# COMPETITIVENESS AND INNOVATION FRAMEWORK PROGRAMME

**CIP-ICT-PSP-2013-7 Pilot Type B**


# WP3 – Service platform integration and deployment in cloud infrastructure

# D3.3.2: Sematic tagging and open data publication tools

Deliverable Lead: PSNC

Deliverable due date: 29/02/2016

Actual submission date: 29/02/2016

Version: 1.2

This project is partially funded under the ICT Policy Support Programme (ICT PSP) as part of the Competitiveness and Innovation Framework Programme by the European Commission under grant agreement no. 621074

## Document Control Page

| | |
|---|---|
| **Title** | D3.3.2 Semantic tagging and open data publication tools |
| **Creator** | Marcin Krystek (PSNC) |
| **Description** | This document introduces the current prototype of the semantic annotation service |
| **Publisher** | FOODIE Consortium |
| **Contributors** | Raul Palma (PSNC), Emilio Rubiera (CTIC) |
| **Creation date** | 01/02/2016 |
| **Type** | Text |
| **Language** | en-GB |
| **Rights** | copyright "FOODIE Consortium" |
| **Audience** | ☐ internal<br>☒ public<br>☐ restricted |
| **Review status** | ☐ Draft<br>☐ WP leader accepted<br>☐ Technical Manager accepted<br>☒ Coordinator accepted |
| **Action requested** | ☐ to be revised by Partners<br>☐ for approval by the WP leader<br>☐ for approval by the Technical Committee<br>☐ for approval by the Project Coordinator |
| **Requested deadline** | |

**About the project**

FOODIE project aims at creating a platform hub on the cloud where spatial and non-spatial data related to agricultural sector is available for agri-food stakeholders groups and interoperable. It will offer: an infrastructure for the building of an interacting and collaborative network; the integration of existing open datasets related to agriculture; data publication and data linking of external agriculture data sources, providing specific and high-value applications and services for the support of planning and decision-making processes.

FOODIE project is addressed to four basic groups of users: a) stakeholders from the agriculture sector as end-users of final applications, b) public sector for communication with farmers about taxation, subsidies and regulation, c) researchers for large scale experimentation on real data and d) ICT companies for the development of new applications for agriculture and food sector, mainly using implemented tools

FOODIE specifically works on three pilots:

- Pilot 1: Precision Viticulture (Spain) will focus on the appropriate management of the inherent variability of crops,
- Pilot 2: Open Data for Strategic and Tactical Planning (Czech Republic) will focus on improving future management of agricultural companies (farms) by introducing new tools and management methods,
- Pilot 3: Technology allows integration of logistics via service providers and farm management including traceability (Germany).

**Contact information**

Miguel Angel Esbrí

Project Coordinator

Atos Spain, Madrid, Spain

E-mail: miguel.esbri@atos.net

URL: http://www.foodie-project.eu

Twitter: https://twitter.com/FOODIE_Project

# Table of Contents

## Glossary

The glossary of terms used in this deliverable can be found in the public document "FOODIE_Glossary.pdf" available at: http://www.foodie-project.eu

## Abbreviations and Acronyms

| Abbreviation / Acronym | Description |
|---|---|
| API | Application Programming Interface |
| CPU | Central Processing Unit |
| DBA | Database Administrator |
| DBaaS | Database as a Service |
| DDD | Domain-Driven Design |
| DNS | Domain Name System |
| GTM | Global Transaction Manager |
| HA | High Availability |
| HDD | Hard Disk Drive |
| HTTP | Hypertext Transfer Protocol |
| IaaS | Infrastructure as a Service |
| IDE | Integrated Development Environment |
| JAR | Java ARchive |
| MPP | Massive Parallel Processing |
| OS | Operating System |
| POM | Project Object Model |
| RAM | Random Access Memory |
| RDBMS | Relational Database Management System |
| RDF | Resource Description Framework |
| SLA | Service Level Agreement |
| SQL | Structured Query Language |
| TCP | Transmission Control Protocol |
| TDD | Test – Driven Development |
| VM | Virtual Machine |
| API | Application Programming Interface |

*Table 1: Abbreviations and Acronyms*

## Executive Summary

This document introduces the semantic annotation service.

