# Seventh Framework Programme
# ICT-2009-6.4
# Information and Communication Technology

# Tagging Tool based on a Semantic Discovery Framework

**Project ID: 247893**

## Deliverable D3.1.3

**Version 1.2.7**

## Semantic Service Environment and Framework Architecture V3

Due date of deliverable: 30/04/2012

Internal release date: 15/06/2012
Actual submission date: --/06/2012

# Document Control Page

| | |
|---|---|
| **Title** | Semantic Service Environment and Framework Architecture V3 |
| **Creator** | cismet GmbH |
| **Editor** | Pascal Dihé |
| **Description** | This document provides the final specification of the TaToo Framework Architecture. |
| **Publisher** | TaToo Consortium |
| **Contributors** | CIS, TPZ, ATOS, IDSIA, AIT, JRC, MU |
| **Type** | Text |
| **Format** | MS Word |
| **Language** | EN-GB |
| **Creation date** | 25.03.2011 |
| **Version number** | 1.2.7 |
| **Version date** | 14.06.2012 |
| **Last modified by** | Pascal Dihé |
| **Rights** | Copyright ‘TaToo Consortium‘.<br>During the drafting process, access is generally limited to the TaToo Partners. |
| **Audience** | □ internal<br>☒ public<br>□ restricted,<br>access granted to: |
| **Review status** | □ Draft<br>☒ WP Leader accepted<br>□ PCO quality controlled<br>□ Co-ordinator accepted<br>Where applicable:<br>□ Accepted by the GA<br>□ Accepted by the GA as public document |
| **Action requested** | □ to be revised by Partners involved in the preparation of the Project Deliverable<br>□ to be revised by all TaToo Partners<br>□ for approval of the WP Leader<br>☒ for approval of the PCO (Quality Manager)<br>☒ for approval of the Project Co-ordinator<br>□ for approval of the General Assembly |
| **Requested deadline** | 18/06/2012 |

| Revision history | | | |
|---|---|---|---|
| **Version** | **Date** | **Modified by** | **Comments** |
| 0.1 | 19/04/2011 | PDi | Draft Structure |
| 0.2 | 09/05/2011 | PDi | 1st internal release |
| 0.3 | 27/05/2011 | PDi | + Roadmap & Tasks |
| 0.4 | 17/06/2011 | PDi | + architecture updates, general descriptions, etc |
| 0.5 | 17/07/2011 | BBo | + AIT Contributions |
| 0.6 | 19/07/2011 | TPa, JMF, SYu | + ATOS Contributions |
| 0.7 | 20/07/2011 | GAv, LPe | + TPZ Contributions |
| 0.8 | 21/07/2011 | PDi | + Administration |
| 0.8.1 | 22/07/2011 | BBo | + Visualisation |
| 0.8.2 | 25/07/2011 | PDi | Draft ready for preliminary review |
| 0.8.3 | 26/07/2011 | PDi | Update after PCO review |
| 0.9 | 30/07/2011 | PDi, BBo, GAv, LPe | + final contributions |
| 1.0 | 31/07/2011 | PDi | Final version ready for QA |
| 1.0 | 09/08/2011 | GSc, TPa | QA |
| 1.1 | 02/04/2012 | PDi | Updates for V3 |
| 1.1.3 | 20/05/2012 | TPa | + initial ATOS Contributions |
| 1.1.4 | 21/05/2012 | PDi | + CIS Contributions |
| 1.1.5 | 22/05/2012 | PDi | + TPZ Contributions |
| 1.1.6 | 23/05/2012 | SNe | + IDSIA updates of the evaluation components |
| 1.1.7 | 24/05/2012 | SNe | + IDSIA Updates on Linked Data |
| 1.1.8 | 25/05/2012 | PDi | + CIS Contributions, integration of content |
| 1.1.9 | 29/05/2012 | PDi | Tagging service & Tagging Core Use Case updated |
| 1.2.0 | 31/05/2012 | LPe | + conceptual description of Tagging |
| 1.2.1 | 01/06/2012 | PDi | Integration and updates |
| 1.2.2 | 03/06/2012 | TPa | + final ATOS Contributions |
| 1.2.3 | 08/06/2012 | GAv, PDi | + final TPZ Contribution |
| 1.2.4 | 11/06/2012 | BBo | + final AIT Contribution |
| 1.2.5 | 12/06/2012 | PDi | + Annexes E & F |
| 1.2.6 | 13/06/2012 | SPF, RGü | Review by SPF & RGü |
| 1.2.7 | 14/06/2012 | PDi | Final version for QA |
| | | | |
| | | | |

*THIS DOCUMENT IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS DOCUMENT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.*

----------------

## Table of Contents

## Index of Figures

## Index of Diagrams

## Index of Tables

# 1. Management summary

The present document has been produced by the consortium of the European Project FP7-247893 Tagging Tool based on a Semantic Discovery Framework (TaToo) and corresponds to the deliverable D3.1.3 - Semantic Service Environment and Framework Architecture V3 of WP3 - Specification. It is the final revision of the initial architecture specification presented in D3.1.1 (TaToo-D311, 2010) and the enhanced specification presented in D3.1.2 (TaToo-D312, 2011).It describes the third and last iteration of the TaToo Semantic Service Environment and Framework Architecture including the achievements and progress made in V1, V2 and V3. It serves as the basis for the third software design and implementation phase of TaToo performed in WP4 – Implementation and thus as blueprint for the reference implementation of the TaToo Framework.

The aim of the TaToo Framework is to provide an infrastructure to bridge the discovery gap between environmental resources and users. The TaToo Framework shall enable experts as well as arbitrary users to share trusted and reliable information and to allow easy semantically-enhanced discovery and tagging of existing environmental information. The architecture outlined in this document contributes to the fulfilments of this aim by establishing the conceptual foundations of the TaToo Framework.

## 1.1. Purpose of this document

The goal of this document is to specify the TaToo Framework Architecture on an implementation independent level while considering the functional and non-functional requirements identified during the requirements analysis phase. It provides implementation independent descriptions and specifications of components and describes the basic information flows with the help of a set of Core Use-Cases focusing on TaToo's base functionality.

This document establishes the conceptual framework for the discovery and tagging of resources in TaToo while also considering comprehensively the aspects of multilingualism, security, evaluation, and Linked Data principles. Thereby it describes the TaToo discovery process emphasizing in its relations with other Information Retrieval (IR) systems and the advantages of the semantically-enhanced TaToo tagging process.

It provides several annexes that contain besides the descriptions of all the TaToo Framework components detailed implementation independent (functional) specifications of discovery and tagging related components and a description of the architectural processes applied and the architectural design decisions taken.

## 1.2. Intended audience

The target readers of this document are individuals interested in the TaToo project as well as Work Package- and Task Leaders of the TaToo project (WP4 - Implementation, WP5 - Validation Scenarios) involved in the specification and implementation of TaToo Services and Tools and the TaToo Framework.

## 1.3. Structure of the document

The structure of the document and the relationships between the different chapters and annexes is as follows.

- **Chapter 1** consists of this executive summary, which offers an overview and explains the overall purpose of this document.
- **Chapter 2** lists the abbreviations and acronyms used throughout the document and provides definitions of important terms that are required to understand this document.
- **Chapter 3** provides a short introduction to the architectural design process and explains the basic architectural principles of the TaToo Framework Architecture.
- **Chapter 4** provides an overview of the TaToo System from the discovery and tagging point of view. It describes how TaToo fits into the IR literature and establishes the boundaries of the ontology based discovery in TaToo. It continues with the description of a tagging system based on ontologies and a description of the TaToo approach for evaluation of resources and annotations and finishes with a description on how Linked Data principles are applied in TaToo.
- **Chapter 5** specifies Cross Tiers (and cross issues) of the architecture which addresses common aspects and concepts with influence on all architectural components. These aspects thoroughly considered in the third iteration of the architecture as well as in the reference implementation encompass security, multilinguality, system administration and harvesting of resources.
- **Chapter 6** is the actual specification of the TaToo Framework Architecture and includes an introduction to the most important components, the definitions of the high level Building Blocks, and the Core Use-Cases.
- **Chapter 7** summarises the results of the architecture work performed and provides a conclusion.
- **Chapter 8** recognises that research was funded by the European Community.
- **Chapter 9** lists the references and bibliography used in writing this document.

This document has six annexes:

- **Annex A** (TaToo-D313a, 2012) provides a template for the functional specification of TaToo Public Services.
- **Annex B** (TaToo-D313b, 2012) provides a template for the functional specification of TaToo Components. In general, the Component Specification Template is used for Tools (User Components) with graphical user interface or any other components that are not realised as services (e.g. certain Core Components).
- **Annex C** (TaToo-D313c, 2012) contains the final functional specifications of discovery related User Components, Public Services and Core Components according to the templates provided in Annex A and B.
- **Annex D** (TaToo-D313d, 2012) contains the final functional specifications of tagging, evaluation and linking related User Components, Public Services and Core Components according to the templates provided in Annex A and B.

- **Annex E** (TaToo-D313e, 2012) explains the scope of the TaToo Framework Architecture and discusses in detail the architectural design process, its objectives, and supporting elements along with the considerations and decisions that resulted in the pursued approach. It furthermore provides a comprehensive overview on the design decisions taken by the TaToo Consortium while specifying the TaToo Framework Architecture.
- **Annex F** (TaToo-D313f, 2012) contains the functional descriptions of the components identified during the first, second and third iteration of the TaToo Framework Architecture specification. Component descriptions provide a brief overview on the purpose and capabilities of a certain component and act as the basis for functional specifications (Annex C and D).

## 1.4. Updates in V3

The major achievements and results of the third iteration of the TaToo Framework Architecture are listed in the following table:

| V3 Achievements | New and updated |
|---|---|
| Overall reduction of the size of the deliverable to provide a more focused view on the TaToo Framework Architecture. | all |
| Provision of a comprehensive overview on the design decisions taken by the TaToo Consortium. | Annex E |
| Short summary of the architectural approach followed in TaToo and introduction of Annex E which provides the updated and complete description of this approach. | 3<br>Annex E |
| Significant reduction of section 4 and reduction of the Information Retrieval state-of-the-art description. The NeOn methodology just mentioned and referenced. The choice of the hybrid ontology model summarised. | 4, 4.1 |
| Improvement of the discovery description and more details added on the ranking, expansion and limiting algorithms. | 4.2 |
| Improved description of the tagging model applied by TaToo and removal of general and informal concepts. | 4.3 |
| Introduction of Linked Data principle to provide the possibility to link resources by establishing explicit typed links between them and specification of the new related components. | 4.5, 6.3.10, 6.3.16<br>Annex D & E |

| V3 Achievements | New and updated |
|---|---|
| Update of the specification of the Cross Tiers and issues of the of the architecture and removal of obsolete concepts which are not considered extensively in the reference implementation. | 5 |
| Improved the concept and description of multilingualism. | 5.2 |
| Clarified the concept of system administration. | 5.3 |
| Revision of the Resource Harvester and the Harvester Connectors description as a general one to adhere to the general plug-in mechanism to add new connectors for resource harvesting. | 5.4, 6.3.5 |
| Update of the Functional Purview of the TaToo Framework Architecture and inclusion of new V3 components (Linking Components, Ontology Manager) | 6 |
| Translation of new and updated requirements gathered during the third iteration requirements analysis process to additional system functionalities which are reflected in the new and updated component specifications. | 6.3<br>Annex C, D and E |
| Introduction of the Geotagging, the Advanced Tagging and the Tags Editing Portlet. Update of the Tagging Service, RDF Tagger Component and the Filtering Component. | 6.3.4, 6.3.7, 6.3.13<br>Annex D and E |
| Update of the Search Tools, the Discovery Service and the Discovery Core Components | 6.3.8, 6.3.3, 6.3.14<br>Annex C and E |
| Update of the User Context Manager component as providing functionality for the user context. | 6.3.20<br>Annex E |
| Updated the explanation and the sequence diagrams of the tagging, discovery, evaluation and security use cases. | 6.4.1, 6.4.2, 6.4.3, 6.4.4, 6.4.5 |
| Introduction of the Ontology Manager Service. | 6.3.11<br>Annex E |
| Introduction of Annex F which contains the updated and improved component descriptions. | Annex F |

**Table 1.1: Updates in V3**

# 2. Abbreviations and Glossary

This chapter lists the abbreviations and acronyms used throughout the document and provides definitions of important terms that are required for the understanding of this document.

## 2.1. Abbreviations and acronyms

| | |
|---|---|
| **API** | Application Program Interface |
| **AuthN** | Authentication |
| **AuthZ** | Authorization |
| **BB** | Building Block |
| **DO** | Domain Ontology |
| **DoW** | Description of Work |
| **FAO** | United Nations Food and Agriculture Organization |
| **FOAF** | Friend of a Friend |
| **GA** | General Assembly |
| **GENESI-DEC** | Ground European Network for Earth Science Interoperations – Digital Earth Community |
| **GRDDL** | Gleaning Resource Descriptions from Dialects of Languages |
| **GUI** | Graphical User Interface |
| **HTML** | HyperText Markup Language |
| **HTTP** | Hyper Text Transfer Protocol |
| **ICT** | Information and Communication Technology |
| **IR** | Information Retrieval |
| **ISO** | International Standardisation Organisation |
| **JSR** | Java Specification Request |
| **MERM** | Minimum Environmental Resource Model |
| **NUTS** | Nomenclature of Territorial Units for Statistics |
| **OAI** | Open Archive Initiatives |
| **OASIS** | Organization for the Advancement of Structured Information Standards |
| **OGC** | Open Geospatial Consortium |
| **ORCHESTRA** | Open Architecture and Spatial Data Infrastructure for Risk Management |
| **OWL** | Web Ontology Language |
| **OWL-DL** | Web Ontology Language – Description Logics |
| **SSO** | Single Sign-On |
| **PCO** | Project Control Officer |

| | |
|---|---|
| **QA** | Quality Assurance |
| **QoS** | Quality of Service |
| **RDB** | Relational Data Base |
| **RDF** | Resource Description Framework / Resource Description Format |
| **RDFa** | Resource Description Framework – in – attributes |
| **RDFS** | Resource Description Framework Schema |
| **RDQL** | RDF Data Query Language |
| **REST** | Representational State Transfer |
| **RFC** | Request for Comments |
| **RM-OA** | Reference Model for the ORCHESTRA Architecture |
| **RQL** | RDF Query Language |
| **RSS** | Really Simple Syndication |
| **SAC** | Semantic Access Control |
| **SANY** | Sensors ANYwhere |
| **SIOC** | Semantically-Interlinked Online Communities |
| **SOA** | Service Oriented Architecture |
| **SOAP** | Simple Object Access Protocol |
| **SPARQL** | Simple Protocol and RDF Query Language |
| **SQL** | Simple Query Language |
| **SSO** | Single Sign-On |
| **UML** | Unified Modelling Language |
| **URI** | Uniform Resource Identifier |
| **URL** | Uniform Resource Locator |
| **W3C** | World Wide Web Consortium |
| **WP** | Work Package |
| **WS** | Web Service |
| **WSDL** | Web Service Description Language |
| **XML** | eXendible Markup Language |
| **XSD** | XML Schema Definition |
| **XSLT** | eXtensible Stylesheet Language Transformations |

## 2.2. Glossary

**Access** (RFC 2828, http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#access)
To interact with a system entity in order to manipulate, use, gain knowledge of, and/or obtain a representation of some or all of a system entity's resources

**Access Control** (SANY-SA, 2009)
Ability to enforce a policy that identifies permissible actions on a particular resource by a particular subject.

