



## Open Budgets

### Fighting Corruption with Fiscal Transparency

Project Number: 645833

Start Date of Project: 01.05.2015

Duration: 30 months

## Deliverable 4.2

# Analysis of the Required Functionality of OpenBudgets.eu

Dissemination Level	Public
Due Date of Deliverable	Month 2, 30.06.2015
Actual Submission Date	03.08.2015
Work Package	WP 4, OpenBudgets.eu Requirements, Platform Architecture Integration and Development
Task	T 4.1
Type	Report
Approval Status	Draft
Version	0.4
Number of Pages	22
Filename	D4.2 - Analysis of the required functionality.docx

**Abstract:** A report defining the required functionality of OpenBudgets.eu for acquiring, transforming, storing, viewing, analysing, sharing and publishing of data. This report will include the system architecture and high-level information interchange between the mentioned functions

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	29.06.2015	Kick-off Meeting	Sören Auer
0.2	15.07.2015	Internal WP4 Meeting	Fahrettin Gökgöz
0.3	20.07.2015	Internal WP4 Meeting	Fahrettin Gökgöz
0.4	20.07.2015	Wording update	Jaana Takis

## Author List

Organisation	Name	Contact Information
UBONN	Fahrettin Gökgöz	gokgozf@gmail.com
Fraunhofer	Sören Auer	soeren.auer@iais.fraunhofer.de
Fraunhofer	Jaana Takis	jaana.takis@iais.fraunhofer.de



---

# Executive Summary

[Text, no more than 300 – 500 words]



---

## Abbreviations and Acronyms

<b>API</b>	Application Programming Interface
<b>BDP</b>	Budget Data Package
<b>CSV</b>	Comma Separated Values
<b>Data Store</b>	A data store is a repository of a set of data objects.
<b>LOD</b>	Linked Open Data
<b>OLAP</b>	OLAP is an acronym for online analytical processing. OLAP is a computer-based technique for analysing business data in the search for business intelligence.
<b>QB</b>	Data Cube Vocabulary
<b>RDBMS</b>	Relational Database Management System
<b>RDF</b>	Resource Description Framework
<b>REST</b>	Representational State Transfer
<b>S3</b>	Simple Storage Service
<b>SRS</b>	Software Requirement Specification
<b>Triple Store</b>	A triple store is a framework used for storing and querying RDF data. It provides a mechanism for persistent storage and access of RDF graphs.



---

# Table of Contents

<b>1</b>	<b>INTRODUCTION</b>	<b>7</b>
1.1	PURPOSE	7
1.2	SCOPE	7
1.2.1	Objectives	7
1.3	OVERVIEW	8
<b>2</b>	<b>DESCRIPTION</b>	<b>9</b>
2.1	PRODUCT PERSPECTIVE	9
2.1.1	Detailed Methodology	9
2.1.2	System Interfaces	10
2.1.3	Memory Constraints	10
2.1.4	Operations	10
2.1.5	Site Adaptation Requirements	10
2.2	USER CHARACTERISTICS	10
2.3	FUNCTIONAL REQUIREMENTS SPECIFICATION	11
2.3.1	Journalism Use Case	11
2.3.2	Transparency Use Case	12
2.3.3	Participation Use Case	13
<b>3</b>	<b>SPECIFIC REQUIREMENTS</b>	<b>14</b>
3.1	FUNCTIONAL REQUIREMENTS	14
3.2	NON-FUNCTIONAL REQUIREMENTS	21
3.3	DESIGN CONSTRAINTS	22
3.3.1	Standards Compliance	22
3.4	SOFTWARE SYSTEM ATTRIBUTES	22
3.4.1	Reliability	22
3.4.2	Availability	22
3.4.3	Maintainability	22
<b>4</b>	<b>CHANGE MANAGEMENT PROCESS</b>	<b>22</b>
<b>5</b>	<b>REFERENCES</b>	<b>22</b>
<b>6</b>	<b>APPENDICES</b>	<b>22</b>
6.1	NAMING CONVENTION FOR REQUIREMENTS	22



---

## List of Figures

Figure 1 : Current OpenBudgets Architecture Blueprint..... 10  
Figure 2: Journalism Use Case..... 11

## List of Tables

**No table of figures entries found.**



---

# 1 Introduction

Fiscal transparency is at the core of the OpenBudgets.eu project. Whilst the lack of transparency causes distrust in the public administration and is the enabler of corruption, then transparency on the other hand enhances accountability and fights corruption. It is the governments especially who should be setting the example in this as a way of managing the trust of its people. Hence, for any government striving towards participatory democracy, fiscal transparency is the precondition to public engagement on issues like budget priorities and public spending.

