D2.3 SaaS Generic Software Components

Report

Project acronym:	Mobile-Age
Project full title:	Mobile-Age
Grant agreement no.:	693319
Responsible:	TT
Contributors:	TT, UPM, FTB
Document Reference:	D2.3
Dissemination Level:	PU
Version:	1.3
Date:	2017-10-09

History

Version	*Date*	*Modification reason*	*Modified by*
0.1	2016-11-15	Initial Draft and Table of Contents (ToC)	TT
0.2	2016-12-18	Revision according to comments to align better with D2.1 and D2.2.	TT
0.3	2016-12-19	Input from Freddy, added new images to illustrate maps	TT
0.4	2016-12-20	Input from Frank on maps module, revision of flow chart.	TT
0.5	2017-01-09	Start to work in comments.	TT
0.6	2017-01-13	Revised summary, OGD annotation, reorganised structure in line with D2.2, added schema for venue OGD. Heading numbers.	UPM, TT
0.6.2	2017-01-24	Revised executive summary and intro. Added info on attributes in OGD database. Text polish. Components diagram.	TT
0.7	2017-01-25	Added SaaS integration overview, OGD provider annotation GUI. Take out map module.	TT, UPM
0.8	2017-02-08	Update on annotation chapter	UPM, TT
0.9	2017-02-10	Added annex on data transformation	TT
0.91	2017-02-14	Added comment from AUTH	TT, UPM
1.0	2017-02-17	Final proof read and edited version	ULANC
1.1	2017-07-06	Updates according to review outcomes.	TT
1.2	2017-07-14	Revised version submission	ULANC
1.3	2017-10-06	Updates according to review outcomes	TT, UPM
	2017-10-09	Final proof read version	ULANC

Table of contents

History **2**

Table of contents **3**

List of figures **4**

List of tables **5**

Executive summary **6**

1 Introduction **7**

1.1 Aims and objectives 7

1.2 Placement relative to other tasks 8

2 The Mobile-Age ecosystem: Stakeholders and Users **9**

3 Methodology **12**

3.1 Approach 12

3.2 Co-creation aspects 12

3.3 Contributors 12

3.4 Glossary 13

4 The SaaS components in the development cycle **14**

5 Outline of the SaaS components **16**

5.1 Semantic Annotation Component 17

5.2 OGD Sources Search and Data Transformation Component 21

5.2.1 OGD Sources Search 21

5.2.2 Data transformation component 22

5.2.3 Service search component 23

5.3 Concept lookup component 24

6 SaaS components integration outline and status **26**

7 Innovation Aspects (comparison to the state of art) **29**

7.1 Novel aspects of the Semantic Annotation Module 29

7.2 Novel aspects of the OGD and services search 29

7.3 Novel aspects of the Mobile-Age Concept lookup 30

8 Key findings (conclusions) **35**

9 References **36**

Annex 1: OSCPSEP database Schema for venues **38**

Annex 2: Data Transformation Example **42**

List of figures

Figure 1- Relationship with other deliverables 8
Figure 2 - Overview of the Mobile-Age users and stakeholders in co-creation of open data-based public services 10
Figure 3 - Overview of the Mobile-Age users Mobile-Age users and stakeholders in co-creation of open data-based public services 11
Figure 4 The SaaS components in the Workflow of the application development use-case. ... 14
Figure 5 Outline of the SaaS covered in this report 16
Figure 6 The current user interface for the OGD annotation. 18
Figure 7 Sub-components of The Semantic Annotation Component 18
Figure 8 Example SPARQL Query. 21
Figure 9 Example lookup of "Front end" on the Mobile-Age website (www.mobile-age.eu). The meaning of the concepts is retrieved from the Mobile-Age term base. In this way, we can maintain just one glossary for each part of the project. 24
Figure 10 Map of park bench locations in Bremen. The image shows a lookup of the concept "backrest" from a textual description of a park bench location. Note that two records are provided, one tailored to the Bremen use-case and one general, from Wordnet. 26
Figure 11 OGD Data Transformation and storage 27

List of tables

Table 1: List of contributors 12
Table 2 : SaaS components attributes. 17
Table 3 OGD Sources Search Attributes........ 21
Table 4 Service Search Component Attributes 23
Table 5 Concept lookup attributes 25
Table 6 Components status (updated for status 2017-07-07) 28
Table 7: The Concept lookup compared to a selection of related in-line dictionary lookup applications, as of 2017-10-05. 32

Executive summary

This document reports on work carried out within the context of Mobile-Age Work Package 2 (WP2). We outline two SaaS components, the components for Semantic Annotation of OGD and the OGD data Search and transformation component. The selection of components is indicated in the DoW and input to subsequent adjustments of them will be collected from co-creation sessions, and from the insights gained from the technical development.

The Semantic Annotation Component will implement ways to annotate relevant OGD sources so that they can be correctly mapped to common formats used in the OSCPSEP database.

During phase I of the fieldwork undertaken at two field sites (Bremen and Zaragoza), several data sets that are of interest to the users (older citizens) have been identified. These data sets include information that is relevant to the users, for example descriptions of pharmacies or location of park benches.

The OGD data Search and transformation component prepares a set of solutions to collect data from OGD sources using APIs, search engine indexing, and web scraping. For Software developers, the search can be used to find OGD data sets and related services. The search also covers a concept lookup service. This enables a unified search for concepts across selected websites to implement a loose integration of public services.

The Mobile-Age SaaS components are intended to support Software developers to use OGD and to create t more accessible applications with less effort. The approach will enable the Software developers to use the SaaS components with a reference to an endpoint provided by the OSCPSEP.

This deliverable is part of the output from Work Package 2 (WP2), in particular Task 2.3, and aims to describe the services provided via the OSCPSEP, and explains how Software developers and data providers can benefit from such services. The OSCPSEP requirements are covered in D2.1 OSCPSEP Requirements, while technical specifications are outlined in D2.2 Interim OSCPSEP technical specifications, which include an overview of the architecture and interfaces of the SaaS components.

1 Introduction

Mobile-Age aims to explore and implement innovative ways to support older citizens to access and use public services through personalized mobile technologies based on open government data. To promote this aim, a set of SaaS components will be provided for use in the OSCPSEP and for integration into applications for older citizens that make use of open data services.

The SaaS components design has been chosen to enable separation of concerns. In this way, we can more easily combine existing, adapted and new functionality with different intellectual property schemes and different technical implementations. This is helpful to provide services for platform developers and to enable Software developers to combine services when creating their apps.

The SaaS components are prepared to support Software developers in composing their applications and data providers to enter their data in three ways:

1. Data management services: OGD transformation, data annotation, dataset search, service search, dataset-app integration, and data record samples.
2. To facilitate the composition of the applications with ready made accessible front-end components: Geographical Map service, Termer service. (Front-end components will be covered in D4.1).
3. To understand user behaviour: Analytics. The Analytics component will be covered in D2.4.

This document consists of 4 chapters. Chapter 1 provides the background and introduces the document. Chapter 2 gives an overview of the Stakeholders and users. Chapter 3 presents an overview of the components' functionality and Chapter 4 gives an outline of the current status of the components and describes how these components can be combined.

We refer to Chapter 3 of D2.2 for the extended analysis of five key scenarios, the (i) Create a new application, (ii) acquire open datasets, (iii) acquire services, (iv) acquire reusable front ends. The scenario (v) request behaviour analytics is covered in D2.4 Behaviour Analytics & Workflow Software components, and not in this document.

The Analytics service is described in D2.4: Behaviour Analytics & Workflow Software Components and the reusable front end components are prepared and described in D4.1.

1.1 Aims and objectives

The information on the SaaS components in chapter 2 serves the objective to provide a short introduction to the component functionality, enabling OGD providers and Software developers external to the project to locate and quickly understand the purpose of the SaaS components, and points them to related online demonstrations. This is followed by descriptions of search components to find services and datasets. The services to search among will be related to the use of OGD, such as a service to calculate the route between two defined positions on a map.

1.2 Placement relative to other tasks

The SaaS components are the results of the task T2.3 OSCPSEP SaaS components & API Development. The SaaS components are being prepared to be used in the OSCPSEP platform and therefore relate to the tasks of T2.1 OSCPSEP requirements and specifications and also the T2.2 OSCPSEP PaaS Infrastructure Development and Deployment for further details on endpoints and integration.

In the subsequent process the outcomes from D2.3 will be used in WP3 (Mobile Services Co-Creation Activities & Evaluation) and WP4 (Development of front end components and demonstrator applications).

Figure 1 provides a diagrammatic representation of the links of D2.2 to other deliverables in the project.

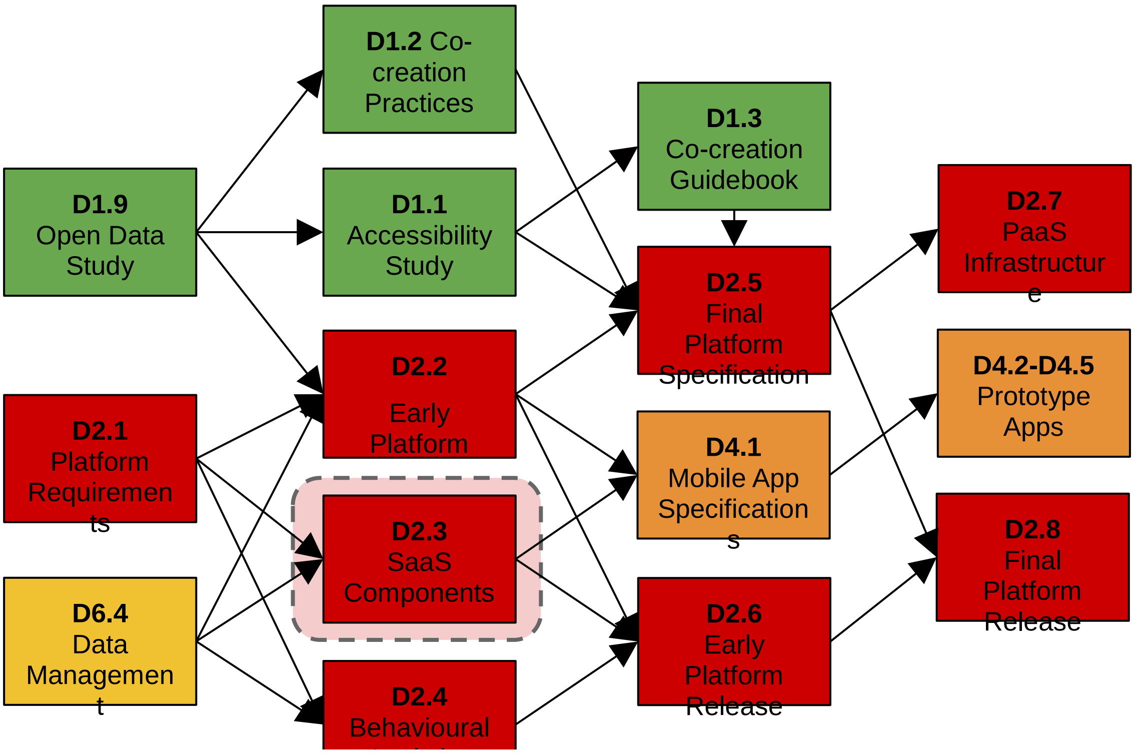

D1.2 Co-creation Practices
D1.9 Open Data Study
D1.1 Accessibility Study
D1.3 Co-creation Guidebook
D2.7 PaaS Infrastructure
D2.5 Final Platform Specification
D2.2 Early Platform
D4.2-D4.5 Prototype Apps
D2.1 Platform Requirements
D4.1 Mobile App Specifications
D2.3 SaaS Components
D2.8 Final Platform Release
D6.4 Data Management
D2.6 Early Platform Release
D2.4 Behavioural

Figure 1- Relationship with other deliverables

2 The Mobile-Age ecosystem: Stakeholders and Users

Here we provide an overview of the roles of Mobile-Age users. Please note that their roles may be overlapping.