**Architectural Layer**
see Tier.

**Authentication** (OASIS SOA-RM, http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf)
Concerns the identity of the participants in an exchange. Authentication refers to the means by which one participant can be assured of the identity of other participants.

**Authorisation** (OASIS SOA-RM, http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf)
Concerns the legitimacy of the interaction. Authorisation refers to the means by which an owner of a resource may be assured that the information and actions that are exchanged are either explicitly or implicitly approved.

**Building Block**
A set of components, often residing in the same Tier, that work together in order to provide a certain functionality of the TaToo System. High Level Building Blocks may span multiple tiers.

**Cross Tier**
Cross Tiers address common aspects of the architecture with influence on components regardless on which horizontal Tier they reside. Examples for such common aspects are administration and security.

**Discovery** (derived from W3C:
http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#discovery)
Act of locating a machine-processable description of a resource that may have been previously unknown and that meets certain functional, informational or qualitative criteria. It involves matching a set of functional and other criteria with a set of resource descriptions.
Despite there is no commonly accepted distinction between search and discovery, in the scope of TaToo, search is the process of expressing the need for information by the user, and discovery is the process performed by the system to retrieve the results.

**Framework** (http://www.opengeospatial.org/resources/?page=glossary)
An information architecture that comprises, in terms of software design, a reusable software template, or skeleton, from which key enabling and supporting services can be selected, configured and integrated with application code.

**Framework Infrastructure Component**
A Framework Infrastructure Component is a component that contributes to the general infrastructure of the TaToo Framework but is neither related to search/discovery nor to tagging and evaluation/validation directly. Furthermore Framework Infrastructure Components are not associated with a single Building Block or Tier but may be member of different Building Blocks. Examples of infrastructure components are Core Components like the Clearinghouse or the Harvester, Public Services like Security Services and User Components like Administration Tools or the TaToo Web Portal.

**Folksonomy** (Collins English Dictionary - Complete & Unabridged 10th Edition)
A system of classification that makes use of terms that occur naturally in the language of users of the system.

**Formal Specification**
A formal specification is the implementation dependent complement of a functional specification considering implementation specific aspects not provided in a functional specification. A formal specification of a service has to provide a machine-processable description of the service's interfaces including operations and parameters types, for example a WSDL (WSDL, 2001) file. Thus it facilitates the development of a client accessing an instance of the service.

**Functional Specification**
A functional specification is a specification focusing on the purpose and the functionalities to be provided through the service's interface or the component's operations or GUI. A functional specification is implementation independent which means that operations are specified on an abstract level not defining specific data types or schemas. It is the basis for a Formal Specification.

**GeoNames** (https://en.wikipedia.org/wiki/Geonames)
GeoNames is a geographical database available and accessible through various Web services. Beyond names of places in various languages, data stored include latitude, longitude, elevation, population, administrative subdivision and postal codes.

**High Level Building Block**
High Level Building Blocks are the topmost Building Blocks of the TaToo System. They encompass the User Components Building Block, the Public Services Building Block and the Core Components Building Block.

**Horizontal Tier**
see Tier.

**Interface** (ISO 19119:2005)
Named set of operations that characterise the behaviour of an entity. The aggregation of operations in an interface, and the definition of the interface, shall be for the purpose of software re-usability. The specification of an interface shall include a static portion that includes definition of the operations. The specification of an interface shall include a dynamic portion that includes any restrictions on the order of invoking the operations.

**Interoperability** (ISO 19119:2005)
Capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units

**Knowledge Base**
see Semantic Repository

**Linked Data** (http://linkeddata.org/faq)
The term Linked Data refers to a set of best practices for publishing and connecting structured data on the Web. Key technologies that support Linked Data are URIs, HTTP and RDF.

**Manageable Component**
A manageable component supports the possibility to dynamically modify it's behaviour, appearance or internal configuration without touching the component implementation itself. Any TaToo component, whether it is a User Component, a Public Service or a Core Component, can be considered a manageable component.

**Meta-Information** (RM-OA, 2007)
Descriptive information about resources in the universe of discourse. Its structure is given by a meta-information model depending on a particular purpose. Note: A resource by itself does not necessarily need meta-information. The need for meta-information arises from additional tasks or a particular purpose (like catalogue organisation), where many different resources (services and data objects) must be handled by common methods and therefore have to have/get common attributes and descriptions (like a location or the classification of a book in a library).

**Microformat** (http://microformats.org/wiki/what-are-microformats)
Microformats are simple conventions for embedding semantics in HTML to enable decentralized development.

**NUTS** (https://en.wikipedia.org/wiki/NUTS)
The Nomenclature of Territorial Units for Statistics or Nomenclature of Units for Territorial Statistic (NUTS) is a geocode standard for referencing the subdivisions of countries for statistical purposes. The standard is developed and regulated by the European Union, and thus only covers the member states of the EU in detail.

**Portal** (http://looselycoupled.com/glossary/portal)
Web access point. A portal consists of web pages that act as a starting point for using the Web or web-based services.

**Portlet** (Java Portlet Specifications: JSR 168 and JSR 286, http://developers.sun.com/portalserver/reference/techart/jsr168/)
Portlets are web-based components managed by portlet containers that supply dynamic content. Portals employ portlets as pluggable user-interface components, a presentation layer, for information systems.

**Search**
see Discovery.

**Search Profile**
The search profile provides a tool-independent description of how to query a RDF graph in an end-user manner.

**Semantic Annotation** (derived from SOA4ALL, http://www.soa4all.eu/glossary.html)
A semantic annotation is additional information that identifies or defines a concept in a semantic model in order to describe a resource.

**Semantic Repository**
A Semantic Repository, also known as triplestore or RDF repository, is a data layer component allowing the storage of RDF triples.

**Service** (ISO/DIS 19119)
A service is a distinct part of the functionality that is provided by an entity through interfaces.

**Tag**
see Semantic Annotation.

**TaToos**
see Semantic Annotation.

**TaToo Component**
Generic term for an abstract or concrete object that provides some functionality. Component is a composite that may occur in the composition of another component. Examples: Tool, Service, Tagging Service, Clearinghouse, and so forth.

**TaToo Core Component**
Internal component realising the core functionality of the TaToo System which is supposed not exposing a public interface.

**TaToo Framework**
see TaToo System.

**TaToo Framework Architecture**
Overall Architecture of the TaToo Framework. Consists of an implementation independent Functional Purview and an Implementation Purview.

**TaToo Functional Purview**
The implementation-independent Functional Purview is the major part of the TaToo Framework Architecture. It provides mainly the descriptions and functional specifications of all TaToo Components and illustrates their interactions with help of a set of Core Use-Cases.

**TaToo Implementation Purview**
The Implementation Purview complements the Functional Purview by implementation and technology specific concerns. It's focus lies on how a certain component is realised in detail and on the formal specification of the component's interfaces and the (semantic) information exchanged.

**TaToo Public Service**
Service exposing the functionality of the TaToo System through a public, well-defined and formally specified interface. Resides on the Service Tier. Example: Tagging Service.

**TaToo Security Frontend**
The TaToo Security Frontend is responsible to provide authentication and authorization functionalities for User Components that want to communicate with secured (access controlled) Public Services.

**TaToo System**
Deployed instances of all TaToo Components acting together as a unit.

**TaToo Tool**
A front-end component generally but not necessarily with a graphical user interface that allows or supports interaction with a human user. It resides on the Presentation Tier and acts as a client for the TaToo Public Services. Examples: the TaToo Portal, a client API library, a TaToo Toolbar.

**TaToo User Component**
Is used as a synonym for TaToo Tool; see TaToo Tool.

**Tier**
Tiers are a way to logically separate general concerns like presentation and business logic.

**Web Service** (http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#webservice)
A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format.

# 3. Introduction

Following the conceptual and architectural work performed in the SANY (SANY-SA, 2009) and ORCHESTRA (RM-OA, 2007) projects and in respect to high-level architectural objectives like implementation independence and the ability to cover multiple technologies, the architectural design process of the TaToo Framework has been divided into two different workpackages: WP3 – **Specification** and WP4 – **Detailed Design and Implementation**.

Likewise the TaToo Framework introduces an implementation independent Functional Purview and an implementation depend Implementation Purview. The Functional Purview defines among others the descriptions and functional specifications of TaToo services and components. The Implementation Purview describes the technologies selected for the implementation and provides a concrete realisation of the TaToo Framework (reference implementation) consisting of software, ontologies (semantic resource model), and documentation as well as implementation specifications.

The Functional Purview is represented by the document at hand while the Implementation Purview is represented by the WP4 - Implementation deliverable D4.1.3 Semantic Framework Implementation Prototype V3 (TaToo-D413, 2012).

The main architectural objectives of the Functional Purview can be summarised as

- to be implementation independent and to be able to cover multiple technologies
- to define the conceptual foundations for TaToo’s Semantic Resource Model
- to define TaToo’s approach to semantic discovery
- to define TaToo’s approach to semantic tagging
- to define the Cross Tiers and issues of the architecture
- to describe TaToo Components, their role and interactions
- to serve as basis for detailed software design and implementation

Thereby the TaToo Framework Architecture is clearly separated into different tiers and divided into Core Components that are “hidden” within the TaToo Framework, and components that are publicly visible (Public Services and User Components) and can be used in external applications. Those publicly visible TaToo Components are designed as server/client applications and use open standards interfaces and data encodings wherever possible.

More information on TaToo’s architectural design process as well as a summary of design decisions concerning the implementation independent part of the architecture can be found in Annex E of this document (TaToo-D313e, 2012).

# 4. TaToo's semantic tagging and discovery concepts

Semantic tagging and discovery are the core functionalities of TaToo.

## 4.1. Usage of ontologies in TaToo

This chapter explains the usage of ontologies in TaToo, including a description of the selected ontology development methodology, and introduces the concept of the Minimum Environmental Resource Model (MERM).

The TaToo project aims at searching for, discovering and tagging environmental resources. In recent years there has been a trend, framed in the context of the Semantic Web, to improve search processes by fostering an intensive use of domain knowledge encoded as ontologies. TaToo aims to capitalise on the principles of formal tagging by investigating the ability to add valuable information in the form of semantic annotations, facilitating future usage and discovery, and kicking off a beneficial cycle of information enrichment.

But why use semantic and not informal annotations? The key reason is that informal annotations are relatively good for human interpretation, but fail to help machines to understand the meaning of and perform more advanced tasks based on those annotations. Annotations should then be formal and shared in the domain. The Semantic Web was first introduced by Berners-Lee (2001) with the aim of providing meaningful web content for machines by bringing formal structure to web resources. Gruber (1993) introduced the concept of an ontology as "a formal explicit specification of a shared conceptualisation." Thus in the Semantic Web ontologies play the role of a formal (machine-understandable) and shared (in a domain) backbone. Ontologies are becoming a clear way to deal with integration and semantic discovery of web resources in the environmental domain.

Ontologies are usually related to resources through annotations stored as meta-information, so semantic search is usually meta-information-based, not content-based. In general, queries on those systems are expressed in query languages suitable for ontologies such as SPARQL and executed against a Knowledge Base, returning a set of ontology instances that satisfy the queries.

One of the core goals of TaToo is that of allowing cross-subdomain discovery of resources in the environmental domain. The environmental domain is in fact composed of several subdomains such as the ones represented in the TaToo use cases. There is no single standard ontology or shared vocabulary that encompasses the entire environmental domain, so the TaToo ontology framework faced the challenge of being capable of dealing with annotations from different ontologies while enabling a certain degree of cross-domain discovery capabilities, and at the same time offering a unified, extensible and relatively simple ontology framework.

TaToo follows a simplified hybrid ontology approach for semantic interoperability elaborated by Wache (2001). The hybrid approach is based on using a common shared ontology and a set of local or application ontologies that are mapped uniquely to the shared vocabulary. In this case the local ontologies would extend the vocabulary to the needs of a given domain and interoperability is achieved based on the shared ontology.

In TaToo, there is also the need to uniformly describe environmental information resources. In the scope of TaToo an e**nvironmental resource is a web resource defined using a URI and a small set of objective properties** (author, creation, etc.), **plus** a set meta-information related to the subjective **annotations about the topics of what the resource is about** in the domain (resource about cancer, about pollutants, about climate, about building construction, etc.) **and the nature of the resource** (the resource is a web page, a web service, a time series, etc.). For this purpose TaToo has defined the **Minimum Environmental Resource Model (MERM),** an ontology containing the resource and annotation model, and the **BRIDGE** ontology that encompasses some common elements (time, geographical elements, etc.) and the set of alignments needed to ensure the cross-domain objective of the TaToo ontology framework. .

TaToo ontologies are based on existing W3C standards, particularly RDF (RDF, 2004), RDFS (RDFS, 2004), and OWL (OWL, 2004). In the ontology engineering process several shared vocabularies have been reused, such as DC (DCMI), FOAF (FOAF), SIOC (SIOC), etc. TaToo does not provide an ontology engineering tool, but instead relies on existing tools for ontology engineering, such as the NeOn Toolkit (NeOn-Toolkit) or Protégé (Protégé). Ontology development is a complex, expensive and time-consuming process that includes different activities, such as specifying requirements, information extraction, logical modelling, etc. In order to efficiently manage this process, it is convenient to use a methodology and its supporting tools. The development of ontologies in TaToo has followed the **NeOn Methodology** (Suárez-Figueroa, 2008), and **adopted the usage of the NeOn Toolkit as the main (but not only) ontology editing tool.** It is worth noticing that the NeOn methodology has been applied in a lightweight way, meaning that some of the activities have been carried out just as a guideline in the ontology development process, but not the whole set of steps and supporting tools or forms have been fully implemented.

## 4.2. Search and discovery in TaToo

This chapter provides a conceptual description of the discovery in TaToo. The TaToo discovery process makes intensive use of domain knowledge (encoded as ontologies) to improve search processes. Ontologies are usually related to resources through annotations stored as meta-information, so TaToo discovery is meta-information-based, not content-based. Queries on TaToo are normally expressed, from the system's point of view, in the SPARQL query language and executed against the TaToo Knowledge Base using the service layer explained in the deliverable.

In general, the process of discovery and / or search can be defined as the retrieval by a system of a set of resources that satisfy a need expressed by a user. Despite the fact that there is no commonly accepted distinction between search and discovery, **in the scope of TaToo, search is defined as the process of expressing the need for information by the user, while discovery is the process performed by the system to retrieve the results.**

Information Retrieval (IR) is the field of computer science that deals with searching for information in resources, searching for resources themselves and searching for metadata which describes resources, across several means.

IR is a multidisciplinary field in which both resources (text, data, video, audio, etc.) and the scenarios (Internet, databases, text corpora, etc.) can be very varied. Thus, throughout this chapter, we describe the approach followed in TaToo for search and discovery.

In Figure 4.1, a typical IR system architecture is presented.