In this project the aim is to develop a generic framework and concrete tools for supporting financial transparency, which will serve as a clearinghouse for financial data at all levels of public administration (i.e. local, regional, national, European and international levels). The project will build on and integrate existing initiatives (OpenSpending.org, OffenerHaushalt.de etc.) with a similar objective and provides a one stop shop for European citizens on information about allocated budgets and spending. The project will define a semantic knowledge model for budgets and spending and thus enable the integration of corresponding data at different levels of administration. Each item will be available for discussion and debate by the public. The framework will provide a toolbox for the creation and sharing of budget and spending visualizations, comparative data analysis and the discovery of trends. Governments and public administrations can hence benefit from receiving targeted suggestions, comparative benchmarks and scenarios.

## 1.1 Purpose

The purpose of this SRS document is to provide a description of the requirements for the OpenBudgets.eu and understanding the functionalities demanded by end users (the stakeholders). The foundations of the project are provided in this SRS, and it can be used to design and construct the OpenBudgets.eu.

This SRS will be a guide to Software engineers, research teams and consultants building on the OpenBudgets.eu. The Stakeholders can also use this SRS to observe the direction of this project and to what degree their expectations are met.

## 1.2 Scope

The OpenBudgets.eu focuses on the analysis and comparison of public finance data, typically published as part of governmental Open Data initiatives. Its scope is currently limited to three pilot use cases, targeting three different applications related to public spending: journalism, anti-corruption initiatives and private citizenship engagement.

### 1.2.1 Objectives

The framework and specific tools developed as part of OpenBudgets.eu must satisfy the following objectives:

- (1) Financial Data publishing and integration employing Linked Open Data.

This enables public administrations, in the role of data providers, to publish financial data in a reusable, machine-processable, linked-data format. Aside from enabling its integration and re-use across multiple domains, the use of a standard format also facilitates the comparison of data from different municipalities or regions. More importantly, it allows all the stakeholders involved or interested in budget planning or spending, to interpret the data using the same tools and methods, thus supporting financial transparency in public budgeting and spending.

- (2) Exploratory and comparative Financial Data Analysis



---

The provision of visualisation, analytics and exploration tools offer the different stakeholders an opportunity to scrutinize and interpret financial data related to a region of interest, to compare allocations and transactions between multiple regions, to visualise detected trends and budget projections and investigate anomalies and activities, which have been flagged as suspicious.

(3) Interactive Budget Management

This involves the stakeholders, particularly the local citizens, in the budget planning and auditing phases, by allowing them to interact with the process by providing opinions and suggestions on setting budget priorities, providing feedback on the published transactions, and generally aim for participatory budget management.

(4) Comprehensive OpenBudgets Framework

The development of a suitable framework, which follows the privacy-by-design paradigm and integrates the required analysis tools, data formats, APIs and user interfaces for the different stakeholders to publish and consume open budget data centrally, easily and intuitively.

(5) Cross-domain Application

This applies the OpenBudgets.eu solution to a number of pilot studies in the data journalism, transparency and participatory budgeting, in order to demonstrate the potential impact of this initiative within multiple related domains.

(6) Sustainable establishing of OpenBudgets.eu as a Software-as-a-Service

Realises a comprehensive financial transparency platform for public administrations and stakeholders with defined Service Level Agreements and an operational business model for the longterm sustainability of the platform beyond the actions end.

## 1.3 Overview

This SRS is organized as the following - in the second chapter the overall description of the project is given from the high level perspective and the third chapter goes into more (technical) details.





---

## 2 Description

In this section the general architecture and background for the requirements are introduced.

### 2.1 Product Perspective

#### 2.1.1 Detailed Methodology

The high level architecture is based on open sourcing.

