Technical Test Report 1

MARCH 2016

DELIVERABLE

Project Acronym: **SDI4Apps**
Grant Agreement number: **621129**
Project Full Title: **Uptake of Open Geographic Information Through Innovative Services Based on Linked Data**

D4.5.1 TECHNICAL TEST REPORT 1

Revision no. 06

Authors: D. Tarini, A.Iembo, N. Zanetti (Hyperborea)
S.R. Bergheim (AVINET)
M. Tuchyna (SAZP)
P. Vohnout (BOSC)
K. Charvat (CCSS)
O. Čerba and T. Mildorf (University of West Bohemia)

Project co-funded by the European Commission within the ICT Policy Support Programme		
Dissemination Level		
P	Public	X
C	Confidential, only for members of the consortium and the Commission Services	

REVISION HISTORY

Revision	Date	Author	Organisation	Description
01	04/11/2015	Daniele Tarini	Hyper	Initial TOC, methodology upgrade
02	28/12/2015	Daniele Tarini and Alfredo Iembo	Hyper	First draft of Section 1 report
03	18/01/2016	S.R. Bergheim, M. Tuchyna, P. Vohnout, O. Čerba	Avinet, Sazp, CCSS, UWB	Refinement Section 1 of Report
04	22/02/2016	D. Tarini, S.R. Bergheim, M. Tuchyna, P. Vohnout, O. Čerba	Hyper, Avinet, Sazp, CCSS, UWB	Contributions to tests of main modules
05	14/03/2016	D. Tarini, S.R. Bergheim, M. Tuchyna, P. Vohnout, O. Čerba	Hyper, Avinet, Sazp, CCSS, UWB	Refinement of main modules tests
06	31/03/2016	A.Iembo, D. Tarini, N. Zanetti	Hyper	Consolidation and delivery of the report

TABLE OF CONTENTS

Revision History 3
Table of Contents 4
List of Acronyms 6
List of Tables 7
Executive Summary 8
1 Introduction 9
2 TECHNICAL TEST METHODOLOGY UPDATE 10
2.1 Introduction 10
2.2 Platform Services under Test 10
3 MODULES TEST SCENARIOS 11
3.1 Map Module 11
3.1.1 Functional Test 11
3.1.2 Non-functional Test 13
3.2 Information retrieval module 14
3.2.1 Functional Test 14
3.2.2 Non-functional test 15
3.3 Advanced Visualization Module 16
3.3.1 Functional Test 16
3.3.2 Non-functional Test 16
3.4 Mobile Module 17
3.4.1 Functional Test 17
3.4.2 Non-functional Test 17
3.5 Analytical and Modelling Module 18
3.5.1 Functional Test 18
3.5.2 Non-functional Test 18
3.6 Data Harmonization Module 19
3.6.1 Functional Test 19
3.6.2 Non-functional Test 19
3.7 Multilingual Module 20
3.7.1 Functional Test 20
3.7.2 Non-functional Test 20
3.8 Semantic Tools for Linked Open Data Harmonization Module 21
3.8.1 Functional test 21
3.8.2 Non-functional Test 21
4 TECHNICAL TEST REPORT 22

4.1 Map Module 22
4.1.1 Functional Report 22
4.1.2 Non Functional / Performance Efficiency Report 24
4.2 Information Retrieval Module 27
4.2.1 Functional Report 27
4.2.2 Non-functional Report 28
4.3 Advanced Visualization Module 29
4.3.1 Functional Report 29
4.3.2 Non-functional Report 30
4.4 Mobile Module 32
4.4.1 Functional Report 32
4.4.2 Non-functional Report 32
4.5 Analytical and Modelling Module 34
4.5.1 Functional Report 34
4.5.2 Non-functional Report 35
4.6 Data Harmonization Module 35
4.6.1 Functional Report 35
4.6.2 Non-functional Report 35
4.7 Multilingual Module 36
4.7.1 UFT-8 36
4.7.2 Attribute xml:lang and RFC 5646 / BCP 47 Tags for the Identification of Languages 36
4.7.3 System Moses 37
4.7.4 Functional Report 37
4.7.5 Non-functional Report 38
4.8 Semantic tools for LOD data harmonization module 39
4.8.1 Functional report 43
4.8.2 Non-functional Report 43
5 Conclusion 44

LIST OF ACRONYMS

SaaS	Software as a Service
SDI	Spatial Data Infrastructure
SME	Small and Medium Enterprises
IaaS	Infrastructure as a Service
PaaS	Platform as a Service
IoT	Internet of the Things

LIST OF TABLES

Table 1: Map module functional test 13
Table 2: Map module non-functional test 14
Table 3: Information retrieval module functional test 15
Table 4: Information retrieval module non-functional test 15
Table 5: Advanced visualization module functional test 16
Table 6: Advanced visualization module non-functional test 17
Table 7: Mobile module functional test 17
Table 8: Mobile module non-functional test 18
Table 9: Analytical and modelling module functional test 18
Table 10: Analytical and modelling module non-functional test 19
Table 11: Data harmonization module functional test 19
Table 12: Data harmonization module non-functional test 20
Table 13: Multilingual module functional test 20
Table 14: Multilingual module non-functional test 20
Table 15: Semantic tools for LOD data harmonization module functional test 21
Table 16: Semantic tools for LOD data harmonization module non-functional test 21
Table 17: Map Module Functional Report - Data publishing tool Upload 22
Table 18: Map Module Functional Report - Data publishing tool Publish 22
Table 19: Map Module Functional Report - Data publishing tool Styler 23
Table 20: Map Module Functional Report - Search for a Dataset in the Catalogue 23
Table 21: Map Module Functional Report - Search for a composition 23
Table 22: Map Module Functional Report - Create a map composition 24
Table 23: Map Module Non Functional Report - Web Test for 5 users * 4 requests per user 25
Table 24: Map Module Non Functional Report - Web Test for 10 users * 4 requests per user 25
Table 25: Map Module Non Functional Report - Web Test for 15 users * 4 requests per user 25
Table 26: Map Module Non Functional Report - Web Test for 5 users * 4 requests per user 26
Table 27: Map Module Non Functional Report - Web Test for 10 users * 4 requests per user 26
Table 28: Map Module Non Functional Report - Web Test for 15 users * 4 requests per user 27
Table 29: Information Retrieval Module - Functional tests 28
Table 30: Information Retrieval Module - Non Functional tests 29
Table 31: Advanced Visualisation Module - Functional tests 30
Table 32: Advanced Visualisation Module - Non Functional tests 32
Table 33: Mobile Module - Functional tests 32
Table 34: Mobile Module - Non Functional tests 34
Table 35: Analytical and Modelling Module - Functional tests 35
Table 36: Analytical and Modelling Module - Non Functional tests 35
Table 37: Data Harmonisation Module - Functional tests 35
Table 38: Data Harmonisation Module - Non Functional tests 36
Table 39: Multilingual module functional test 38
Table 40: Multilingual module non functional test 38
Table 41: Multilingual module non functional test: prior de-cuplication 38
Table 42: Multilingual module non-functional test: After de-duplication 39
Table 43: Existing vocabularies implemented to SPOI data model 41
Table 44: Semantic tools module functional test 43
Table 45: Semantic tools module non-functional test 43

EXECUTIVE SUMMARY

This report D4.5.1 “Technical Test Report 1” provides an overview of tests and their results from the first phase of the project implementation.

The focus and methodology exploited for testing the extended functionality are similar to those adopted in WP3. According to the DoW and also to the comment provided by Reviewers (please refer to the first Review comments provided after the 1st Project Review Meeting), the report includes:

- testing and quality assurance of the integrated sub-systems and components towards the overall user-requirements;
- resolution of problems identified throughout the testing activities.

In particular, a scenario based test plan for every component was prepared by identifying both functional and non-functional tests. The CCSS Redmine system was used for management of the testing, bug tracking and the process of bug fixing.

Furthermore, it is worth pointing out that the Code Camps and Hackathons organised by the Consortium contributed to gather feedback from both internal and external developers and therefore to the consolidation of the overall testing process for the first release of the project technical report.

