Small or medium-scale focused research project (STREP)

ICT SME-DCA Call 2013
FP7-ICT-2013-SME-DCA

Data Publishing through the Cloud:
A Data- and Platform-as-a-Service Approach to Efficient
Open Data Publication and Consumption
DaPaaS

Deliverable D5.2

Use case implementation, v1

<table>
<thead>
<tr>
<th>Date</th>
<th>31.10.2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Ivan Berlocher (Saltlux), Seonho Kim (Saltlux), Tony Lee (Saltlux)</td>
</tr>
<tr>
<td>Dissemination level</td>
<td>PU</td>
</tr>
<tr>
<td>WP</td>
<td>WP5</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
</tbody>
</table>
Document metadata

Quality assurors and contributors

<table>
<thead>
<tr>
<th>Quality assuror(s)</th>
<th>Amanda Smith (ODI) and Brian Elvesæter (SINTEF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contributor(s)</td>
<td>DaPaaS Consortium</td>
</tr>
</tbody>
</table>

Version history

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>30.09.2014</td>
<td>Initial outline and Table of Contents (TOC)</td>
</tr>
<tr>
<td>0.2</td>
<td>27.10.2014</td>
<td>Ready for internal peer review</td>
</tr>
<tr>
<td>0.3</td>
<td>30.10.2014</td>
<td>Integrated version with suggested edits from the internal peer review</td>
</tr>
<tr>
<td>0.4</td>
<td>31.10.2014</td>
<td>Revision addressing reviewers’ comments</td>
</tr>
<tr>
<td>1.0</td>
<td>31.10.2014</td>
<td>Final formatting and layout</td>
</tr>
</tbody>
</table>
Executive Summary

The main goal of the DaPaaS project is to provide an integrated Data-as-a-Service (DaaS) and Platform-as-a-Service (PaaS) environment, together with associated services, for open data, where 3rd parties can publish and host both datasets and data-driven applications that are accessed by end user data consumers in a cross-platform manner.

This report represents Deliverable D5.2 "Use case implementation, v1" of the DaPaaS project. Based on the outcome of D1.2, D2.2 and D4.1 this deliverable provides the details of the implementation of the DaPaaS use case.

As outlined in the previous version of this document - Deliverable D5.1 "Use case definition and requirements analysis", the use case implements PLUQI (Personalized and Localized Urban Quality Index). PLUQI is a customizable index model that can be used to represent and visualize the level of well-being and sustainability for given cities based on individual preferences.

The PLUQI prototype has been implemented as a web application using the datasets and APIs from the DaPaaS platform. This implementation illustrates the semantic integration of heterogeneous open data from the Korean Statistical Information Service (KOSIS)¹ and data from social sensors linked through the CITI-SENSE² platform.

---

¹ [http://kosis.kr/](http://kosis.kr/)
² [http://www.citi-sense.eu/](http://www.citi-sense.eu/)
Table of Contents

EXECUTIVE SUMMARY ................................................................................................................................... 3
TABLE OF CONTENTS...................................................................................................................................... 4
LIST OF ACRONYMS......................................................................................................................................... 5
LIST OF FIGURES .............................................................................................................................................. 6
LIST OF TABLES ................................................................................................................................................ 7
1 INTRODUCTION ....................................................................................................................................... 8
2 PLUQI USE CASE PROTOTYPE ............................................................................................................ 9
   2.1 DATASETS ................................................................................................................................................. 9
   2.2 ONTOLOGY MODEL .................................................................................................................................. 10
   2.3 INDICES ................................................................................................................................................... 12
       2.3.1 Z-Score .......................................................................................................................................... 13
   2.4 WEB APP ................................................................................................................................................. 14
       2.4.1 Home Page .................................................................................................................................... 14
       2.4.2 Rank of Cities Page ....................................................................................................................... 15
       2.4.3 Compare Cities Page ..................................................................................................................... 16
   2.5 APPLICATION API .................................................................................................................................... 17
       2.5.1 Get City Data................................................................................................................................. 17
       2.5.2 Get Sub Index Details ................................................................................................................... 18
       2.5.3 Get City Rank ................................................................................................................................ 19
       2.5.4 Compare Cities .............................................................................................................................. 19
   2.6 OPEN SOURCE LIBRARIES USED .............................................................................................................. 21
   2.7 APPLICATION DEPLOYMENT .................................................................................................................... 21
3 FUTURE WORK ....................................................................................................................................... 22
4 APPENDIX A: REQUIREMENTS ADDRESSED BY PLUQI............................................................. 23
### List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>CSV</td>
<td>Comma Separated Values (format)</td>
</tr>
<tr>
<td>DaaS</td>
<td>Data-as-a-Service</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>HTTPS</td>
<td>Hypertext Transfer Protocol Secure</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation (format)</td>
</tr>
<tr>
<td>PaaS</td>
<td>Platform-as-a-Service</td>
</tr>
<tr>
<td>PLUQI</td>
<td>Personalized and Localized Urban Quality Index</td>
</tr>
<tr>
<td>REST</td>
<td>Representational state transfer</td>
</tr>
<tr>
<td>RDF</td>
<td>Resource Description Framework</td>
</tr>
<tr>
<td>SLA</td>
<td>Service Level Agreement</td>
</tr>
<tr>
<td>SOA</td>
<td>Service Oriented Architecture</td>
</tr>
<tr>
<td>SPARQL</td>
<td>SPARQL Protocol and RDF Query Language</td>
</tr>
<tr>
<td>SSH</td>
<td>Secure Shell</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modeling Language</td>
</tr>
<tr>
<td>XML</td>
<td>eXtensible Markup Language</td>
</tr>
</tbody>
</table>
List of Figures