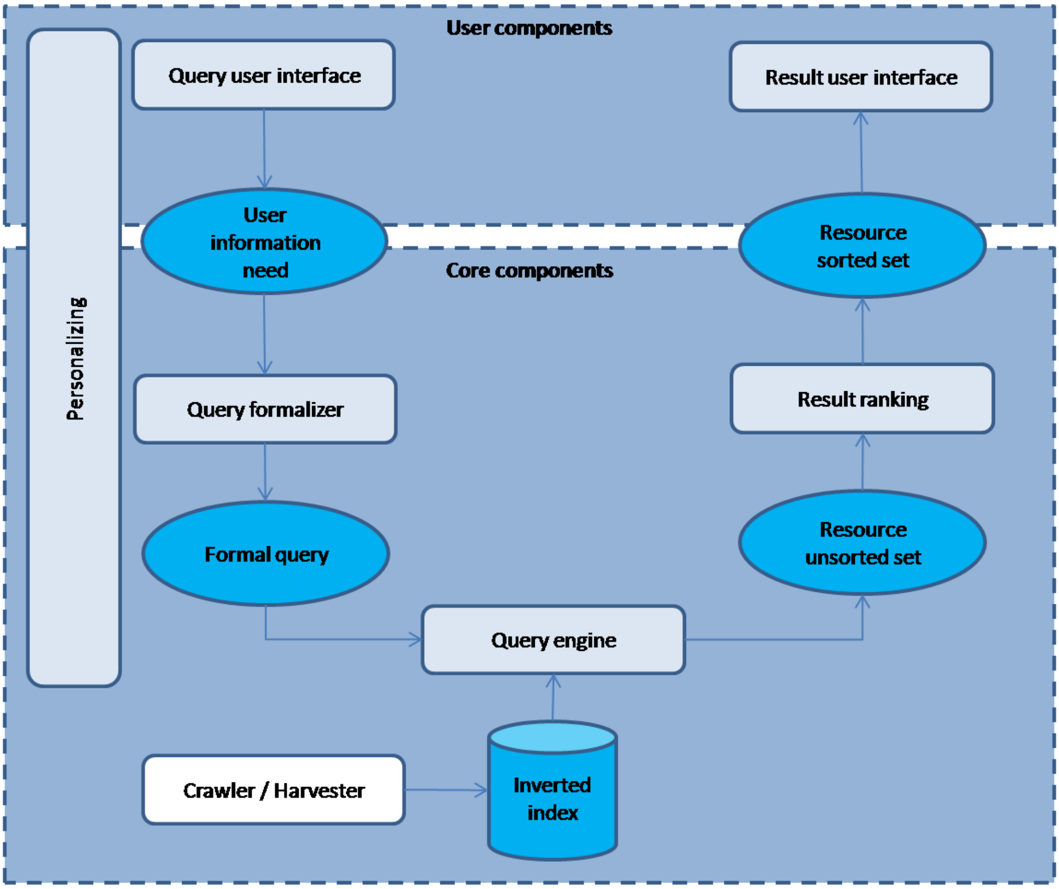

*Figure 4.1: Typical IR system architecture*

The query issued by the client is usually passed to the query formalizer by the query user interface. The query formalizer formalizes the user information need into a formal query executable by a query engine. The query engine executes the formal query against a previously constructed inverted index. The inverted index is populated by the crawler and / or harvester processes or manually by users adding information to the system.

The query engine provides a set of relevant resources as result of the carried out formal query. The relevant resources go through the process of result ranking, which organizes the retrieved resources in order of relevance. Finally, sorted resources are returned to the results user interface. Throughout all these processes, different techniques can be used for personalization, which aim to improve search results based on user preferences and context.

This generic discovery process has been harmonized to adopt the principles of the TaToo architecture, resulting in the diagram shown in Figure 4.2.

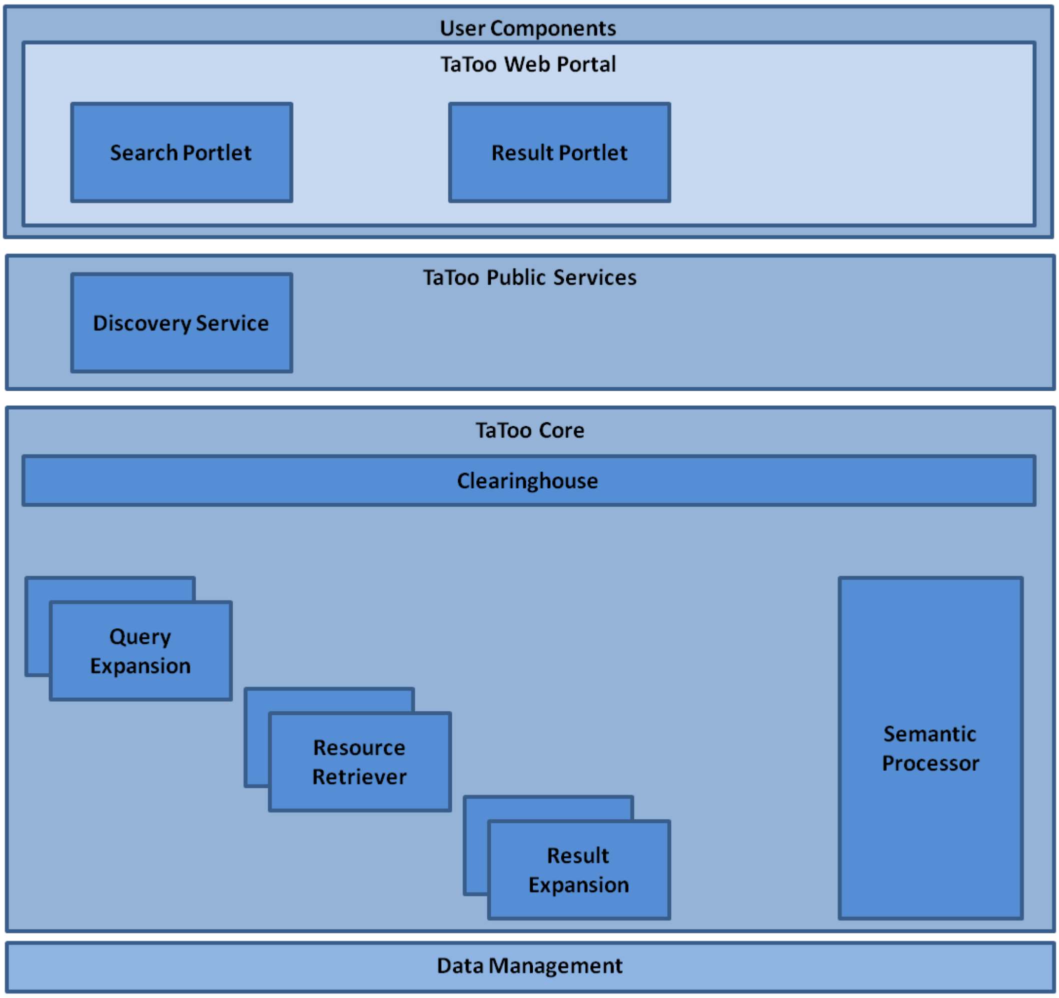

*Figure 4.2: TaToo discovery architecture*

As can be seen in Figure 4.2 search and discovery components take part in the three architecture Building Blocks: User Components, TaToo Public Services, and TaToo Core Components.

The reference implementation of the User Components providing access to the system is a web portal, specified, designed, and implemented as a set of portlets. Figure 4.2 presents the discovery portlets

- Search portlets allow users to query the TaToo System in different ways.
- Results portlets show the user the relevant resource annotations for a given query.

The service oriented architecture adopted in TaToo is extensible and allows the future development of different discovery user interfaces making use of the TaToo Discovery Services. TaToo Public Services provide public access to TaToo functionality.

The TaToo Core Components provide the business functionality of TaToo. Three Core Components related to discovery have been identified:

- The **Query Expansion** (QE) component, which is responsible for transforming the user's information need into a set of SPARQL queries executable by the Semantic Processor. TaToo provides a reference implementation of the QE

- The **Resource Retriever** (RR) component, which is responsible to retrieve more meta-information about the resources and related annotations in order to rank and limit the results.

- The **Results Expansion** (RE) component, which is responsible aggregating the results and preparing the final resource objects that become the output of the discovery process.

One important aspect to consider in the discovery process is **how to rank and expand/limit the results of a query**.

In Information Retrieval, ranking is the effort to measure how relevant a result is to a user's information need in order to present it to the user in a meaningful order. In this sense, it is noteworthy that the ranking process does not add or remove results from a given query, but only allows sorting the results to present them in a better way to the user. Therefore, the ranking phase in TaToo takes place after query execution and before returning the results to the public services.

**Ranking** is therefore about measuring how relevant a resource is. There are several ways to calculate this relevance measure: relevant to the user, relevant to the context, relevant to a query, relevant to a domain, etc. Several relevance measures can be merged (even using different weights for doing so) in order to provide a more reliable relevance measure. In order to do so, one can adopt a vector space model, so every relevance measure is a real number between 0 and 1, and the merging process is a normalized sum of the elements of the vector. In the same way, weights can be assigned to different elements of the vector by multiplying by a real number between 1 and 0.

In the scope of TaToo, the ranking model followed allows the implementation of several relevance measures. At least one implementation (relevance to the query) will be provided as reference implementation. There are two clear ways of ranking resources in TaToo:

1. Ranking based on the relevance of the annotations to the query: The Query Expansion component translates the user information needs into SPARQL queries that are executed towards a semantic repository. SPARQL queries are Boolean queries, as a resource either belongs to the result set or not. On the other hand, TaToo queries are executed towards subjective annotations, provided by several users, so it is possible that some annotations from a resource are relevant to the query while the rest aren't. In this sense, a measure of the relevance to a given query makes sense. Therefore, the relevance to the query measure calculates how important the set of annotations belonging to a resource is, compared to the total set of annotations.

2. Ranking based on the evaluations: The resources and the annotations in TaToo can be evaluated. Therefore the system might retrieve an evaluation measure for each of the resources and/or annotations in the result set and rank them according to that criteria.

**Expansion** is the process of expanding the terms of a query in order to retrieve not only results that matches with the original query, but also to an extended set of elements related more or less loosely with the original query. This process is usually done when no results matches the original query or the result set is not big enough, and can be triggered either automatically or after an iteration with the client.

**Expansion** is the process of expanding the terms of a query in order to retrieve not only results that matches with the original query, but also to an extended set of elements related more or less loosely with the original query. This process is usually done when no results match the original query or the result set is not big enough, and can be triggered either automatically or after an iteration with the client.

In TaToo, the expansion of the query is done within the Query Expansion component. It is worth noticing that the use of ontologies and semantic inference within the TaToo Knowledge Base already produce an expansion by inferring implicit knowledge. Nevertheless, the cross-domain alignments existing between the Domain Ontologies based in the use of SKOS properties allow expansion of the queries when the query topics are related to other topics. This process will only be triggered when the list of resources retrieved by the first query is not too long; 100 resources is the arbitrary limit imposed for this type of trigger.

**Limiting** the resources is the opposite of expanding the query results. The TaToo discovery services allow the specification of the maximum number of resources to be retrieved by a query. The reasons for limiting are basically related to the performance of the discovery process and the adequacy of the results. Ranking in addition to limiting the resources allows the system to present to the user a manageable set of relevant results, ruling out the less relevant and therefore focusing the search.

The discovery strategy in TaToo consists of using an intelligent mixture of ranking/expansion/limiting algorithms to get the most relevant resources without overloading the user with irrelevant results and without penalizing the response time. Nevertheless the SPARQL endpoint service allows one to query the TaToo Knowledge Base with no specific limitations, so the client will always be able to express complex queries that retrieve as many resources as they like.

More information about the Discovery Core Components can be found in section 6.3 and the orchestration of the components within the discovery process is explained in section 6.4.2.

## 4.3. Tagging in TaToo

In the Web context, the term ‘tagging’ concerns the attachment of user-defined labels, tags or descriptive keywords, to resources in order to describe, organize and share these resources. Tagging democratizes the process of classification as tags are not exclusively provided by a central authority, normally owning the resources.

To allow a semantically enhanced resource discovery process, the tagging process has to be performed restricting the set of possible tags to those belonging to the identified Domain Ontologies. These concepts are defined conceptualising the relevant domains of interest and used by tagging expert users (also referred as *taggers*) in those domains.

The process of **Tagging, in the scope of TaToo, is defined as the process of associating structured meta-information based on the MERM model with an environmental resource** either provided by the end users, or already stored in the TaToo Knowledge Base repository resulting from a search process.

Tagging is strictly bound to TaToo Search and Discovery strategies, so to improve the accuracy of results, the taggers will be allowed only to use concepts from shared Domain Ontologies. This approach is called closed tagging, in which the taggers are forced to use a specific set of tags.

TaToo Ontology engineering is based on the adoption of an upper ontology, namely Bridge Ontology, that is composed from already known and available ontologies in the semantic community (SKOS, DC, GeoNames, NUTS, FOAF, W3C Annotations, and so on), and from MERM, that is used internally to the TaToo Framework as the application ontology.

When the end user performs a Tagging operation in TaToo, he is providing a subject that is an URI uniquely identifying a resource, a predicate providing an ontology defined property and finally an object, that is an ontology class individual from a domain ontology provided on the front end. These three elements provided by the tagger are composing RDF triples that describe the resource. The presented triples must moreover adhere to the model defined in MERM. This means that together with the user provided data, other meta-information and RDF triples are stored, either bound to the Resource model or to the Annotation model.

As mentioned above, the MERM Ontology defines two different upper classes that influence the Tagging chain in TaToo:

- **Resource**, that is the class identifying an environmental resource stored in the TaToo Knowledge Base;

- **Annotation**, that is the class defining all the meta-information associated to the environmental resource.