1 INTRODUCTION

According to the DoW, this document aims at providing "an overview of the tests performed and their results from the first phase of the project implementation", i.e. the first batch of technical tests run on the project platform. As such, it be mainly focused on testing the platform advanced services and provide the appropriate reports on the outcomes.

However, according to the comments provided by the Reviewers after the review technical meeting held in Luxembourg the 6th of May 2015, the current report also includes actions undertaken by project partners to complete the overall technical test methodology (D4.4) which has been already submitted at M12.

2 TECHNICAL TEST METHODOLOGY UPDATE

2.1 Introduction

D4.4 gave the guidelines about the test methodology as it was meant to be applied to the platform advanced services. Below, main technical test methodology aspects (derived by former work) are reported:

1. identification of functional and non-functional requirements to test
2. definition of the test matrix, i.e. a list of tests addressing the requirements, where each test states:
    a) the success condition addressing one of the requirements
    b) the indicator(s) that can be measured to verify the condition
    c) the metric of the indicator
    d) the thresholds of success to be used for each metric

In particular, following the recommendations provided by reviewers during the first review meeting, this chapter included "the validation methodology still with clear indication of the testing data and criteria, with metrics for quantitative assessment".

By quoting the DoW "A scenario based test plan for every component will be prepared. The CCSS Redmine system is used for management of the testing, bug tracking and the process of bug fixing", testing data and criteria for each module will be described as scenarios.

2.2 Platform Services under Test

Documents DoW, D3.3.1, D4.2 and D4.4 define the advanced services (module, components) that the project's platform should provide. Here is a list as a summary of all of them:

- Map module
- Information retrieval module
- Advanced visualization module
- Mobile module
- Analytical and modelling module
- Data harmonization module
- Multilingual module
- Semantic tools for LOD data harmonization module

3 MODULES TEST SCENARIOS

This section details what should be tested for each item under test defined in the previous section. Each of the following section defines all the parameters needed to set up the tests.

3.1 Map Module

Scenario:

User's needs: representation of system objects on a map. This action is performed through a map visualisation module able to show different layers and backgrounds of the identified objects which are of interest for the user. The module would also allow to represent the different layer objects along with the visualisation of their related geometries.

From the non-functional perspective, the module would be able to respond in a proper short time to ensure an acceptable user experience (according to the identified Metric & Scale as reported in the tests table of the module below) and be easily integrated into a third party application.

3.1.1 Functional Test

Success Condition	Indicators	Metric and Scale
Invoking the API to insert the map within the HTML page results in the HTML page filled with the map	Whether the map is present in the HTML DOM	If the map is present the test is successful.
Geographical objects visualized in the map are correctly geo-located	Correspondence between CRS and geographical item	If the marker matches with Point of Interest the test is successful
The geometry of the geographical objects is properly defined.	Correspondence between different kind of geometry (line, point and polygon) and geographical representation	If the geometry definition correspond to geographical representation test is successful
List of map “compositions” are loaded from metadata system.	Appearance and behaviour of map and “Compositions” panel on http://ng.hslayers.org/examples/compositions/	List of map “compositions” are loaded from metadata system.
Composition list is filtered by part of title, description or current map extent.	Appearance and behaviour of map and “Compositions” panel on http://ng.hslayers.org/examples/compositions/	Composition list is filtered by part of title, description or current map extent.
List of features are loaded from SPARQL endpoint and displayed on the map.	Appearance of map on	List of features are loaded from SPARQL endpoint and displayed on the map.
“Zoom to” layer extent is provided for vector layers.	Appearance and behaviour of map on	“Zoom to” layer extent is provided for vector layers.
“Zoom to” layer extent is provided for WMS raster layers.	Appearance and behaviour of map on	“Zoom to” layer extent is provided for WMS raster layers.

List of layers defined in the map module configuration are visible in the "layer manager" panel.	Appearance of "Layer manager" panel on	List of layers defined in the map module configuration are visible in the "layer manager" panel.
Feature attributes are listed in "Info" panel after clicking the feature.	Appearance of "Infol" panel on	Feature attributes are listed in "Info" panel after clicking the feature.
Feature attributes can be edited and are stored in triple store.	Data in tripple store http://app.hslayers.org/spoi_admin/	Feature attributes can be edited and are stored in triple store.
It is possible to change the style of vector layer features (icon, colour) when "styles" module is enabled.	Appearance of "Layer manager", "Style" panels and map on	It is possible to change the style of vector layer features (icon, colour) when "styles" module is enabled.
Mapping module functions are provided to container application through javascript API.	Output in the web browsers console on any website which uses mapping module.	If a global JavaScript variable "hslayers_api" is present which can be checked by typing "hslayers_api" in browsers console on the website and if the output contains list of functions grouped by modules, then the test if successful.
It is possible for a user to add custom Web Map Services through GUI panel.	Appearance and behaviour of "Add external data" panel http://ng.hslayers.org/examples/datasources/?hs_panel=ows	For the test to be successful, the following steps have to execute correctly 1) User clicks on + icon and "Add external data" panel appears 2) User chooses data format "WMS" 3) User enters http://gis.lesprojekt.cz/wms/transport/traffic_volumes?request=GetCapabilities&service=WMS&version=1.3.0 in URL textbox and clicks Link/Connect button, a form which contains GetCapabilities response is displayed 4) User ticks some layer in the generated list at the bottom of "Add external data" panel and clicks "+" button 5) Layers which displays traffic intensity on the roads of Liberec region is displayed on the map
Legend of WMS layer is shown in "Layer manager" panel.	Appearance and behaviour of "Layer manager" panel http://ng.hslayers.org/examples/datasources/	Legend of WMS layer is shown in "Layer manager" panel.

It is possible to add vector layers by drag&drop functionality on the map (klv, gpx, geojson files must be supported).	Appearance and behaviour of map. http://ng.hslayers.org/examples/datasources/	It is possible to add vector layers by drag&drop functionality on the map (klv, gpx, geojson files must be supported).
Measurement tool must provide measurement of polygon area or line distance by taking into account current map projection.	Appearance and behaviour of Measurement panel. http://ng.hslayers.org/examples/datasources/?hs_panel=measure	Measurement tool must provide measurement of polygon area or line distance by taking into account current map projection.
Changes in the map (centre position, zoom, visible layers) must be stored described in URL parameters, if "Permalink" module is loaded.	URL of any web app which uses mapping module. http://ng.hslayers.org/examples/datasources/	Changes in the map (centre position, zoom, visible layers) must be stored described in URL parameters, if "Permalink" module is loaded.
"Share map" tool should generate embeddable iframe code with the current map configuration.	Appearance and behaviour of "Share map" panel. http://ng.hslayers.org/examples/datasources/?hs_panel=permalink	"Share map" tool should generate embeddable iframe code with the current map configuration.
User has to be able to search for a place by its name. Results of the search has to be displayed in a list and on the map. After clicking on a result map has to be centred on the place coordinates and resolution has to be set depending on the size and category of the place.	Display and behaviour of "Seartch" field on any map module enabled site. http://ng.hslayers.org/examples/datasources/	User has to be able to search for a place by its name. Results of the search has to be displayed in a list and on the map. After clicking on a result map has to be centred on the place coordinates and resolution has to be set depending on the size and category of the place.

Table 1: Map module functional test

3.1.2 Non-functional Test

Success Condition	Indicators	Metric and Scale
The map is loaded in a time acceptable for user experience (responsiveness)	The time T needed to load the map and insert it into HTML page by an automated test procedure	if T < 1 sec test is successful
It's easy for the developer to use the module and integrate it into his application (usability)	A user survey set of questions addresses the issue	The average score in the user survey is at least 4 (on a 1-5 scale)
The map module doesn't conflict with the js modules already present in the app (doesnt use global variables)		

The "look & feel" of the map module is configurable to the design of container application.		
Map module has to support responsive design principles and have a different layout on mobile and other small screen devices.		
Map module GUI has to support localization.		
SPARQL queries generated by map module have to run in an acceptable time frame.	The time it takes to load features in geographical bounding box.	The features from SPARQL endpoint have to be loaded in less than 3 seconds in a 20x20km geographical bounding box.