# 1 Introduction

Semantic tagging and open data publication tools is a system of services which can be used to extract additional knowledge from unstructured data such as plain text or text files in different formats, transform (semi-) structured data into semantic format, and publish the generated data according the Linked Data principles.

In particular, semantic annotations enrich the source data with a context that is linked to some structured knowledge of a domain or application, which can then be exploited by different applications and services. For instance, this knowledge can be used to enrich the description of the resources presented to the user by the Marketplace platform, as described in D3.5.1[1].

Since the newly discovered knowledge is described by standard ontologies, stored in machine-readable format and accessible through standard APIs and protocols, it can also be used for further machine processing allowing better integration with existing knowledge bases and their publication in the Linked Open Data Cloud[2], discovering and understanding relations and dependencies between resources (e.g., Marketplace products), as well as the implementation of all other kind of user scenarios.

Similarly, semantic data generated through the transformation of (semi-)structured data sources, such as tabular data or XML, can then be exploited to discover additional knowledge through the integration with other data elements via semantic links.

# 2 Components Description

Semantic tagging and open data publication tools compose together a system in which the following components can be distinguished:

## 2.1 Semantic Annotation Service

Semantic Annotation Service is a core component of the system. It provides a simple REST API designed for creating, updating and retrieving annotations. Annotation Service orchestrates other services in order to control and fully perform data analysis process, creates semantic form of the generated data and persists semantic data using semantic store.

## 2.2 Annotation Tools

These are external tools used as libraries or services that allows keywords extraction, sentence disambiguation and text meaning identification. The results generated by these tools are then used to create semantic representation of the annotation. Current implementation of the Semantic Annotation Service uses the following annotation tools:

### 2.2.1 AgroTagger

AgroTagger[3] is a keyword extractor that uses the ARGOVOC thesaurus as its set of allowable keywords. It also identifies concepts from AGROVOC vocabulary in terms of Linked Open Data. AGROVOC is a controlled vocabulary covering all areas of interest of the Food and Agriculture Organization (FAO) of the United Nations, including food, nutrition, agriculture, fisheries, forestry, environment etc. It is published by FAO and edited by a community of experts.

### 2.2.2 Babelfy

Babelfy[4] is a unified, multilingual, graph-based approach to Entity Linking and Word Sense Disambiguation based on a loose identification of candidate meanings coupled with a densest subgraph heuristic, which selects high-coherence semantic interpretations.

## 2.3 Data Semantization Service

The data semantization service is a tool implemented to carry out chiefly conversions from a heterogeneous range of data into RDF.

Mainly focused on the transformation of tabular data in different formats, it is expected to add as well means to transform XML files into RDF, in future implementations.

It is an open source tool with a simple Web interface where users upload the file to be transformed and thanks to the mediation of a small lightweight piece of code that identifies the structure of the added data, it is capable of generating an output file fully compliant with the RDF family language.

This early version offers only a manual tool with no guidelines or best practices for the end user; however, in the next releases, it is planned that the conversion become more user-friendly and even show semi-automatic capabilities. Besides, the number of input format supported will be increased.

Thanks to this tool, users will be able to transform their unstructured or semi-structured data into more friendly formats and pave the way to share this data in an easier way or simply get that data in a format fully compliant with the linked open data guidelines.

## 2.4 Data Access and Publishing Components

Semantic annotations and other semantic data generated and stored by the semantic services are accessible using the following frameworks and API's:

### 2.4.1 Silk

Silk[5] is a linked data integration framework for integrating heterogeneous data sources. It allows generating links between related data items within different Linked Data sources and applying data transformations to structured data sources. Linked Data publishers can use Silk to set RDF links from their data sources to other data sources on the Web. Accordingly, in FOODIE, Silk is used to discover (and store) links between the semantic data in the semantic store and other key knowledge bases in the Linked Open Data Cloud.