Figure 1: An example of open data - Highschools in Korea ................................................................. 10
Figure 2: Classes model from PLUQI use case ................................................................................... 11
Figure 3: Properties of Classes from PLUQI use case ....................................................................... 11
Figure 4: Instance example of PLUQI use case .................................................................................. 12
Figure 5: Instance example serialized in turtle format ........................................................................ 12
Figure 6: PLUQI – Home page ............................................................................................................ 14
Figure 7: City detail indices for Seoul ............................................................................................... 15
Figure 8: PLUQI – Rank of cities page .............................................................................................. 16
Figure 9: PLUQI – Compare cities page ............................................................................................ 17
List of Tables

Table 1: Themes, parameters and indicators selected for the current study........................................ 13
Table 2: PLUQI functional requirements addressed.............................................................................. 23
1 Introduction

This report represents supporting documentation for the prototype developed for Deliverable D5.2 “Use case implementation, v1”. The goal of this deliverable is to provide:

- An initial implementation of the use case which will make use of the DaPaaS platform released at the end of the first year.

The rest of this report is structured as follows:

- Section 2 describes the development of the PLUQI use case prototype.
- Section 3 outlines future work for the second version of the PLUQI use case.
- Appendix A summarises how the main PLUQI functional requirements are addressed.
2 PLUQI Use Case Prototype

2.1 Datasets

As stated in Deliverable D5.1, PLUQI (Personalized Local Urban Quality Index) is defined as a customizable index model that shows the level of well-being and sustainability for cities, based on individual preferences.

Sustainability is a notion that is too macro-level and multi-faceted to be measured by any single metric. Several domains (or dimensions) and measurable and available metrics have to be defined and then aggregated into a single summary indicator.

The well-being index model has been benchmarked from several models (for details, see Deliverable D5.1):

- OECD Better Life Index
- Gallup-Healthways Well-Being Index (WBI)
- The Urban Sustainability Index
- UK Sustainable Cities Index
- London Ward Well-Being Scores
- UK experimental subjective well-being estimates

For this use case, we define the model in the following way:

1. Environmental needs and efficiency: green space, water and electricity consumption, air quality.
2. Level of opportunity: job, education, re-education, economic dynamic.
3. Cultural satisfaction: number of theatres, museums, art centres, etc.
4. Daily life satisfaction: weather, transportation, community, living density, shopping space, entertainment venues.
5. Safety and security: number of police station, crime per capita, fire station.

For the PLUQI use case implementation, a survey of available datasets has been undertaken for South Korea.

The Korean Statistical Information Service (KOSIS)\(^3\) is providing numerous datasets in Excel format. For the implementation of the prototype, each dataset used in PLUQI has been chosen based on the following rules:

- Data should have the same granularity of all locations (Korean Cities) and
- Data should be as recent as possible

The datasets (in row format) are transformed into RDF so that the data can be accessible in a uniform manner - through SPARQL queries. For PLUQI, a simple ontology has been designed for representing Quality Index from various sources (e.g., open data, open API and social data) and various formats (e.g., CSV, Excel, JSON and RDF).