Table 2: Map module non-functional test

3.2 Information retrieval module

Scenario:

User needs: use and publish the data, therefore query the system to search and retrieve data of interest according to both alphanumeric and spatial query/search criteria. Search functionalities would allow to filter the query according to attributes, metadata, geometries of interest. The module would also allow to perform full-text searches.

User's needs: representation of system objects on a map. This action is performed through a map visualisation module able to show different layers and backgrounds of the identified objects which are of interest for the user. The module would also allow to represent the different layer objects along with the visualisation of their related geometries.

From the non-functional perspective, the module would be able to respond in a proper short time to ensure an acceptable user experience (according to the identified Metric & Scale as reported in the tests table of the module below) and be easily integrated into a third party application.

3.2.1 Functional Test

Success Condition	Indicators	Metric and Scale
Information retrieval module finds the terms specified in the query in one or more textual or digital objects	terms are identified in textual objects	if the terms specified in the request are underlined in the text, the test is successful
Information retrieval module identifies geographical objects corresponding to query parameters	geographical objects are represented on the map	if the parameters specified in the query correspond to the represented geographical objects, the test is successful

Object search by attributes	Object are retrieved according to specified filters	Only objects having attribute matching the filter value are retrieved
Object search by bounding box	Object are retrieved according to specified filters	All retrieved objects have geometry in the bounding box value
Metadata retrieval	Object classification (categories) is retrieved	All available categories are returned
Object search by metadata (categories)	Object are retrieved according to specified filters	Only object related to specified categories are retrieved
Full text search	Retrieve object containing a string in any attribute	Only object having attributes whose value contains the string are retrieved
Retrieve objects	Fetch from platform objects given the IDs	Only objects with the provided IDs are fetched and returned
Helper class	Helper offers same API as REST API	Helper overhead in response time is less than 20% compared to direct API call
Pagination	Module offers pagination	Default pagination in implemented but user can set its own value for page size

Table 3: Information retrieval module functional test

3.2.2 Non-functional test

Success Condition	Indicators	Metric and Scale
The information resulting from the query are loaded in a time acceptable for user experience (responsiveness)	The time T needed to show the query results and insert them in the HTML/DOM page by an automated test procedure	if T < 1 set, test is successful
It must be easy for developers to adopt and use the functions in their own applications. This condition is met if documentation, coding practises and commenting are systematically adhered to.	Availability of standard documentation Availability of non-minified, human readable version of JavaScript Availability of in-line code comments Adherence to good practise in JavaScript design patterns	1 if present/compliant, 0 if missing or not working. Max score 4/4

Table 4: Information retrieval module non-functional test

3.3 Advanced Visualization Module

Functional requirements: The advanced visualization module must provide utility methods to quickly and efficiently create chart and map based visualizations of statistical data. It is mainly target at numerical multi-dimensional quantitative data but may also be used to visualize single ordinal dimensions. All common graph types/styles must be easy to add to visualize, i.e. pies, bars, lines, scatter, ordinal map etc.

Non-functional requirements: The module must be responsive with large data sets. Since it it implemented as a client-side technology it needs to support a wide array of browsers. Initial drawing speed and 'refresh' of graphs need to be very fast. The API must be well documented so that it is easy to use for developers who would like to integrate it with their applications.

3.3.1 Functional Test

Success Condition	Indicators	Metric and Scale
Must support all chart types in specification	Can draw pie chart Can draw bar chart Can draw scatter chart Can draw line chart	1 if present and working, 0 if missing. Max score 4/4
Must support all map types in specification	Can draw symbol map Can draw live data map Can draw choropleth map Can draw prism map Can draw heat map Can draw bubble-pie map	1 if present and working, 0 if missing. Max score 6/6
Must support coordinated views	Can add multiple visualizations to a single coordinated view Can filter one graphic and reflect the modified data in other coordinated views	1 if present and working, 0 if missing. Max score 1

Table 5: Advanced visualization module functional test

3.3.2 Non-functional Test

Success Condition	Indicators	Metric and Scale
It must be easy for developers to adopt and use the functions in their own applications. This condition is met if documentation, coding practises and commenting are systematically adhered to.	Availability of standard documentation Availability of non-minified, human readable version of JavaScript Availability of in-line code comments Adherence to good practise in JavaScript design patterns	1 if present/compliant, 0 if missing or not working. Max score 4/4
Visualizations must be able to handle a large number of data records, i.e. > 5 000	Ability to handle 100 records Ability to handle 1 000 records Ability to handle 5 000 records	'Ability to handle' is defined as loading dataset from JSON file, parsing it and making

records for processing on the client on any device.	Ability to handle 10 000 records Ability to handle 15 000 records	two visualizations added to a coordinated view
Visualizations must draw and update with an acceptable response time and speed, i.e < 250 ms	Drawing speed Update speed	The indicator will be measured in milliseconds. Since the operations can be run on 'any' client device, a representative execution environment for the tests cannot easily be achieved. However, this removes scalability from the equation since each operation is run synchronously on its own CPU.

Table 6: Advanced visualization module non-functional test

3.4 Mobile Module

Functional requirements: The mobile module must provide a mechanism that permits use of maps in apps in offline mode. The module must make it possible to download a feature dataset, edit it while offline and then check-in the edited data again. The module must give access to device sensors such as camera and GPS.

Non-functional requirements: The module must be interoperable with the Cordova framework for cross-browser hybrid mobile applications. Any map functionality must be extensions to the OpenLayers 3 mapping framework. It must be easy to integrate the library with other code; i.e. it must be well-documented and well structured.

3.4.1 Functional Test

Success Condition	Indicators	Metric and Scale
Must enable offline background maps	Functional offline maps web service	1 if is present and works, 0 if not implemented or not working. Max score 1.
Must enable retrieval of GeoJSON data for offline use/editing	Functional synchronization web service	1 if is present and works, 0 if not implemented or not working. Max score 1.

Table 7: Mobile module functional test

3.4.2 Non-functional Test

Success Condition	Indicators	Metric and Scale
It must be easy for developers to adopt and use the functions in their own applications. This condition is met if documentation, coding practises and commenting are systematically adhered to.	Availability of standard documentation Availability of non-minified, human readable version of JavaScript Availability of in-line code comments Adherence to good practise in JavaScript design patterns	1 if present/compliant, 0 if missing or not working. Max score 4/4

Service must scale with size of area/volume of data.	Capacity to extract 1 km^2 Capacity to extract 2 km^2 Capacity to extract 5 km^2 Capacity to extract 10 km^2 Capacity to extract 25 km^2	Time in milliseconds + uncompressed size of extract in Kilobytes
Check in of data must be quick, i.e. less than 5 seconds	Capacity to check-in 1 km^2 Capacity to check-in 5 km^2	Time in milliseconds + uncompressed size of data transferred from client to server in Kilobytes
Background maps must draw with minimal lag, i.e. < 200ms per tile	Draw speed per tile	Milliseconds per tile

Table 8: Mobile module non-functional test

3.5 Analytical and Modelling Module

Scenario:

User needs: data analysis according to predefined models for analysis and based on both spatial and alphanumeric attributes. Following main scenarios have been identified on transportation, land use, cadastral data and community services:

- identification of the part of territory which is of interest for the analysis to be performed
- collection and aggregation of data for the area of interest
- analysis model application
- elaborated data delivery for their representation on the client side

From non-functional perspective, the module would be able to asynchronously perform the elaborations and allow promptly pass the elaborated results to the client for the final delivery. In fact, analysis models might require long processing times that are not compatible with synchronous application mechanisms.

Finally the module would be easily integrated into a third party application.