### 2.4.2 Sparql

Sparql[6] is as standard query language for RDF data. FOODIE provides access to RDF data through a SPARQL endpoint that is exposed by the semantic store.

### 2.4.3 Faced Search and Browser

Faced Search and Browser is based on Virtuoso Faceted Browser[7], which allows simple text search and results browsing. In FOODIE, we use this interface mainly to enable simple navigation of RDF data through their links.

### 2.4.4 Semantic Annotation Service

Semantic Annotation Service provides dedicated API which allows annotation retrieval in the simplified JSON form. Currently this API is being used by the Marketplace portal[8], which showcases its usage through external applications.

## 2.5 Storage for Annotation Service

The Semantic Annotation service processes and produces various types of data. In order to meet specificity of the data type as well as different requirements and constraints regarding data processing procedure, three different storage components are used.

### 2.5.1 Semantic Store

For storing semantic data the OpenLink Virtuoso[9] server is used. Virtuoso has built in support for SPARQL query language and inference engine. It also provides good performance, geospatial support and capabilities to generate linked data from different data sources types via Sponger. Sponger is also transparently integrated into Virtuoso's SPARQL Query Processor delivering URI de-referencing within SPARQL query patterns and is directly accessible via REST interfaces.

### 2.5.2 RDBMS

For storing simple relational data such as mappings, resource processing status and statistics the relational database management system is used. The current implementation of Semantic Annotation Service uses sqlite[10] implementation as simple, lite and providing satisfying performance solution. Since sqlite provides standard SQL interface it can be easily replaced with other, more advanced solution if required.

### 2.5.3 File system

Semantic Annotation Service allows annotating unstructured files. All files send to the service are stored temporally in the file system. They are available only for the annotation procedure and will be removed after procedure successfully finishes.

# 3 System Design

Semantic tagging and open data publication tools compose together a system for which three main layers can be identified:

- API layer – implemented as set of REST services. Each service provides different functionality for the client application or final user. API's denotes overall system functionality.
- Service layer – it is set of different web services or libraries directly used by Semantic Annotation Service or operating on data generated by Semantic Annotation Service
- Storage layer – provides persistence for all data processed or generated by the Semantic Annotation Service