Local/regional governments: These can be managing the co-creation activities, define features of the applications, and serve as experts for a specific service domain. In many cases, local governments are also the data owners.

Software developers: These can be independent developers or companies, or working for IT-departments in public authorities or civil society organisations such as the Open Knowledge Foundation. They develop the applications using the platform and they participate in the co-creation activities, adjusting the applications to accommodate for the participants' requests and demonstrating the results in an iterative process.

Older adults: They are a key stakeholder of Mobile-Age and the main users of the mobile applications being developed. They may participate in the core project group or engage in the broader co-creation activities.

Service providers such as government, social welfare organisations, religious congregations or NGOs may be part of the core project group or engaged for specific input. Some of the service providers may also provide (open) data.

Intermediaries include professionals and non-professionals that may support the co-creation activities by providing input for specific tasks in the co-creation process. They may become users of the applications developed.

Facilitators are experienced individuals in the work with older adults and/or groups. They support the co-creation activities through e.g. running workshops, focus groups, interviews.

Other organisations & individuals comprise for example senior citizen organisations, senior citizens' clubs (e.g. computer clubs) but also media and journalists that may report about the co-creation activities, and thereby support engagement as well as dissemination.

Figure 2 provides an illustration of these roles.

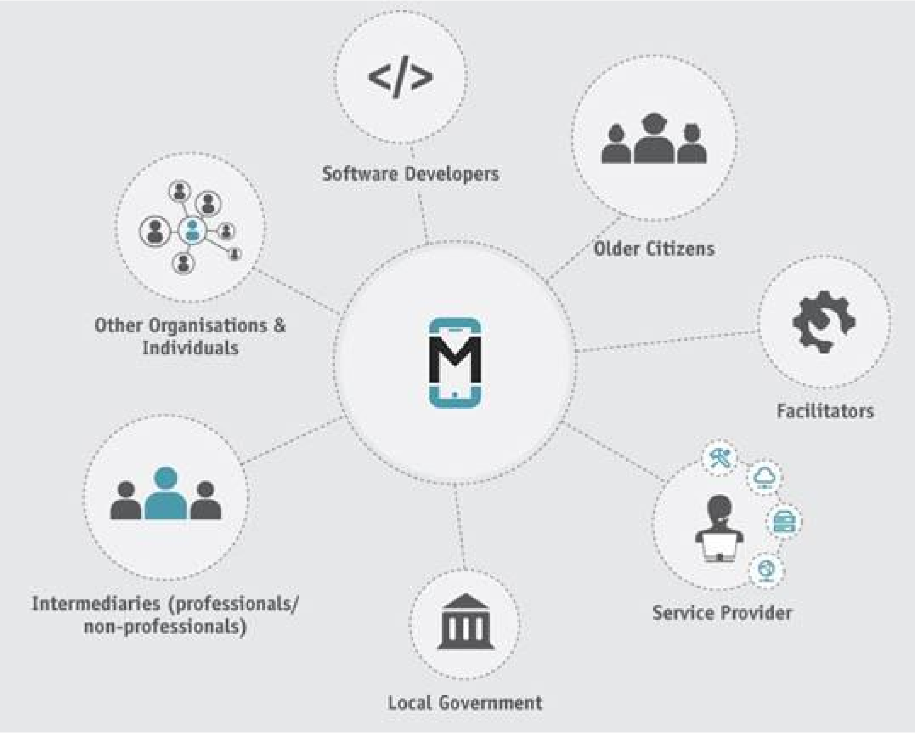

Software Developers
Older Citizens
Other Organisations & Individuals
Facilitators
Intermediaries (professionals/ non-professionals)
Service Provider
Local Government

Figure 2 - Overview of the Mobile-Age users and stakeholders in co-creation of open data-based public services

Based on how these entities interact with the Mobile-Age ecosystem, they can be categorized in two main groups:

- ***Platform Users*: These are users that make direct use of the platform and consist of the following:
    - **Software Developers**
    - **Service Providers**
- ***End Users***: These are users of the mobile applications developed using the Mobile-Age platform.
    - **Older adults**
    - **Local /regional governments**
    - **Intermediaries (professional and non-professional)**
    - **Facilitators**
    - **Other Organizations and Individuals**

Figure 3 illustrates this categorization.

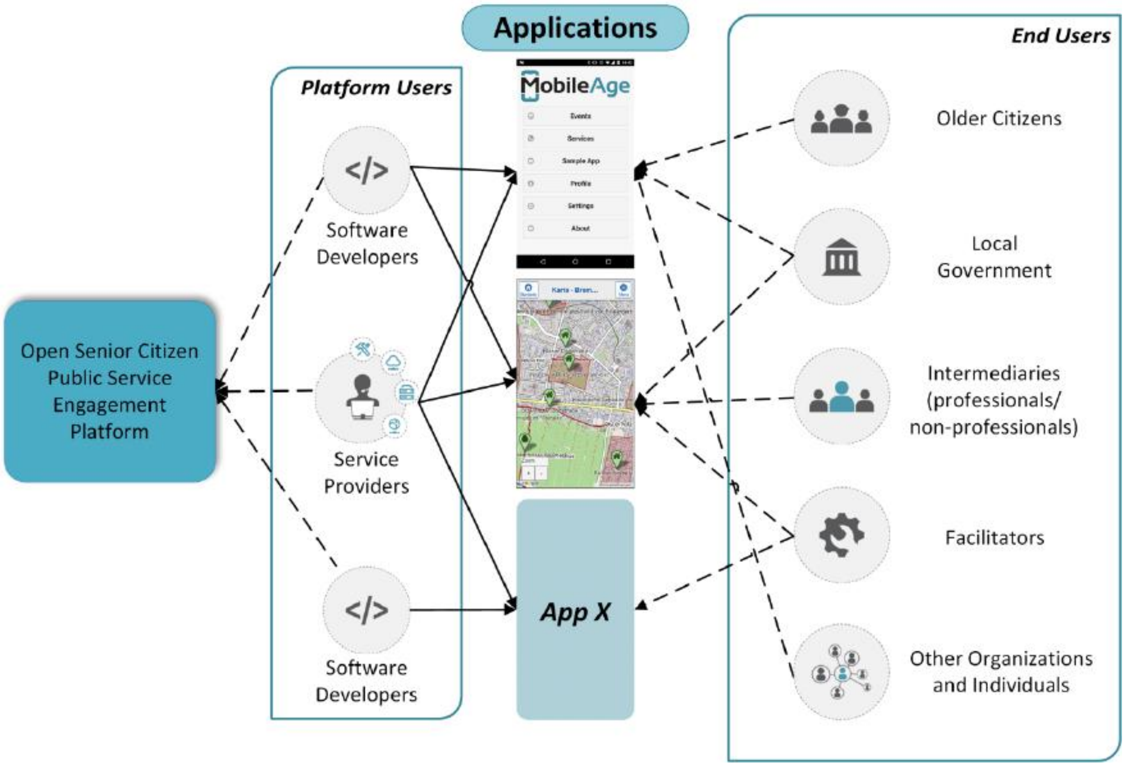


Figure 3 - Overview of the Mobile-Age users Mobile-Age users and stakeholders in co-creation of open data-based public services

3 Methodology

3.1 Approach

The software development methodology used to prepare the SaaS components is an agile one. Departing from the key scenarios given in Chapter 3 of D2.2, we have iteratively prepared the components. The components functionality is tested frequently internally in the developers team using Gitlab to keep track of issues, enhancements and their resolutions.

This caters for a transparent, accountable and participatory development process.

3.2 Co-creation aspects

For the Semantic Annotation component and for OGD Sources Search and Data Transformation Component the co-creation is initially based on the requested data sets from the fieldsites activities. The basic functionality of these components is therefore designed to support the annotation, and the search of these datasets. The further functionality to enable the use of the datasets in the mobile apps, is designed with the developers using the OSCPSEP platform in mind.

So far the testing of the functionality prepared for developers is done internally in the team so that another developer than the originator of a new feature tests it. An overview of the contributors report is given in the next section.

For the Concept lookup component the co-creation is carried out with two local user groups in Norway. This is to allow more frequent physical meetings with users located nearby Tingtun. The first group consist of asylum seekers and refugees who represent multilingual users with a wide range of computer skills. The second group consist of IT students who can focus more on the testing of privacy and security issues.

3.3 Contributors

Table 1 lists the contributors for each chapter and section in this document, along with their affiliation.

Table 1: List of contributors

Document Chapter	*List of Sections*	*Contributors Information*	
		Name	*Affiliation*
1. Introduction	*All*	Mikael Snaprud Freddy Priyatna	TT UPM
2. Mobile-Age Users	*All*	Manolis Falelakis Michail Papamichail	AUTH AUTH
3. Outline of the SaaS components	*3.1*	Freddy Priyatna	UPM
	3.2	Freddy Priyatna Mikael Snaprud Daniel Aasen	UPM TT TT

	3.3	Mikael Snaprud Daniel Aasen	TT TT
4. SaaS components integration outline and status	*All*	Daniel Aasen Mikael Snaprud Freddy Priyatna Manolis Falelakis Papamichail Michail Christopher Bull Frank Berker Frank Reins	TT TT UPM AUTH AUTH ULANCS – SCC FTB FTB
5. Conclusions	*All*	Mikael Snaprud Freddy Priyatna	TT UPM

3.4 Glossary

Please refer to the Mobile-Age “Glossary search page” for acronyms, abbreviations, and glossary records. This page is temporarily made available at

https://glossary.tingtun.no/search/?api=Mobileage

The service will be refined and made available on a URL on the Mobile-Age project website at a later date.

When helpful we have tried to use the concepts from ISO/IEC TR 20000-10:2015(en) Information technology — Service management — Part 10: Concepts and terminology [1].

The following naming conventions are used in this document:

- **System** – represents the internal OSCPSEP services, modules and tools that are being utilized by the Users
- **Services** – represents the SaaS components outlined in this deliverable.
- **Data source** – an online location or organisation with one or more data sets.
- **Data owner** – an organisation who owns and usually maintains a data set.
- **Data set** – a set of data, e.g. the location of pharmacies in Zaragoza, which can be retrieved from a data source.

4 The SaaS components in the development cycle

SaaS components will be used both to support the OGD providers and to facilitate the Software developer interaction with the OSCPSEP services, and their use of components in future applications Software development. When implemented in a Mobile-Age app, the SaaS components can be used by the older citizens.

We assume that developers and OGD providers have registered in order to gain access to the OSCPSEP as outlined in D2.1, chapter 5.3.1 Application development.