3.5.1 Functional Test

Success Condition	Indicators	Metric and Scale
The analytics and modelling module must elaborate data request managed by SDI4APPS data repository and model	Point of interest are represented from a geographical point of view within a boundary defined by the request	If the results of the analysis correspond to a well defined geographical scale and data are analysed properly corresponding to the model structure, the test is successful

Table 9: Analytical and modelling module functional test

3.5.2 Non-functional Test

Success Condition	Indicators	Metric and Scale

It's easy for the developer to use the module and integrate it into his application (usability)	A user survey set of questions addresses the issue	The average score in the user survey is at least 4 (on a 1-5 scale)
The module offers a wide range of status reporting during the polling of the response (reliability)	An automated procedure set up to result in an error (forced crash or out of bounds analysis) is run on the module	If the polling results in an appropriate error are reported to the pooler, the test is successful

Table 10: Analytical and modelling module non-functional test

3.6 Data Harmonization Module

Scenario:

User needs: gather hetherogenous data (e.g. different formats) from different data sources (namely: CSV, shapefile, WFS, etc.) to be harmonised and integrated in the platform of interest

Following main scenarios have been identified on Land-Use and transportation data even through the very proactive cooperation with OpenTransportNet project:

- specifications gathering of accessible data
- specifications gathering of data formats and semantics
- harmonisation steps definition
- harmonisation execution
- harmonized data publication

From non-functional perspective, the module would be sufficiently flexible to allow the implementation of harmonization steps with data retrieved from different sources, in different formats, etc.

3.6.1 Functional Test

Success Condition	Indicators	Metric and Scale
The module allow to design a transformation	A data harmonization process is designed.	The resulting designed harmonization process is able to produce harmonized dataset coherent with requirements.
The module allow to transform spatial data	The data are harmonized according designed process.	The datasets are properly transformed.
The module allows to discard syntactically incorrect data	If a data-set contains not valid information (such as topology – e.g. polygons with overlap), the module reports the error.	The transformed data-set does not include syntactically incorrect data

Table 11: Data harmonization module functional test

3.6.2 Non-functional Test

Success Condition	Indicators	Metric and Scale

The harmonization module allows the acquisition of data from standard formats	The module allows the acquisition and transformation of data from standard services and GIS formats	At least standard Web Services OGC WMS and WFS, shapefile are supported
The module manages both source and target data types	The module allows the transformation of data types from the source to target	At least Integer, float, data, OGC_GEOMETRY string are supported
Harmonised data sets are available	Number of harmonised data sets	At least one data set and a transformation are available
The module allows to harmonise different data-sets into the resulting final one	Number of source data-sets to be harmonised	At least two datasets to be harmonised

Table 12: Data harmonization module non-functional test

3.7 Multilingual Module

Scenario:

User needs: platform data multilinguality.

Typical module usage scenarios are based on data collected from EUR lex, for Czech and English (ref. D.4.1.1) and provide the translation of not only single terms but both phrases and text.

From the non-functional perspective, the module would support multiple languages and at least English.

3.7.1 Functional Test

Success Condition	Indicators	Metric and Scale
The module is able to translate terms	The term is translated in the target language, if available	The translation is correct
The module is able to translate both sentences and text.	Test is translated in the target language, if available	The translation is correct with respect to the semantics of the sentence and geographical names at least for catchphrases 1000

Table 13: Multilingual module functional test

3.7.2 Non-functional Test

Success Condition	Indicators	Metric and Scale
The modules supports different languages	Main languages are supported	At least English language is supported
The module allows the translation of different terms	Number of available terms	At least 10M terms are available

Table 14: Multilingual module non-functional test

3.8 Semantic Tools for Linked Open Data Harmonization Module

Scenario:

User needs: LOD data available in the platform.

Main identified scenarios of the module usage are based on SmartPOI (SPOI):

- Data acquisition
- semantics annotation and link to other data-sets (LOD production)
- Publication of LOD data sets

Non functional requirements are the number of used ontologies

3.8.1 Functional test

Success Condition	Indicators	Metric and Scale
Source Data set is transformed in Open Data format	The data set is published in Open Data format	The final data set is available in 5-stars form
The data of each dataset are classified according to a standard classification system (e.g. ontologies)	Number of categories	At least 5 categories per data set

Table 15: Semantic tools for LOD data harmonization module functional test

3.8.2 Non-functional Test

Success Condition	Indicators	Metric and Scale
Standard vocabularies (e.g. ontologies) are used	Number of standard vocabularies	At least 3 standard vocabularies are used (for identification, properties, geometries, etc.)
Available LOD data sets	Number of available SPARQL endpoints or data dumps	At least one SPARQL endpoint

Table 16: Semantic tools for LOD data harmonization module non-functional test

4 TECHNICAL TEST REPORT

This section provides the results of testing as defined in the previous chapter. Each of the following sections refers to a section in the previous chapter, where test details are provided.

4.1 Map Module

The module was tested and reported in form of test-cases, with success/fail conditions, along with a section dedicated to quantitative results.

4.1.1 Functional Report

This test refers to Data Publishing tool - Upload Scenario

Success Condition	Indicators	Metric and Scale
Uploading a file containing geodata, likewise uploading of zipped shapefile Prerequisites: User is logged in and Data Publishing tool is open	The file is unzipped and shown in the Files panel Error message is shown	Data was successfully uploaded

Table 17: Map Module Functional Report - Data publishing tool Upload

This test refers to Data Publishing tool - Publish Scenario.

Success Condition	Indicators	Metric and Scale
Publish uploaded geodata as a map layer through different steps: -Select the uploaded file and choose 'Publish' -Fill in the publication form -The 'Publish as' field must have the default "As new" value. -Click 'Publish' Prerequisites are: User is logged-in, Data publishing tool is open, Some geodata is successfully uploaded	1. The system processes the request and finally new data appears in the Data panel and new layer appears in the Layers panel. 2. The data in Data panel and the layer in the Layers panel is shown in the group it has been published to. Error message is shown	Data was successfully published

Table 18: Map Module Functional Report - Data publishing tool Publish

This test refers to Data Publishing tool - Styler Scenario.

Success Condition	Indicators	Metric and Scale

Open the published layer in Styler, through different steps: - Select the published layer - Click 'Styler' in the layer menu. Prerequisites are: User is logged-in, Data publishing tool is open, Some geodata is successfully uploaded and published.	1. The Styler opens. 2. The layer can be selected in the Layers panel. 3. Select the layer and the layer is shown. Error message is shown	Layer was successfully open in the Styler and styled was allowed.

Table 19: Map Module Functional Report - Data publishing tool Styler

This test refers to Search for a dataset in the Catalogue Scenario.

Success Condition	Indicators	Metric and Scale
Find required dataset, through: - Fill form with search terms - Click on search - Wait for result - Examine results Prerequisites are: Catalogue client application is opened	Search result contains searched dataset Search doesn't work	Dataset was successfully found using search

Table 20: Map Module Functional Report - Search for a Dataset in the Catalogue

This test refers to Search for a composition in the Map Compositions Directory Scenario.

Success Condition	Indicators	Metric and Scale
Open desired composition, through: - Zoom in to desired location - Filter compositions by keywords - Select desired composition Prerequisites are: Thematic Map Viewer app opened	Data from composition is displayed Composition can't be found	Desired composition was successfully displayed

Table 21: Map Module Functional Report - Search for a composition

This test refers to Create a map composition – Add layer to map Scenario.

Success Condition	Indicators	Metric and Scale
Add published layer to map, through: - Select the published layer. Click 'Add to map'. - Click on save composition button - Fill information about composition - Save composition on server Prerequisites are: User is logged-in, Data publishing tool is open, Some geodata is successfully uploaded and published. It may be styled.	Composition is saved on server Error message is shown	Composition was successfully stored

Table 22: Map Module Functional Report - Create a map composition

4.1.2 Non Functional / Performance Efficiency Report

Tests were done using jmeter.

Layman

The conducted Web Test covers the GET services of the Layman component.

The base url of the Service endpoints (first column) presented in all the tables below is

http://portal.sdi4apps.eu/cgi-bin/layman/layman/. The tested services provide the required functionality of the Layman component, namely: getting list of data (/fileman), list of tables (/data), list of published layers (/layed).

