



## OpenGovIntelligence

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

---

### Deliverable 3.5

### OpenGovIntelligence ICT tools

---

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

## Deliverable factsheet

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

<b>Deliverable title:</b>	OpenGovIntelligence ICT tools
<b>Deliverable number:</b>	D3.5
<b>Official submission date:</b>	31 October 2018
<b>Actual submission date:</b>	31 October 2018

<b>Editor(s):</b>	Arkadiusz Stasiewicz (NUIG)
<b>Author(s):</b>	Dimitrios Zeginis (CERTH) Evangelos Kalampokis (CERTH) Konstantinos Tarabanis (CERTH) Efthimios Tambouris (CERTH) Arkadiusz Stasiewicz (NUIG) Mohamed Adel (NUIG) Agustín García Pereira (NUIG) Paul Hermans (ProXML) Bill Roberts (SWIRRL) Rick Moynihan (SWIRRL)
<b>Reviewer(s)</b>	Bill Roberts (SWIRRL)

<b>Abstract:</b>	This deliverable list the final software components delivered as a result of the OpenGovIntelligence project.
------------------	---

## Effort of Participating Partners

	<i>Name</i>	<i>Short Name</i>	<i>Role</i>	<i>Person Months</i>
1.	Centre for Research & Technology - Hellas	CERTH	Participant	7
2.	Delft University of Technology	TU Delft	None	0
3.	National University of Ireland, Galway	NUIG	Leader	17
4.	Tallinn University of Technology	TUT	None	0
5.	ProXML bvba	ProXML	Participant	1.75
6.	Swirrl IT Limited	SWIRRL	Participant	7
7.	Trafford council	TRAF	None	0
8.	Flemish Government	VLO	None	0
9.	Ministry of 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

<i>Version</i>	<i>Date</i>	<i>Revised by</i>	<i>Reason</i>
0.1	06-Sep-2018	A. Stasiewicz (NUIG)	Initial Draft
0.2	20-Sep-2018	A. Stasiewicz (NUIG)	Tools list update
0.3	24-Oct-2018	A. Stasiewicz (NUIG)	General updates
0.4	26-Oct-2018	L. Porwol (NUIG)	General updates
0.5	31-Oct-2018	A. Stasiewicz (NUIG)	Addressing internal review comments
1.0	31-Oct-2018	CERTH	Submission to EC

**Statement of originality:**

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

## Table of Contents

<b>DELIVERABLE FACTSHEET .....</b>	<b>2</b>
<b>EFFORT OF PARTICIPATING PARTNERS.....</b>	<b>3</b>
<b>REVISION HISTORY .....</b>	<b>4</b>
<b>TABLE OF CONTENTS .....</b>	<b>5</b>
<b>LIST OF ABBREVIATIONS.....</b>	<b>6</b>
<b>EXECUTIVE SUMMARY .....</b>	<b>7</b>
<b>1 LIST OF SOFTWARE COMPONENTS .....</b>	<b>8</b>
1.1 CUBIQL - A GRAPHQL SERVICE FOR QUERYING LINKED DATA CUBES.....	9
1.2 JSON API FOR DATA CUBE (SPECIFICATION & IMPLEMENTATION).....	9
1.3 TABLE2QB AND GRAFTER.....	10
1.4 DATA CUBE BUILDER .....	10
1.5 ASSISTED CUBE SCHEMA CREATOR .....	10
1.6 DATA CUBE RDF VALIDATOR .....	10
1.7 DATA CUBE AGGREGATOR .....	11
1.8 LOSD MACHINE LEARNING COMPONENT .....	11
1.9 DATA CUBE EXPLORER .....	11
1.10 SPARQL CONNECTOR FOR EXPLORATORY.....	12
1.11 DATA SCIENCE STUDIO SPARQL PLUGIN .....	12
1.12 PILOT SPECIFIC TOOLS.....	12
<b>2 CONCLUSION .....</b>	<b>13</b>

## List of Abbreviations

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

<i>Abbreviation</i>	<i>Description</i>
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 lists and provides access to the components delivered as the OpenGovIntelligence ICT tools. Specification details and information about usage are documented in Deliverable D3.6 – “Report on OpenGovIntelligence ICT tools” (referred to as D3.6). Listed components reflect the status of work of OpenGovIntelligence project in Month 33. Despite the fact, that the development in the frame of the OpenGovIntelligence project has finished, further updates to the selected tools are expected.

Based on the primary purposes, delivered tools are grouped in three main categories:

- (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.

In general, the developed tools cover parts of the OpenGovIntelligence Architecture and are part of related ecosystem of tools aligned with OpenGovIntelligence innovation framework for public service co-creation driven by the exploitation of Linked Open Statistical Data, which is documented in deliverable D2.2 – “OpenGovIntelligence Framework” (referred to as D2.2).

Please note, that OpenGovIntelligence ICT tools is a suite of both open source and commercial tools.

## 1 List of Software Components

Below we list the tools, which were delivered **during the project lifecycle** as well as links by which they can be accessed. In general, developed tools can be accessed at the GitHub repository available at: <http://github.com/OpenGovIntelligence>.