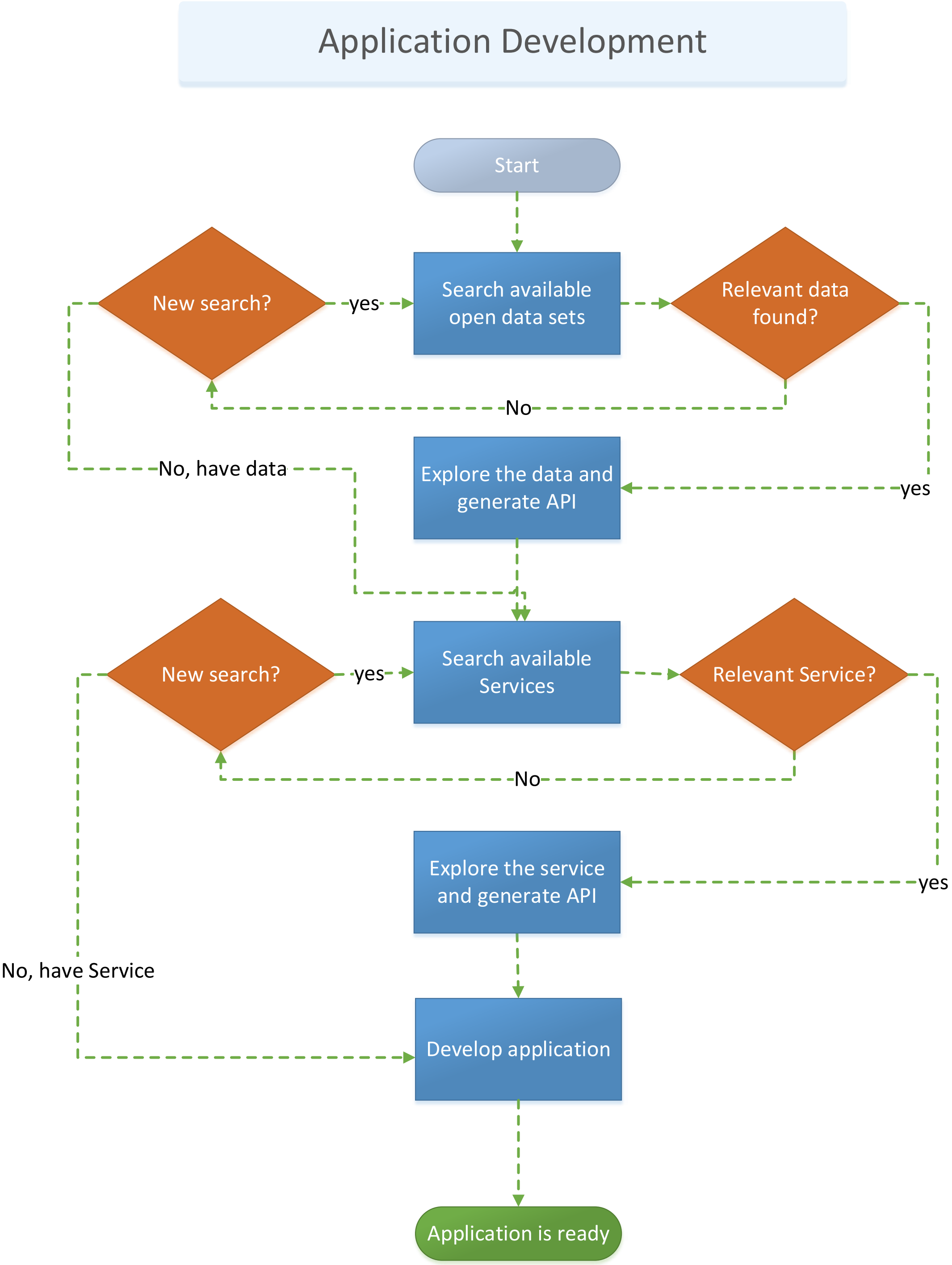


Figure 4 The SaaS components in the Workflow of the application development use-case.

The SaaS components in the workflow for the application development use case are shown in Figure 4.

The search for data will match the search string against a selection of the OGD data source attributes and against the associated annotation data. More details on the searched attributes and the annotations are given in chapter 2.1.

Currently the number of services is small and we therefore will initially present them in a list to provide an overview for the Software developers. For the case with a larger number of services, the workflow to search and explore services is similar to the search for front-end components however, annotation data will be omitted.

5 Outline of the SaaS components

This chapter provides general information regarding the Mobile-Age SaaS components. An outline of the components is given in Figure 5. The figure shows how the data sets are retrieved from the external OGD source (e.g. from Bremen or Zaragoza) via the OGD transformation to be made available in the OGD repository for searching. The figure also indicates the concept lookup connected to an external concept source with for example, Mobile-Age concepts. The Dictionary is hosted by Tingtun. The functionality provided by both the OGD Search and the Concept lookup components are available from the OSCPSEP platform and are presented in the Client browser. Access from Software developer applications is not shown in this figure.

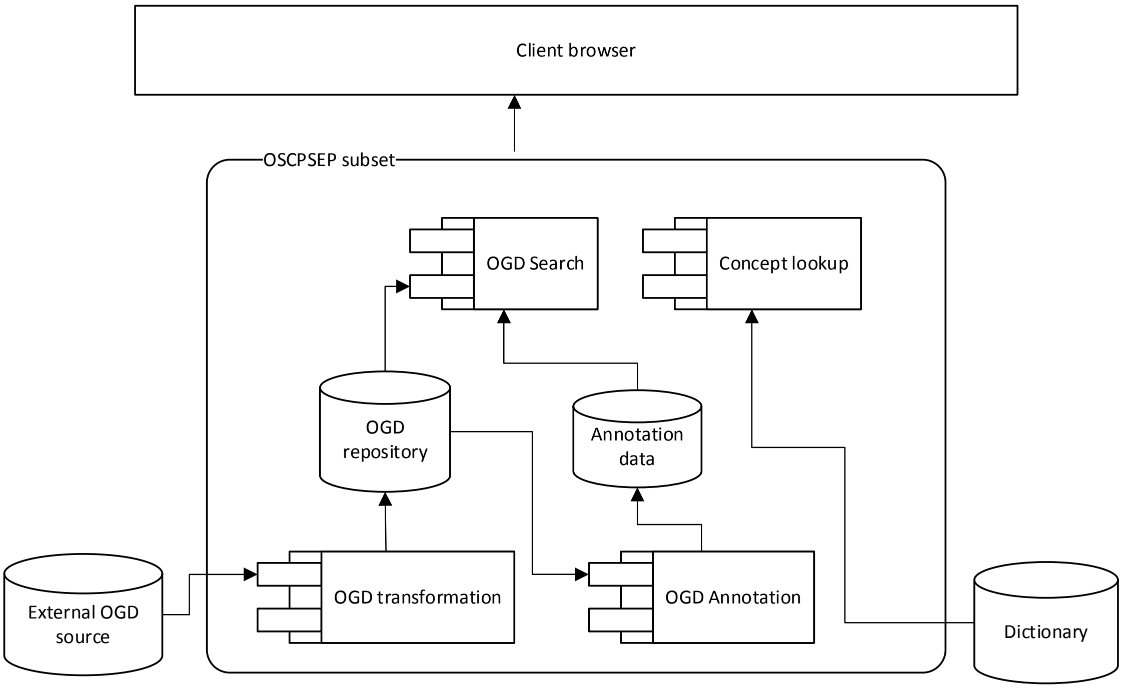


Figure 5 Outline of the SaaS covered in this report.

The Semantic Annotation Component will implement ways to annotate relevant OGD sources so that they can be correctly mapped to common formats used in the OSCPSEP database.

The OGD Search and Data Transformation Component provides a set of solutions to collect data from OGD sources using APIs, search engine indexing, and web scraping.

The concept lookup service provides a unified search for concepts across selected websites to implement a loose integration of public services.

In Table 2 the Component Attributes are introduced as a common format used for SaaS component descriptions.

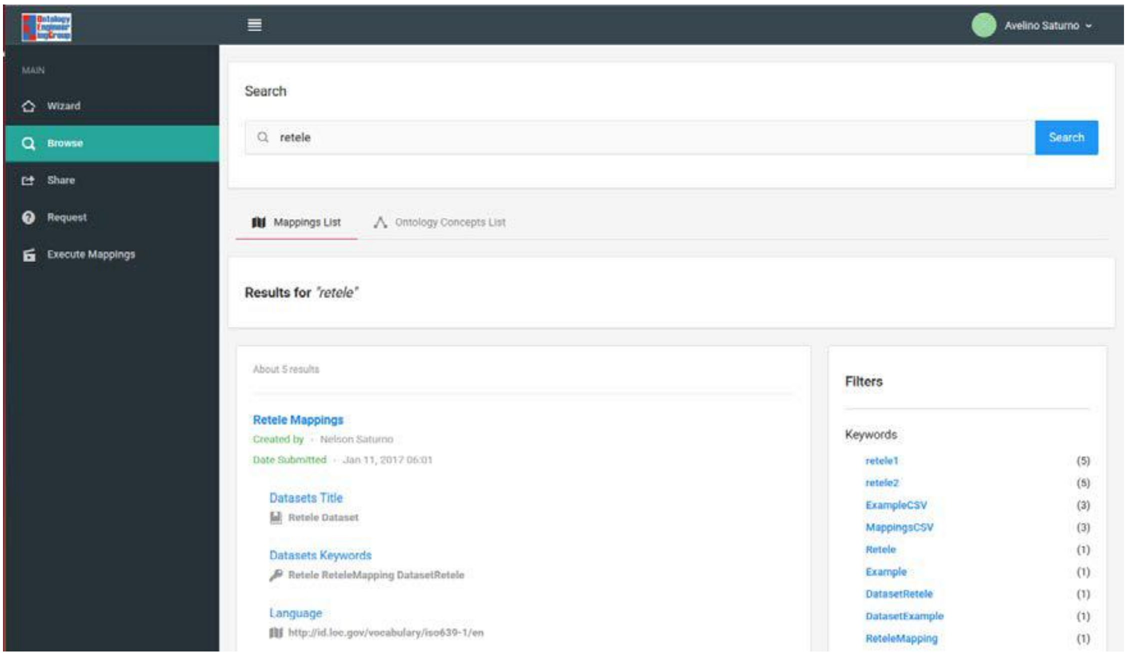
Table 2 : SaaS components attributes.

Attribute	*Description*
Name	Name of SaaS component
Purpose	What problem the SaaS component will solve
Integration points	Where can the SaaS be connected. OSCPSEP / Mobile-Age Apps to be developed / Web browser
Integration method	e.g. Javascript in web page / Javascript in App code / Browser addon to Firefox or Chrome.
Use in OSCPSEP	e.g. concept lookup to support Software developers as part of the Software developers interface in the OSCPSEP.
Use in other SaaS components	e.g. Integration into the maps
Use in Mobile-Age apps	How to activate and use in app.
Relation to the scenarios in D2.2	List related to scenarios of D2.2 chapter 3.
Use of OGD	How does the SaaS use OGD, if relevant.
Responsible Partner	Partner with lead role to develop SaaS component.