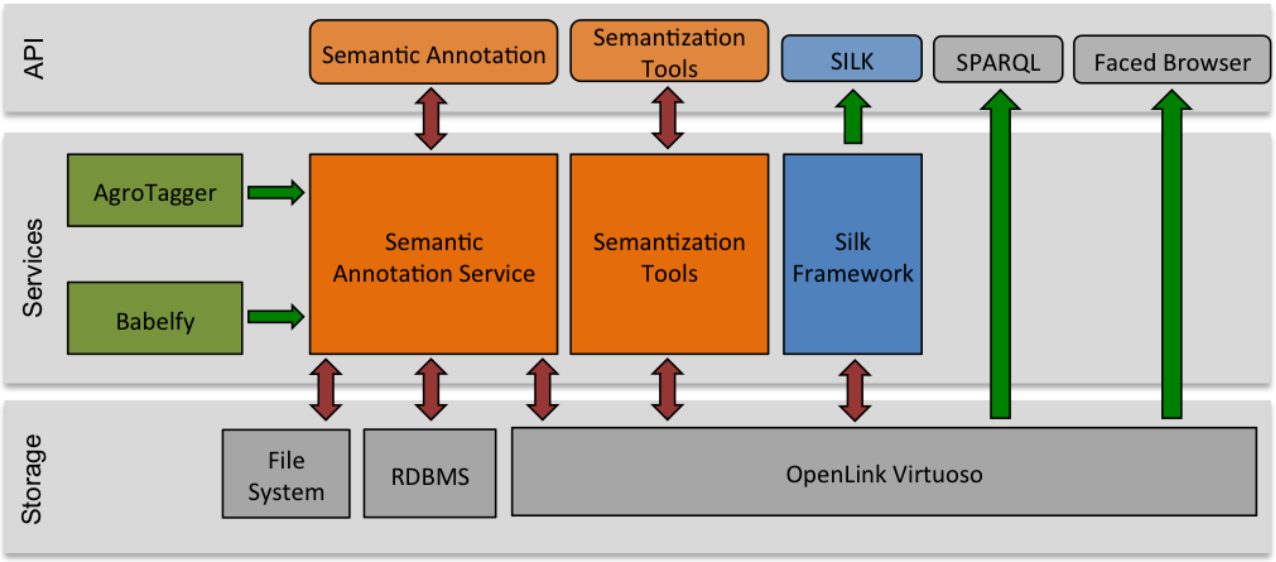


*Figure 1 Sematic tagging and open data publication tools system design*

Arrows on the Figure 1 denotes direction of the data flow direction.

## 4 Data Model for Annotations

Annotations created by the Semantic Annotation Service are modeled using Modular Unified Tagging Ontology (MUTO)[11]. It is designed specifically for tagging and folksonomies. It allows representing public and private tagging, simple and auto generated tags and others. It is also easily extensible since all concepts defined in MUTO ontology inherits from other more general ontologies like SKOS[12], SIOC[13] or vocabularies as RDFS[14].

The general structure of the MUTO ontology and main concepts are presented in the following diagram.

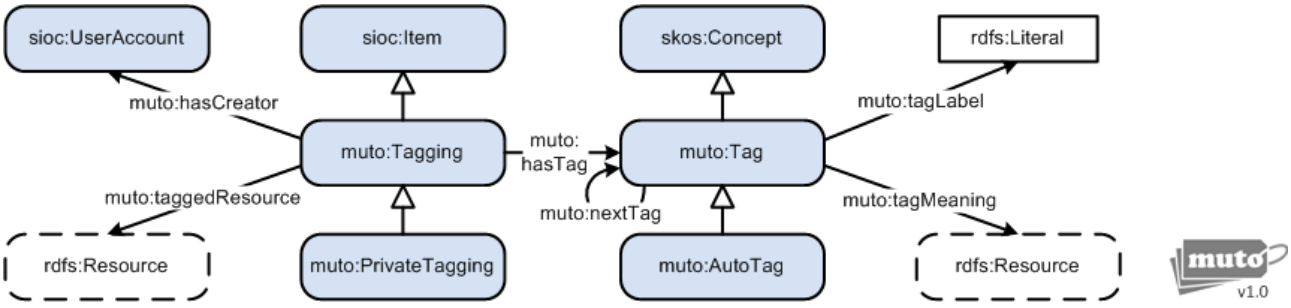


*Figure 2 MUTO ontology general design*

The central concept of the MUTO ontology is muto:Tagging. muto:Tagging instance is created as response for create annotation request. For each tagging a tagged resource is also created. Tagged resource instance represents an object that is actually annotated. It can be anything (file, set of files, text description) that can be addressed or identified by the client. For each tagging instance there is a set of muto:Tag instances created. Each muto:Tag instance represents a singe concept identified by the annotation tools. muto:Tag instance has its default label defined as muto:tagLabel. If the identified concept has its representation in the Linked Open Data Cloud, an additional skos:Concept instance is created and related with tag using muto:tagMeaning. This object can be also enriched with labels in different languages if such are available in LOD Cloud.

Tags identified by automatic annotation tools are created as muto:AutoTag instances. Tags defined by user manually are stored as muto:Tag instances. muto:PrivateTagging and sioc:UserAccount are currently not used.

# 5 Implementation Details

## 5.1 Semantic Annotation Service

### 5.1.1 REST API

Semantic Annotation Service REST API provides the following functionality:

| Method | Description |
|---|---|
| **POST /tagging** | Creates new tagging for requested list of files and text description. Tagging will consist of set of tags (annotations) identified by annotation tools. It also creates tagged resource instance. If resource instance URI is provided in the request, the method updates existing tagging. |
| **GET /tagging** | Retrieves tag labels for the resource and language specified in the request. |
| **POST /tag** | Adds new tag to the existing tagging. |
| **GET /statistics/meaning** | Retrieves statistics showing which concepts are most commonly identified in the annotated resources. |

More technical description of the Semantic Annotation Service REST API can be found in D3.2.2. Open API specification.