The given architecture in the Figure 1 is explained in the following steps

- **Acquisition, Clean, Load:** Data for the budgets could come from different sources and in different formats, gathering the data and cleaning, and loading to the system is done by this module
- **Write API + UI:** Between cleaning the data and loading to the system, it is necessary to post process the cleaned data to compute defined analytics. Such as access controls, loggings, revisions, and etc. Write API + UI module is responsible for these tasks
- **Data Store:** Persistence of the data is an important step. Gathered data will be stored into provided data stores, relational data bases, triple stores, or flat files. Once the data is loaded to data store, it enables more than one option to follow. Analytics module, or Read API are the ones that are depend on the data store module.
- **Analytics:** Analytics module is where more data inside is produced. In this module arbitrary queries can be done on the data, and also the aggregation by given dimensions is possible. Additionally text search service is provided by this module
- **Read API:** Read API is the module where the platform reaches outside world. In this module, search, and aggregate components are responsible for using either directly the data store for raw data, or analytics for more complex data. REST API is providing the processed data for visualization, browsing & searching, and analysis & Presentation.
- **Browse & Search:** Once the data is processed and read through the read API, now each transaction can be browsed, using browse transaction component, or can be searched using text search component with in the Browse & Search module.
- **Visualization:** Visualization module depends on read API and Data store, and provides variety of visualization methods to data. Additionally the module offers newcomers guidance with the how to use the API component.
- **Analysis & Presentation:** Although the analytics and visualizations are done in the previous modules, there is still a need for another layer to combine all the knowledge and present. Analysis & Presentation module is responsible for creating proper representation of the data

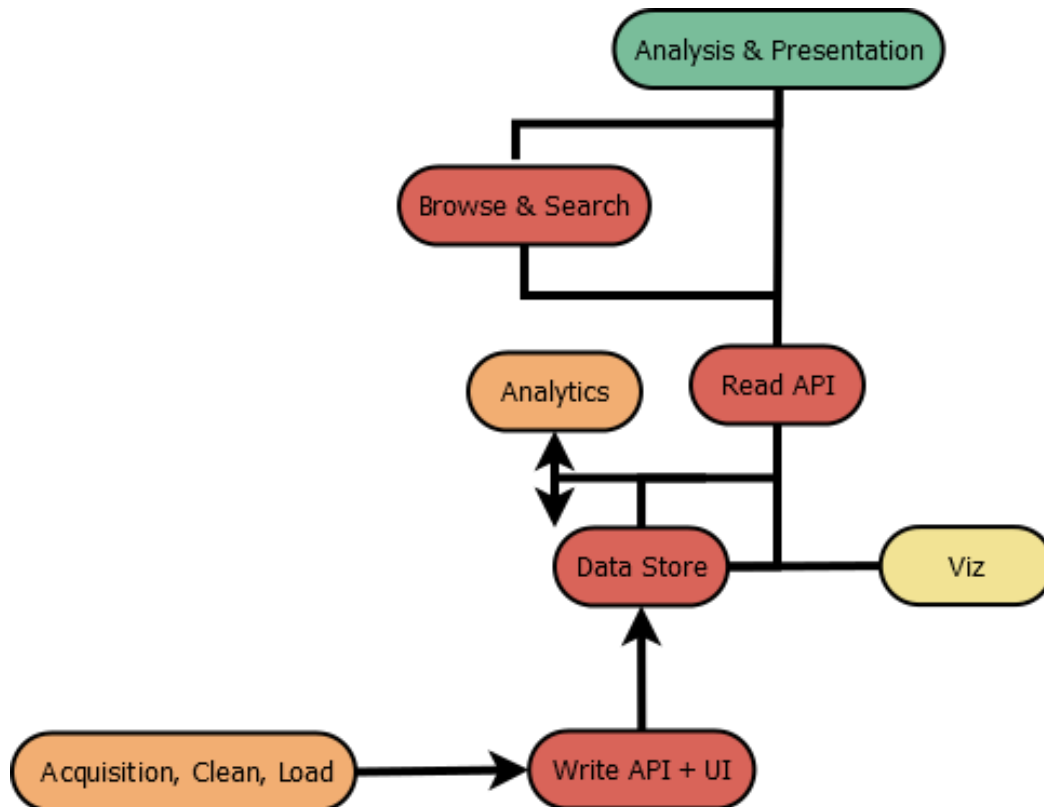


Figure 1 : Current OpenBudgets Architecture Blueprint

## 2.1.2 System Interfaces

To be defined as part of D4.1.