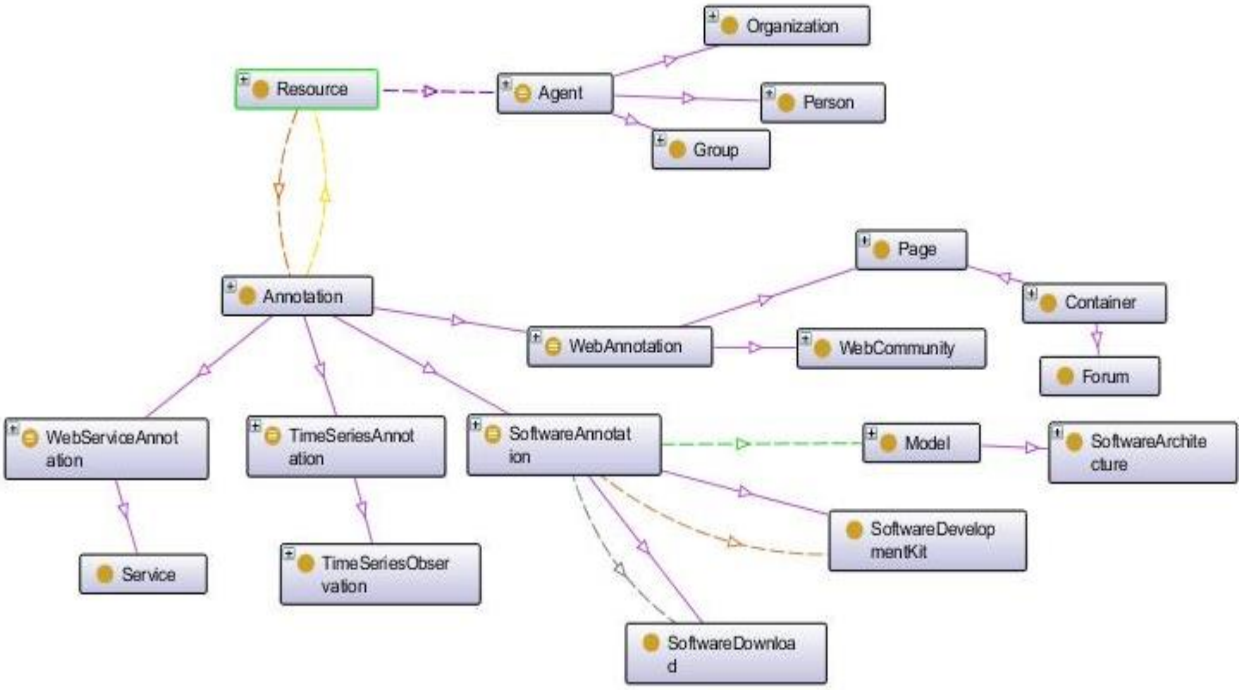

*Figure 4.3: TaToo Tagging MERM Model*

Figure 4.3: presents the MERM Structure and relationships between the merm:Resource ontology class and the *merm:Annotation* class engineered for tagging purpose in TaToo. As shown in the graph ontology representation, a Resource is bound with different relationships to two classes, besides the Annotation class also to the FOAF:Agent class that is provided by the FOAF (Friend of a Friend) Ontology.

The Annotation class is composed of different subclasses that are used to distinguish between different types of resources and different types of tagging allowed in TaToo. This approach enables a deep tagging, meaning the ability to perform complex tagging tackling particular needs, such as Web Annotations, Web Service Annotations, Timeseries Annotations, and so on.

The Annotation class presents the following subclasses:

- **WebServiceAnnotation**: used to identify any type of Web Service annotation;
- **TimeSeriesAnnotation**: used to identify any kind of TimeSeries annotations;
- **SoftwareAnnotation**: used to identify any type of Software related meta-information;
- **WebAnnotation**: used to annotate any Web resource, a Web Page a Forum Page, and so on.

The presented Annotation class structure makes it possible to plug in new subclasses defining new types of Annotations, so that TaToo can extend tagging to address new environmental resources.

In the presented MERM Model, the Resource class is also bound to a FOAF:Agent class that defines the user that is performing the Tagging operation, namely Provider in TaToo. A resource or annotation provider can be either a FOAF:Person or FOAF:Group of FOAF:Organization.

In Figure 4.4: below different ontology properties, either provided in MERM, in the Bridge Ontology or in the OWL language, are defining the relationships between Resource, Annotation and Agent.

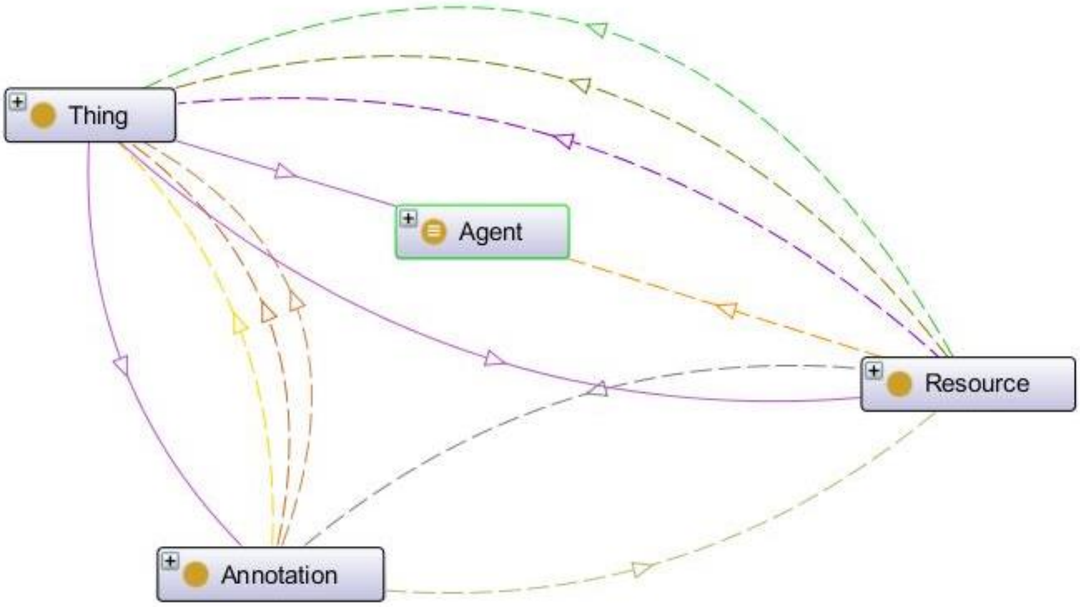

*Figure 4.4: Relationships between Resource, Annotation, Agent*

In particular, Resource and Annotation have a bijective relationship that is defined based on two MERM Ontology properties: *merm:hasAnnotation*, from Resource to Annotation, and *merm:annotates*, from Annotation to Resource. The relationship between Resource and Agent is defined using the property *merm:publisher*. Moreover MERM defines new properties for Resource meta-information: *datePublished*, *dateResourceProvided*, *has_resource_provider*, while for Annotation meta-information: *dateAnnotated*, has*_annotation_provider* and *subject*.

The presented TaToo tagging structure also affects predicate and object provided by the user in the context of their domain, thus selecting ontology properties and concepts individuals from their domain. To be consistent in the Knowledge Base repository, the predicate and the object provided by the end users have to be ontology classes aligned with the Bridge Ontology. The tagger will be able to use a predicate that is either an ontology property specified in MERM or in a Domain Ontology, given that this property is a sub-property of *merm:Subject*. Concerning the object of the triple statement provided by the user, an individual of a domain ontology class, to adhere to TaToo tagging and to be discoverable at a later time, the individual must be an instance of an ontology class that is subclass or *sameAs* of the Bridge Topic class.

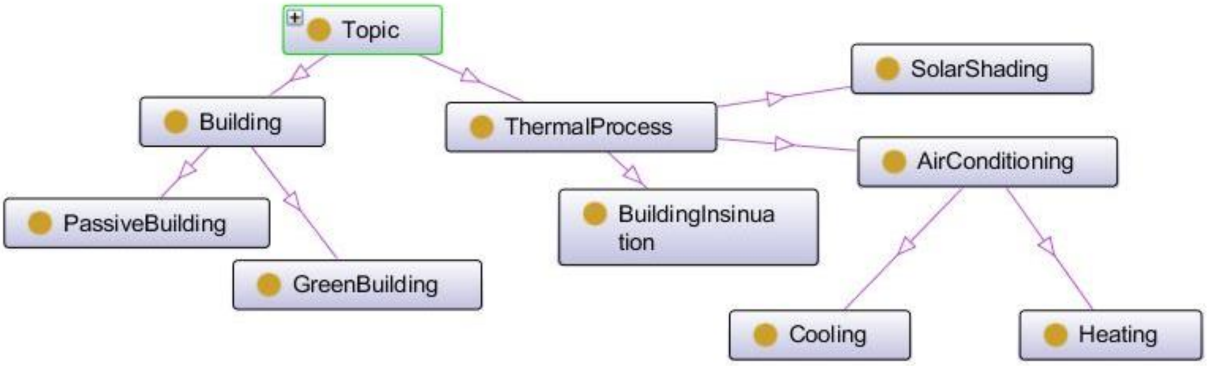

*Figure 4.5: Domain Ontology tagging in TaToo*

Figure 4.5: presents an excerpt of a Domain Ontology alignment, in this case taken from the AIT domain. Based on the described TaToo tagging model, the user will be able to annotate resources using any instance of the classes Building, GreenBuilding and so on.

To properly drive the Tagging process within the TaToo Framework, the end user will be prompted on the TaToo User Interface with only the right properties / predicates and values / objects that are adhering to the alignment with the Bridge Ontology.

## 4.4. Evaluation of resources and annotations

One of the main issues regarding the current Web is how to assess the quality of information resources. Numerous guides and approaches regarding this subject have been proposed (Smith, 1997; Vaughan, 2006; Phillips, 2004). A resource quality can be evaluated with respect to a number of evaluation criteria, each of which is determined by a number of indicators. An example evaluation criterion, which is quite often applied, is the resource currency and completeness. The corresponding indicators for this criterion include the following questions:

1) When was the resource created?

2) When was it last updated, and is this done regularly?

3) Does the resource evolve over time?

4) Is the resource's information complete?

In order to provide a comprehensive evaluation of the managed environmental resources, **the TaToo system enables the evaluation of not only the actual resources but also their annotations**. In other words, TaToo considers both environmental resources and resource annotations as information entities that can be evaluated. Therefore we distinguish between two main types of TaToo evaluations:

1) Resource Evaluation, and

2) Annotation Evaluation.

Regardless of the evaluation type (i.e., resource evaluation, annotation evaluation), each TaToo evaluation is uniquely identified by the evaluation URI and is represented by a set of RDF tags that are stored in the TaToo RDF repository. Figure 4.6 illustrates a part of the MERM ontology that specifies the TaToo evaluation schema. As we can see from the figure, TaToo evaluations are characterised by the evaluation criteria, evaluation metric and evaluation value. Moreover, each evaluation is also characterised by the evaluator whose profile information is modelled within the TaToo user profile. In addition, the resource evaluations are linked to the instances of the *merm:resourceInfo* class while the annotation evaluations and the tag evaluations are linked to the instances of the *merm:Annotation* class.

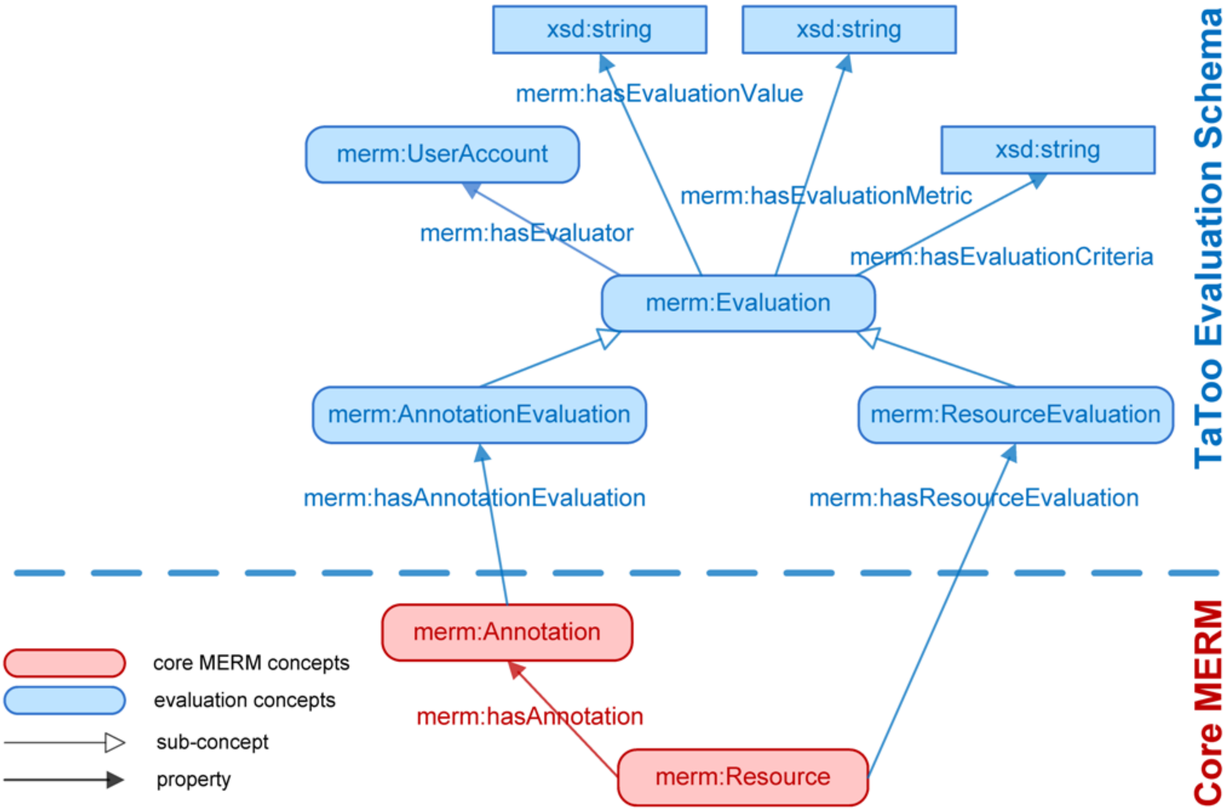

*Figure 4.6: A part of the MERM ontology providing the TaToo evaluation schema*

A usage of the TaToo evaluations is within the TaToo search and discovery process. The TaToo system can use them in a process of filtering and ranking of search results (e.g., ranking the results based on their evaluations for a given evaluation criterion). Moreover, the resource evaluations help TaToo users decide what resources fit the best to their needs.

In section 6.4.4 we present the two use cases, each of which corresponds to one of the two types of the TaToo evaluations. Both use-cases start by a user browsing resources from the list of retrieved search results.

## 4.5. Linked Data principles in TaToo

The term Linked Data refers to a set of best practices for publishing and interlinking data, widely known as the Linked Data principles:

1. Use URI as names for things.

2. Use HTTP URIs, so that people can look up those names.

3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL).

4. Include links to other URIs, so that they can discover more things.

The RDF data model is introduced to be the underlying data representation model for the Linked data. Two principal types of RDF triples are *Literal Triples* and *RDF Links*. While the *Literal Triples* are used to describe the properties of resources, the *RDF Links* describe the relationship between two resources.

TaToo resources, that is, resources that are annotated and evaluated by the TaToo approach, are uniquely identified and can be easily linked according to the Linked Data principles. In addition to the TaToo annotation and evaluation approaches, the TaToo Linked Data approach enhances the overall TaToo tagging and discovery approach by adding a possibility to link resources by establishing explicit typed links (i.e., RDF links) between them.

Considering that the TaToo approach has already provided formalisms (i.e., MERM) and mechanisms (i.e., tagging operations) for handling and storing resource URIs, the key issue of the introduction of the TaToo Linked Data approach is the selection of the link types that will be available for linking TaToo resources. The link types are formally defined as ontological object type properties.

With respect to the link types that will be used to link resources the TaToo system is going to support a predefined set of the link types (i.e., properties). That set of link types will include both upper-level link types and domain-level link types. The upper-level link types will include object properties for standardised or well-known upper level ontologies. Some candidates for the TaToo upper-level link types include:

1. `rdfs:seeAlso` - A related resource that provides further information about the subject resource.

2. `rdfs:isDefinedBy` - A related resource that represents a definition of the subject resource.

3. `rdfs:member` - A related resource that is a member of the subject resource.

4. `owl:sameAs` - URI references of the subject and object resources refer to the same thing.

5. `owl:differentFrom` - Subject and object resources are different.

6. `dcterms:hasPart` - A related resource that is included either physically or logically in the subject resource.

7. `dcterms:isPartOf` - A related resource in which the subject resource is physically or logically included.