5.1 Semantic Annotation Component

The Semantic Annotation Component is responsible for managing the metadata associated to the OGD sources in the OSCPSEP. It also enables a more efficient search since searching the content of the datasets would require a full-text search consuming computing resources, and taking a significant amount of time to generate the search result. Another benefit of annotating OGD sources is the ability to deal with heterogeneous terminologies/languages (e.g., pharmacy vs farmacy vs farmacia) that are used in the OGD sources.

Two profiles of OSCPSEP users can make use of this component: Data Providers and Software developers. An OGD Data Provider uses the Semantic Annotation Component to upload annotations that correspond to its datasets. A Software developer can use the Semantic Annotation Component to find required datasets. For example, Figure 6 shows the current web-based user interface for the OGD annotation when a Software developer tried to find annotations that correspond to his needs.



retele

Results for *"retele"*

About 5 results

Retele Mappings

Created by · Nelson Saturno

Date Submitted · Jan 11, 2017 06:01

Datasets Title

Retele Dataset

Datasets Keywords

Retele ReteleMapping DatasetRetele

Language

http://id.loc.gov/vocabulary/iso639-1/en

retele1	(5)
retele2	(5)
ExampleCSV	(3)
MappingsCSV	(3)
Retele	(1)
Example	(1)
DatasetRetele	(1)
DatasetExample	(1)
ReteleMapping	(1)

Figure 6 The current user interface for the OGD annotation.

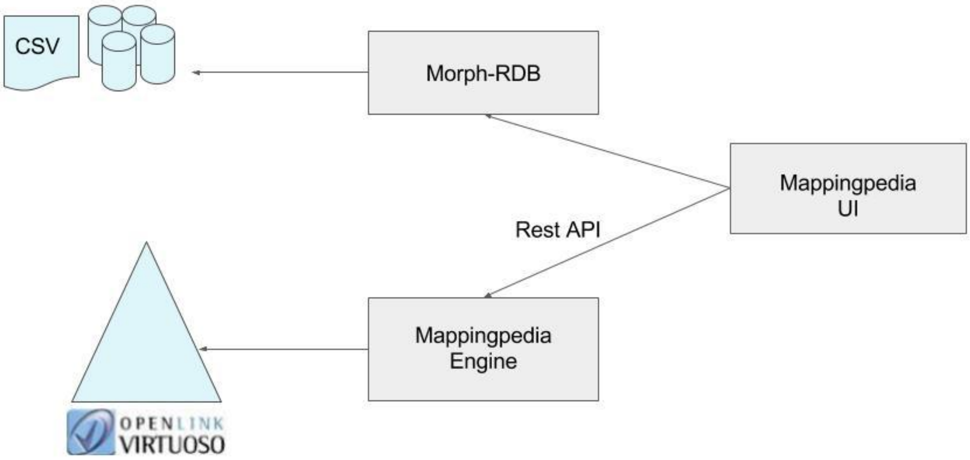


Figure 7 Sub-components of The Semantic Annotation Component

As illustrated in Figure 7, the Semantic Annotation Component consists of several sub-components: mappingpedia-engine, mappingpedia-UI, and morph-RDB. The mappingpedia-engine provides a REST API interface and is used in collaboration with Virtuoso Triple Stores to manage annotations. The mappingpedia UI is used as a web interface for the mappingpedia-engine. Additionally, the mappingpedia-engine facilitates the option of transforming datasets that are in the format relational database or CSV, into RDF.

Table 2 describes the attributes of the Semantic Annotation Component.

Attribute	***Description***
Name	Semantic Annotation Component
Purpose	Used to annotate OGD datasets
Integration points	Connection to the datasets
Integration method	JDBC connection and REST APIs
Use in OSCPSEP	By REST APIs or through the web interface
Use in other SaaS components	Interacts with the Search Module via the datasets
Use in Mobile-Age apps	Not directly used by apps, but via the Search component
Relation to scenarios in D2.2	Provide Data, Use Open Dataset
Use of OGD	The annotations of OGD datasets are maintained by this component
Responsible Partner	UPM

The Semantic Annotation Component supports annotating not only in a dataset level (such as the owner or keyword or the dataset), but also to a more detail level, such as tables and columns of the dataset. We use R2RML [2] for annotating Relational Databases (RDB) in the OGD sources. R2RML is a W3C recommendation to describe the relationship between a RDB to an ontology. One benefit of annotating a RDB dataset in OGD sources using R2RML is that it offers the capability to annotate beyond the dataset granularity level. In other words, we can annotate the tables and columns in the RDB dataset. An R2RML Mapping Document contains a set of Triples Map. Triples Map (`rr:TriplesMap`) are used to specify the relationship between a database's table with an ontology class (using property `rr:class`). The relationship between a database column and an ontology properties is specified via property `rr:predicateMap`.

In addition to providing a dataset and its annotation, a user may optionally provide her own ontology, that will be used by the Semantic Annotation Component, in a process that is called "*mapping saturation*". For example, given a mapping that annotates table "`FarmaciaDeComunidad`" as a type of "`CommunityPharmacy`" and an ontology that specifies that "`CommunityPharmacy`" is a subclass of "Pharmacy", then Semantic Annotation Component will also create a new mapping that annotates the table " `FarmaciaDeComunidad` " as a type of "`Pharmacy`" .

Annotating Datasets. The Semantic Annotation Component uses the classes and properties provided by DCAT [3] to annotate a dataset. Two main DCAT classes that are used are: Dataset and Distribution. A Dataset (dcat:Dataset) represents a single OGD database that is provided by the data owner. Some of the Dataset' properties are: the name and description of the dataset (dct:title and dcat:description), keyword that describes the dataset (dcat:keyword) , language (dct:language). Distribution (dcat:distribution) is used to describe how a database is distributed using properties such as media type (dcat:mediaType), access to the dataset (dcat:accessURL) , size of the dataset (dcat:byteSize) and when the dataset was last updated (dct:modified). Some of these properties, such as media type, dataset size, last modified date can be calculated by the module. For other properties, such as title, description, publisher, the data owner has to provide them manually, and this task is supported by the module, either via web interface or REST API calls.

Annotating Event. The Event class from schema.org (www.schema.org) initiative can be used to annotate events. Some important properties of this class that can be used are common properties such as name (schema:name) and description (schema:description); temporal related properties such as schema:startDate, schema:endDate and schema:duration; spatial related properties (schema: location); together with properties to specify the persons and organizations related to the event, such as sponsor (schema:sponsor), performer (schema:performer) and organizer (schema:organizer).

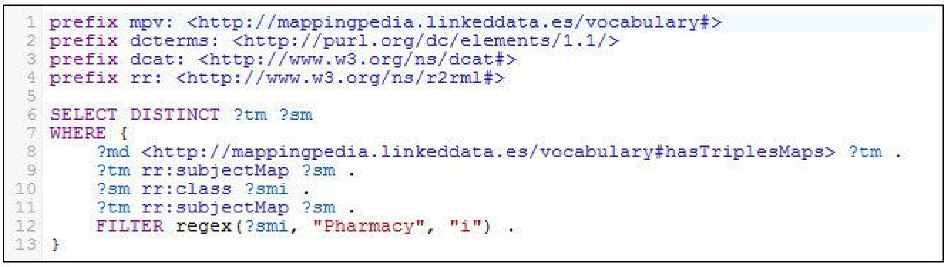
Annotating Location. The Basic Geo (WGS84 lat-long) Vocabulary (https://www.w3.org/2003/01/geo/#example) is a vocabulary that can be used to represent spatial properties. An important class of this vocabulary is a Point (geo:Point) that can stores geographic location using longitute and latitute properties (geo: long and geo: lat)

Storing and Retrieving Annotations. Annotations are represented as RDF triples and the Semantic Annotation Component uses Virtuoso Triple Store [4], to store them. Virtuoso is a solution for data access, integration, and relational database management (SQL Tables and/or RDF based Property/Predicate Graphs).

The following table contains information of the endpoint that can be called to upload a new annotation to the Semantic Annotation Module. We refer the reader to the Section 3.11.2 of the Deliverable D2.2 for other operations.

Endpoint	/ogd/resource/annotation/dataset/:dataset_id/upload-mappings
Method	POST
Description	Upload a new annotation for a specific dataset :datasetid
Headers	Application/json
Data	**mapping_text {String}** : The annotation to be uploaded
Parameters	**dataset_id {Number}** : The id of the dataset to be annotated.
Example	/ogd/resource/annotation/dataset/d001/mappings/upload-mapping

Since annotations are represented as RDF triples, all REST calls to the Semantic Annotation Component must be translated to SPARQL queries executed on the Virtuoso Triple Store. Every URL of the REST calls is associated with a SPARQL template and the Semantic Annotation Components instantiates this template with the values from the URL parameters. For example, the listing in Figure 8 is a SPARQL [5] query sent to the component to obtain all the annotations corresponding to the Pharmacy-related classes.



```
prefix mpv: <http://mappingpedia.linkeddata.es/vocabulary#>
prefix dcterms: <http://purl.org/dc/elements/1.1/>
prefix dcat: <http://www.w3.org/ns/dcat#>
prefix rr: <http://www.w3.org/ns/r2rml#>

SELECT DISTINCT ?tm ?sm
WHERE {
    ?md <http://mappingpedia.linkeddata.es/vocabulary#hasTriplesMaps> ?tm .
    ?tm rr:subjectMap ?sm .
    ?sm rr:class ?smi .
    ?tm rr:subjectMap ?sm .
    FILTER regex(?smi, "Pharmacy", "i") .
}
```

Figure 8 Example SPARQL Query.

5.2 OGD Sources Search and Data Transformation Component

This chapter covers the components for search and transformation of OGD.

5.2.1 OGD Sources Search

The OGD data sources search module is responsible for identifying external data sets that provide open government data. This module stores a list of data sets that expose open government data together with the relevant information (metadata) of each set, such as license, access point, contributors/providers among others. These data can be linked to the annotations from the Semantic Annotation component above. The data sets can be presented in different forms including access via a web service, CSV file, or data on a webpage.

The OGD search will return a list of datasets from the OSCPSEP repository including the above annotations (metadata), according to the search criteria. To help the Software developers explore that the datasets correspond to their needs, a sample of records will also be provided.

The relations among the concepts of the annotations will allow the users to search for data sets more efficiently. Similarly, the concepts will be linked across languages so that the search scope can be extended to cover more languages.

The initial service will only connect to external sources via API. We will try to avoid storage of third party data as far as possible. A reference via API is expected to avoid liability issues for the Mobile-Age consortium and to assure access to the latest version of the data. We will also add a disclaimer to the Mobile-Age services to clarify what users can expect from the services.

Please note that, for the data sources where an API is provided, we will provide a data conversion to enable the Software developers to access the data from different sources in the same way as from a unified API.

Table 3 OGD Sources Search Attributes

Attribute	***Description***
Name	OGD Sources Search Module
Purpose	Used to search for OGD datasets.

Integration points	Connection to the datasets.
Integration method	JDBC connection and REST APIs
Use in OSCPSEP	By REST APIs
Use in other SaaS components	Interacts with the Annotation Module via the datasets
Use in Mobile-Age apps	Not directly used by apps, but via the Search module
Relation to scenarios in D2.2	Scenario – Acquire Open Datasets
Use of OGD	The search enables Software developers to find OGD datasets.
Responsible Partner	TT

5.2.2 Data transformation component

This component will implement ways to transform relevant OGD sources so that they can be correctly mapped to common formats used in the OGD repository.