Indicator	Measurement							Status
Get services of the Layman component	Service	Samples	Average (msec)	Min (msec)	MAx (msec)	Throughput		Error
						Requests/sec	KB/sec	
	/fileman	20	595	388	723	6.6	3,23	0
	/data	20	672	490	937	5.9	9.36	0
	/layed	20	1312	824	1462	3.2	61.61	0

Table 23: Map Module Non Functional Report - Web Test for 5 users * 4 requests per user

Indicator	Measurement							Status
Get services of the Layman component	Service	Samples	Average (msec)	Min (msec)	MAx (msec)	Throughput		Error
						Requests/sec	KB/sec	
	/filemanager	40	530	370	799	13.9	6.79	0
	/data	40	641	510	1130	12.5	19.96	0
	/layed	40	1273	770	1801	6.5	126.15	0

Table 24: Map Module Non Functional Report - Web Test for 10 users * 4 requests per user

Indicator	Measurement							Status
Get services of the Layman component	Service	Samples	Average (msec)	Min (msec)	MAx (msec)	Throughput		Error
						Requests/sec	KB/sec	
	/filemanager	60	546	379	1040	19.1	9.32	0
	/data	60	770	560	1220	15.9	25.39	0
	/layed	60	1509	901	2129	7.6	146.22	0

Table 25: Map Module Non Functional Report - Web Test for 15 users * 4 requests per user

Web Test 3: MIcKA

The conducted Web Test covers the GET services of the MIcKA component. The base url of the Service endpoints (first column) presented in all the tables below is http://portal.sdi4apps.eu/php/catalogue/libs/cswclient/cswClientRun.php. The tested services provide namely: getting list of map compositions, list of datasets.

Indicator	Measurement							Status
Get services of the Micka component	Service	Samples	Average (msec)	Min (msec)	MAx (msec)	Throughput		Error
						Request s/sec	KB/sec	
	mapCompositions	20	82	70	136	18.1	277.93	0
	datasets	20	341	298	426	9.1	277.95	0

Table 26: Map Module Non Functional Report - Web Test for 5 users * 4 requests per user

Indicator	Measurement							Status
Get services of the Micka component	Service	Samples	Average (msec)	Min (msec)	MAx (msec)	Throughput		Error
						Request s/sec	KB/sec	
	mapCompositions	40	105	70	210	32.2	494.55	0
	datasets	40	406	292	1510	16	486.15	0

Table 27: Map Module Non Functional Report - Web Test for 10 users * 4 requests per user

Indicator	Measurement							Status
Get services of the Micka component	Service	Samples	Average (msec)	Min (msec)	MAx (msec)	Throughput		Error
						Request s/sec	KB/sec	
	mapCompositions	60	152	72	344	37.6	576.92	0
	datasets	60	469	296	1193	19.3	587.47	0

Table 28: Map Module Non Functional Report - Web Test for 15 users * 4 requests per user

4.2 Information Retrieval Module

The module was tested exploiting the data and component available to:

http://sdi4apps.hyperborea.com for community services and POI in the Florence area.

5000 objects in 40 dataset (40 categories), geometries in WGS84.

Testing by an HTML form, allowing calls to REST API and corresponding methods in the javascript helper library.

Check of functional success/failure (e.g. match between filters and result) has been carried out comparing results with data in DB.

4.2.1 Functional Report

Success Condition	Indicators	Metric and Scale
Terms are identified in textual objects	Searching for content containing 50 random geographical names in the region covered by the content. Names extracted from content. Each search yielded results. All resources contained complete or partial matches for the query terms.	Passed
Geographical objects are represented on the map	Qualitative search for five known places within subject matter coverage of content: Lucca, Pisa, Bologna, Florence, Genova. All objects queried provided geographical locations in WGS84 lat/long format. When	Passed

	overlaid on a map, the objects appeared at the expected location.	
Object search by bounding box	All retrieved objects have geometry in the bounding box value, no failure (100 test)	Passed
Object search by point and radius (buffer)	All retrieved objects have geometry in the “radius” buffer over point, no failure (100 test)	Passed
Object classification (categories) is retrieved	All available categories are returned, no failure (100 test)	Passed
Object search by category	Only object related to specified categories are retrieved, no failure (100 test)	Passed
Object full text search	Only object having attributes whose value contains the string are retrieved, no failure (100 test)	Passed

Table 29: Information Retrieval Module - Functional tests

4.2.2 Non-functional Report

Indicator	Measurement	Status
The time T needed to show the query results and insert them in the HTML/DOM page by an automated test procedure	Tested 5 simultaneous threads for 60 seconds with 127 requests on each thread issued at random intervals. Average response time under single requests, ~740 ms per request Average response time under stress ~1 seconds per request	Passed
Availability of standard documentation	Documentation is present	Passed
Availability of non-minified, human readable version of code	Human-readable code available	Passed
Availability of in-line code comments	Code comments used systematically	Passed
Adherence to good practise in design patterns	Design patterns adhered to	Passed
Time needed to retrieve categories (REST API and Helper)	40 categories retrieved: average 690 ms (REST API) 705 (Helper) overhead 2%. num. of test 100.	Passed

Time needed to retrieve object by categories (REST API and Helper)	70 object found for a category, 10 returned (pagination at 10): average 890 ms (REST API) 950 (Helper) overhead 8%. num. of test 100.	Passed
Time needed to retrieve objects from a full-text search (REST API and Helper)	23 object found for a full text search over “Michelangelo”, 10 returned (pagination at 10): average 900 ms (REST API) 964 (Helper) overhead 7%. num. of test 100.	Passed
Time needed to retrieve objects from a bounding box search (REST API and Helper)	120 object found for a geographical search with boundingBox=11.2 43.7 11.4 43.8, 10 returned (pagination at 10): average 939 ms (REST API) 992 (Helper) overhead 6%. num. of test 100.	Passed
Time needed to retrieve objects from a point and radius (buffer)	96 object found for a geographical search with POINT(11.3 43.75) and radius=5000, 10 returned (pagination at 10): average 851 ms (REST API) 867 (Helper) overhead 2%. num. of test 100.	Passed

Table 30: Information Retrieval Module - Non Functional tests

4.3 Advanced Visualization Module

The module was tested using data extracts from Smart Points of Interest data (http://sdi4apps.eu/spoi/).

4.3.1 Functional Report

Indicator	Measurement	Status
Can draw pie chart	Functionality present and working: 1	Score 4/4

Can draw bar chart	Functionality present and working: 1	Passed
Can draw scatter chart	Functionality present and working: 1	
Can draw line chart	Functionality present and working: 1	
Can draw symbol map	Functionality present and working: 1	Score 5/6
Can draw live data map	Functionality present and working: 1	Passed
Can draw choropleth map	Functionality present and working: 1	Note: Missing map type (prism map) planned for second release
Can draw prism map	Functionality not yet implemented: 0	
Can draw heat map	Functionality present and working: 1	
Can draw bubble-pie map	Functionality present and working: 1	
Can add multiple visualizations to a single coordinated view	Functionality present and working: 1	Score 2/2
Can filter one graphic and reflect the modified data in other coordinated views	Functionality present and working: 1	Passed

Table 31: Advanced Visualisation Module - Functional tests

4.3.2 Non-functional Report

Indicator	Measurement	Status
Availability of standard documentation	1: JSDoc reference documentation is available	Score 4/4
Availability of non-minified, human readable version of JavaScript	1: A non-minified development/debugging version of the library is available	Passed
Availability of in-line code comments	1: Code comments exist for all public modules, methods and properties and are systematically applied in compliance with the JSDoc commenting standard	