\(^3\)http://kosis.kr
Other sources of data come from social sensors through the CITI-SENSE\(^4\) platform. A Social Sensor is a software agent that provides observations about an environment via communicating to other agents. The role is to understand citizen's experience about their environment. By analyzing a stream of messages from a citizen, an agent can detect citizen's intention and expectation. Then, by using ontologies, data can be linked semantically to other open data, geo-spatial information and other physical or social sensors data. A social sensor provides good integrated indicators about various aspects of the environment.

Social sensors are defined for getting data from 7 cities and 9 regions: Seoul, Busan, Daegu, Incheon, Gwangju, Daeyeon, Ulsan, Gyeonggi-do, Gangwon-do, Chungcheongbuk-do, Chungcheongnam-do, Jeollabuk-do, Jeollanam-do, Gyeongsangbuk-do, Gyeongsangnam-do, Jeju-do.

### 2.2 Ontology Model

The PLUQI ontology schema is depicted in Figure 2. It can represent sub-indices and its values with location and date-time information. It reuses other standard ontology models such as WGS84 and Korean Administrative Divisions ontology\(^5\).

---

\(^4\) [http://www.citi-sense.eu/](http://www.citi-sense.eu/)

\(^5\) [http://lod.seul.go.kr](http://lod.seul.go.kr)
Figure 2: Class model from PLUQI use case

Figure 3: Properties of classes from PLUQI use case
Following the schema, we can describe the information of high schools in Seoul as an instance of Level of opportunity, in Turtle format as following:

```turtle
:v1 rdf:type :Value ,
   owl:NamedIndividual ;
   :measure 151 ;
   :time "2011-00-00T00:00:00Z"^^xsd:dateTime ;
   rdfs:comment "type:total"^^xsd:string ;
   :location :Seoul .
   :number_of_coed_high_schools rdf:type :Level_of_opportunity ,
   owl:NamedIndividual ;
   :hasValue :v1 .
```

To assist and simplify the RDFization process, the Grafter\(^6\) tool provided by the Work Package 4 has been used. Future versions of this tool enhanced by new functionalities and a Graphical User Interface (GUI) will further facilitate this process.

2.3 Indices

In order to calculate PLUQI indices, several data representing different variables need to be gathered. After this data is collected, normalization, aggregation and weighting of the underlying variables are needed. Normalization is usually applied to single variables in order to make them comparable, i.e., transforming the various scales of variables into one unique scale. The normalized indicators are then

\(^6\) http://grafter.org
aggregated using specific formulas (e.g. arithmetic mean). If one indicator is more “important” than another, the former is assigned a stronger weight than the latter within the aggregation procedure.

Scientifically sound methods for normalization (to make data ‘comparable’), weighting (to specify the ‘correct’ interrelationships), and aggregation (to get the ‘right’ functional relationship) are pre-requisites for the construction of meaningful indices.

In order to implement the PLUQI application we referred to the paper “An approach to measure sustainability: Comparative Evaluation of States of India”\(^7\). As the authors stated in their survey, datasets were often incomplete and at different scales. Sustainability indicators are often criticized because they aim to ‘measure the immeasurable’. Anyways, in this survey authors did provide a clear methodology to deal with datasets and we apply their proposed methodology in our implementation of PLUQI.

2.3.1 Z-Score

To make various parameters comparable a methodology has been adopted in the current implementation. Various indicator values computed for different cities in South Korea have been normalized using Z-score.

Z-score is a dimensionless quantity derived by subtracting the population mean from an individual raw score and then dividing the difference by the population standard deviation. This conversion process is called standardizing or normalizing. The standard score indicates how many standard deviations an observation is above or below the mean.

\[
Z = \frac{X - \mu}{\sigma}
\]

where:

- \(X\) is a raw score to be standardized
- \(\sigma\) is the standard deviation of the population
- \(\mu\) is the mean of the population.

Z-score values computed for each of the parameters have been averaged for different themes (assuming equal weightage). Finally, a sustainability index has been formed using average of the averaged Z-score values of different themes.