While the development of the majority of the tools were initialised during the OpenGovIntelligence project, some of them were using outputs of the DaPaaS<sup>1</sup> and OpenCube<sup>2</sup> projects as the technical starting point. Details are presented in Table 1. OGI ICT Tools , visualised in Figure 1. OpenGovIntelligence LOSD Tool Ecosystem and discussed further in D3.6 and Deliverable D4.5 - Pilots and Evaluation Plan – V3 (referred to as D4.5).

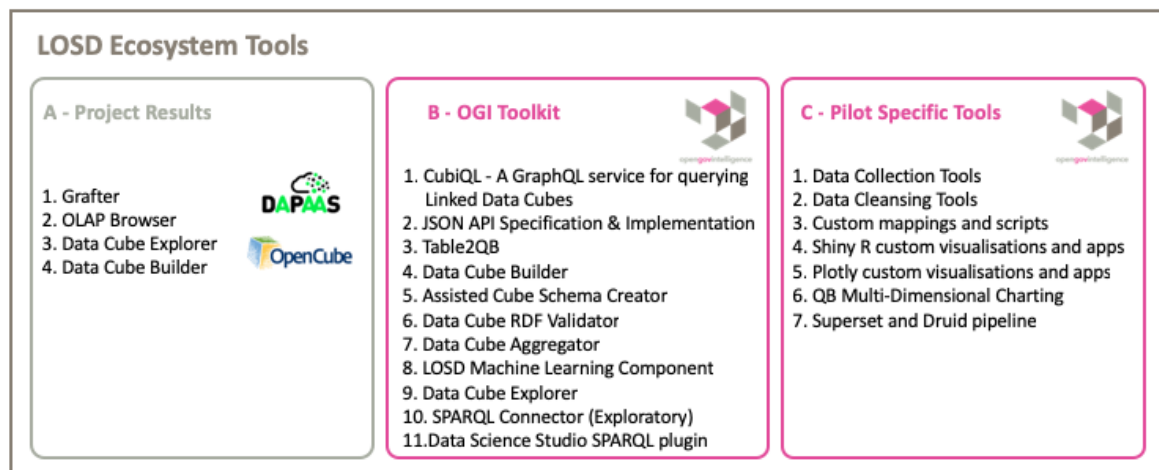


Figure 1. OpenGovIntelligence LOSD Tool Ecosystem

Table 1. OGI ICT Tools

No.	Name	Development in Year 3	Foundation
1	CubiQL - A GraphQL service for querying Linked Data Cubes	Yes	OpenGovIntelligence
2	JSON API For Data Cube (Specification & Implementation)	No	OpenGovIntelligence

<sup>1</sup> <https://project.dapaas.eu>

<sup>2</sup> <http://opencube-project.eu>



3	Table2QB And Grafter	Yes	DaPaaS
4	Data Cube Builder	No	OpenGovIntelligence
5	Assisted Cube Schema Creator	No	OpenGovIntelligence
6	Data Cube RDF Validator	Yes	OpenGovIntelligence
7	Data Cube Aggregator	Yes	OpenCube
8	LOSD Machine Learning Component	No	OpenGovIntelligence
9	Data Cube Explorer	Yes	OpenGovIntelligence
10	SPARQL Connector for Exploratory	Yes	OpenGovIntelligence
11	Data Science Studio SPARQL plugin	Yes	OpenGovIntelligence

### 1.1 CubiQL - A GraphQL service for querying Linked Data Cubes

CubiQL - A GraphQL service for querying Linked Data Cubes - standing for Cube Query Language and pronounced like 'cubicle' aims to (i) facilitate the development of Linked Open Statistical Data (LOSD) tools through a style of interaction familiar to web developers and (ii) offers a uniform way to access LOSD. The API can be installed on top of a RDF repository and offer basic and advanced operations on RDF Data cubes assuming that:

- i) they are stored using the RDF Data Cube Vocabulary,
- ii) they follow a specific application profile (common practices) and
- iii) are accessible through a SPARQL endpoint.

The returned results are in JSON format. The implementation adopted the GraphQL query language for APIs. This approach enables the flexible expression of data queries giving to the API clients the power to ask for exactly what they need and thus enabling the development of powerful tools.

Available at: <https://github.com/Swirrl/cubiql>

### 1.2 JSON API for Data Cube (specification & implementation)

**NOTE:** This work has been superseded by CubiQL, a GraphQL service for querying Linked Data Cubes, as described in Section 1.1 above. However, some of the OGI tools are (optionally) compatible with JSON-qb.

This component is the JSON QB specification provided by OpenGovIntelligence project together with its initial implementation. It aims to provide an easy to use API for web developers that use statistical data stored in the form of RDF Data cubes. The API implementation can be installed on top of any RDF repository and offer basic and advanced operations on RDF Data cubes.

**Available at:** <https://github.com/OpenGovIntelligence/json-qb>

**Available at:** <https://github.com/OpenGovIntelligence/json-qb-api-implementation>

### 1.3 Table2qb and Grafter

The Table2qb tool, takes data in a specific tabular structure, either as a CSV or Excel file, and converts it into an RDF Data Cube: representing the data as a series of observations with dimensions, attributes and measures, and generating the associated Data Structure Definition.