Adherence to good practise in JavaScript design patterns	1: The code is structured into JavaScript modules. Only a single object is introduced into the global namespace: s4a. Scope is isolated for all modules Strict mode is used for scripts. Code validates using JSHint/Lint	
Ability to handle 100 records	Simulation of user behavior using JavaScript test method 100 iterations Average time: ~1 ms Min: < 1 ms Max: 3 ms	Passed
Ability to handle 1 000 records	Simulation of user behavior using JavaScript test method 100 iterations Average time: 2.86 ms Min: 2 ms Max 4 ms	Passed
Ability to handle 5 000 records	Simulation of user behavior using JavaScript test method 100 iterations Average time: 12.95 ms Min: 11 ms Max 25 ms	Passed
Ability to handle 10 000 records	Simulation of user behavior using JavaScript test method 100 iterations Average time: 25.43 ms Min: 23 ms Max: 41 ms	Passed
Ability to handle 100 000 records	Simulation of user behavior of user behavior using JavaScript test method 100 iterations Average time: 303.11 ms Min: 281 ms Max: 371 ms	Passed

Maps/charts drawing speed	Measured average page-load time (including network latency) 100 iterations Average speed: ~542 ms Max: ~783 ms Min: ~308 ms	Passed
Maps/charts update speed	Measured reported update time (reported by API, excluding network latency) 100 iterations Average speed: ~247 ms Max: ~311 ms Min: ~223 ms	Passed

Table 32: Advanced Visualisation Module - Non Functional tests

4.4 Mobile Module

The module was tested using data extracts from Open Street Map and tile data prepared using TileMill

4.4.1 Functional Report

Indicator	Measurement	Status
Functional offline maps web service	Tested web service and found the following methods to be working as specified: – Upload – Download – List – Query	Passed
Functional synchronization web service	Tested web service and found the following methods to be working as specified: – CheckOut – CheckIn – GetConflicts – Resolve	Passed

Table 33: Mobile Module - Functional tests

4.4.2 Non-functional Report

Indicator	Measurement	Status

Availability of standard documentation	1: JSDoc reference documentation is available	Score 4/4
Availability of non-minified, human readable version of JavaScript	1: A non-minified development/debugging version of the library is available	Passed
Availability of in-line code comments	1: Code comments exist for all public modules, methods and properties and are systematically applied in compliance with the JSDoc commenting standard	
Adherence to good practise in JavaScript design patterns	1: The code is structured into JavaScript modules. Only a single object is introduced into the global namespace: s4a. Scope is isolated for all modules Strict mode is used for scripts. Code validates using JSHint/Lint	
Capacity to extract 1 km2	Simulation of user behavior using JavaScript test method 100 iterations Average time: 54 ms Median: 48 ms Min: 2 ms Max: 816 ms	Passed
Capacity to extract 5 km2	Simulation of user behavior using JavaScript test method 50 iterations Average time: 268 ms Median: 35 ms Min: 2 ms Max: 5 746 ms	Passed
Capacity to extract 10 km2	Simulation of user behavior using JavaScript test method 25 iterations Average time: 800 ms Median: 337 ms Min: 28 ms Max: 12 600 ms	Passed Note! A limit of area size should be imposed as the performance does not scale linearly.
Capacity to check-in 1 km2	Simulation of user behavior using JavaScript test method	Passed

	100 iterations Average time: 66 ms Median: 62 ms Min: 32 ms Max: 782 ms	
Capacity to check-in 5 km2	Simulation of user behavior using JavaScript test method 50 iterations Average time: 295 ms Median: 326 ms Min: 30 ms Max: 5 779 ms	Passed
Capacity to check-in 10 km2	Simulation of user behavior using JavaScript test method 25 iterations Average time: 878 ms Median: 412 ms Min: 28 ms Max: 14 133 ms	Passed Note! A limit of area size should be imposed as the performance does not scale linearly.
Draw speed per tile	Simulation of user behavior using JavaScript test method. Loading 1 000 random tiles from database covering 10 x 10 kilometers. Average time: 220 ms Median: 216 ms Min: 180 ms Max: 439 ms	Passed Note! Draw speed for tiles read from SQLite databases is slower than reading image files directly. There is a notable lag compared to a file data source. It is recommended to experiment with caching of image data on the device for the second release.

Table 34: Mobile Module - Non Functional tests

4.5 Analytical and Modelling Module

Main usage scenarios of the module are based on transportation, land use, cadastral data and community services.

4.5.1 Functional Report

Indicator	Measurement	Status
The analytics and modelling module must elaborate data request managed by SDI4APPS data repository and model	Point of interest are represented from a geographical point of view within a boundary defined by the request	Passed

Table 35: Analytical and Modelling Module - Functional tests

4.5.2 Non-functional Report

Indicator	Measurement	Status
It's easy for the developer to use the module and integrate it into his application (usability)	A user survey set of questions addresses the issue	4/5: Passed
The module offers a wide range of status reporting during the polling of the response (reliability)	An automated procedure set up to result in an error (forced crash or out of bounds analysis) is run on the module	Passed

Table 36: Analytical and Modelling Module - Non Functional tests

4.6 Data Harmonization Module

USED DATA:

http://sdi4apps.eu/open_land_use/

http://opentransportmap.info/

http://sdi4apps.eu/spoi/

4.6.1 Functional Report

Indicator	Measurement	Status
Correctly transformed dataset	All harmonized dataset where produced with all input data	Passed
Correctly transformed geometries	All geometries where correctly transformed	Passed
Reject invalid data	All invalid geometries where rejected during the transformation	Passed

Table 37: Data Harmonisation Module - Functional tests

4.6.2 Non-functional Report

Indicator	Measurement	Status
Supported geodata formats	Module supports WMS, WFS, shapefile formats	Passed
Supported types	String, integer, float, date, OGC geometry type are supported	Passed
Harmonized dataset or transformations available	Test dataset are all available (2 dataset, 1 transformation)	Passed

Number of input dataset concurring to one harmonized dataset	4	4 - Passed

Table 38: Data Harmonisation Module - Non Functional tests

4.7 Multilingual Module

One of the elemental question during data harmonization and semantic description is related to using more various languages. In the pilot applications of SDI4apps (mainly in Open Smart Tourist Data) is multiliguality taken in consideration with use three technologies or standards:

1. UTF-8 (UCS Transformation Format with 8-bit code units)
2. Attribute xml:lang and RFC 5646 / BCP 47 Tags for the Identification of Languages
3. System Moses

4.7.1 UFT-8

UTF-8 is a character encoding capable of encoding all possible characters in Unicode (a computing industry standard for the consistent encoding, representation, and handling of text expressed in most of the world's writing systems). The encoding is variable-length and uses 8-bit code units. It was designed for backward compatibility with ASCII, and to avoid the complications of endianness and byte order marks in the alternative UTF-16 and UTF-32 encodings. The name is derived from: Universal Coded Character Set + Transformation Format—8-bit.

UTF-8 is the dominant character encoding for the World Wide Web, accounting for 86.1% of all Web pages in January 2016. The W3C recommends UTF-8 as the default encoding in XML and HTML.

UTF-8 encodes each of the 1,112,064 valid code points in the Unicode code space (1,114,112 code points minus 2,048 surrogate code points) using one to four 8-bit bytes (a group of 8 bits is known as an octet in the Unicode Standard). Code points with lower numerical values (i.e., earlier code positions in the Unicode character set, which tend to occur more frequently) are encoded using fewer bytes. The first 128 characters of Unicode, which correspond one-to-one with ASCII, are encoded using a single octet with the same binary value as ASCII, making valid ASCII text valid UTF-8-encoded Unicode as well. And ASCII bytes do not occur when encoding non-ASCII code points into UTF-8, making UTF-8 safe to use within most programming and document languages that interpret certain ASCII characters in a special way, e.g. as end of string.

Example of coding declaration in XML header (SPOI RDF data):

<?xml version="1.0" encoding="utf-8"?>

Implementation of UTF-8 is also evident from following example (in Ad 2. section, where Roman or Latin alphabet, Cyrillic alphabet and Korean alphabet /hangul/ are used).

4.7.2 Attribute xml:lang and RFC 5646 / BCP 47 Tags for the Identification of Languages

To distinguish various languages in data attributes (labels, descriptions, web resources), which are coded in some XML-based format, the standard xml:lang attribute is used. It bears also on data published in RDF format (such as Smart POIs).