The data transformation component can take input in different formats (including, XML JSON, and also JSON-LD) and will transform the data into the internal format of the OSCPSEP OGD repository for data related to locations. The transformation is guided by a configuration entered by the OGD data provider which contains mapping of the retrieved data attributes to the OGD repository data attributes. The configuration defines what data attributes to collect data from for the transformation component. The transformation component can also handle transformation from different coordinate systems to latitude and longitude. The currently supported transformation of coordinates systems are UTM and what the EPSG Geodetic Parameter Dataset supports [6]. When the data is transformed, it is stored in the OGD repository.

The database tables to store the venue, data sets, and data about the data transformation is presented in Annex 1. An example to illustrate the transformation from external to internal data is outlined in Annex 2.

The OGD data sets will be provided to the developed apps either as the result of an API call to the OSCPSEP database or through an OSCPSEP API which forwards the request to an external OGD source API. These API calls are described in D2.2. The data from the API call is returned as JSON or possibly as XML if needed. An example data set is shown below.

Search for hospitals in Zaragoza:
URL:http://ma.tingtun.no/datasets/category/hospitals/es/zaragoza/

JSON response:

{"result": [{"category": 23, "categoryName": "Hospitales", "publisher": "zaragoza", "issued ": null, "hasVersion": null, "accessRights": "Public data", "landingPage ": "https://www.zaragoza.es/sede/servicio/equipamiento/", "language": "es", "modified ": "2016-11-04T09:17:02Z", "id": 22, "keywords": null, "description": "Data source from zaragoza", "spatial ": "Spain, Zaragoza", "accrualPeriodicity": null, "versionNotes ": null, "identifier": null}]}

The choice to store data in the OSCPSEP or to access an external OGD source will depend on the format of the source and the properties of the API. For example, if the OGD source response time is too long then the data can be stored locally (or cached) to improve the response time for the application. Other reasons to store data in the OGD repository include that the data may be contain inconsistencies in format, or not open yet, or not consistently accessible or maintained.

Webscraping will be implemented based on the import.io [7] functionality or similar. This functionality can extract data from a webpage with a simple configuration and convert into a JSON response from an API call. The service is freely available for up to 500 queries (OGD sources download) per month.

Venue coordinates are stored in the original format and, if needed, converted to longitude, latitude. Currently the conversion from UTM to latitude/longitude is supported.

The venue types are mapped to an identifier in the OSCPSEP database which will enable straightforward translation of this data into additional languages.

5.2.3 Service search component

Currently the number of relevant services is not known. Depending on the number of services we will either opt for presentation in a list extracted from the database or a search resembling the search functionality to find OGD data sets.

If the list of services requires a simple search, then a service search functionality will be designed in a similar way like the OGD data set search. This will ensure that the user interaction is consistent across these two search services. The actual search will be carried out based on the available metadata about the services. When services are located, the user may - when available - also explore a demonstrator and example implementations to better assess the capabilities of the service.

The attributes table is very similar to the one for the OGD search.

Table 4 Service Search Component Attributes

Attribute	***Description***
Name	Service Search Component
Purpose	Used to search for services related to the use of OGD.
Integration points	Connection to the services descriptions table.
Integration method	JDBC connection and REST APIs
Use in OSCPSEP	By REST APIs
Use in other SaaS components	No expected use in other SaaS components

Use in Mobile-Age apps	Not directly used by apps, but via the Search module
Relation to use cases in D2.2	Scenario – Acquire Services
Use of OGD	The search enables Software developers to find services.
Responsible Partner	TT

5.3 Concept lookup component

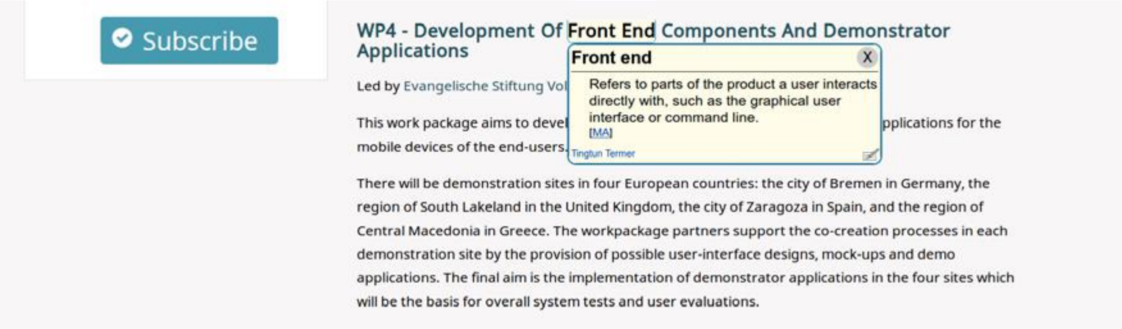
The concept lookup component enables the user to retrieve definitions of concepts or translations of words and yet remain in the interaction with the application. This can both facilitate navigation and tailor the lookup results to the content from which the lookup is initiated.

This is particularly helpful for the Older adults who may have challenges both with the understanding of specific terms and with navigation of multiple browser windows.

The descriptions of terms can be plain text, or include images. The solution can present definitions from one or more sources in the results list from one lookup. Existing sources can be manually entered (typed in), imported or integrated via an API.

For language pairs where general dictionaries are available the module can support translation of words to help non-native users.

The Concept lookup component is based on a service owned by Tingtun.



WP4 - Development Of Front End Components And Demonstrator Applications

Led by Evangelische Stiftung Vol

This work package aims to devel pplications for the mobile devices of the end-users.

Front end

Refers to parts of the product a user interacts directly with, such as the graphical user interface or command line.
[MA]

Tingtun Termer

There will be demonstration sites in four European countries: the city of Bremen in Germany, the region of South Lakeland in the United Kingdom, the city of Zaragoza in Spain, and the region of Central Macedonia in Greece. The workpackage partners support the co-creation processes in each demonstration site by the provision of possible user-interface designs, mock-ups and demo applications. The final aim is the implementation of demonstrator applications in the four sites which will be the basis for overall system tests and user evaluations.

Figure 9 Example lookup of "Front end" on the Mobile-Age website (www.mobile-age.eu). The meaning of the concepts is retrieved from the Mobile-Age term base. In this way, we can maintain just one glossary for each part of the project.

Figure 9 shows the description of the concept «Front End» retrieved from the Mobile-Age term base.

The use of the same glossaries through the same user interface across multiple public websites can deliver a unified search to implement a loose integration among them for the older adults.

Table 5 Concept lookup attributes

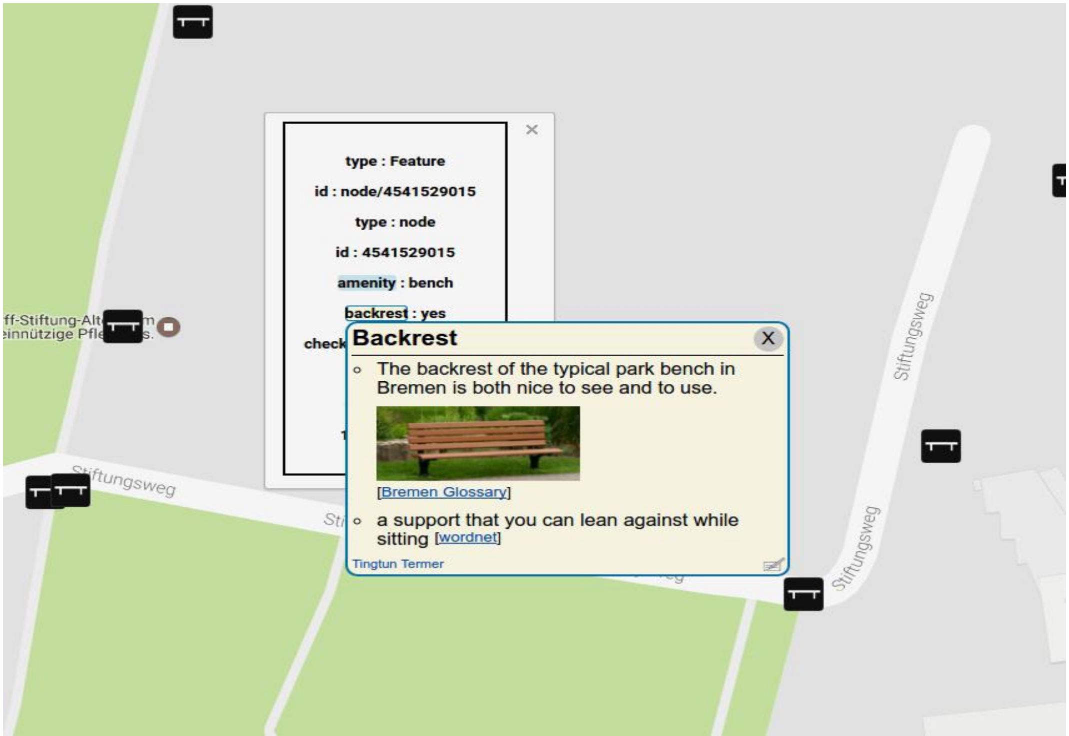
Attribute	***Description***
Name	Concept lookup Component
Purpose	Connect texts on web to glossaries to provide definitions of technical terms or to provide translations to another language.
Integration points	Website / Mobile-Age Apps to be developed / Web browser
Delivery form	Javascript in web page / Javascript in App code / Browser addon to Firefox or Chrome.
How to integrate	Javascript
Use in OSCPSEP	Glossary to support Software developers.
Use in other SaaS components	Integration into the accessible maps and potentially other SaaS components
Use in Mobile-Age apps	Can connect to apps based on webpages to enable users to lookup from glossaries.
Relation to scenarios in D2.2	Use Front Ends
Use of OGD	Can integrate OGD glossary contents, and can extend OGD sources projected on geographical maps.
Responsible Partner	TT

6 SaaS components integration outline and status

The SaaS components will be integrated both into the OSCPSEP platform and can be combined in the applications prepared by Software developers for older citizens to use.

The OGD transformation component is used to support the transformation and import of external data sources. This will subsequently, for example, enable a Software developer to project a selection of datasets on a map implemented by the map component. The Analytics component can be integrated into apps as to inform Software developers about the actual app usage as input for further improvements.

The Concept lookup component can enable users to define and maintain their own layer of information to a map. Users can enter their own content, which can support groups of Mobile-Age users to maintain layers of information, for example on top of city maps as indicated in Figure 10. This layer can contain both text and images.



type : Feature

id : node/4541529015

type : node

id : 4541529015

amenity : bench

backrest : yes

check

Backrest

- The backrest of the typical park bench in Bremen is both nice to see and to use.

[Bremen Glossary]

- a support that you can lean against while sitting [wordnet]

Tingtun Termer

Figure 10 Map of park bench locations in Bremen. The image shows a lookup of the concept "backrest" from a textual description of a park bench location. Note that two records are provided, one tailored to the Bremen use-case and one general, from Wordnet.