Sustainability Index = Average (ZEnv, ZOpportunity, ZCultural, …)

Sustainability index values have been analysed and interpreted for various states of Korea to adjudge their performance.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Theme</th>
<th>Indicators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environmental needs and efficiency</td>
<td>green space</td>
<td>count, area</td>
</tr>
<tr>
<td></td>
<td>Air quality</td>
<td>Yellow Sand &amp; PM10 (social sensor)</td>
</tr>
<tr>
<td></td>
<td>Temperature</td>
<td>Social Sensor</td>
</tr>
<tr>
<td></td>
<td>Humidity</td>
<td>Social Sensor</td>
</tr>
<tr>
<td>Level of opportunity</td>
<td>Education</td>
<td>Number of schools</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number of professors</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number of students</td>
</tr>
<tr>
<td>Cultural satisfaction</td>
<td>Cultural facilities</td>
<td>Number of theatres</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number of museums</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number of Art galleries</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number of science museums</td>
</tr>
<tr>
<td>Daily life satisfaction</td>
<td>Traffic equipment</td>
<td>number of railroad stations</td>
</tr>
</tbody>
</table>

\(^7\) [http://umconference.um.edu.my/upload/163-1/Paper%2048.doc](http://umconference.um.edu.my/upload/163-1/Paper%2048.doc)
number of railroad line
Length of road
Area of road
number of railroad stations

<table>
<thead>
<tr>
<th>Happiness</th>
<th>Social Sensor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Safety and security</td>
<td>Criminality</td>
</tr>
<tr>
<td>Anxiety</td>
<td>Social Sensor</td>
</tr>
</tbody>
</table>

| 2.4 Web App |
This section describes the web application prototype, which aims to demonstrate the strengths of DaPaaS in real applications. The web application visualizes data about PLUQI.

These indices are visualized in form of different graphs, e.g., map, bar chart and table, with intuitive user interactions. In this prototype, data of Korean cities has been used.

PLUQI indices are calculated from the indicators in the table above. Each index can have different weight/importance for different users. The web application consists of three pages (Home, Rank of cities and Compare cities) as described below.

2.4.1 Home Page
The main page is the starting point of the application from where users can see and interact with the PLUQI visualizations.

Figure 6: PLUQI – Home page
The main page has following parts (marked as (1) to (4) in the above picture):

- Menu bar (1): on top of page, which allows user to navigate between pages.
Index weight setting (2): For example, ‘environment’ may have higher importance to user A but less importance to user B. These sliders are used to set the individual’s preferred weight of each index. From here, PLUQI is calculated from the set of these weights and their corresponding index scores.

PLUQI chart (3): visualizes the PLUQI of cities. A higher bar means there is a higher PLUQI.

Map chart (4): visualizes cities’ PLUQI on map. The size of the circle represents PLUQI.

Page interactions:
- Changing sliders and clicking refresh button in (2) will redraw the PLUQI chart (3) with newly calculated values.
- By clicking a bar in PLUQI chart (3), the corresponding city will be highlighted on map (4).
- By clicking a city on map (4): the corresponding city in PLUQI chart (3) is highlighted and detail indices of the city will be shown below the map.

The city detail indices look as follows:

![City detail indices for Seoul](image)

The city detail view can be interacted as follows:
- Clicking an index will detail data of the index.
- Clicking ‘view city details’ will bring the city detail pop-up.

### 2.4.2 Rank of Cities Page

This page helps users see how sustainable development indices changed through time.
The user can select the preferred index and time period to how cities are ranked based on the selected conditions.

2.4.3 Compare Cities Page

This page allows user to compare cities. Indices of the compared cities are visualized in a bar chart. Each group of bars presents an index of cities side by side, and a higher bar means there is a higher score.
2.5 Application API

In order to fetch data from the DaPaaS Platform and pass it to the visualization component, an intermediate API has been defined as follows:

2.5.1 Get City Data

**HTTP method**
GET CityData

**Description**
Get list of cities and their information.

**Parameters**

| weights | Set of 5 weights for 5 sub-indices (education, environment, healthcare, cultural_satisfaction, traffic_satisfaction). |

**Response**

**Cities**: An array contains cities data. If there is an empty slot, the slot will be set as empty object.

**Sample Response**:

```json
{
  "response": {
    "name": "Seoul",
    "PLUQI": 9,
    "Indices": {
      ...
    }
  },
  ...
}
```
2.5.2 Get Sub Index Details

HTTP method
GET SubIndexDetailData (city, [sub-indexname])

Description
Get the details about a city’s sub-indexes, with evidences data.