8. `dcterms:isReferencedBy` - A related resource that references, cites, or otherwise points to the subject resource.

9. `dcterms:isReplacedBy` - A related resource that supplants, displaces, or supersedes the subject resource.

10. `dcterms:isRequiredBy` - A related resource that requires the subject resource to support its function, delivery, or coherence.

Besides the upper-level link types, the predefined set of the TaToo link types will contain some domain-level link types which will be selected from the TaToo Domain Ontologies (i.e., MU, AIT, and JRC).

# 5. Cross tiers and cross issues of the architecture

The so-called cross tiers address common aspects of the architecture that have an influence on components regardless of on which horizontal tier they reside. They can be seen as tiers in the architecture and drawn vertically spanning the space taken by the other horizontal tiers. Cross issues refers to more than one tier in the architecture but they cannot really be considered as tiers from an architectural point of view.

## 5.1. Security

The Security tier is one of the Cross Tiers augmenting the core TaToo Framework Architecture. In the TaToo project the role of the security tier is related to the management of TaToo users and different user communities. Moreover, the Security Tier is directly involved in the process of identifying owners of tags, in determining the privileges for editing and deletion of existing tags as well as in the access control of tools provided by TaToo.

The general objective of the Security tier is to support a user friendly system on the Presentation tier and adopt a solid security framework on the Service Tier level, which deals with user identity provision and Authentication and Authorization mechanisms.

Authentication and Authorization represent two different tasks involved in regulating access to TaToo components. In TaToo they represent two different levels of the security tier. The Authentication level provides the identification of subjects operating in the system. **In TaToo the Authentication** adopts **the Single Sign-On**, which is an emerging security concept in favour of the end user. The Authorization level, once the user is identified, is responsible for checking the user identity and / or user attributes against defined policies in the TaToo Security Front-end as a basis for an access control decision.

The TaToo project aims to deliver a widely adopted framework from different domains, leading to heterogeneous types of users accessing the framework. To regulate the variety of users accessing the framework, domains or communities will have to be bridged. In particular to be bridged together, the communities have to “trust” each other. The concept of trust in the security knowledge indicates “the willingness of a party to be vulnerable to the actions of another party based on the expectation that the other will perform a particular action important to the truster, irrespective of the ability to monitor or control that other party” (Mayer, 1995). Thus in TaToo the trust between different domains will be used to build a confederated organization. In TaToo these bridged communities will be called Federations (Federations). In particular a Federation is a collection of collaborating organizations that use a common framework for exchanging identity attributes, supporting security processes and infrastructures, and providing overarching policy governing the interaction.

### 5.1.1 Single Sign-On

The Single Sign-On (SSO) security concept has raised interest in the context of informatics security applied to Web Sites, Web Portals and in general Web based Graphical User Interfaces. The functionality provided from the Single Sign-On consists in minimizing the number of times that the user is required to authenticate when accessing a resource. In particular, SSO ensures that the users have to authenticate only once when accessing the Web resource, granting access to a set of potentially large number of resources.

These characteristics have increased the adoption of the SSO concept when implementing security frameworks for Web applications. The TaToo Framework is then extended by using this concept in the TaToo Web Portal and in the communications between the Presentation Tier and the Public Service Tier, applying SSO Authentication to the Tagging, Discovery and Evaluate /Validate Web Services. Finally, the single signed on session is replicated on the Web Service client.

**The Single Sign-On** identifies **the user through a credential based on the Security Assertion Markup Language (SAML) standard.** The Single Sign-On architecture also involves a third actor during the authentication, the Identity Provider. The Identity Provider in the SSO architecture is responsible for introducing the trust in the authentication mechanism. In particular there are a number of service providers, the TaToo Public Services, that can make use of assertions about a subject in order to control access and provide customized service, and accordingly they become the relying parties of an asserting party, which is the Identity Provider.

The user token identifier containing this information in TaToo will be translated in the Knowledge Base using a particular ontology, called FOAF. “FOAF is devoted to linking people and information using the Web. Regardless of whether information is in people's heads, in physical or digital documents, or in the form of factual data, it can be linked. FOAF integrates three kinds of network: *social networks* of human collaboration, friendship and association; *representational networks* that describe a simplified view of a cartoon universe in factual terms; and *information networks* that use Web-based linking to share independently published descriptions of this inter-connected world.” (FOAF).

The FOAF mapping is used within the TaToo semantic framework as well, outside of the security scope and with no means of authorization, but to keep track of information such as author of the annotation, or executor of the discovery.

## 5.2. Multilinguality

Multilingualism from the point of view of TaToo consists in providing to the end users the chance to use the platform not only in English but also with other languages. Multilingualism was not considered in V1 and was implemented only partially in V2.

As a research project, TaToo is aiming to have a proof of concept of the multilinguality. Therefore the set of languages provided by the system will be limited to the ones agreed upon by the project partners, avoiding an excessive localization effort. Nevertheless, the system will provide the means or methodology to include new languages.

Multilingualism enhancement creates multiple requirements in different layers of the TaToo platform, namely at the ontology, service, core components and user interface levels.

### 5.2.1 Ontology level

One of the major problem for multilingualism is to ensure the continuity of the multilinguality across all ontology terms and instances. There are lots of entities in ontologies that ideally should be multilingual in the TaToo ontology framework. Enhancing ontologies with localization is the first major task in this process. This task is also a hot topic in the research community. The United Nations Food and Agriculture Organization (FAO) especially worked on multilingualism with different languages and different alphabets and précised certain requirements for multilingual ontologies [Multilingual1, 2007]. Some of them are:

- Represent lexicalizations in several languages that refer to the same object; for example ‘cow’ (English), ‘vache’ (French), ‘vacca’ (Italian) and ‘mucca’ (Italian) all refer to the same animal;
- specify ISO standard codes for languages and countries for lexical items;
- Represent relationships between lexical items, within and across languages. These relationships include synonymy (e.g. ‘vacca’ and ‘mucca’ are synonyms for cattle), acronyms (‘BSE’ stands for the English ‘Bovine spongiform encephalopathy’, but ‘ESB’ stands for the French name of the same disease, ‘Encéphalopathie spongiforme bovine’), spelling variances, different types of names (e.g. ‘Gabonese Republic’ is the official long name, and ‘Gabon’ is the official short name of the same country);
- Account for the use of the same terms in different contexts, that are translated by different terms in other languages (e.g. ‘benzene’ is considered a pollutant in a specific area, and a fuel in another sphere);
- Use UTF-8 as default character encoding.

These requirements are consistent with the TaToo requirements and OWL supports such representations. RDFSchema metadata contain methods such as “rdfs:comment” or “rdfs:label” which are usable for multilingual metadata storage. Also for a richer representation SKOS vocabulary can be chosen. Finally there are defined modules for multilingualism such as LIR model (Multilingual2, 2008).

**The decision made in TaToo is to use the RDFSchema metadata rdfs:label to add multilingual metadata storage in the TaToo Ontologies (including Domain Ontologies).**

Therefore, an automatic method to generate the required labels in several target languages would be desired. In this direction, The NeOn Tookit offers a specific plug-in for the LIR model, but this model unfortunately does not apply to “rdfs:label”.

The TaToo platform's information space is handled with W3C recommended semantic technologies Web Ontology Language (OWL) and Resource Description Format (RDF) encoded with XML (XML, 2006). In OWL and RDF the ontology engineer is able to specify the language of the literal format that is linked to any entity. An example can be seen below:

```
<nsExample:InstanceExample>
        <nsExample:datatypeExample xml:lang="en">Literal</nsExample:datatypeExample>
</nsExample:InstanceExample>
```

The example above signifies that the example instance "ns:Example:InstanceExample" is linked with the "Literal" via the example datatype "nsExample:datatypeExample" and the language is English (stated with the attribute xml:lang="en"). xml:lang attribute is generic for XML documents and sets as language identifiers in RFC 4646 format (W3C, 2008).

In the TaToo Ontological Framework some of the Dublin Core elements vocabulary is imported for describing and giving titles to resources. But these representations are handled by TaToo tagging and discovery services. On the other hand Discovery and Ontology Manager Services are configured to be interoperable with multilingual labels for each language.

Labels are simple tags in ontologies. This can be done with the "rdfs:label" property. Any URI (class, property or instance) can be tagged with a label in OWL+RDF/XML format.

Some specific classes and properties should have labels in desired languages in order to be interoperable with TaToo platform. These are:

- Annotation types (subclasses of "merm:Annotation")
- Subject types (subproperties of "merm:Subject")
- Topic types (subclasses of bridge: "bridge:Topic")

Therefore any domain ontology that is going to enter under the hierarchy of one of these terms must be tagged with at least a label in the English language. In the scope of the v3, some of the Domain Ontologies will also cope with labels in other languages, such as German, Italian, or Czech.

Below is an example of an OWL class ("exampleClass") using English and Spanish labels:

```
<owl:Class>
        <rdf:resource>#exampleClass</rdf:resource>
        <rdfs:label xml:lang="en">example class</rdfs:label>
        <rdfs:label xml:lang="es">clase de ejemplo</rdfs:label>
</owl:Class>
```

### 5.2.2 Service level

The TaToo services operations should provide a way to identify the target language in which the client wants to retrieve/input meta-information in the signature of most of the operations. This information will be used by the core components.

**Each Service operation handling multilingual information from ontologies has an additional parameter which identifies the language (locale) of the user and is used to retrieve literals in the specified language.**

### 5.2.3 Core components level

The TaToo core components have to take into account the target language when retrieving/adding information from the TaToo repositories. However, not all the meta-information would be available in the desired language. In this case, the system will always try first to retrieve the labels in the target language, and if it is not present in the default language (English or the Id of the desired element).

### 5.2.4 User interface level

The reference implementation of the TaToo user interface (the TaToo Portal and portlets) is multilingual by nature with respect to handling the metadata stored in the TaToo Knowledge Base. Interaction with the TaToo services should be done using the methods provided by the service layer specifying the target language as the desired language of the user.

The rest of the user components in the TaToo Portal or other clients, however, are not meant to be multilingual, as they are reference implementations. Therefore, although multilingualism is recommended for all user-related components it is not required.

## 5.3. System administration

System administration is a cross cutting issue and it mainly encompasses in the context of TaToo the administration of the TaToo Components (Manageable Components). Other tasks common to system administration, such as the administration of deployed service instances, the management of ontologies, user and policy management, administration of the Knowledge Base, etc. can only be performed by system administrators that have direct access to the underlying infrastructure of the TaToo System. Those tasks are usually carried out with the help of tools provided by the operation system, web server, servlet container, etc. and are therefore not considered in TaToo’s system administration approach which is based on the concept of Manageable Components.

Any TaToo component, whether it is a User Component, a Public Service or a Core Component, can be considered a manageable component if it provides the possibility to dynamically modify its behaviour, appearance or internal configuration without touching the component implementation itself. A TaToo Public Service, for example, should allow a system administrator to specify the URL of a corresponding Clearinghouse instance(s) by editing a configuration file rather than changing the URL in the implementation directly.

Due to the implementation independent nature of the TaToo Framework Architecture, where the same component can be implemented using different approaches, programming languages, libraries, tools and products, etc., each featuring a vast amount of possible configuration options and methods, manageable components do not specify a dedicated configuration interface. Instead we follow a generic approach for the configuration of manageable components in TaToo that is independent of the actual configuration methods supported by the component implementations.

Therefore we introduce an Administration Service and a corresponding Administration Portlet (see section ), which provide simple but nevertheless effective methods to equip components that do not provide themselves any management interfaces, with remote administration capabilities supported by a GUI. The Administration Portlet in conjunction with the Administration Service can be used by the system administrator to conveniently update the configuration of manageable components that are supported by the implementation of the service and the portlet.

## 5.4. Harvesting

As already stated, meta-information can be provided by actual resource owners or by users through tagging (information enrichment process) while accessing, visualising, and taking advantage of discovered resources. This has some major advantages:

- Meta-information is valuable as provided by the resource owner, who best knows about resources he is publishing;
- Users have information, from their domain of interest, which is not yet available while resources are published, and not obvious for resource owners.

In general, resource owners can directly provide meta-information acting as general users (using TaToo tools for doing it), or meta-information can be collected by harvesting. Harvesting is the process of collecting meta-information associated to a resource. Resources and associated meta-information are then made available to users through the semantic discovery processes at a second time.

While harvesting, sources of meta-information are accessed and meta-information associated with resources is stored in the TaToo Knowledge Base (Semantic Repository). The harvesting process can take place at the beginning (when the TaToo system is deployed) to create an initial set of meta-information, once in a while (possibly based on configurable policies), and/or triggered by the Clearinghouse to get updated content when considered advisable.

Several sources of resource meta-information can be taken into account: catalogues, Web services, web pages, and so forth. Considering that sources can be different and new types of sources could be considered in the future, the architecture portion dealing with the harvesting process is based on a central component extendible with a set of plug-ins.

The Resource Harvester is the central entity while the Resource Connectors (in literature also referred as adaptors) are the plug-ins providing ad-hoc harvesting functionality. A connector is foreseen for each type / instance of resource to be harvested. Seeing that Resource Connectors are plug-ins to the Resource Harvester, adding a new resource type / instance to the system is a matter of implementing a new Resource Connector plug-in.

To get meta-information from a catalogue it is necessary to know the format and schema used for storing meta-information in the catalogue in order to realise a component able to convert this format into RDF and to map the catalogue's inherent semantics to TaToo concepts which are concepts of ontologies (Domain Ontologies) known to the TaToo System.

Meta-information can also be collected from Web Services (e.g. realised as RESTful services, or SOAP services) and from Web pages. This is possible in the case that the content is structured somehow and a mapping of the semantics can be made. Normally the content is represented through XML, in that case an XML to RDF converter is required (e.g. GRDDL, or XSLT), or through structured formats such as RDFa (XML based as well).

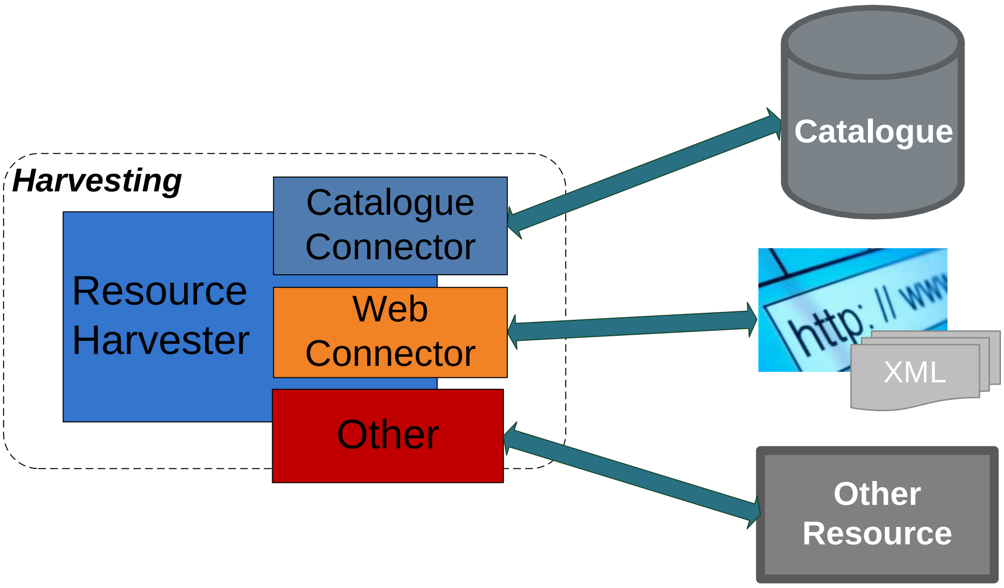

*Figure 5.1: Components for Harvesting Resources*

In Figure 5.1 the harvesting architecture is presented. The figure shows the Resource Harvester and the Catalogue and Web connectors types. The general box 'Other' puts the focus on the fact that new connector types can be identified and implemented as plug-ins for harvesting other types of resources (e.g. Linked Data).The architecture foresees for Harvesting the following different connector types:

- **Catalogue connectors**
  A catalogue could be a set of web URIs containing RDF (e.g. RDF / XML), an OGC RESTful web services, or other type. Catalogues can be of different format and respect different standards: OAI (Open Archives Initiative) has not provided yet an unique widely adopted structure.
- **Web pages connectors**
  Web Page Connector harvest Web pages containing RDFa structured content. If a

page contains RDFa structured content, this content can be harvested and stored as RDF in the system. Web pages providing interesting content on domains of interest are identified by their URI. When the web connector performs the harvesting, the resource list store is accessed to get the list of web pages (URIs) to harvest and the format used to structure the content.