## 2.1.3 Memory Constraints

- User : ~2GB
- Developer: ~4-8GB
- Organization: ~>8 [Depending on the data size]

## 2.1.4 Operations

Operations are not defined

## 2.1.5 Site Adaptation Requirements

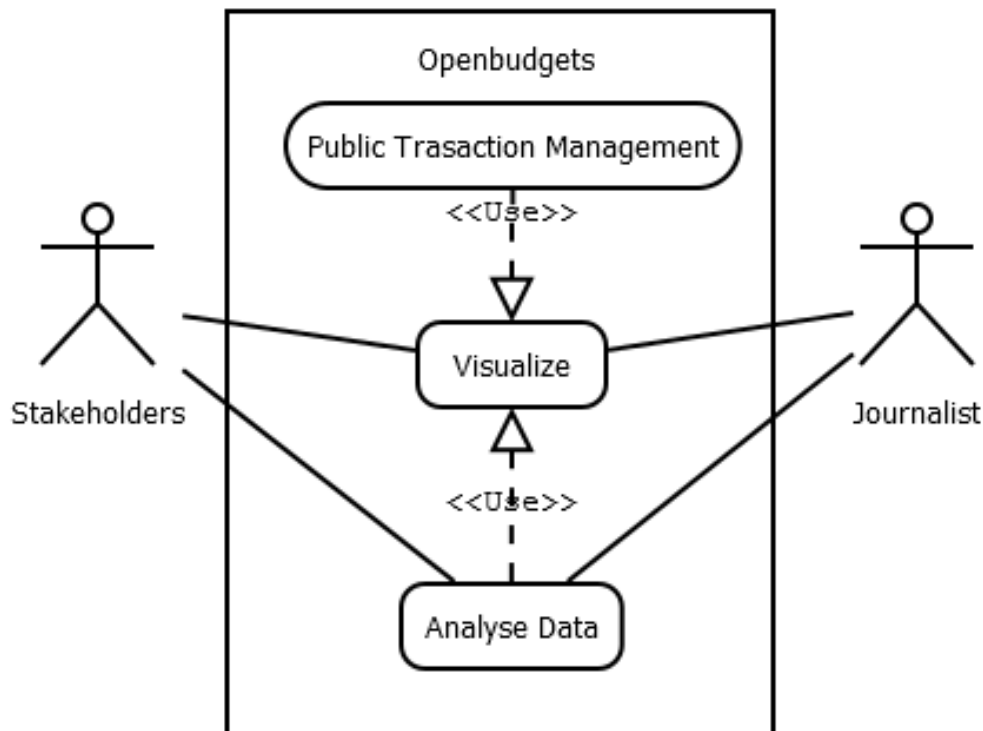
Site Adaptation Requirements are not defined

## 2.2 User Characteristics

- Educational : Citizens
- Experience : Public Administrators
- Technical : Developers and Scientists

## 2.3 Functional Requirements Specification

### 2.3.1 Journalism Use Case



**Figure 2: Journalism Use Case**

Journalism focuses particularly on the elements of fostering debate, creation of narrative around the numbers contained in budgets and spending data, and examining incentive structures to encourage citizens to contribute to community-driven efforts.

The Journalism use case is about making the budget and spending decisions easy to understand and to communicate to the general public. Creating and sharing visualizations, producing comparative data analyses with a well-defined toolbox is one of the ways to achieve these goals and this is where the scope of the OpenBudgets.eu is limited to within this use case.

Therefore, the following items are of main interest to the journalists

- Enhancing accountability within public sectors and as a result increase the leverage of civil society on the administrative decision process;
- Augmenting citizens' trust in public administration;
- Comparative analysis for better budget management;
- Enabling all stakeholders to visualize public transactions.