Sample Response
(each sub index, will have different JSON object format)
For Education:

```
{  
  "response": {  
    "name": "education",  
    "weight": 9,  
    "details": {  
      "number_of_schools": 538,  
      "number_of_graduates": 89500  
    }  
  }  
}
```

For Environment:

```
{  
  "response": {  
    "name": "Environment",  
    "weight": 9,  
    "details": {  
      "temperature": 32,  
      "humidity": 90,  
      "PM10": 50,  
      "green_spaces": 85  
    }  
  }  
}
```
2.5.3 Get City Rank

**HTTP method**
GET CityRank (time interval)

**Description**
Get city rank data.

**Sample Response**
```
{
    "response": [
        {
            "cities": [
                {
                    "node": "Seoul"
                },
                {
                    "node": "Daejeon"
                },
                {
                    "node": "Incheon"
                }
            ]
        }
    ]
}
```

2.5.4 Compare Cities

**HTTP method**
GET Sub-Indices Rank Data

**Description**
Get cities information to compare them.

**Sample Response**
```
{
    "response": [
        {
            "city": "Seoul",
            "values": [
                {
                    "index": "PLUQI",
                    "value": 10
                },
                {
                    "index": "Education",
                    "value": 5
                },
                {
                    "index": "Environment",
                    "value": 8
                },
                {
                    "index": "Healthcare",
                    "value": 7
                }
            ]
        }
    ]
}
```
"index": "Cultural Satisfaction",
"value": 9
},

{ "index": "Traffic Satisfaction",
"value": 4
}
]

"city": "Daejeon",
"values": [ 

{ "index": "PLUQI",
"value": 9
},

{ "index": "Education",
"value": 6
},

{ "index": "Environment",
"value": 9
},

{ "index": "Healthcare",
"value": 7
},

{ "index": "Cultural Satisfaction",
"value": 9
},

{ "index": "Traffic Satisfaction",
"value": 4
}
]

"city": "Daegu",
"values": [ 

{ "index": "PLUQI",
"value": 8
},

{ "index": "Education",
"value": 6
},

{ "index": "Environment",
"value": 5
},

{ "index": "Healthcare",
"value": 7
},

{ "index": "Cultural Satisfaction",
"value": 9
},

{ "index": "Traffic Satisfaction",
"value": 4
}
]
2.6 Open Source Libraries Used

The PLUQI application uses the following 3rd party libraries:

- **Leaflet**\(^8\), which is a modern open-source JavaScript library for mobile-friendly interactive maps. This library is used to visualize data on the map in Main page.

- **NVD3**\(^9\), which is an attempt to build re-usable charts and chart components for d3.js without taking away the power that d3.js. This Javascript library is used to draw charts.

2.7 Application Deployment

The application intended to run on Tomcat so PLUQI has been developed in JSP / Java and packaged in a war file. DaPaaS provides a Graphical User Interface (GUI) for uploading a war file packaging the application. Through this GUI, metadata are provided for managing and retrieving the application.

\(^8\) [http://leafletjs.com](http://leafletjs.com)

\(^9\) [http://nvd3.org](http://nvd3.org)
3 Future Work

This first prototype illustrated the usage of functionalities provided by DaPaaS platform. The PLUQI use case did use services (APIs) for importing datasets, transforming and publishing them in RDF format, then querying them uniformly for creating an API for feeding the visualization part. Then the application was deployed on the DaPaaS platform. The application can be retrieved by search based on metadata provided during registration.

During data importing, it is clear that an easy interface for importing, transforming data is important for minimizing the cost of development, data quality checking and maximizing the reuse of Open Data. Datasets are often missing values, containing lot of spelling errors, incomplete etc. In order for the PLUQI use case to be more attractive, more data is planned to be integrated and additional features for social support should be taken in account.

As Open Data is currently at the beginning stage in South Korea, we could not find as many datasets as expected but we plan to apply the same process for others cities, especially for cities in Europe. Also, because DaPaaS is still on-going work, we could not use some of datasets expressed in JSON but with next versions of DaPaaS as more data formats will be supported we will integrate them in the next version. One future work planned is to provide also a mobile application.
### 4 Appendix A: Requirements Addressed by PLUQI

Deliverable D5.1 outlined a set of requirements for the use case. The tables below summarize how the main PLUQI (Personalized Local Urban Quality Index) functional requirements are addressed by the current version of the use case implementation.