The attribute xml:lang is defined by XML 1.0 as a common attribute that can be used to indicate the language of any element's contents. This includes any human readable text, as well as other content (such as embedded objects like images or sound files) contained by the element in which it appears. The xml:lang value applies to any sub-elements contained by the element. It also applies to attribute values associated with the element and sub-elements (though using natural language in attributes is not best practice). The value of the xml:lang attribute is a language tag defined by RFC 5646 / BCP 47 Tags for the Identification of Languages (http://www.rfc-editor.org/rfc/bcp/bcp47.txt).

Example of use xml:lang (SPOI RDF data):

<rdfs:label>평양직할시 - Pyongyang</rdfs:label>

<rdfs:label xml:lang="de">Pjöngjang</rdfs:label>

<rdfs:label xml:lang="en">Pyongyang</rdfs:label>

<rdfs:label xml:lang="fr">Pyongyang</rdfs:label>

<rdfs:label xml:lang="it">Pyongyang</rdfs:label>

<rdfs:label xml:lang="pt">Pyongyang</rdfs:label>

<rdfs:label xml:lang="ru">Пхеньян</rdfs:label>

4.7.3 System Moses

Moses (information was taken over from Moses web page – http://www.statmt.org/moses/) is an implementation of the statistical (or data-driven) approach to machine translation (MT). This is the dominant approach in the field at the moment, and is employed by the online translation systems deployed by the likes of Google and Microsoft. In statistical machine translation (SMT), translation systems are trained on large quantities of parallel data (from which the systems learn how to translate small segments), as well as even larger quantities of monolingual data (from which the systems learn what the target language should look like). Parallel data is a collection of sentences in two different languages, which is sentence-aligned, in that each sentence in one language is matched with its corresponding translated sentence in the other language. It is also known as a bitext.

The training process in Moses takes in the parallel data and uses occurrences of words and segments (known as phrases) to infer translation correspondences between the two languages of interest. In phrase-based machine translation, these correspondences are simply between continuous sequences of words, whereas in hierarchical phrase-based machine translation or syntax-based translation, more structure is added to the correspondences. The extra structure used in these types of systems may or may not be derived from a linguistic analysis of the parallel data. Moses also implements an extension of phrase-based machine translation know as factored translation which enables extra linguistic information to be added to a phrase-based systems.

The two main components in Moses are the training pipeline and the decoder. There are also a variety of contributed tools and utilities. The training pipeline is really a collection of tools (mainly written in perl, with some in C++) which take the raw data (parallel and monolingual) and turn it into a machine translation model. The decoder is a single C++ application which, given a trained machine translation model and a source sentence, will translate the source sentence into the target language.

In the SDI4Apps project the Moses was tested to translate longer texts (for example the descriptions of POIs or legend related to places labelled by POIs). SDI4apps partners also provided several texts from geographical domain to training pipeline. In the present day the results of translation are not satisfying completely. It is caused by an occurrence of specific terms, geographical (which should be not translated although they contain common words), old terms and names in case of legends. During next year the cooperation with authors of Moses (experts institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University in Prague, Czech Republic) will continue. Also more relevant results are expected.

Data used:

- http://eur-lex.europa.eu/oj/direct-access.html
- http://eur-lex.europa.eu/search.html

Quantitative indicators are taken from D.4.1.1 and here reported.

4.7.4 Functional Report

Indicator	Measurement	Status
Word correctly translated	All available words	Passed

Sentence correctly translated	2000 sentences correctly translated	Passed

Table 39: Multilingual module functional test

4.7.5 Non-functional Report

Please note that the parameters for the non-functional test are many and those not allow to present a readable table if reported in the same template as exploited before. Nevertheless, same approach as for the other module tests was adopted and below we report the Indicator, i.e. Corpora table prior de-cuplication and the Measurement performed in table 41 Multilingual module non-functional test: After de-duplication. According to the results presented below, Status of this test is: Passed.

Indicator	Measurement	Status
Parameters as esplicated in table 40: "Multilingual module non functional test: prior de-cuplication"	Parameters as esplicated in table 41: "Multilingual module non functional test: after de-cuplication"	Passed

Table 40: Multilingual module non functional test

	Segments	English Words	Czech Words
Direct Access, Parallel	5.19M	36.10M	28.75M
Search Access, Parallel	10.38M	91.98M	122.76M
Direct Access, Monolingual	20.96M	--	141.12M
Direct Access, Monolingual	20.93M	158.74M	--
Search Access, Monolingual	40.10M	--	304.49M
Search Access, Monolingual	45.24M	633.99M	--

Table 41: Multilingual module non functional test: prior de-cuplication

After de-duplication at the segment level, we have, approximately figures as indicated in the table below.

	Segments	English Words	Czech Words
Direct Access, Parallel	1.54M	26.83M	20.42M
Search Access, Parallel	2.92M	63.25M	95.68M

Direct Access, Monolingual	5.36M	--	97.38M
Direct Access, Monolingual	5.15M	107.83M	--
Search Access, Monolingual	10.77M	--	169.32M
Search Access, Monolingual	13.94M	462.76M	--

Table 42: Multilingual module non-functional test: After de-duplication

4.8 Semantic tools for LOD data harmonization module

Data used:

http://sdi4apps.eu/spoi/ (Smart POI)

Description of the process

The following text describes how we used semantic tools for Linked Open Data harmonization module, which are used in Open Smart Tourist Data pilot (development of Smart Points of Interest data set). But the provided information and tools are so general that it can be re-used in other cases and situations.

The application of single-purpose scripts proved to be the right solution for harmonization steps used in SDI4Apps. The reason consists in an occurrence of huge number of very heterogeneous input data, including non-structured data and data provided by non-experienced persons. Therefore the main semantic tools used in harmonization do not represent concrete software products of services, but existing vocabularies. They increase an information value of each piece of data, because they refer to relevant information resources containing explaining features, metadata and other semantic information.

The following list shows the main existing vocabularies implemented to SPOI data model:

- **DC (Dublin Core)** – http://dublincore.org/

  The Dublin Core Schema is a small set of vocabulary terms that can be used to describe web resources (video, images, web pages, etc.), as well as physical resources such as books or CDs, and objects like artworks. The full set of Dublin Core metadata terms can be found on the Dublin Core Metadata Initiative (DCMI) website. The original set of 15 classic metadata terms, known as the Dublin Core Metadata Element Set] are endorsed in the following standards documents: IETF RFC 5013, ISO Standard 15836-2009, NISO Standard Z39.85.

  Dublin Core Metadata may be used for multiple purposes, from simple resource description, to combining metadata vocabularies of different metadata standards, to providing interoperability for metadata vocabularies in the Linked Data cloud and Semantic Web implementations.

- **FOAF (Friend of a Friend)** – http://www.foaf-project.org/

  FOAF is a machine-readable ontology describing persons, their activities and their relations to other people and objects. FOAF is a descriptive vocabulary expressed using the Resource Description Framework (RDF) and the Web Ontology Language (OWL). Each profile has a unique identifier (such as the person's e-mail addresses, a Jabber ID, or a URI of the homepage or weblog of the person), which is used when defining these relationships. The FOAF project, which defines and extends the vocabulary of a FOAF profile, was started in 2000 by Libby Miller and Dan Brickley. It can be considered the first Social Semantic Web application, in that it combines RDF technology with 'Social Web' concerns.

- **GeoSPARQL** (SPARQL Protocol and RDF Query Language) –

http://www.opengeospatial.org/standards/geosparql

GeoSPARQL is a standard for representation and querying of geospatial linked data for the Semantic Web from the Open Geospatial Consortium (OGC). The definition of a small ontology based on well-understood OGC standards is intended to provide a standardized exchange basis for geospatial RDF data which can support both qualitative and quantitative spatial reasoning and querying with the SPARQL database query language.

The Ordnance Survey Linked Data Platform uses OWL mappings for GeoSPARQL equivalent properties in its vocabulary. The LinkedGeoData data set is a work of the Agile Knowledge Engineering and Semantic Web (AKSW) research group at the University of Leipzig, a group mostly known for DBpedia, that uses the GeoSPARQL vocabulary to represent OpenStreetMap data.