The table2qb tool builds on the more generic Grafter library, which was first developed in a previous EU funded project ('DaPaaS'). Grafter has been extended and improved during the OpenGovIntelligence project.

#### Availability

Grafter can be obtained from <https://github.com/Swirrl/grafter>.

Table2qb can be obtained from <https://github.com/swirrl/table2qb>.

### 1.4 Data Cube Builder

Data Cube Builder is a tool for transforming non-RDF data sources to RDF Data Cube. It is built on top of TARQL<sup>3</sup>. Data Cube Builder can be used through multiple interfaces such as desktop UI, command line, web user interface and as a web service.

**Available at:** <https://github.com/OpenGovIntelligence/data-cube-builder>

### 1.5 Assisted Cube Schema Creator

Assisted Cube Schema Creator is a tool created in order to simplify the RDF mapping process. It is built on top of spreadsheet concept, Open Refine and Data Cube vocabulary. Based on user inputs it generates RDF data structure definition used by e.g. Data Cube Builder.

**Available at:** <https://github.com/OpenGovIntelligence/qb-assisted-schema-creator>

### 1.6 Data Cube RDF Validator

Data Cube RDF Validator is a collection of integrated test cases to confirm compatibility of the data (e.g. generated using table2qb) with CubiQL, a GraphQL service for querying Linked Data Cubes.

---

<sup>3</sup> <http://tarql.github.io>

This tool run collection of test cases against a SPARQL endpoint. The endpoint can be either a HTTP(s) SPARQL endpoint or a file or directory on disk. Test cases can be specified as either a SPARQL query file, or a directory of such files.

The repository contains versions of the well-formed cube validation queries defined in the RDF data cube specification. These are defined as SPARQL SELECT queries rather than the ASK queries defined in the specification to enable more detailed error reporting.

**Available at:** <https://github.com/Swirrl/rdf-validator>

## 1.7 Data Cube Aggregator

The Data Cube Aggregator aggregates data across dimensions of a cube. The aggregate functions supported are the AVG, SUM, MIN, MAX and COUNT. This implementation aims in computing aggregations from raw RDF data. Thus, taking as input the raw RDF data it creates a cube that contains the corresponding aggregated observations.

**Available at:** <https://github.com/OpenGovIntelligence/json-qb-api-implementation>

(integrated with JSON-QB API)

## 1.8 LOSD Machine Learning component

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

**Available at:** <https://github.com/OpenGovIntelligence/qb-machine-learning-component>

## 1.9 Data Cube Explorer

Data Cube Explorer is a web-based tool that catalogues and presents details of available data cubes to the users. It provides search interface and enables user to preview cube data using different visualisation tools: pivot table, OLAP Browser<sup>4</sup> and Cube Visualiser.

**Available at:** <https://github.com/OpenGovIntelligence/data-cube-explorer>

---

<sup>4</sup> <https://github.com/OpenGovIntelligence/qb-olap-browser>

### 1.10 SPARQL connector for Exploratory<sup>5</sup>

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 JSON configuration file. It has become part of a largely deployed product. Details about the Exploratory extensions are available at: <https://docs.exploratory.io/import/extensions.html>

**Available at:** [https://github.com/OpenGovIntelligence/exploratory\\_sparql\\_plugin](https://github.com/OpenGovIntelligence/exploratory_sparql_plugin)

### 1.11 Data Science Studio SPARQL plugin

Data Science Studio of Dataiku Is a Collaborative Data Science tool for self-service analytics and machine learning. The 'Data Science Studio SPARQL plugin' developed during the project offers a connector to SPARQL endpoints as data source.

This plugin allows to add datasets from SPARQL SELECT queries to the DSS data science analytics flows. More info at: <https://www.dataiku.com/>

**The plugin is available at:** [https://github.com/OpenGovIntelligence/datasciencestudio\\_sparql\\_plugin](https://github.com/OpenGovIntelligence/datasciencestudio_sparql_plugin)

### 1.12 Pilot specific tools

In majority of the Pilots trials, there was a need to develop additional, Pilot Specific Tools in order to fulfil the use scenarios requirements. In general, those tools covers parts of the OpenGovIntelligence Architecture and are part of related ecosystem of tools aligned with OpenGovIntelligence innovation framework. However, due to nature of the requirements and in some cases its unique functions, it is not possible to clarify Pilot Specific Tools as a part of OGI Toolkit. These tools are described in more details at D3.6.

---

<sup>5</sup> <https://exploratory.io>

## 2 Conclusion

This deliverable lists the components delivered in the final phase of OpenGovIntelligence project (Month 33). The detailed descriptions of the new and updated components are provided in the Deliverable D3.6 – “Report on OpenGovIntelligence ICT tools”.

In the final stage of the development, most of the efforts were focused on the development of a CubiQL: new data access methods, performance optimisation, user documentation improvements, examples and validation tools.

Despite the fact, that the development in the frame of the OpenGovIntelligence project has finished, further updates to the selected tools are expected.