The current experimental map component is distributed as a JS code module. The actual map tiles are retrieved from Google maps. On top of the Google maps layer the current map component has a navigation GUI with improved accessibility, and the ability turn on/off the indication of points of interest. The OGD data can be imported from XML files and shown on top of the map. The API to integrate the map component in Mobile-Age apps is indicated in D2.2 chapter 2.12 Reusable front-ends Module. The details of the API are not yet elaborated.

The concept lookup component can be integrated in the reusable front-ends components by adding a line of HTML code to refer to a Javascript. The concept lookup sends the concept to look up, and an API key to the concept lookup request relay which forwards the request to the external concept source. The lookup component generates a popup with the response from the concept lookup server.

The actual integration between the components is realised by means of REST APIs apart from the Analytics component and the Concept lookup component where the integration is based on inclusion of a Javascript.

The initial integration experiments so far show that the SaaS approach can facilitate the re-use and consistency of services across applications. The approach facilitates the development and use of the components. The separation of concerns can also support convenient integration of open source components with proprietary components.

The use of the OGD Data Transformation Module is indicated in Figure 11 .

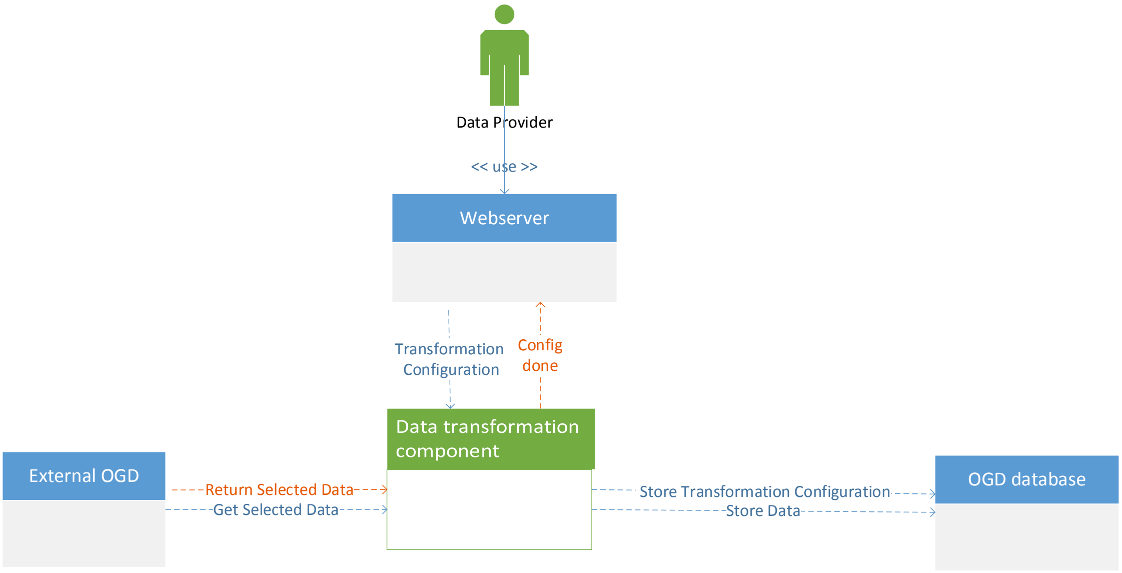


Figure 11 OGD Data Transformation and storage.

Please refer to D2.2 for a high level description of the APIs of the components. A description of the OGD API generated by Swagger [8] is presented at http://83.212.100.226/swagger/.

More details will be available in D2.4 for the Behaviour analytics module and in D4.1 Technical requirements and specification of demonstrator applications will cover the API for the front-ends module.

The status of the components is indicated in Table 6. Three of the SaaS components can now be accessed online for an initial test. The Map component can present a map from Bremen with several points of interest, the concept lookup component can be used to read this deliverable with access to a glossary, and the OGD Sources Search and Data Transformation component can, for example, present a list of hospitals in Zaragoza. Further examples can be composed by referring to the Swagger page.

The initial test of the OGD sources search has been carried out by the Software developers in Mobile-Age. The OGD sources search service is prepared by Tingtun and tested by AUTH, ULANC, and by UPM. The tests have been carried out in two online meetings as the services have been improved.

The OGD annotation component is prepared by UPM and tested by AUTH, and by TT in a similar way to the OGD sources search.

The purpose of these initial tests has been to see that the basic functionality of the services is in line with the specifications and expectations of the developers in the project.

The Concept lookup component has been tested both with the developers and with partners from the co-creation tasks of the project. The tests have shown that the Concept lookup component can be used to present concepts on webpages, PDF documents and on the maps provided by the Map component.

Additional testing will be carried out, with additional users, once the platform has been launched.

Table 6 Components status (updated for status 2017-07-07)

Component	*Status*	*Online availability*
Semantic Annotation Component	Demonstrator ready for initial test	Not yet.
OGD Sources Search and Data Transformation component	Proof of concept ready for initial test	Yes, e.g. Get datasets for pharmacies
Service search component	Proof of concept ready for initial test	Not yet.
Map component optimized for seniors	Demonstrator ready for initial test	Yes, e.g. Map with beautiful sites.
Concepts lookup component	Demonstrator ready for initial test	Partly, e.g. this document with access to Mobile-Age glossary. Or with the map component as shown in Figure 7.

7 Innovation Aspects (comparison to the state of art)

7.1 Novel aspects of the Semantic Annotation Module

Most of the Open Government Datasets that are provided have not been annotated and the lack of annotation may hinder the capability of developers to find and understand the content of the datasets. The Semantic Annotation Module is included as part of the Mobile-Age Platform as an online collaborative environment for the annotations of Open Government Datasets. In comparison to existing systems, the Semantic Annotation Module is a novelty given several aspects:

- A repository to share and browse annotations that have been formulated in the W3C Recommendation R2RML language and its extensions (such as RML).
- An environment that helps its users to obtain annotated data without the need to installing any of R2RML and its extensions engines.
- An environment that allows any of its users to request annotations for the Open Government Datasets that have not been annotated.
- An environment that allows any of its users to response to the annotation request by providing the corresponding mappings.

The complete description of the Semantic Annotation Module can be found in the demo paper of Extended Semantic Web Conference 2017 [9].

Noy et al [10] proposed a repository for storing ontology mappings. Unlike the aforementioned approach the semantic annotation module supports mappings between data source and ontologies that allows the publications of semantically enriched open data.

There is some prior art on semantic annotation of OGD sources, such as [11]. However, these approaches seem to assume that the data is available as Linked Open Data (LOD), which is often not the case for the sources relevant for Mobile Age.

7.2 Novel aspects of the OGD and services search

The OGD and Services Search is a meta search to enable the user to find OGD sources or public services in several repositories with one search while using the OSPSEP to develop apps.

There are several examples of meta search applications both for general search on the web, for OGD and for public services. The tight integration with the OSCPSEP will allow the developers to be more efficient in their work when searching for OGD sources and public services.

The search for OGD sources is built around Schema.org which has been made possible because we have annotated the content of OGD sources with the aforementioned vocabulary.

A challenge here is to map search phrases in multiple languages to English and to Schema.org. To deal with this, we have prepared an approach that works as follows:

1. A user may enter a search term in any language. If the search term is in English, we process to Step 2. Otherwise, we translate that term into English using a restful translation service. This service has a language detection and can provide a translation of words and phrases.

2. If the result from Step 1 is a concept of Schema.org, we process to Step 2. Otherwise, we match the result from Step 1 against the English items in Schema.org using a variant of string comparison based on the Levenshtein distance [12]. If there is a match then return the found item(s). This would resemble the "did you mean" functionality found in common search engines.
3. If the result from Step 2 matches with a class of Schema.org, then we proceed to send the request to the external OGD sources (i.e., Zaragoza or Bremen) and return the relevant hits (that include the matched class itself together with its subclasses) from them. The returned hits information is designed to enable the user (developer) to quickly decide if a given source or service is relevant to explore further. See the OGD and Services Search section for more details on this. Otherwise, a warning message is presented that informs the user that no OGD sources have been annotated with the search term.

The following example illustrates how the above process works:

1. A user searches for "Sitio". Because the search term is not an English word, the OGD and Services module translates this into "Place".
2. Since "Place" is a class of Schema.org, there is no need to perform any string similarity techniques.
3. The OGD and Services module returns datasets that have been annotated with "Place" or any of its subclasses, such as "Landmark" or "Historical Building".

The main contribution from this module is the way in which it can provide a seamless experience for the developer to increase productivity when using the OSCPSEP platform.

7.3 Novel aspects of the Mobile-Age Concept lookup

Effective use of online services often requires users to have access to relevant dictionaries and translations. Spelling errors in the search query and searching for terms that are not in the Termbase is among the most common mistakes users make when using dictionary portals [11]. An in-line lookup service will eliminate the need for users to type the concepts as they are fetched directly from the text.

The Mobile-Age Concept Lookup provides support for the older adults, and can also support developers, e.g. to read specific technical terms in specifications. The Mobile-Age concept lookup, based on Tingtun Termer [14], extends the state of the art for terminology approaches with a novel combination of features in an in-line lookup service.

The features include:

- Accessible dynamic highlighting of dictionary concepts in online texts. Accessible to meet the criteria for WCAG 2.0 level AA [15].
- Support for lookup both from HTML and from PDF.
- Support for general and special dictionaries in the same text.
- Ability to connect selected dictionaries to a given text.
- Easy addition of new concepts, and a way to create and share dictionaries among users, leaning on the crowdsourcing ideas from [16].
- Interoperability across web browsers and platforms including smartphone, tablet, and PC

There are several existing in-line lookup applications as presented in Table 7. The table indicates the coverage of some of the features of the Mobile-Age Concept Lookup compared to the existing related applications, as of 2017-10-05. From the discussions and testing with older adults we note that navigability and ease of use are essential for wider adoption. The ability to use PDF is essential given the wide use of PDF in government already. The recent development to replace paper mails with PDF documents sent to a government hosted mailbox will further raise the demand for a Concept Lookup that can be used with this format.

Table 7: The Concept lookup compared to a selection of related in-line dictionary lookup applications, as of 2017-10-05.

Name	*Platform*	*Dict. provided*	*Add new word*	*Add more dict.*	*PDF*	*Comment*
Amazon Kindle dictionary	Hardware	Oxford dictionary of English + more.	y	y	y	Dictionary lookup is not available for Kindle for PC.
Apple iPhone lookup	Hardware	Apple or other	n	y	n	Only works on iOS. Earlier called "define"
Google Chrome translate	Chrome extension	Google translate	n	n	n	
Wiktionary and Google Translate	Firefox Browser plugin	Wiktionary.org formerly Google dictionary	Y via Wiki-ped-ia	n	n	Connected to advertisements. Require plug in installation and just one dictionary option.
Dictionary.com pop-up extension	Chrome, Firefox, IE.	Dictionary.com	n	n	n	Has ads.
TheFreeDictionary.com Extension	Chrome, Safari, Kindle	TheFreeDictionary.com	n	n	n	Ads or pro version. Not really in-line, opens new tab for each lookup.
Web dictionary widget	JavaScript on page	Wiktionary.org	Y via Wikt-ion-ary	n	n	Has ads.
Wordweb	Windows	Wordweb, Oxford, Collins + more	y	y	y	In-line lookup only on Windows.
Mobile-Age Concept Lookup Service	Chrome, Firefox or JavaScript on page.	Wordnet, and user defined.	y	y	y	Works across Smartphones, Tablets, and PCs

The dictionary functionality for the related services provided on iPhone and Kindle is limited to their hardware platforms. Dictionary lookup is not available for Kindle for PC. There are several similar applications e.g. Terminology Dictionary for iOS. One thing they have in common is that they are restricted to their operating system.