- **Other types of connectors**
  Other types of connectors can be realised depending on the resource sources to be harvested. For example connectors to harvest Linked Data (possibly directly from a SPARQL end point) or connectors to harvest information from proprietary data sources, e.g. directly from a data base, csv file, etc.

Two examples for concrete catalogue connectors developed for TaToo are:

- GENESI-DEC (Ground European Network for Earth Science Interoperations – Digital Earth Community) is an FP7 project establishing open data and services access, allowing Digital Earth Communities to seamlessly access, produce, and share data, information, products, and knowledge. GENSI-DEC Resources are Catalogues exposing data and meta-data on earth observation and climate changes from the OGC (Open Geospatial Consortium) community, in particular data from SeaDataNet, DLR, ESA and JRC can be harvested, where:
  - SeaDataNet: provider of Marine data, including boats routes, marine buoy measures, and so on;
  - DLR: provider of Air traffic data and earth observation, grounding data;
  - ESA: provider of Spatial data, satellites and so on;
  - JRC: provider of earth observation data.

  The GENESI connector is used to harvest meta-information from GENESI-DEC catalogues. GENESI Catalogues are namely Datasets Series and Service, based on RDF / XML.
- The JRC Connector is used to harvest meta-information from JRC. JRC is one of the three Validation Scenarios for TaToo. JRC is making available their resources together with associated meta-information as RDF in order to be easily harvested by TaToo. Meta-information is expressed through concepts from the JRC Domain Ontology. JRC Catalogues are RDF / OWL Catalogues based on TaToo JRC Domain Ontology.

The complete list of connectors implemented and resources harvested is documented in D4.1.3 deliverable D4.1.3 Semantic Framework Implementation Prototype V3 (TaToo-D413, 2012). A general use case for harvesting is presented in section 6.4.3 together with the other Core Uses-Cases of TaToo.

# 6. Functional Purview of the TaToo Framework Architecture

Conceptually, the TaToo Framework is designed as n-Tier architecture composed of two main High Level Building Blocks:

1. User Components
2. TaToo System Components

The System Components are further divided into:

- **TaToo Public Services**
  the externally visible face of the TaToo Core (TaToo System Components). To achieve maximum interoperability, these services must be accessible over standardised web service interfaces.
- **TaToo Core Components**
  which provide the business logic, data and meta-information management. These components are only accessed by other System Components and by the TaToo Public Services. Consequently, interoperability is less of a concern and the System Components need not be accessible over a standardised web service interface.

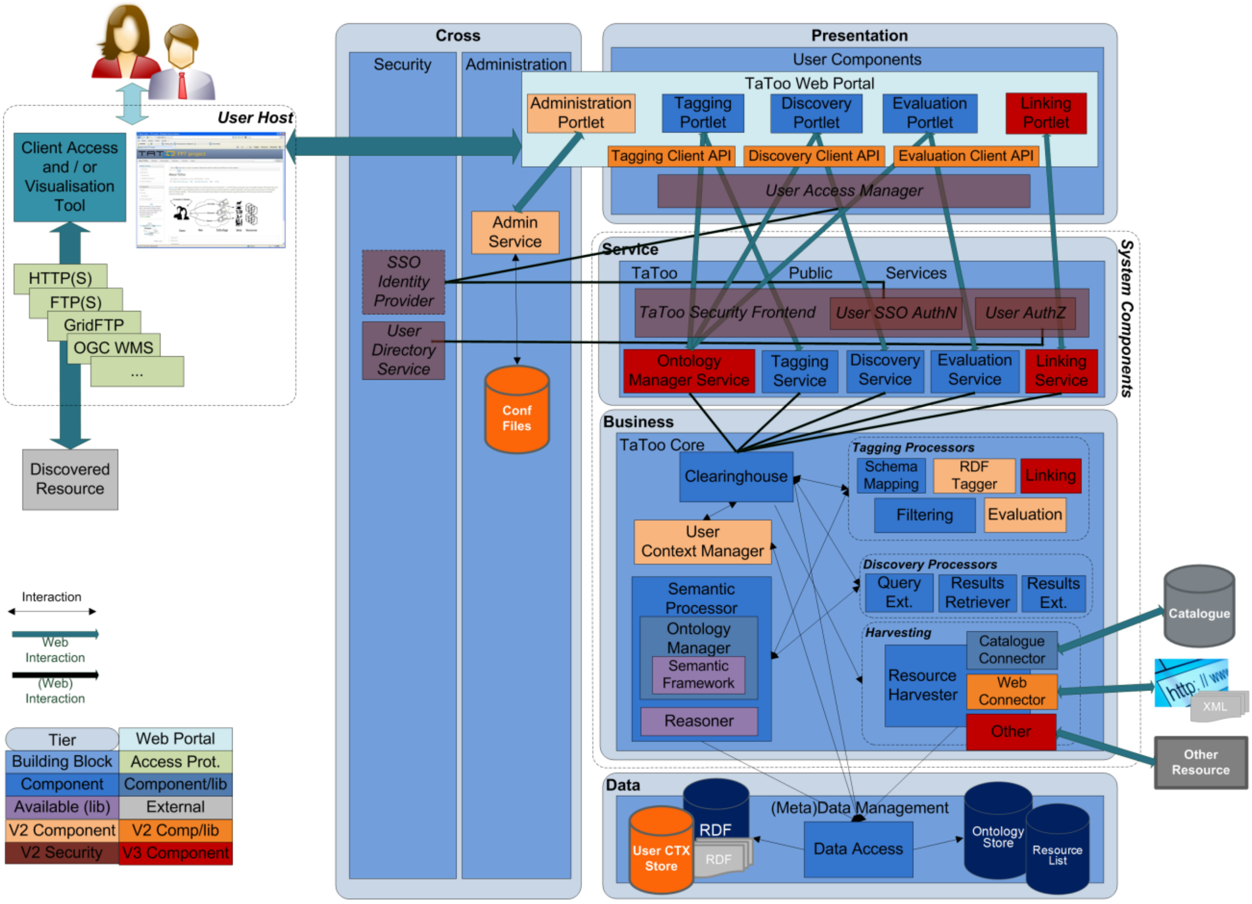

*Figure 6.1: Functional Purview of the TaToo Framework Architecture V3*

Figure 6.1 provides a complete overview on the Functional Purview of the TaToo Framework Architecture, showing also the Cross Tiers Security and Administration introduced in the second iteration. The diagram is further explained in the subsequent chapters where only the relevant details are shown.

The Functional Purview addresses general and implementation independent aspects only, and provides precise and detailed functional specification of each component in relation to identified functional requirements.

## 6.1. TaToo User Components High Level Building Block

The User Components High Level Building Block represents the front-end and client-side part of the TaToo Framework Architecture. It comprises the TaToo Tools which act as clients of the TaToo Public Services. A User Component, or TaToo Tool, is a component residing at the TaToo Presentation layer. It is a component that can in general be directly used by the user, which is particularly true in the case in which the component provides a GUI (Graphical User Interface). The user interacts with the component in order to take advantage of the functionality offered by TaToo.

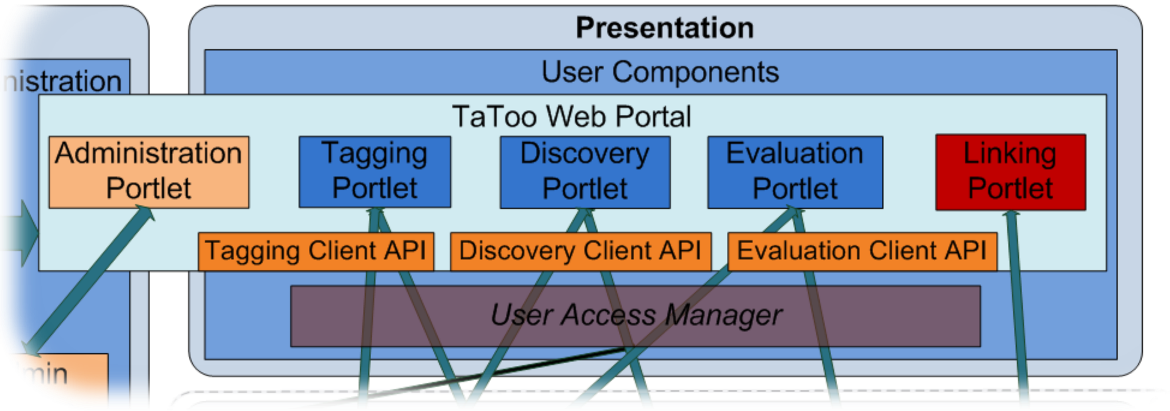

*Figure 6.2: TaToo User Components High Level Building Block*

In Figure 6.2, the Building Block presents four types of functional components. Essentially, the User Components can be classified as Tagging Tools, Discovery Tools , Evaluation Tools, Linking Tools and Administration Tools.

The open and service-oriented approach taken for the service Tier of the TaToo Framework in conjunction with the provision of well defined and self-describing service interfaces facilitates the development of User Components of various types. A User Component can be realised, for example, as:

- A Web-Portal, which combines a complete set of the TaToo functionality;

- A portlet allowing the use of specific TaToo functionality on a portal container , e.g. a Tagging Portlet;
- An external application to be installed on the user’s machine;
- A browser plug-in to be installed in the user’s browser;
- An API allowing the users to write their own application without the burden of dealing with web services. Each set of APIs provides a stub (a web client) to invoke TaToo Public Services transparently;

User Components shall be developed such that they can be interactively combined in order to enable users to create, modify, delete and update annotations. The composition is open, allowing third party users and developers to exploit existing or develop new Public Service and User Components supporting their specifically targeted workflows.

### 6.1.1 The TaToo Web Portal

TaToo also has to provide a set of reference specifications and implementations capable of satisfying the most important user needs in order to achieve broad user acceptance within the scientific community.

This is particularly true for User Components which represent the user front-end, which is of vital importance for successful and widespread user adoption. Therefore, among various types of tools that may be made available by the TaToo Consortium or by third parties, TaToo has to offer a central and user-friendly entry point (e.g. as a kind of One Stop Shop), the TaToo Web Portal. The users essentially access the portal to:

- search for resources;
- add tags and annotations on discovered resources;
- evaluate existing resources;
- link resources to other relevant resources.

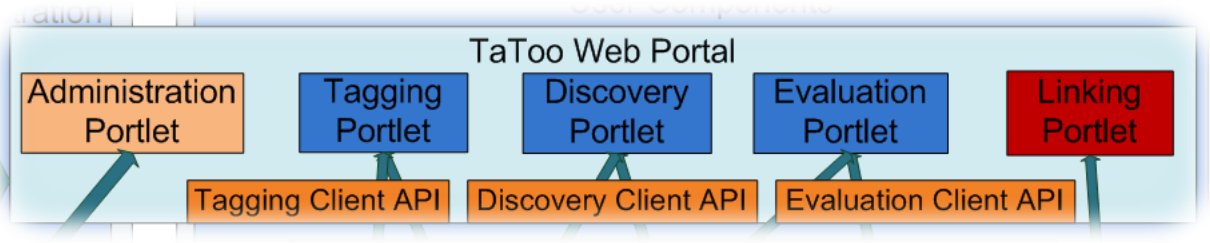

*Figure 6.3:TaToo Web Portal*

As shown in Figure 6.3 the portal takes a special position in the User Components Building Block. Being itself by definition a User Component it is in fact a composition of different User Components which are realised as portlets. Due to the open nature of the TaToo Architecture, user-specific tools may of course coexist with the portal.

However, the focus of the presentation Tier's reference implementation lies on the portal and the respective components supporting concrete functionalities (portlets). The main reason for the decision of the TaToo consortium to follow the portal based user interface approach as opposed to an approach that relies on set of browser extensions is the heterogeneous nature of web browser landscape. Many different versions of web browser from a few different vendors are currently in use. Supporting just the most widely used browser versions would already increase the development effort significantly. Furthermore developing for browsers includes in some cases very specific and non-standard technologies.

## 6.2. TaToo System Components High Level Building Block

The System Components High Level Building Block represents the server side part of the TaToo Framework Architecture and comprises the TaToo Public Services and the Core Components Building Blocks. The programmatic functionality of the TaToo System accessible by User Components is provided through the public interfaces of the TaToo Public Services (Service Tier). The fundamental meta-information/ontology management, processing, collection, harvesting and storage capabilities are realised by the Core Components Building Blocks (Business Tier) and Data Tier Components.

In contrast to the Public Services, Core Components do not expose any public interface outside the System Components Building Block and thus are not supposed to be accessed by the User Components directly.

Both the TaToo Public Services and the Core Components do not provide an interface to a human user but rather to a software component requesting an operation at the service interface level. All aspects dealing with user interaction and visualisation are addressed in the User Components High Level Building Block (Presentation Tier) as described in the previous chapter.

### 6.2.1 TaToo Public Services Building Block

The TaToo Public Services Building Block is realised as a set of web services which make the functionality of the TaToo System available through their interfaces.

Currently the following services can be identified: Tagging Service, Discovery Service, Evaluation Service, Ontology Manager Service, and Linking Service.