### 2.3.2 Transparency Use Case

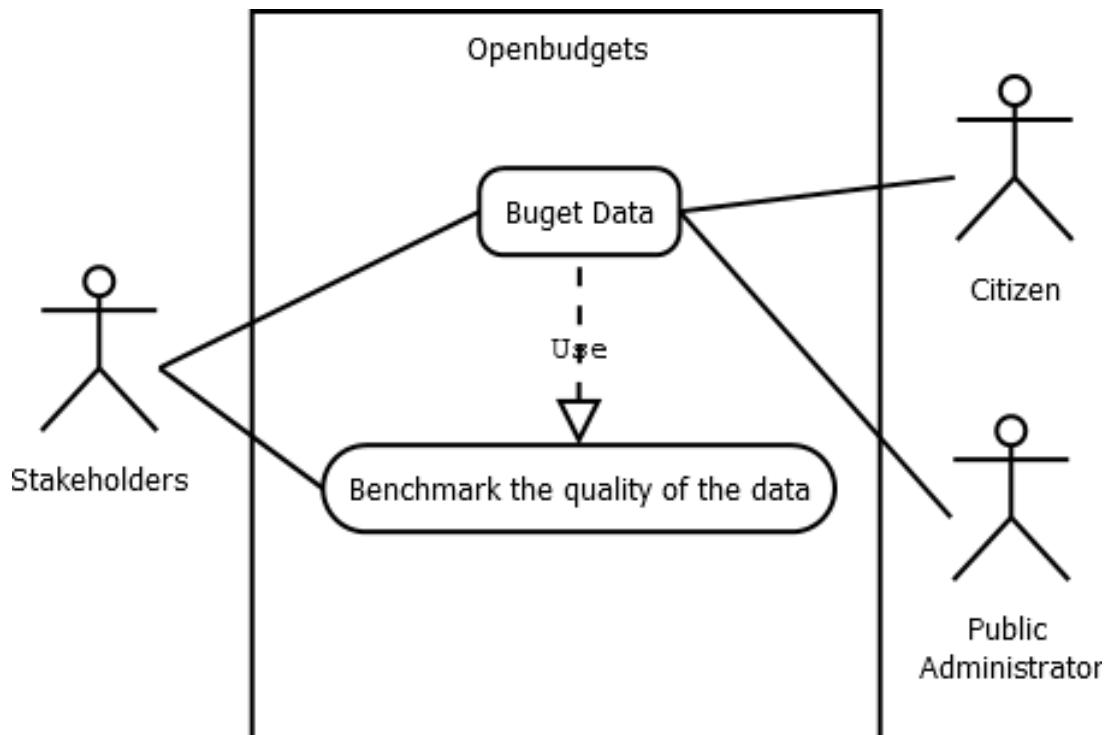


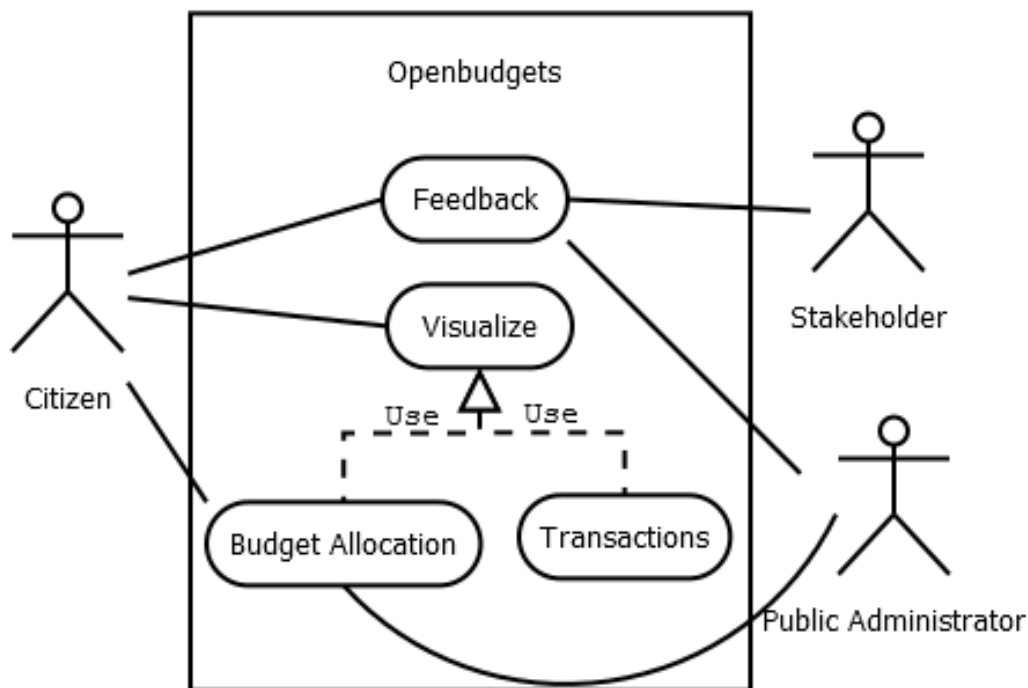
Figure 3: Transparency Use Case

Civil society organizations do not just use existing open budget platforms to analyze budgets or to investigate particular cases. Open budget platforms are also important ways of advocating for more open and better-structured budget and spending information provided by public authorities that can populate such platforms. The aim of the use case is to create an ‘advocacy test bed’ which will assess the needs of one specific set of end-users of the platform – EU policy makers – and incorporate this feedback into the development of the platform, thereby helping to build political support for this platform in particular and open budget data in general.