Some Important limitations with existing browser plugins are that they can use only a limited set of dictionaries and that none of them are designed to allow the user to add new dictionaries or to read PDF documents.

Both the Kindle reader and the Wordweb application let the user add new concepts and read PDF documents. However, their use is limited to the Amazon Kindle hardware or to the Windows operating system, respectively.

The Mobile-Age Concept Lookup overcomes these limitations. In addition, it is designed to improve the relevance of the returned information by several measures:

- Provide seamless access to combinations of general and specialized dictionaries (terminologies). The concepts from terminologies will be highlighted while the remaining text can be referenced for a lookup in a general dictionary. Any range of the text can be selected with a key combination to request the general concept meaning whereas concepts included in specialised dictionaries are highlighted for the user to interact with. For example, this feature can be used to read a patient information leaflet.
- Support preselection of dictionaries to use on a given text. The approach enables user to configure glossaries to use on different content and authors to preselect dictionaries to be used on their text.
- Present only the meaning of concepts without advertisements. The presentation of advertisements together with the meaning can cause confusion and navigability issues. Personalised advertisements may also give rise to privacy issues since they will have to rely on personal information collected about the users.

To meet the requirements raised by the General Data Privacy Regulation (GDPR) [17] the Concept Lookup has been designed to collect and store as little information as possible about the users and their use. And to conform to the Web Accessibility Directive (WAD) [18] the user interface is prepared to support accessibility as derived from WCAG 2.0, including tab navigation, zooming, and screen readers. Moreover the user interface is designed to enable users to build on their prior experience with related services in line with the recommendations from [19].

We have searched the web and the literature for electronic dictionaries and digital glossary research and products. We have also been in contact with translators in the European Parliament, the Norwegian Language Council, NATO, ISO, Termcat in Spain, TNC in Sweden and several further experts on terminology and related technology.

The findings from these investigations confirm that the Concept lookup, based on Tingtun Termer, will bring the state of the art a significant step ahead with a novel combination of features with support for accessibility, openness, and interoperability. The openness enables users to add and to share new dictionaries and concepts. The interoperability assures that both PDF and HTML documents can be used and that the Concept lookup will work across apps on both Android and iOS. The accessibility will assure WCAG 2.0 conformance and enable more of the older adults to use the service.

The Mobile-Age Concept lookup is planned to be used in the project as part of the user interface for the developers using the OSCPSEP platform, and as a reusable component to be included in the Mobile-Age apps. This technology has also been used as the basis for two recent publications [20], [21] and an invited talk [22].

8 Key findings (conclusions)

The SaaS components will be used to provide services both within the OSCPSEP platform and within the apps to be elaborated by the Software developers. They are designed to enhance each other in terms of functionality. The front-end components prepared to meet the needs of older citizens and to aid discussions in the co-creation process across the Mobile-Age project field sites.

We refer to the key scenarios in chapter 3 of D2.2, which entail the scenario Acquire open datasets. Before we can run this scenario, the data sets will be made available in the OSCPSP OGD database. This is supported by the Data transformation component with the Semantic Annotation Component. The experimental integration of the components for data transformation has shown that we are able to connect to OGD sources in Zaragoza and in Bremen and to transform the data for venues and store in the OGD repository. The annotation component is able to annotate datasets, and to connect to widely used vocabularies.

We have then verified that the datasets in the repository can be searched in line with the Acquire Open Datasets scenario in D2.2. This will be helpful for more efficient search and use of OGD across language borders.

The report also indicates a variant of the Module interaction scenario from chapter 5 of D2.2, with an integration between the preliminary version of the Reusable map front-end component and the Concept lookup component. The latter is also used to demonstrate how Mobile-Age concepts can be looked up from this report.

The experience so far from the implementation and initial tests confirm that the SaaS approach is helpful and viable for the Mobile-Age deployment. The approach can facilitate the re-use and consistency of services across applications. Consistency can be simply improved by re-use and the REST APIs can help to both connect and to isolate the components. The impact of the benefits in practice from flexible integration, isolation of concerns, and efficient re-use are yet to be verified.

9 References

[1] International Standard ISO/IEC TR 20000-10:2015(en) Information technology — Service management — Part 10: Concepts and terminology, https://www.iso.org/obp/ui/#iso:std:iso-iec:tr:20000:-10:en (accessed 2017-10-05)

[2] Das, S., Sundara, S., & Cyganiak, R. (2012). R2RML: RDB to RDF Mapping Language. W3C Recommendation.

[3] Maali, F., Erickson, J., & Archer, P. (2014). Data catalog vocabulary (DCAT). W3C Recommendation.

[4] Virtuoso – a hybrid server to deal with data representations as Relational Database, graphs, several formats such as JSON, XML, etc. https://virtuoso.openlinksw.com/ (accessed 2017-10-05)

[5] SPARQL – RDF Query language, http://www.w3.org/TR/sparql11-query/ (accessed 2017-10-05)

[6] EPSG Geodetic Parameter Dataset, https://www.epsg-registry.org/, (accessed 2017-01-25)

[7] Import.io is a service to convert data on webpages to JSON, https://www.import.io/ (accessed 2017-10-05)

[8] SWAGGER is an open source framework to build and to document RESTful APIs. http://swagger.io/ (accessed 2017-10-05)

[9] Priyatna, F., E. Ruckhaus, N. Mihindukulasooriya, O. Corcho and N. Saturno, MappingPedia: A Collaborative Environment for R2RML Mappings. ESWC 2017 Demo session. Portoroz, Slovenia. June 2017.

[10] Noy N.F., Griffith N., Musen M.A. (2008) Collecting Community-Based Mappings in an Ontology Repository. In: Sheth A. et al. (eds) The Semantic Web - ISWC 2008. ISWC 2008. Lecture Notes in Computer Science, vol 5318. Springer, Berlin, Heidelberg

[11] Sorrentino S., Bergamaschi S., Fusari E., Beneventano D. (2013) Semantic Annotation and Publication of Linked Open Data. In: Murgante B. et al. (eds) Computational Science and Its Applications – ICCSA 2013. ICCSA 2013. Lecture Notes in Computer Science, vol 7975. Springer, Berlin, Heidelberg

[12] A measure for the difference between two strings of text. https://en.wikipedia.org/wiki/Levenshtein_distance (accessed 2017-10-05)

[13] Heinisch-Obermoser, B., (2016). Web Interfaces of Terminological Databases that are Available on the Internet from a Usability Perspective, TERM BASES AND LINGUISTIC LINKED OPEN DATA, eds. Hanne Erdman Thomsen, Antonio Pareja-Lora & Bodil Nistrup Madsen, TKE 2016 12th International conference on Terminology and Knowledge Engineering.

[14] Project partner Tingtun website http://tingtun.no/ (accessed 2017-10-05)

[15] The Web Content Accessibility Guidelines from W3C, http://www.w3.org/TR/WCAG20/ (accessed 2017-10-05)

[16] Karsch, Barbara I. 2015. “Terminology Work and Crowdsourcing.” In Handbook of Terminology, edited by Hendrik Kockaert and Frieda Steurs, 289–303. Amsterdam: Benjamins.

[17] Regulation on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) http://eur-lex.europa.eu/eli/reg/2016/679/oj (accessed 2017-10-05)

[18] Regulation on the accessibility of the websites and mobile applications of public sector bodies http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016L2102 (accessed 2017-10-05)

[19] Nielsen, Jakob. 2010. Usability engineering. Amsterdam: Morgan Kaufmann.

[20] Snaprud, M.H., B.E Munkvold, J. Radianti, Terminology to save lives, Nordterm 2017, Kongsberg, Norway, 12-15.June, 2017, Conference proceedings to appear 2017.

[21] Snaprud, M.H., J. Radianti, Better Access to Terminology for Crisis Communications, First IFIP Conference on Information Technology in Disaster Risk Reduction (ITDRR 2016), Sofia, Bulgaria, November 16 – 18. 2016.

[22] Snaprud, M.H., Language technologies from the user's perspective, invited talk at Language technologies for Europe, EC Round Table, Luxembourg, 13. December 2016, https://ec.europa.eu/digital-single-market/en/news/language-technologies-round-table, (accessed 2017-10-05).

Annex 1: OSCPSEP database Schema for venues

Venue

Attribute	Type	comment
id	IntegerField	Primary key
name	TextField	name of venue
external_id	TextField	ID given in external source
email	TextField	email to venue
data_url	TextField	URL from which data was retrieved
phone_number	TextField	venue phone number
fax_number	TextField	venue fax
url	TextField	venue URL
last_updated	Datetime	Data record last updated
address	TextField	venue address
street	TextField	venue address
po_box	TextField	venue address
po_box_number	TextField	venue address
house_number	TextField	venue address
location	TextField	venue address
post_number	TextField	venue address
post_name	TextField	venue address
coordinates_type	TextField	lat/long UTM etc.
coordinates	TextField	string with coordinates values
longitude	FloatField	longitude float number
latitude	FloatField	latitude float number
dataset	FK(Dataset)	Foreign key to dataset
description	TextField	description of the venue
opening_hours	TextField	opening hours as string
accessibility	TextField	text description about accessibility of the venue
image_url	TextField	image of venue
generatedName	TextField	If there is no name/title attribute in the retrieved data, a generated name created.

Dataset

Attribute	Type	comment
id	IntegerField	Primary key
description	TextField	This property contains a free-text account of the Dataset. This property can be repeated for parallel language versions of the description.
keyword	TextField	This property contains a keyword or tag describing the Dataset.
publisher	FK(Datasource)	Foreign Key to Datasource
category	IntegerField	This property refers to a category of the Dataset. A Dataset may be associated with multiple themes.
accessRights	TextField	This property refers to information that indicates whether the Dataset is open data, has access restrictions or is not public. A controlled vocabulary With three members (:public, :restricted, :non-public) will be created and maintained by the Publications Office of the EU.
accrualPeriodicity	TextField	This property refers to the frequency at which the Dataset is updated.
identifier	TextField	This property contains the main identifier for the Dataset, e.g. the URI or other unique identifier in the context of the Catalogue.
landingPage	TextField	This property refers to a web page that provides access to the Dataset, its Distributions and/or additional information. It is intended to point to a landing page at the original data provider, not to a page on a site of a third party, such as an aggregator.
language	CharField	This property refers to a language of the Dataset. This property can be repeated if there are multiple languages in the Dataset.
issued	Datetime	This property contains the date of formal issuance (e.g., publication) of the Dataset.
modified	Datetime	This property contains the most recent date on which the Dataset was changed or modified.
spatial	TextField	This property refers to a geographic region that is covered by the Dataset.
versionInfo	TextField	This property contains a version number or other version designation of the Dataset.
versionNotes	TextField	This property contains a description of the differences between this version and a previous version of the Dataset.

		This property can be repeated for parallel language versions of the version notes.
license	TextField	Added to DCAT: license for the data such as public domain, Creative Commons etc.
mobilage_updated	Datetime	Added to DCAT: timestamp for last time the data was retrieved from external source.