**Table 2: PLUQI functional requirements addressed**

<table>
<thead>
<tr>
<th>ID</th>
<th>Functionality</th>
<th>Description</th>
<th>Priority</th>
<th>Addressed</th>
</tr>
</thead>
<tbody>
<tr>
<td>FU-01</td>
<td>User Management</td>
<td>System shall provide functionalities for managing user information (CRUD operations on users).</td>
<td>Core</td>
<td>Covered by the DaPaaS Platform requirements (IO-02): Addressed in Deliverable D1.2.</td>
</tr>
<tr>
<td>FU-02</td>
<td>User Management</td>
<td>System shall support authentication from other Services like Facebook, Google, Twitter, LinkedIn etc.</td>
<td>Optional</td>
<td>Covered by the DaPaaS Platform requirements (DP-10, AD-01, EU-01): Addressed in Deliverable D2.2.</td>
</tr>
<tr>
<td>FU-03</td>
<td>Import Dataset</td>
<td>System shall support various connectors/adapters for importing datasets of various types (textual, tabular or RDF data) and formats (e.g. txt, html, rdf, csv, xls).</td>
<td>Core</td>
<td>Covered by the DaPaaS Platform requirements (DP-01): Addressed in Deliverable D1.2 and D2.2.</td>
</tr>
<tr>
<td>FU-04</td>
<td>Import Dataset</td>
<td>System shall support import for single or multiple files.</td>
<td>Optional</td>
<td>Import of multiple files is to be addressed in the next version of the prototype.</td>
</tr>
<tr>
<td>FU-05</td>
<td>Import Dataset</td>
<td>System shall support import for a directory of files.</td>
<td>Optional</td>
<td>To be addressed in the next version of the prototype.</td>
</tr>
<tr>
<td>FU-06</td>
<td>Import Dataset</td>
<td>System should provide support to bulk upload in parallel.</td>
<td>Optional</td>
<td>To be addressed in the next version of the prototype.</td>
</tr>
<tr>
<td>FU-07</td>
<td>Import Dataset</td>
<td>System should provide support to resume upload in case of interruption (temporary network or system failure).</td>
<td>Optional</td>
<td>To be addressed in the next version of the prototype.</td>
</tr>
<tr>
<td>FU-08</td>
<td>Import Dataset</td>
<td>System shall support various connections to Open API Services like Twitter Streaming/Search API, RSS etc.</td>
<td>Core</td>
<td>To be addressed in the next version of the prototype.</td>
</tr>
<tr>
<td>FU-09</td>
<td>Import Dataset</td>
<td>System shall support various connections to Legacy Systems like RDMS/EDMS/CRM/Sear</td>
<td>Optional</td>
<td>Partially covered by the DaPaaS Platform requirements (DP-01): Addressed in Deliverable D1.2.</td>
</tr>
<tr>
<td>FU-10</td>
<td>Import Dataset</td>
<td>System shall support connections to other already published data from other Open data platforms (C-KAN, D-KAN, Socrata).</td>
<td>Core</td>
<td>To be addressed in the next version of the prototype.</td>
</tr>
<tr>
<td>FU-11</td>
<td>Refresh Dataset</td>
<td>System shall be able to refresh/update data for sources of real-time datasets (like Twitter, Sensors data) or frequently changing datasets (RSS).</td>
<td>Core</td>
<td>Related to Citi-Sense Platform integration.</td>
</tr>
<tr>
<td>FU-12a</td>
<td>Dataset Aggregation</td>
<td>System shall provide functionalities for aggregating several datasets.</td>
<td>Core</td>
<td>To be addressed in the next version of the prototype.</td>
</tr>
<tr>
<td>FU-12b</td>
<td>Dataset Mashup</td>
<td>System shall provide functionalities to link to other datasets.</td>
<td>Core</td>
<td>Covered by the DaPaaS Platform requirements (DP-04): Addressed in Deliverable D1.2.</td>
</tr>
<tr>
<td>FU-13</td>
<td>Data Manipulation</td>
<td>System shall provide functionalities to clean/refine datasets.</td>
<td>Core</td>
<td>Covered by the DaPaaS Platform requirements (DP-05): Addressed in Deliverable D1.2.</td>
</tr>
<tr>
<td>FU-14</td>
<td>Data Storage</td>
<td>System shall support various storage repositories depending on properties of datasets (Triple Store, RDBMS, NoSQL, Search Engine).</td>
<td>Core</td>