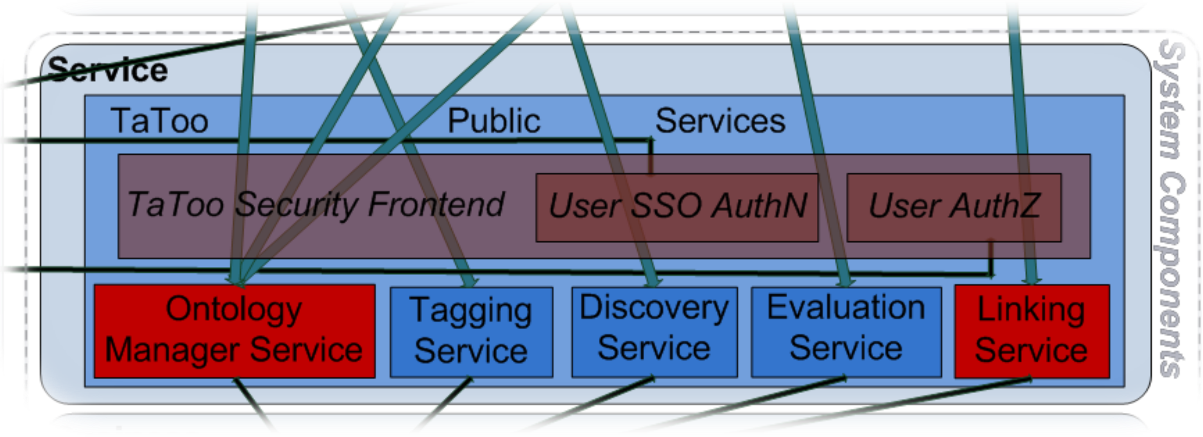

*Figure 6.4: TaToo Public Services Building Block*

### 6.2.2 TaToo Core Components Building Block

As already stated the Core Components Building Block provides the business logic of the TaToo System (see Figure 6.5).

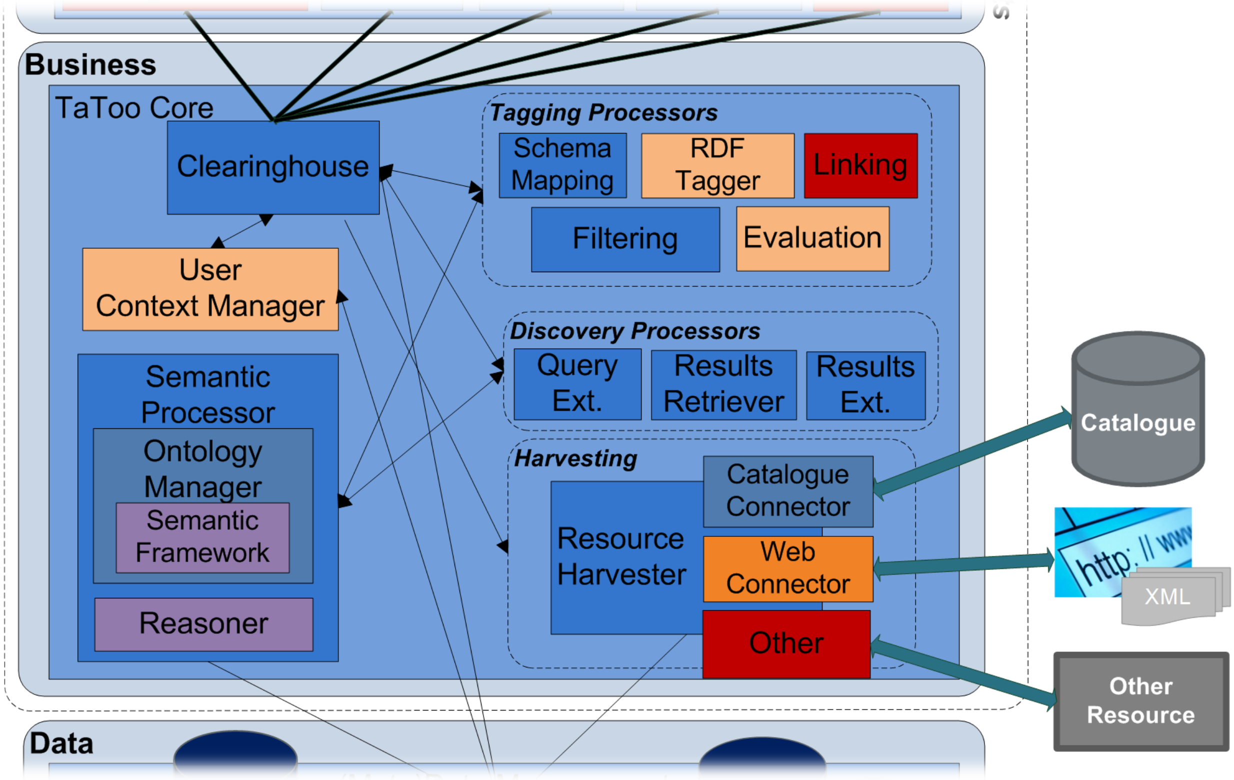

*Figure 6.5: TaToo Core Components Building Block*

The main components of this Building Block are the Clearinghouse, User Context Manager Semantic Processor, Ontology Manager, Reasoner, Tagging Processors, Discovery Processors and Resource Harvester (together with a set of Resource Connectors).

As shown in Figure 6.6 the aforementioned Core Components generally only process the data, leaving the task of data storage and management to separate "Data Tier" components.

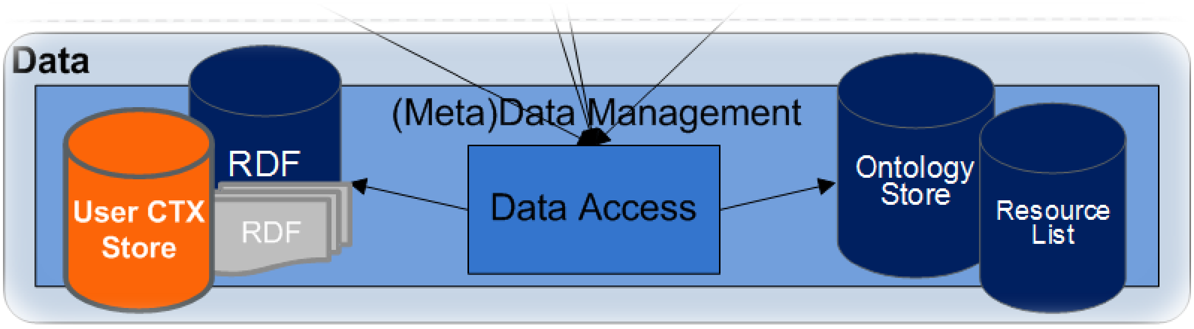

*Figure 6.6: Data Tier*

Within the Data Tier, the (meta)data management is carried out by a Data Access Component which allows managing repositories supporting different data types such as:

- Resource inventories (e.g. catalogue / web sites lists),
- Ontologies,
- User Information,
- RDF-Triples.

Depending on the data type, the Data Access Component may use various types of storage in the background, such as:

- Relational databases,
- Triplestores,
- File system.

The Clearinghouse, which is the central authority for managing semantic information accessed both by TaToo Web Services and Core Components and thus representing a semantic middleware, should be realised as a web service and thus formally be specified during the design and implementation phase. Other Core Components, components of the data Tier in particular, interfacing with the Clearinghouse but not with Public Services, may support different types of interfaces and protocols. In general, standards-based or widely adopted software solutions will be used to realise the different core component's functionality.

## 6.3. Identified components

The list of identified components is described in the following sections. The descriptions are intended to give a brief overview of the role and purpose of the components on an abstract and general level, including the general functionality to be provided.

Please note, that already identified concrete instances of “abstract components” that represent only a set of “similar” components are not part of this list nor of the architecture diagram (Figure 6.1). Examples for such “abstract components” are the Tagging Processors and Discovery Processors. Thus this section provides the general definition of a Tagging Processor or a Discovery Processor without going into the details of a concrete Tagging or Discovery Processor instance.

### 6.3.1 Clearinghouse

The Clearinghouse is a Core Component that represents the central entry point to the Business Tier from the Service Tier. It drives the interaction between other Core Components and in particular it provides access to semantic information through the Semantic Processor, thus representing a semantic middleware between TaToo Public Services and other Core Components. It offers the TaToo functionalities to the TaToo User Components through TaToo Public Services, e.g. during the manual user-driven tagging and discovery processes. Access to other Core Components from TaToo Public Services is always performed through the Clearinghouse. The Clearinghouse has also control over other Core Components, e.g. it triggers the automatic harvesting process which collects meta-information from web resources (e.g. catalogues). A potential performance bottleneck of the Clearinghouse Component in the TaToo System will be addressed by deploying multiple instances of the component. This is possible since the Clearinghouse is implemented as a Web service.

### 6.3.2 Semantic Processor

The Semantic Processor is a Core Component dealing with semantic information and semantic annotations. It supports the tagging and discovery processes by providing functions to

- retrieve ontologies on the basis of user-defined properties like domain or context;
- store and retrieve semantic annotations for resources (RDF-Triples);
- search for semantic annotations by providing a SPARQL query interface.

The Semantic Processor provides its functionality to Core Components either directly or through the Clearinghouse, and to higher tiers through the Clearinghouse only. The semantic functionality is realised by the following component:

The **Ontology Manager** consists of an existing **Semantic Framework** to take advantage of a set of useful APIs to manipulate RDF, support SPARQL queries, and support other functionalities to access and manage the following semantic elements:

- TaToo Ontologies: management (create, read, update, delete) and retrieval of both schemas and instances of current TaToo Ontologies. In the future additional ontologies and functionalities to facilitate the integration of new ontologies may be added;
- TaToo meta-information: storage and retrieval of meta-information in RDF-Triples format;
- A Reasoner to infer new knowledge from the available RDF-Triples & ontologies and to check for incongruent (or inconsistent) information while managing ontologies;
- A triplestore which is a data base for the storage and retrieval of RDF-Metadata

### 6.3.3 Discovery Processors

The Discovery Processors are a set of components that include all Core Components specific for supporting the discovery functionality. The type of the components under the Discovery Processors can be different. For each type of component, TaToo will allow the existence of different instances or implementations. The instance or implementation suitable e.g. for a given discovery strategy will be decided by the Clearinghouse. The components belonging to the Discovery Processors are the Query Expansion Component, the Resource Retrieval Component and the Result Expansion Component. These components implement the discovery business logic and provide the discovery related supporting functionality to the Clearinghouse.

### 6.3.4 Tagging Processors

Tagging Processors are a set of Core Components implementing the tagging, linking and evaluation business logic and providing related supporting functionality to the Clearinghouse.

TaToo core functionalities to store and retrieve semantic annotations are realised by the *RDF Tagger* Component which directly interacts with the selected Knowledge Base implementation either through specific API calls or SPARQL queries. The RDF Tagger covers the complete functionality offered by the Tagging Service. This includes for example the association of annotations with resources, the retrieval of annotations associated with a resource and the deletion of annotations. Filtering capabilities, such as filtering for a certain annotation provider (the user that provided the annotation) are carried out by a dedicated Tagging Processor, the Filtering Component.

TaToo core functionalities that enable the TaToo Linked Data approach are realised by a component called *Linking Processor*. In particular, this component provides functionalities to create RDF links that establish an explicit relationship between TaToo resources determined by specified link (relationship) types, create RDF links that establish explicit similarity links, to store those RDF links into the TaToo Knowledge Base (i.e., RDF repository), to retrieve all resources linked to a given TaToo resource (by a given link type) and to retrieve all resources similar to a given TaToo resource.

The Schema Mapping Component is a Tagging Processor intended to perform a translation from some simple XML tagging format to RDF/XML Triples ready to be stored in the kb given the transformation rules in form of an XSLT file (formalised transformation rules).

TaToo core functionalities for the evaluation of resources and annotations is provided by the Evaluation Processor. It is responsible for generating the TaToo evaluations based on the evaluation information received as part of the evaluation request.

### 6.3.5 Resource Harvester

The Resource Harvester is part of the TaToo Core and is capable of retrieving external meta-information on resources that can be either data (catalogues), or web pages (containing RDFa embedded into HTML) and mapping the resource schemas to known ontologies.

In order to achieve this functionality resource-type specific Harvester Connectors have been designed, which are dedicated Catalogue Harvesters for retrieving the contents of OGC catalogues, and Web Page Harvesters to gather access to web pages content. The Resource Harvester is triggered by the Clearinghouse and has access to the Data Access Component to get the list of resources to be harvested (e.g. the list of web sites).

### 6.3.6 Data Access Component

The Data Access Component belongs to the Data Tier which manages access to all kinds of underlying data stores. Depending on the type of the stored information, different types of data stores may be used. These include relational databases or simple files having simple structures. Examples of data stored by the component are ontologies, users information (profile or context), list of resources to be harvested like URIs identifying catalogues (services or single pages) or web pages.

The Data Access Component allows interacting Core Components to access information with no need to consider how the information required is stored. An interface - API or WSDL - is provided.

### 6.3.7 Tagging Services

The Tagging Service exposes the tagging functionalities of the TaToo Framework to User Components. The related business functionality is provided by specific Core Components, the Tagging Processors. Tagging Processors like the RDF Tagger and the Filtering Component are used to manage annotations and their relations to resources in the Semantic Repository and for filtering annotations according to specific properties (provider, publisher, data, …).

The Tagging Service receives tagging requests from the respective Tagging Tool to associate meta-information (annotations) with resources or to access already available meta-information. It is also be possible to update and remove annotations.

Additional functionalities of the Tagging Service include:

- applying a filter on the list tags retrieved (e.g. filter by annotation provider, date annotated, etc.);
- accessing annotations in different languages.

### 6.3.8 Discovery Services

Discovery Services are TaToo Public Services exposing the semantic search functionalities provided by the Clearinghouse to User Components over the web.

TaToo provides a single discovery service with methods that allow retrieving meta-information from the TaToo Knowledge Base in several ways, from an SPARQL End point to more structured search methods. The discovery process is query driven, allowing the user to selected certain terms from an ontology that are then used for the semantic search. The Discovery Services itself does not implement any business logic.

### 6.3.9 Evaluation Service

The TaToo Evaluation Service is a service of the TaToo System, which belongs to the TaToo Public Services layer. It provides functionalities for a community-based evaluation of the actual TaToo resources, resource annotations, and single annotation tags. The service’s operations are supposed to be invoked by a TaToo Evaluation Portlet, which is a part of the TaToo portal, as well as any other client-side application/tool that employs the TaToo services and provides a user support for the resource and annotation evaluations. After accepting the evaluation requests the service transmits it to the TaToo business layer for processing and generating the TaToo evaluations.

### 6.3.10 Linking Service

With respect to the TaToo Public Services tier, the TaToo Linked Data approach (see section 4.5) is supported by a set of operations that provide access to the Linked Data functionalities of the TaToo system. These operations are provided as methods of a TaToo service called the Linking Service.

### 6.3.11 Ontology Manager Service

The Ontology Manager Service exposes the public functionality to access to some common methods related to the TaToo ontologies. It offers functionality to retrieve filtered information about ontologies from the Semantic Repository. Therefore it provides supporting functionality for clients of the TaToo Framework.

### 6.3.12 TaToo Web Portal

The TaToo Web Portal is a User Component, which in TaToo is considered the main entry point for the TaToo Semantic Discovery Framework and its functionality. However, in principle a wide set of client side components benefiting from TaToo functionality can be identified, designed, and implemented. The basic TaToo portal features can be realised as single portlets, which are in fact Tagging, Discovery Tools and Evaluation Tools. These portlets can be combined in different ways to offer a set of user customisable functionality.

### 6.3.13 Tagging Tools

Tagging Tools are User Components responsible for the tagging of environmental resources, including provisioning of quality assessment. They are part of the TaToo Presentation Tier and, when realised as portlets, part of the TaToo Web Portal. Tagging Tools provide tagging functionality to the user by acting as clients to the Tagging Services.