The main objectives are:

- Understand the needs for EU policy makers – with a focus on Members of the European Parliament (MEPs) - on EU budget and spending information;
- Benchmark the scope and quality of data available to EU policy makers;
- Understand the political, legislative and administrative obstacles to provision of EU budget information on an open data basis;
- Articulate the main changes to legislation or implementation that need to occur to fulfil the needs of EU policy makers and other actors working at EU level.

### 2.3.3 Participation Use Case



**Figure 4: Participation Use Case**

In most cases budget preparation and approval is a closed-door process within administrations. Citizens and other stakeholders usually stay outside the process impeded by factors such as lack of resources to influence decisions, lack of understanding of the different budget concepts and line items, and ignorance of the approval process. On the other side, public administrations, especially at local and regional level, perceive participatory processes as high cost and difficult to manage experiments. Implementing tailor-made digital solutions in small and medium size cities/regions can be expensive and difficult to implement, particularly in an expenditure-reduction context. The focus of the participation case is to identify the solutions for both stakeholders and public administrations. The final aim is to enable real participatory experiences and decision-making processes, thus strengthening the democratic process.

- A tool where citizens can express their budget allocation priorities during the budget approval process, along the lines and within the process defined by each administration concerned. This tool will fundamentally target municipalities;
- A tool where citizens can monitor budget transactions, auditing budget compromised vs. actual spending and giving feedback to the administrations;
- Educational resources for citizens, providing online materials to understand i.e. the budget cycle, terms used or how to influence and monitor the budget. Special attention will be given to:
- Ensuring transparency during the participation process, so every stakeholder can access information about the feedback sent and its effective impact in the decisions made;
- Increasing competences and capabilities, developing the necessary resources for the different participants to make informed decisions;
- Providing structured and valuable information to public administrations, avoiding fragmented or too broad feedback, which does not contribute to making decisions.



## 3 SPECIFIC REQUIREMENTS

Sufficient level of details to design and develop the system according to requirements is provided in this section. Each of the stated requirements should be available to users, administrators, or external systems.

### 3.1 FUNCTIONAL REQUIREMENTS

S. No	Features	Functionality	Description
F001	<b>Data Model</b>	RDF data structure for budgets	Data Cube Vocabulary (QB) to define budget information needs to be defined
F002		RDF data structure for transactions	Data Cube Vocabulary (QB) to define transaction information
F003		Mapping OpenSpending Data Package to RDF	Openspending data to defined vocabulary mapping
F004		Curation for Code lists	Code List management system
F005		Modelling of code lists in RDF	To be based on SKOS or some other.
F006		Data structure for modelling revenues/incomes	Data structure for describing financial data, based on light-weight ontologies.
F007		Budget process model	Budget, and especially participatory budgeting is a process with a strong workflow characteristic. It could help users to understand data if they could see it as a consequent tasks/flows. Ex: Spending planning / expected results > transactions/actual spending > achieved results
F008		Code lists' mappings support	Any localization or external mappings.
F009		Ability to model payer, payee, amount, date, currency	Support for modelling properties of transactions.
F010		Ability to attach concrete targets to spending	Example: if "\$10.000 budget for health" is the budget entry, "building 10 hospitals" would be a concrete target, or "reducing 50% cases of some disease". After the conclusion of the spending (transaction), these targets could be validated by the community.



F011		Link ability	All data items must be linkable, i.e. have URIs. For example, it is necessary to link a spend item to budget line that justifies it.
F012	Data acquisition, loading semantic lifting	Loading of CSV	The platform should support the loading of Comma Separated Value (CSV) files with a columnar structure. It should be possible to select a region in the CSV, which contains the data to be loaded.
F013		Semantic Mapping of CSV	The acquired CSV files should be annotated with mappings allowing them to be transformed into RDF.
F014		Acquisition of metadata	For each loaded data package, metadata such as the public administration the data refers to, the year, the uploading user, the relationship to other datasets etc. should be captured.
F015		Loading from an API	Not all data is in downloadable form only, some can be acquired over a link (saves on manual downloading). Hence one needs a mapping for live data in this case as well.
F016		Visual exploration (RDF)	Graph visualization for relationship identification
F017	Exploration, search	Non Semantic exploration	Faceted browser or a tabulated view of the dataset
F018		Free-text search	Entering a keyword and getting matching results within the dataset <ul style="list-style-type: none"><li>• Across different datasets</li><li>• On particular attributes only</li><li>• On fields names/labels</li></ul>
F019		Semantic search	<ul style="list-style-type: none"><li>• SPARQL endpoint for advanced users</li><li>• Pre-stored common set of queries (which</li></ul>