<td>Covered by the DaPaaS Platform requirements (DP-02): Addressed in Deliverable D1.2.</td>
</tr>
<tr>
<td>FU-16</td>
<td>Dataset Management</td>
<td>System shall offer functionalities for managing access (read/write permissions) depending on user permissions (individual, group, universal).</td>
<td>Optional</td>
<td>Covered by the DaPaaS Platform requirements (DP-07): Addressed in Deliverable D1.2.</td>
</tr>
<tr>
<td>FU-17</td>
<td>Export Dataset</td>
<td>System shall export to various formats (RSS/Atom, text, CSV, RDF, etc) and provide a URL for remote access.</td>
<td>Core</td>
<td>Covered by the DaPaaS Platform requirements (EU-05): Addressed in Deliverable D1.2.</td>
</tr>
<tr>
<td>FU-18</td>
<td>Sociality</td>
<td>System should provide functionalities for creating and managing communities of users.</td>
<td>Optional</td>
<td>To be addressed in the next version of the prototype.</td>
</tr>
<tr>
<td>FU-19</td>
<td>Sociality</td>
<td>System should allow user to follow and/or vote for datasets.</td>
<td>Optional</td>
<td>To be addressed in the next version of the prototype.</td>
</tr>
<tr>
<td>FU-20</td>
<td>Sociality</td>
<td>System should provide connectivity to other social services (Google+, Twitter, Facebook, E-mail) for sharing datasets links.</td>
<td>Optional</td>
<td>To be addressed in the next version of the prototype.</td>
</tr>
<tr>
<td>-------</td>
<td>-----------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>FU-21</td>
<td>Management</td>
<td>System should provide notifications (e.g. change in datasets).</td>
<td>Core</td>
<td>Covered by the DaPaaS Platform requirements (EU-03): Addressed in Deliverable D1.2 and D2.2</td>
</tr>
<tr>
<td>FU-22</td>
<td>Notification</td>
<td>System should provide notification in case of change in datasets.</td>
<td>Core</td>
<td>Covered by the DaPaaS Platform requirements (EU-03): Addressed in Deliverable D1.2 and D2.2</td>
</tr>
<tr>
<td>FU-23</td>
<td>Curation &amp; Management</td>
<td>System should provide functionalities for adding/modifying data in datasets when allowed.</td>
<td>Core</td>
<td>Covered by the DaPaaS Platform requirements (DP-02): Addressed in Deliverable D1.2</td>
</tr>
<tr>
<td>FU-24</td>
<td>Management</td>
<td>System should provide statistics of usage/access of published datasets.</td>
<td>Core</td>
<td>Covered by the DaPaaS Platform requirements (DP-07): Addressed in Deliverable D1.2</td>
</tr>
<tr>
<td>FU-25a</td>
<td>Visualization</td>
<td>System should provide functionalities for viewing full datasets or previewing parts of datasets with adapted visualizations.</td>
<td>Core</td>
<td>Covered by the DaPaaS Platform requirements (EU-04): Addressed in forthcoming Deliverable D3.3.</td>
</tr>
<tr>
<td>FU-25b</td>
<td>Visualization</td>
<td>Visualization should provide support for tabular form, charts (line, plot, histograms etc.) for displaying 2D data, time series, plotting data on a map for geo-spatial data.</td>
<td>Core</td>
<td>Covered by the DaPaaS Platform requirements (EU-04): Addressed in forthcoming Deliverable D3.3.</td>
</tr>
<tr>
<td>FU-26</td>
<td>Search</td>
<td>System should provide federated search to repositories by providing adapted repository querying (Search Engine, RDF-Store, SQL etc.).</td>
<td>Core</td>
<td>Covered by the DaPaaS Platform requirements (DP-03, EU-02): Addressed in Deliverable D1.2.</td>
</tr>
<tr>
<td>FU-27</td>
<td>Discovering</td>
<td>System should provide filter results by tags/criteria (faceted search).</td>
<td>Optional</td>
<td>Covered by the DaPaaS Platform requirements (DP-03, EU-02): Addressed in Deliverable D1.2.</td>
</tr>
<tr>
<td>FU-28</td>
<td>API</td>
<td>System should provide RESTful Open API for all functionalities.</td>
<td>Core</td>
<td>Covered by the DaPaaS Platform requirements (DP-02, AD-01): Addressed in Deliverable D1.2 and D2.2.</td>
</tr>
</tbody>
</table>