All but the two last attributes from DCAT.

Table to connect venue types to their names in different languages.

CategoryName

Attribute	type	Cmt
id	IntegerField	Primary key
category	IntegerField	e.g. 3 to represent hospitals.
name	TextField	name of venue type in given language
Language	CharField	given language, as language code refer to Language codes - ISO 639. See also http://www.iso.org/iso/home/standards/language_codes.htm

Datasource

Attribute	type	cmt
id	IntegerField	Primary key
accessURL	TextField	This property contains a URL that gives access to a Distribution of the Dataset. The resource at the access URL may contain information about how to get the Dataset.
description	TextField	This property contains a free-text account of the Distribution. This property can be repeated for parallel language versions of the description.
api_config	FK(api_config)	Foreign key to api_config
license	TextField	This property refers to the license under which the Distribution is made available.
language	CharField	This property refers to a language used in the Distribution. This property can be repeated if the metadata is provided in multiple languages.

title	TextField	This property contains a name given to the Distribution. This property can be repeated for parallel language versions of the Description.
datasets	JSONField	List of datasets contained in the data source. Represented as reference to CategoryName + optional additional data about the data set.

api_config

Attribute	type	cmt
id	IntegerField	Primary key
format	TextField	The format the data set is represented as, e.g. XML, CSV, JSON.
datatag	JSONField	indication of how to find the venue record in the import data.
tags	JSONField	indication of which fields freom the venue records are to be used.
datanames	JSONField	Mapping between external venue format to OSCPSEP DB schema.
tag_format	JSONField	Data on external data types such as coordinates representation as UTM30T.
source_location	TextField	Indication of whether the source is available online or data is uploaded by the data provider.
downloadURL	TextField	URL to data source if available online
localFilePath	TextField	File path to uploaded file on server.
namespaces	BooleanField	Indication of weather namespaces are used in XML files.
online_api	BooleanField	indicate weather data source has an API
api_name	TextField	A name the data provider gives the API.
combinedAttributes	JSONField	This property is used to describe when two or more retrieved data fields are combined into one value. E.G when the retrieved data do not have a name, combine id and type into a generated name value.

Annex 2: Data Transformation Example

The transformation of external sources to the OSCPSEP OGD data format is carried out in a sequence of steps supported by the data transformation component (DTC). The form of the content will be automatically detected as far as feasible to help the data provider (DP). In this way, the DTC will attempt to identify the data format (XML, JSON, CVS) the data record structure and the field types contained in the records.

1. DP enter the file name or URL pointing to the data.
2. DTC attempt to automatically detect data format. Can be supported by e.g. Python Magic https://github.com/ahupp/python-magic . in case the format cannot be automatically detected, the DP is asked to enter the data format.
3. DTC attempt to automatically detect the data record format. The DP is shown one record of the data. The DP can override this. If the record format cannot be automatically unambiguously determined, then the DP will be prompted to identify tags (XML), and keys (JSON) and if needed select delimiter and separator characters (CSV) while the DTC show the effect of the different selections, in a similar way like when importing a file into a spreadsheet.
4. DP indicate what fields to include and those to ignore from the external data source. Each external field is mapped to an internal one. The internal ones are provided in dropdown options which correspond to the internal OGD database attributes.
    1. One exception to the above is the case where multiple external fields are to be combined into one internal. Then the selection and the sequence of them and the internal field is entered by the DP.
5. DTC attempt to automatically detect the data field types based on information retrieved from the data souce meta data if available such as XML namespaces or JSON-LD @context, or from the mapping for external to internal fields given by the DP. In case the data field types cannot be automatically detected, the DP is asked to enter this.
6. The format of fields that contain geographical coordinates or time/date need to be known before the transfer.
    1. DTC will attempt to identify coordinate system format. This can be supported by e.g. Pyproj. https://pypi.python.org/pypi/pyproj . DP is asked to confirm it is correct. In case the coordinate system format cannot be automatically detected, the DP is asked to enter the coordinate system format.

2. DTC will attempt to identify date/time format. This can be supported by e.g. the Dateutil http://dateutil.readthedocs.io/en/stable/
   DP is asked to confirm it is correct, or change the time format to the correct type.

7. Store the transformation configuration setup in the OGD database.
8. DTC read the transformation configuration as generated by the preceding steps.
9. Apply the transformation configuration on the given data source to carry out the transformation.

Example with data

Transformation Configuration

Format = xml

License = Creative Commons Namensnennung (CC BY 3.0)

Language = de

Publisher = Bremen, Germany

Source_location = local

namespaces = true

source = api/ogd_load_scripts/ogd_data/ (File path or URL)

Config_name = bremen_bank

Online_api = false

```
Datasets = {
    "52": {
        "urlPrams": "Bank.gml", (File name or url ending)
        "release": "01.11.2011",
        "last_updated": "18.02.2015",
        "description": "Der Umweltbetrieb Bremen ist als Eigenbetrieb der Hansestadt Bremen u.a. für die Pflege und Unterhaltung öffentlicher Grünanlagen sowie für die kommunalen Friedhöfe zuständig. Zu weiteren Aufgaben zählen u. a. die Straßenreinigung und der Winterdienst auf den Straßen (Bremen-Nord), die dem kommunalen öffentlichen Verkehr gewidmet sind.\n\nZur Verfügung gestellt werden die Parkbänke. Bei Parkbänken handelt es sich um sogenanntes mobiles öffentliches Mobiliar, wobei mobil hierbei häufig wörtlich zu nehmen ist. Die Standorte unterliegen einer dynamischen Veränderung, die die verschiedensten Ursachen hat. Z. T. ist Vandalismus anzutreffen, teilweise werden Standorte von den Nutzern verändert, teilweise müssen sanierungsbedürftige Bänke abgebaut werden.",
        "origin": "https://ssl5.bremen.de/transparenzportal/sixcms/detail.php?gsid=bremen236.c.4424.de&asl=bremen02.c.734.de&id=4424",
        "Website": "http://transparenz.bremen.de/de/datensatz/bremen236.c.4424.de",
        "Contact": "Belinda Daniel",
        "Email": "Belinda.Daniel@ubbremen.de",
        "Name": "Datensatz Parkbänke in öffentlichen Grünanlagen und auf kommunalen Friedhöfen"}}

DataTag = {"featureMember": {}}
Tags = {
   "OBJECTID": "",
   "pos": "",
   "BEZEICHNUN": "",
   "Baenke": ["id", "true"]}

datanames = {
   "id": "external_id",
   "BEZEICHNUN": "generatedName",
   "OBJECTID": "generatedName",
   "pos": "coordinates"}

combine = {
   "generatedName": ["BEZEICHNUN", "OBJECTID"]}

format = {
   "coordinates": ["EPSG:31493", " "],
   "Meta": {"last_updated": "%d.%m.%Y",
            "release": "%d.%m.%Y"}}
```

One record from the data file

```
<?xml version="1.0" encoding="UTF-8"?>
<gml:FeatureCollection xmlns:gml="http://www.opengis.net/gml"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:fme="http://www.safe.com/gml/fme"
xsi:schemaLocation="http://www.safe.com/gml/fme Bank.xsd">
<gml:boundedBy>
<gml:Envelope srsName="EPSG:31493" srsDimension="2">
<gml:lowerCorner>3467460.05891586 5877034.3363621</gml:lowerCorner>
<gml:upperCorner>3497665.91001599 5897967.61922959</gml:upperCorner>
</gml:Envelope>
</gml:boundedBy>
<gml:featureMember>
<fme:Baenke gml:id="id5e409b8b-63f6-4af8-b343-405497588753">
<fme:OBJECTID>1</fme:OBJECTID>
<fme:BEZIRK>21</fme:BEZIRK>
<fme:OBJEKT>22154</fme:OBJEKT>
<fme:PFLE>5311</fme:PFLE>
<fme:TEIL>123</fme:TEIL>
<fme:OBJART>100</fme:OBJART>
<fme:BEZEICHNUN>WALLANLAGEN
MUEHLE/BLUMENSCHULE</fme:BEZEICHNUN>
<fme:GKZ>300</fme:GKZ>
<fme:AMT>67.3</fme:AMT>
<fme:GROESSE>1</fme:GROESSE>
<fme:DATEN>0</fme:DATEN>
<gml:pointProperty>
<gml:Point srsName="EPSG:31493" srsDimension="2">
<gml:pos>3487138.76699818 5883091.90900774</gml:pos>
</gml:Point>
</gml:pointProperty>
</fme:Baenke>
</gml:featureMember>
```

Further details on the transformation

With this example transformation configuration, the file format is set to XML, it uses namespaces and each venue is under the featureMember tag. This way, the transformation gets all tags and their children with featureMember tag, in this example just one. Also, the configuration only contains one dataset, given in the Datasets dictionary. The key in this dictionary is the category number (52) for the dataset.

Next the DTC goes through the keys in the Tags dictionary of the transformation configuration, for each key in the Tag they are stored in a result dictionary. The keys in the result dictionary are taken from the datanames dictionary. The datanames dictionary is the connection from external source names to the OSCPSEP OGD names. If they are also in the combine dictionary, the separate external fields contents are combined into one internal content.

```
Tags dictionary:
Tags = {
   "OBJECTID": "",
   "pos": "",
   "BEZEICHNUN": "",
   "Baenke": ["id", "true"]}


Datanames dictionary:
datanames = {
   "id": "external_id",
   "BEZEICHNUN": "generatedName",
   "OBJECTID": "generatedName",
   "pos": "coordinates"}
```

With the OBJECTID tag, it finds the value “1” and finds that in the datanames dictionary the key OBJECTID, is generatedName, it is also part of an list in in combine and stored in a separate dictionary containing all combined fields.

Combine dictionary:

```
combine = {
   "generatedName": ["BEZEICHNUN", "OBJECTID"]}
```

When all the data have been collected, the fields in the combine dictionary are combined and puts both OBJECTID and BEZEICHNUN together and stores them in the result dictionary for the dataset with the key generatedName.

A special case is the “id” attribute in the “Baenke” tag,

```
<fme:Baenke gml:id="id5e409b8b-63f6-4af8-b343-405497588753">
```

“Baenke” in the Tags dictionary contains a list with the attribute name, the second value of “Baenke” in this dictionary indicate what key to use in the datanames dictionary. In this case, it is “id” and the associated name is “external_id”.

When that is done, the transformation goes through the format dictionary and transforms the values here. The format dictionary may contain a special key named “Meta”, this key refers to the meta data of the dataset, in this case two date formats about when the dataset was last updated and was released.

The other key is “coordinates”, and it tells the transformation module that the coordinates are given in the EPSG:31493 format, and then uses PyProj to transform this value.

When this is done, the data can be loaded into the OSCPSEP OGD. The data is then immediately available via the OSCPSEP OGD API.

To update already imported data we apply a get or create scheme scheme which is using the external record ID to compare to the already stored record external ID in the OSCPSEP OGD.

NOTE: PyProj does not support the 31493-coordinate format out of the box. The following line was added to the PyProj con.fig file to make this work.

```
<31493> +proj=tmerc +lat_0=0 +lon_0=9 +k=1.000000 +x_0=3500000 +y_0=0
+ellps=bessel +units=m no_def  <>
```