			<p>can be configured with parameters)</p> <ul style="list-style-type: none"><li>• User-friendly interface for non-SPARQL experts</li></ul>
F020		Exploration of processed datasets	Using an “Aggregate API” it should be possible to search over the results of analysis and data aggregates generated at earlier stages.
F021		Metadata	It should be possible to search over additional metadata (e.g. provenance) attached to datasets
F022		Different levels of difficulty in queries	<ul style="list-style-type: none"><li>• lowest level -&gt; drop down menu with base questions and off the shelf visuals</li><li>• intermediary -&gt; access data, filter, pivot tables, queries</li><li>• Advanced -&gt; write on SQL queries and codes, tweak tools, work with data</li><li>• Linked -&gt; Sparql queries, mappings etc.</li></ul>
F023		User-Friendliness of the UI	<ul style="list-style-type: none"><li>• User friendly search interface accessible</li><li>• Advanced level of the user interface</li></ul>
F024		Relevance ranking	Free-text search results will be ordered according to relevance measure.
F025	Visualization	Build custom visualizations	Users can customize the visualizations they build, by selecting columns and relations existing in one or more datasets
F026		Exploit analysis	Provide an interface to make use of the full power that Analysis Tools offer. Let the users decide how to combine data and hide away complex queries





F027		Provide samples	Give beginners some templates to start playing, without requiring them to read any documentation. Give helpful hints and maybe a tutorial. Samples should ideally be real, successful news stories
F028		Suggest first	Don't let the users browse a huge set of visualizations, especially at their first steps. Offer insight on what visualizations would fit their selected data, if they chose to manually select the columns they want to graph.
F029		Do not repeat	Keep track of the previous user's moves in the interface. Let them repeat the same process with updated data
F030		Export and share	Make the information shareable and usable outside of OpenBudgets, including printing
F031		Good quality visualizations	So they can be used in printed media (e.g. by the journalists)
F032		Indisputable visualizations	Provide mechanisms to verify that a visualization came from data provided at a certain point of time by a certain source
F033		Provide geo visualizations	If data has geographical information (coordinates, neighbourhood, city, state), show on maps, together with other relevant geographic information.
F034		Point missing data	Based on previous uploaded data, point missing datasets (e.g.: uploaded 2013 and 2015 budget, point that 2014 is missing)
F035		Preview	Let the users preview what they are going to see
F036	Analytics	Filtering commensurable objects	Aggregate analytics can only operate on a pool of commensurable objects (i.e. objects with comparable "size", in whatever terms).



			The platform should be able to serve data using appropriate filters, e.g., budgets of municipalities with similar population size.
F037		Version tracking of budgets	Analysis of evolution of budgets throughout its preparation phase
F038		Indexing data w.r.t. tabular vs. graph structures	For some types of data, mining from tabular structures (merely enriched by further features) is sufficient. OTOH, some “natively graph-based” data might rather work on graph structures. Each kind of structures would benefit from specific optimized indexing scheme, to assure real-time response.
F039		Outlier detection	Reveal categories that are used disproportionately. Outlier detection can find classification “black holes”, where lot of spending is non-transparently classified.
F040		Extrapolations on data	Ability to outline trends for future budget allocations
F041		Aggregation by time interval	Ability to aggregate (e.g., sum, average) amounts over a user-defined period of time (e.g., quarter).
F042		Temporal trend of the difference between planned and actual spending	How does the difference between planned and actual expenditure differ over time? If it gets smaller, does it imply that the public body improved its estimates?
F043	API	Entity comparison	Query for multiple datasets that match on one or more entities (e.g.: Paris & Athens; Belgium & Germany)
F044		Temporal comparison	Query for “budget” or “spend” over time
F045		Functional comparison	Query for (e.g.: education) over entities and time
F046		Filter by spend	Query for all budget lines in (e.g.: Greek 2015 budget) over (e.g.: 10,000 euro)