In particular, GeoSPARQL provides for:

- a small topological ontology in RDFS/OWL for representation using
  - Geography Markup Language (GML) and well-known text (WKT) literals, and
  - Simple Features, RCC8, and DE-9IM (a.k.a. Egenhofer) topological relationship vocabularies and ontologies for qualitative reasoning, and
- a SPARQL query interface using
  - a set of topological SPARQL extension functions for quantitative reasoning, and
  - a set of Rule Interchange Format (RIF) Core inference rules for query transformation and interpretation.

- **OWL (Web Ontology Language)** – https://www.w3.org/TR/owl2-overview/

  The OWL is a family of knowledge representation languages for authoring ontologies. Ontologies resemble class hierarchies in object-oriented programming but there are several critical differences. Class hierarchies are meant to represent structures used in source code that evolve fairly slowly (typically monthly revisions) whereas ontologies are meant to represent information on the Internet and are expected to be evolving almost constantly. Similarly, ontologies are typically far more flexible as they are meant to represent information on the Internet coming from all sorts of heterogeneous data sources. Class hierarchies on the other hand are meant to be fairly static and rely on far less diverse and more structured sources of data such as corporate databases.

  The OWL languages are characterized by formal semantics. They are built upon a W3C XML standard for objects called the Resource Description Framework (RDF). OWL and RDF have attracted significant academic, medical and commercial interest.The OWL family contains many species, serializations, syntaxes and specifications with similar names. OWL and OWL2 are used to refer to the 2004 and 2009 specifications, respectively. Full species names will be used, including specification version (for example, OWL2 EL). When referring more generally, OWL Family will be used.

- **RDFS (Resource Description Framework Schema)** – https://www.w3.org/TR/rdf-schema/

  RDF Schema is a set of classes with certain properties using the RDF extensible knowledge representation data model, providing basic elements for the description of ontologies, otherwise called RDF vocabularies, intended to structure RDF resources. These resources can be saved in a triplestore to reach them with the query language SPARQL. The RDFS was published by the World Wide Web Consortium (W3C). The RDFS contains common properties such as domain, range, label, comment, seeAlso etc.

- **SKOS (Simple Knowledge Organization System)** – https://www.w3.org/2004/02/skos/

  Simple Knowledge Organization System (SKOS) is a W3C recommendation designed for representation of thesauri, classification schemes, taxonomies, subject-heading systems, or any other type of structured controlled vocabulary. SKOS is part of the Semantic Web family of standards built upon RDF and RDFS, and its main objective is to enable easy publication and use of such vocabularies as linked data.

  The SKOS defines the classes and properties sufficient to represent the common features found in a standard thesaurus. It is based on a concept-centric view of the vocabulary, where primitive objects

are not terms, but abstract notions represented by terms. Each SKOS concept is defined as an RDF resource. Each concept can have RDF properties attached, including:

- one or more preferred index terms (at most one in each natural language)
- alternative terms or synonyms
- definitions and notes, with specification of their language

Concepts can be organized in hierarchies using broader-narrower relationships, or linked by non-hierarchical (associative) relationships. Concepts can be gathered in concept schemes, to provide consistent and structured sets of concepts, representing whole or part of a controlled vocabulary.

Vocabulary	Adopted items	Using
DC	dc:identifier dc:publisher dc:title dc:rights dc:source dc:terms	Properties to describe metadata of particular POI.
FOAF	foaf:mbox foaf:phone foaf:homepage	In the future several of these properties will be transformed from feature level to data set level. Also the ontology PROV-O will be tested.
GeoSPARQL	geos:asWKT geos:sfWithin	Essential contact information (email box, phone number and URL of web page).
OWL	owl:sameAs	In the future the Location Core Vocabulary will be used for a description of addresses (instead of string).
RDFS	rdfs:label rdfs:comment rdfs:seeAlso	First property is used to encoding geometry of POI. Second property represents a topological relation (is located in).
SKOS	skos:exactMatch	Link to a relevant element in other linked data resource (for example DBpedia or GeoNames.org)

Table 43: Existing vocabularies implemented to SPOI data model

In pursuance of SPOI development the RDF vocabulary describing classification of POIs used in Waze navigation system was published. This vocabulary contains ten basic categories of POIs, which are mandatory in SPOI data model. Because of the vocabulary contains URIs as identifier, the category is connected with each POI as link (not as a simple string). In the future we plan to transform to RDF vocabulary also the second classification used in SPOI (adopted from the OpenStreetMap).

The Waze vocabulary is based on SKOS standard. It contains except above mentioned identifier also link to the core element (“Concept”) and English name of the category (taken over from Waze documentation).

<rdf:RDF>

<rdf:Description rdf:about="http://www.openvoc.eu/waze_classification#Car_services">
<rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
<skos:prefLabel xml:lang="en">Car Services</skos:prefLabel>
</rdf:Description>

<rdf:Description rdf:about="http://www.openvoc.eu/waze_classification#Transportation">
<rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
<skos:prefLabel xml:lang="en">Transportation</skos:prefLabel>
</rdf:Description>

<rdf:Description rdf:about="http://www.openvoc.eu/waze_classification#Professional_and_public">
<rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
<skos:prefLabel xml:lang="en">Professional and public</skos:prefLabel>
</rdf:Description>

<rdf:Description rdf:about="http://www.openvoc.eu/waze_classification#Shopping_and_services">
<rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
<skos:prefLabel xml:lang="en">Shopping and services</skos:prefLabel>
</rdf:Description>

<rdf:Description rdf:about="http://www.openvoc.eu/waze_classification#Food_and_drink">
<rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
<skos:prefLabel xml:lang="en">Food and drink</skos:prefLabel>
</rdf:Description>

<rdf:Description rdf:about="http://www.openvoc.eu/waze_classification#Culture_&_entertainment">
<rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
<skos:prefLabel xml:lang="en">Culture & entertainment</skos:prefLabel>
</rdf:Description>

<rdf:Description rdf:about="http://www.openvoc.eu/waze_classification#Other">
<rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
<skos:prefLabel xml:lang="en">Other</skos:prefLabel>
</rdf:Description>

<rdf:Description rdf:about="http://www.openvoc.eu/waze_classification#Lodging">
<rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
<skos:prefLabel xml:lang="en">Lodging</skos:prefLabel>

```
</rdf:Description>

<rdf:Description rdf:about="http://www.openvoc.eu/waze_classification#Outdoors">
<rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
<skos:prefLabel xml:lang="en">Outdoors</skos:prefLabel>
</rdf:Description>

<rdf:Description rdf:about="http://www.openvoc.eu/waze_classification#Natural_features">
<rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
<skos:prefLabel xml:lang="en">Natural features</skos:prefLabel>
</rdf:Description>

</rdf:RDF>
```

4.8.1 Functional report

Indicator	Measurement	Status
Opendata dataset	1 dataset available in 5-star LOD1	Passed
Data categorized	Test data are categorized in 10 categories	Passed

Table 44: Semantic tools module functional test

4.8.2 Non-functional Report

Indicator	Measurement	Status
Opendata dataset	6 onthology used	Passed
Data categorized	1 SPARQL endpoint2	Passed

Table 45: Semantic tools module non-functional test

[1] http://sdi4apps.eu/spoi/
[2] http://data.plan4all.eu/sparql

5 CONCLUSION

The tests results reported in this report are the outcomes gathered by carrying out due tasks according to the DoW and to the recommendations provided by reviewers and the panel of experts. Furthermore, the modules tested in this report were also exploited during the Code Camps and the Hackathons organised by the Consortium which highly benefited of the feedback provided by participating developers in improving the work carried out.

Where relevant future tests will reflect the outcomes of the Abstract tests suites development for the INSPIRE metadata, discovery services, view services (WMS, WMTS) and download services (Atom, WFS)[3].

[3] https://ies-svn.jrc.ec.europa.eu/issues/2685