The tagging creation request may contain plain text description or set of files. Current implementation of Semantic Annotation Service supports processing of the following file types: PDF, DOC, XLS, HTML, TXT

### 5.1.2 Development and frameworks

- Programming language
  Semantic Annotation Service is a Java web application. It requires at least Java 1.7 for development and for runtime environment.
- Application framework
  Semantic Annotation Service uses Spring v.4.1.4 as main application framework, taking advantage of its been management and event processing functionalities as well as support for REST API implementation.
- Building tools
  Semantic Annotation Service is build using Apache Maven v.3.1.3. All project dependencies and deployment process are defined and managed by Maven.
- Web container
  Building process creates war file that can be deployed on any web container. Currently Apache Tomcat v.7.0 is used for the development and production deployment.
- Source code location

Source code of the application is available on git repository: https://git.man.poznan.pl/stash/projects/FOOD/repos/semantic-annotation/browse/SemanticAnnotationRestApi

- Artifactory
  Dependencies that cannot be resolved using public repositories are defined in the following artifactory: http://maven.man.poznan.pl/repository/simple/foodie-releases-public

## 5.2 Data Semantization Service

The Data Semantization Service comprises two components: Tabels Core and Tabels Web.

### 5.2.1 Tabels Core

Tabels Core helds the capability to map from supported tabular formats to RDF documents; the mapping between them should be specified by a transformation program by the means of a DSL (domain specific language) developed with the only purpose of these mappings.

Tabels core was developed in Scala language and using sbt to support build definitions. The source code is available at: https://bitbucket.org/fundacionctic/tabels

### 5.2.2 Tabels Web

Tabels Web is a demonstrator for the interoperability capabilities of a group of technologies related to Linked Data. It integrates online tabels core, tapinos-ws, tapinos-js, undermaps and a bunch of linked data explotation tools.

Tabels Web was developed in Grails. The source code is available at: https://bitbucket.org/fundacionctic/tabels

# 6 Deployment

## 6.1 Service Location

- Semantic Annotation Service is available at:
  http://foodie-vm3.man.poznan.pl/semanticAnnotation
- Data Semantization Service is available at:
  http://prepro.fundacionctic.org/foodie/ (not yet deployed in FOODIE cloud)
- SPARQL endpoint is available at:
  http://foodie-vm1.man.poznan.pl/sparql
- Faced Browser is available at:
  http://foodie-vm1.man.poznan.pl/fct
- SILK endpoint is available at:
- http://foodie-vm3.man.poznan.pl/silk

## 6.2 Semantic Annotation Demo Client

Demo client is a simple web form. It allows user to simply add text description and select files that he wants to annotate. The form values are then sent to the Semantic Annotation Service where tagging is performed. As a

response, user will receive URI of the created resource. User will use this resource URI to retrieve or update existing tagging. Currently available at: http://foodie-vm3.man.poznan.pl/semanticAnnotation/index.html

## References

[1] D3.5.1 Marketplace; Maciej Łabędzki, Patryk Promiński, Adam Rybicki

[2] http://lod-cloud.net/

[3] https://github.com/fcproj/agrotagger

[4] http://babelfy.org/about

[5] http://silkframework.org/

[6] https://www.w3.org/TR/rdf-sparql-query/

[7] http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VirtFacetBrowserInstallConfig

[8] https://www.foodie-cloud.org/

[9] http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/

[10] https://www.sqlite.org/

[11] http://muto.socialtagging.org/core/v1.html

[12] https://www.w3.org/2004/02/skos/

[13] http://rdfs.org/sioc/spec/

[14] https://www.w3.org/TR/rdf-schema/