			spend
F047		Filter by administration type	Query for all (e.g.: budgets) of (e.g.: municipality; municipal department)
F048		Get top-level aggregates	Query for total (e.g.: budget) across all (e.g.: countries) in (e.g.: years 2010 - 2015)
F049		Normalize by key metrics	By population (also breakdown pop. by gender and age); By schools; and so on
F050		Real vs. Nominal	Adjustments for inflation, etc.
F051		Localized data	Return translated data? e.g.: titles for entities, budget lines
F052	<b>SAAS Interface</b>	Kiosk mode	Activity report, and ability to manage software including the lock down
F053		Fully-customizable CI	Customizable continuous integration to have ready to deploy working copy
F054	<b>Use Journalism</b>	Download button all the way	Journalists will need to be able to download and store the data on their own servers at every step of the analysis
F055		Contextual information	What the budget-holder is responsible for, what is on-and-off budget, what does the data say in relation to population and Eurostat data
F056		Proper documentation	Methodology, sources, how the mapping has been done. Annotation up to dataset level
F057		Provenance	Provide strong and clear provenance information, and encourage journalists to explain where data comes from
F058		Red Flag	For data that is not available
F059	<b>Use Transparency</b>	Links to FoI/ATD Tools	Link to ask the AsktheEU.org, fragdenstaat etc. Freedom of information act /Access to Documents



F060		Break Down functionality	Break down into major categories, institutions, etc.
F061		Clear licensing information	To encourage reuse of visualizations or data.
F062		Query by institution, administrative regions	Filtering per dataset as well as aggregates of all data that refers to the institution
F063	<b>Use Participation</b>	List of available investment alternatives	Municipalities will create and update a list of potential investment options for users to pick. (Can citizens propose their own? Not known yet.)
F064		Gathering votes	The platform must be able to store votes from citizens to the proposed investment alternatives. And potentially comments/feedback on these items.
F065		Displaying results	Votes are counted according to the agreed process (see open questions above) and then displayed to other users and to the municipality.
F066		Attach Targets to spending	For citizens, besides spending, it is also important (maybe more important) to see the results of spending, e.g., hospitals built or disease index reduced. For participatory budgeting, it is important to attach spending to concrete results, so that it can be later scrutinized



### 3.2 Non-Functional Requirements

S. No	Features	Description
N001	Ease of Use	Ease of use of accessibility and functionality. Wizards.
N002	Localized labels	Journalists etc. might want to use the figures in articles
N003	Scalability	so the framework is open to further developments in the future
N004	Load balancing	So that data acquisition does not influence how many concurrent users can analyse / view data in a timely manner.
N005	Intuitiveness	Results are easily interpreted and compared
N006	Seamless integration with other platforms	Creating interfaces to connect other known platforms
N007	Responsiveness	Main functionality should be available from mobile devices.
N008	Clear coding	Implement coding standards and promote regular refactoring for further maintainability.
N009	Usage monitoring	Applying tools like Piwik or GA to monitor tool usage and identify problem areas
N010	Bug reporting / feature requests	Environment for bug reporting, or requesting additional features.
N011	Public documentation	Of code functions etc. needed for building a community around the project.
N012	Feature requests	Implement tools (e.g. UserVoice) to prioritize further features to be added.
N013	Quality assessment	conduct quality assessment over the transformed data



---

### 3.3 Design Constraints

The system shall be developed using Open-Source software technologies

#### 3.3.1 Standards Compliance

International Standards for Software Development are followed during the creation of the document are listed below

- **SRS** – Software Requirements Specification: IEEE 830
- **SDS** – Software Design Specification Document: IEEE 1016
- **STD** – Software Test Documentation: IEEE 829
- **SUD** – Software User Documentation: IEEE 1063

### 3.4 Software System Attributes

Objectively verifying product is a process needed to be done in many perspectives, and software system attributes play a role as a requirement in this process.

#### 3.4.1 Reliability

The designed system - at the time of delivery – should meet the required reliability

#### 3.4.2 Availability

The designed system shall be available, up and running throughout the year except the maintenance activities.

#### 3.4.3 Maintainability

The designed system shall have easy operation and management activities throughout its life cycle

## 4 Change Management Process

Changes to this document may be made after approval

## 5 REFERENCES

**There are no sources in the current document.**

## 6 Appendices

### 6.1 Naming Convention for requirements

- Functional requirements F[Number of the requirement]
- Non-functional requirements N[Number of the requirement]