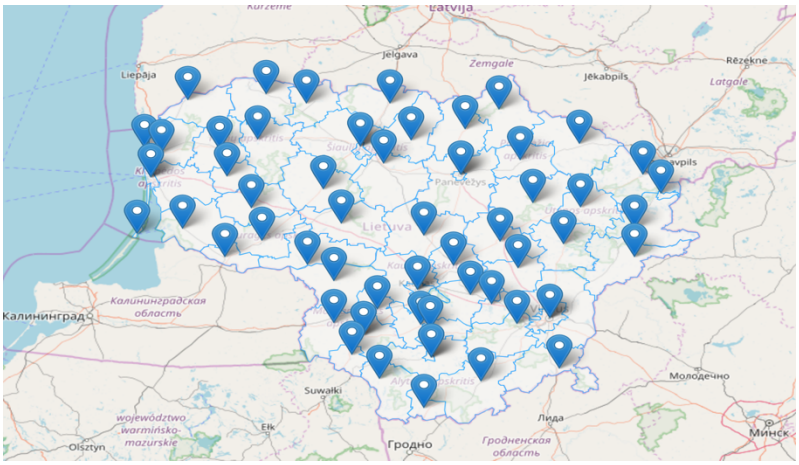


Figure 5 Lithuanian Data Cubes Geo based Visualization screen samples – B

3.4.4 License

The software is available as open source under the MIT License.

3.4.5 Pilots involved

Enterprise Lithuania is the initial user of the tool.

3.4.6 Future development plan

The plans for next year development include:

1. Collecting user feedback on the tool.
2. Updating the tool based on the user feedback and tool testing results.

3.5 SPARQL Connector for Exploratory

SPARQL connector for Exploratory allows to connect the Exploratory Data Science tool to a SPARQL endpoint. This component is provided as R code together with a JSON configuration file. It has become part of a widely deployed product.

3.5.1 Functionality Description

This component enables import of data stored as RDF into the Exploratory system into Exploratory.

3.5.2 Implementation Description

This component is provided as R code together with JSON configuration file.

3.5.3 Availability

Extension is available at: https://github.com/OpenGovIntelligence/exploratory_sparql_plugin

3.5.4 License

For the license details please check https://exploratory.io/terms

3.5.5 Pilots involved

The Flemish Government is the initial users of the tool.

3.5.6 Future development plan

his component has become part of a widely deployed product and already meets its design objectives so no further development of this is required or foreseen.

3.6 JSON API for Data Cube Specification – update

Following evaluation of initial implementations, it was decided to implement a second version of the API using the GraphQL query language for APIs, due its fast growing adoption in the web developer community and because its features are a good match for our requirements to search for and filter RDF Data Cubes.

The set of API methods has been re-specified using the GraphQL approach, and expanded as additional useful functions have been identified

The work on the API is available at https://github.com/Swirrl/graphql-qb

3.7 JSON API for Data Cube Implementation - update

During year 2 of the project, the API has been implemented using GraphQL. Work is continuing to add new data access methods, to optimise performance and to improve user documentation - available at https://github.com/Swirrl/graphql-qb

3.8 Table2QB And Grafter - update

During year 2 of the project, further research has been carried out into the optimal design of the table2qb input format, to make it flexible enough for the range of use cases to be supported, but still easy for statisticians or analysts to produce, without needing technical knowledge of Linked Data.

- Further prototypes have been implemented and tested
- Improvements to the Grafter libraries have been implemented to improve performance and scalability when processing large data cubes.

The table2qb specification can be found at https://github.com/Swirrl/table2qb

Grafter can be found at https://github.com/Swirrl/grafter

3.9 Data Cube Aggregator - update

During year 2 of the project, the Data Cube Aggregator was extended to support the computation of aggregations from raw RDF data. The extension takes as input raw RDF data and creates a cube that contains the corresponding aggregated observations. The current version of the Data Cube Aggregator is integrated with version 1 of the JSON-QB API available at GitHub repository: https://github.com/OpenGovIntelligence/json-qb-api-implementation

Future work is aiming to migrate the Data Cube Aggregator to the new API implementation based on GraphQL.

4 Conclusion

This deliverable provides description of the components delivered in the second phase of OpenGovIntelligence project (Month 21 of the project). Detailed information about usage of previously available tools are documented in Deliverable D3.2 – "Report on OpenGovIntelligence ICT tools – first release".

All the tools available as the result of OpenGovInteligence project are constantly updated based on the evaluation result. The plans for next year development include the usability and performance improvements. It is expected that all project pilots will use suitable tool. Moreover, numerous government organisations external to the project are expected to start using OpenGovInteligence tools in year 3 of the project.

Continuous JSON-QB API development is expected to extend the range of methods supported by the API, deploy in operational environments. One of the major features under development is aggregation across hierarchies. In this case the observations are aggregated across a hierarchy of a dimension. For example, if a cube contains the election results at municipality level, then the Aggregator can compute the results at region and at country level with the prerequisite that the corresponding hierarchy (municipality → region→ country) exists.

This deliverable will be updated together with the ongoing dvelopment of the OpenGovIntelligence tools. The final release of the developed components will be documented in deliverable D3.6 "Report on OpenGovIntelligence ICT tools" planned at October 2018.

References

[1] D1.1 OpenGovIntelligence challenges and needs

[2] D2.1 OpenGovIntelligence framework - first release

[3] D2.2 OpenGovIntelligence framework

[4] D3.1 OpenGovIntelligence ICT tools - first release

[5] D3.2 Report on OpenGovIntelligence ICT tools - first release

[6] D4.1 Pilots and Evaluation plan