Tagging Tools can process one or more resources so the users can add annotations to a particular resource or to a set of resources. Tagging Tools should also offer the functionality of sharing existing TaToos (tags) provided by other users, depending on the user role and respective access control policy.

As part of a portal Tagging Tools can be implemented through the means of different portlets to address specific functionality, e.g. tagging can be presented by a portlet that offers a user the support of a browseable ontology to select the terms to be associated with the resources as meta information.

Independent Client APIs are also available to the user to access the Public Service Tier. Tagging, Discovery and Evaluation Client APIs also provide security levels of Authentication and Authorization with the usage of the embedded User Access Manager component.

### 6.3.14 Search Tools

Search Tools are User Components enabling the search for environmental resources. They are part of the TaToo Presentation Tier and, when realised as a portlet, part of the TaToo Web Portal. Search Tools provide functionality to the user by acting as clients to the Discovery Services and providing a user-interface for expressing user information needs in terms of resources to be discovered.

### 6.3.15 Evaluation Tools

Evaluation tools enable users to express their opinion on TaToo resources and resource annotations. The TaToo evaluation portlet is an example of the TaToo evaluation tools that is provided as a part of the TaToo portal.

Besides the TaToo Evaluation Portlet, the TaToo Evaluation Tools can also include a range of other, client-side applications/tools, that invoke the TaToo Evaluation Service. Finally, the TaToo Evaluation Tools can include some plug-ins to the conventional Web browsers. Figure 6.7 shows an example of a pop-up-based TaToo evaluation interface. As we can see from the figure when the user wants to evaluate a resource, he first selects an evaluation criterion (e.g., usefulness or completeness) and then assesses the evaluation values.

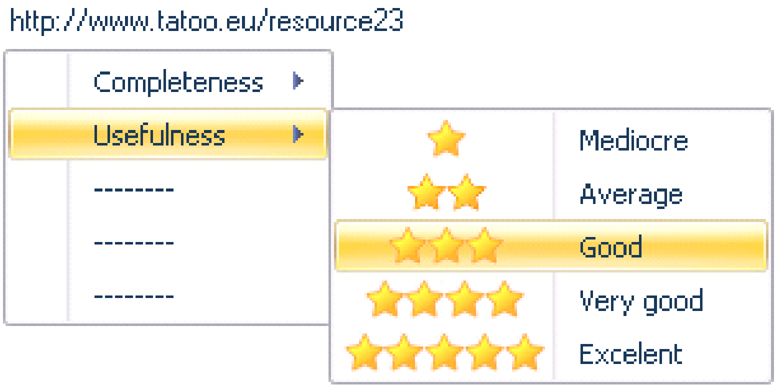

*Figure 6.7: An example pop-up based resource evaluation interface*

### 6.3.16 Linking Tools

The GUI of the TaToo Web portal provides UI elements that enable TaToo users to benefit from the introduced Linked Data functionalities. One group of these UI elements is responsible for adding links between resources, while the other one contains UI elements that are responsible for browsing related resources and navigating across the TaToo resource repository by following the links between the resources.

The UI elements from the first group should be available in both TaToo tagging and TaToo discovery use cases. A user should be able to link a resource to other resources at the time she/he annotates the resource. Moreover, a user should be able to link a resource to other resources at the time she/he browses search results (discovered resources). To satisfy both use cases, but not to replicate the same UI elements, the TaToo system provides a portlet that is independent from the discovery and tagging portlets but is exploited by them. This portlet is called the *TaToo Linking Portlet*.

The UI elements from the second group, that is, resource navigation and browsing should be available in the TaToo discovery use case. Therefore, these UI elements will be implemented as a part of the existing TaToo Discovery portlets by extending their GUI.

### 6.3.17 Administration Services and Tools

Administration Services and Tools support the generic configuration of manageable components (e.g. setting the communication preferences of the Tagging Service) through a graphical user interface that provides textual and visual editing capabilities for remotely stored configuration files.

### 6.3.18 User Access Manager

The User Access Manager is a security component located in the Presentation Tier. The component is responsible for providing the authentication and authorization level on client side to the Public Service Tier. In particular the User Access Manager presents to the TaToo Security Frontend either the user token identifier, a SAML assertion, or the user SSO credentials, username and password. The TaToo Security Frontend will consume the token id or the credentials to verify the identity of the user and retrieve the user attributes. If the user token id and its attributes are matched with the access rights associated to the resource the user is authorized.

The User Access Manager is realised as an API that can be used both from the TaToo Web Portal and from the custom user clients. The API will make use of the existing SSO API to authenticate, and provide the expected functionality in different programming languages addressing TaToo validation scenarios.

### 6.3.19 TaToo Security Frontend

The TaToo Security Frontend is the security framework responsible for the Authentication and Authorization on the Public Service Tier.

The TaToo Security Frontend is composed by different components:

- User Single Sign-On AuthN
- User AuthZ
- Identity Provider
- User Directory Service

The components composing the TaToo Security Frontend are distributed in different architectural locations. The User Single Sign-On AuthN and the User AuthZ are in the Public Service Tier, which is the entry point tier to the TaToo Semantic Framework that will be secured. The Identity Provider and the User Directory Service are in the Security Cross Tier.

The User Single Sign-On AuthN is the module responsible for authenticating the user presenting its credentials to the Security Frontend. The user is identified either by providing the token identifier obtained from the TaToo Web Portal (SAML assertion), or with the SSO username and password. The User SSO AuthN will then involve the Identity Provider that will sign in the user to the system, once for all and to a set of resources, in our case the set of TaToo Public Services.

The User AuthZ, given the retrieved user token identifier, will take advantage of the TaToo User Directory Service to retrieve the list of attributes pertaining to the user. The attributes can indicate the community of the user or its role within the community. These information, token identifier and user attributes, will be used by the User AuthZ to authorize or not the access to the secured resources.

### 6.3.20 User Context Manager

The User Context Manager is situated in the Business Tier of the TaToo Framework. The user does not interact directly with this component, but it is used to manage certain information about the user, such as provided resources, number of annotations and categorisation of the user.

The functionality of the User Context Manager is the following:

- Retrieve a fixed number of resources provided from the user ordered by a specific property (e.g. decreasing date);
- Retrieve a fixed number of annotations provided from the user ordered by different criteria evaluated by the filtering component (SPARQL extension)
- Categorisation of the user (e.g. according to the number of annotations made).

## 6.4. Core Use-Cases

In the following, the basic TaToo Core Uses-Cases showing the core functionality supported by TaToo in the second iteration of the project are described. They give an outline of the TaToo tagging and discovery functionality and provide a clear picture of the role of the tagging- and discovery related components in the architecture.

### 6.4.1 Tagging

Tagging is the process of attaching meta-information to resources. The result of a tagging process, the annotation instances, are always related to ontological concepts, thus allowing the deduction of additional implicit knowledge during tagging (or while performing a search). Diagram 6.1 and Diagram 6.2 illustrate the two most common tagging use-case:

The first Tagging Core Uses-Cases is related to the association of meta-information with a resource:

- The **User** identifies resources to be tagged (directly providing their URIs or as resulting from a search process) in the GUI of one of the **Tagging Portlets** (for example the **Advanced Tagging Portlet**).

- The **User** selects his domain (e.g. Air Quality Modeller, etc.) from a list of supported domains provided by the **Ontology Manager Service** (not shown in Diagram 6.1).

- The **Tagging Portlet** offers a basic list of terms (based on concepts related to available ontologies) associated with the selected domain provided by the **Ontology Manager Service**.

    - The **Tagging Portlet** asks the **Ontology Manager Service** for available Domain Ontologies. The **Ontology Manager Service** asks the **Clearinghouse** for ontologies related to the selected domain.

    - The **Clearinghouse** calls the **Ontology Manager** Core Component.

    - The **Ontology Manager** performs a search for a matching ontology in the **TaToo Knowledgebase** through the **Semantic Processor** and returns the ontologies to the **Clearinghouse**.

    - The ontologies are returned by the **Ontology Manager Service** to the **Tagging Portlet**. The **Tagging Portlet** presents the selectable terms to the user. Thanks to the support for multilingual labels, the terms can be shown to the user in his preferred language.

- The **User** chooses the type of the resource (e.g. Web Page, Web Service, etc.) and a number of terms from the ontology and tags the resource. The user has furthermore the possibility to optionally add additional meta-information like a comment. Then, the user activates the update function provided by the **Tagging Portlet**.

- The **Tagging Portlet** makes a call to the *addAnnotationsToResources* operation of the **Tagging Service** passing as parameters the identification of the resource(s) (including at least the URI), a list of annotations (including at least the selected terms) and the locale of the user which is used to define the language of literal values (e.g. an additional comment provided by the user).

- The **Tagging Service** calls the *addAnnotationsToResources* operation of the **Clearinghouse** and passes the respective parameters.

- The **Clearinghouse** calls the **RDF Tagger** Core Component to store the annotations as RDF-Triples in the **TaToo Knowledgebase**.

- The **RDF Tagger** calls the **Semantic Processor** to store the annotations and returns a detailed status message to **Clearinghouse**.

- A status message is returned through the **Tagging Service** to the **Tagging Portlet**, indicating whether the tagging was successful or not.

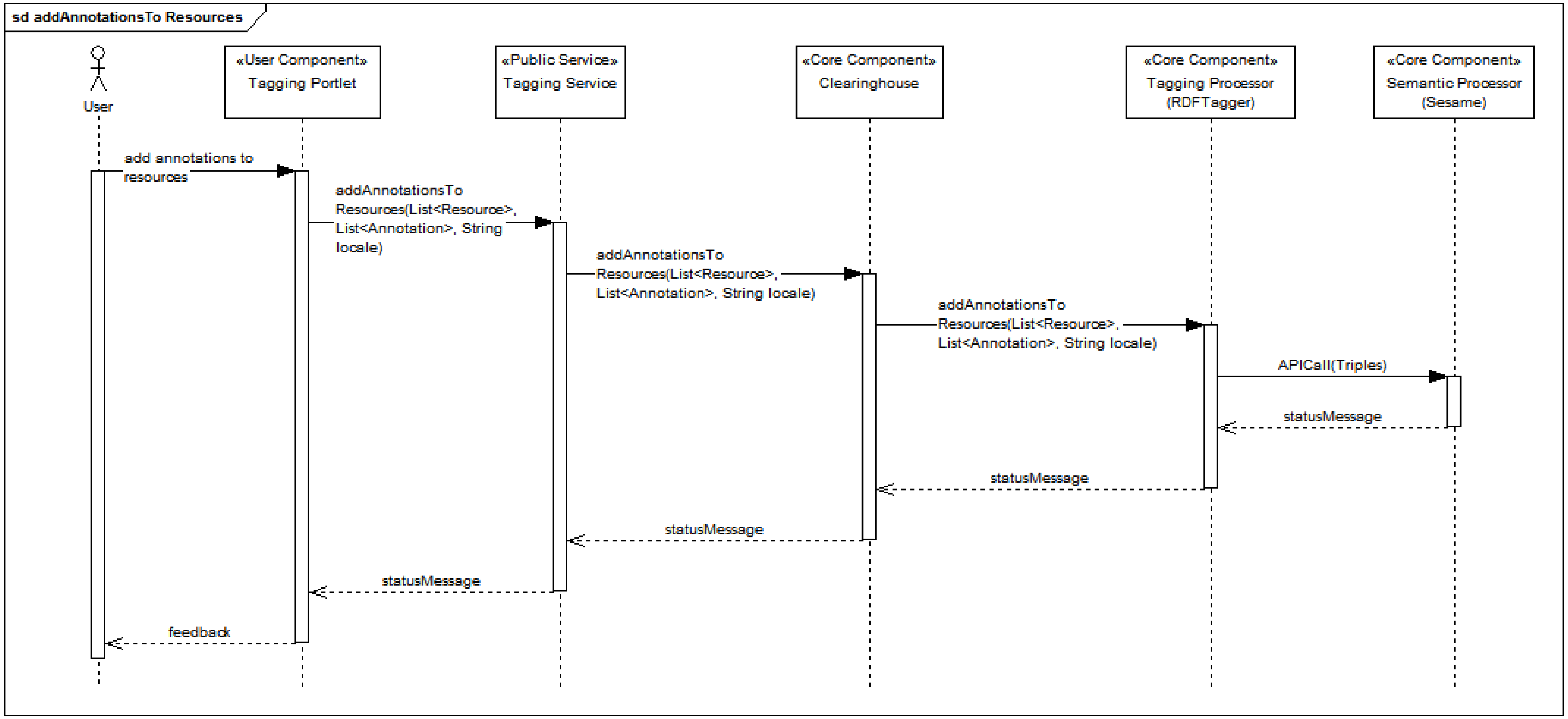
sd addAnnotationsTo Resources
User
«User Component» Tagging Portlet
«Public Service» Tagging Service
«Core Component» Clearinghouse
«Core Component» Tagging Processor (RDFTagger)
«Core Component» Semantic Processor (Sesame)
add annotations to resources
addAnnotationsTo Resources(List<Resource>, List<Annotation>, String locale)
addAnnotationsTo Resources(List<Resource>, List<Annotation>, String locale)
addAnnotationsTo Resources(List<Resource>, List<Annotation>, String locale)
APICall(Triples)
statusMessage
statusMessage
statusMessage
statusMessage
feedback

*Diagram 6.1: addAnnotationsToResources*

The second Tagging Core Uses-Cases is related to the retrieval of meta-information associated with a resource:

- The **User** identifies a resource to show meta-information (annotations) about (directly providing their URIs or as resulting from a search process) in the GUI of one of the **Tagging Portlets** (for example the **Advanced Tagging Portlet**).

- The **User** optionally provides one or more filter conditions, e.g. to show only annotations pertaining to certain user (provider of the annotation) or annotations that received good evaluations.

- The **Tagging Portlet** makes a call to the *getAnnotationsOfResource* operation of the **Tagging Service** passing as parameters the identification of the resource (including at least the URI), the locale of the user which is used to define the language of literal values (e.g. an additional comment provided by the user) and the optional filter conditions.

- The **Tagging Service** calls the *getAnnotationsOfResource* operation of the **Clearinghouse** and passes the respective parameters.

- The **Clearinghouse** calls the **RDF Tagger** Core Component to retrieve the annotations as RDF-Triples in the **TaToo Knowledgebase**.

- The **RDF Tagger** constructs several SPARQL queries to retrieve the annotation triples from the **TaToo Knowledge Base**. The **RDF Tagger** sends the generated SPARQL queries along with the filter conditions provided by the user to the **Filtering Component**.

  - The **Filtering Component** processes SPARQL queries and the filter conditions received from the **RDF Tagger** and produces the augmented queries. The Filtering Component sends the augmented **SPARQL queries** back to the **RDF Tagger**.

- The **RDF Tagger** contacts the **Semantic Processor** to execute the augmented SPARQL queries.

  - The **Semantic Processor** executes the queries against the **TaToo Knowledgebase** and returns the matching triples to the RDF Tagger .

- The **RDF Tagger** constructs annotations from the Triples and returns them to the **Clearinghouse**.

- The annotations belonging to the selected resource and matching the filter conditions are returned by **Tagging Service** via the **Clearinghouse** to the **Tagging Portlet**. The **Tagging Portlet** shows the annotations the user in his preferred language.

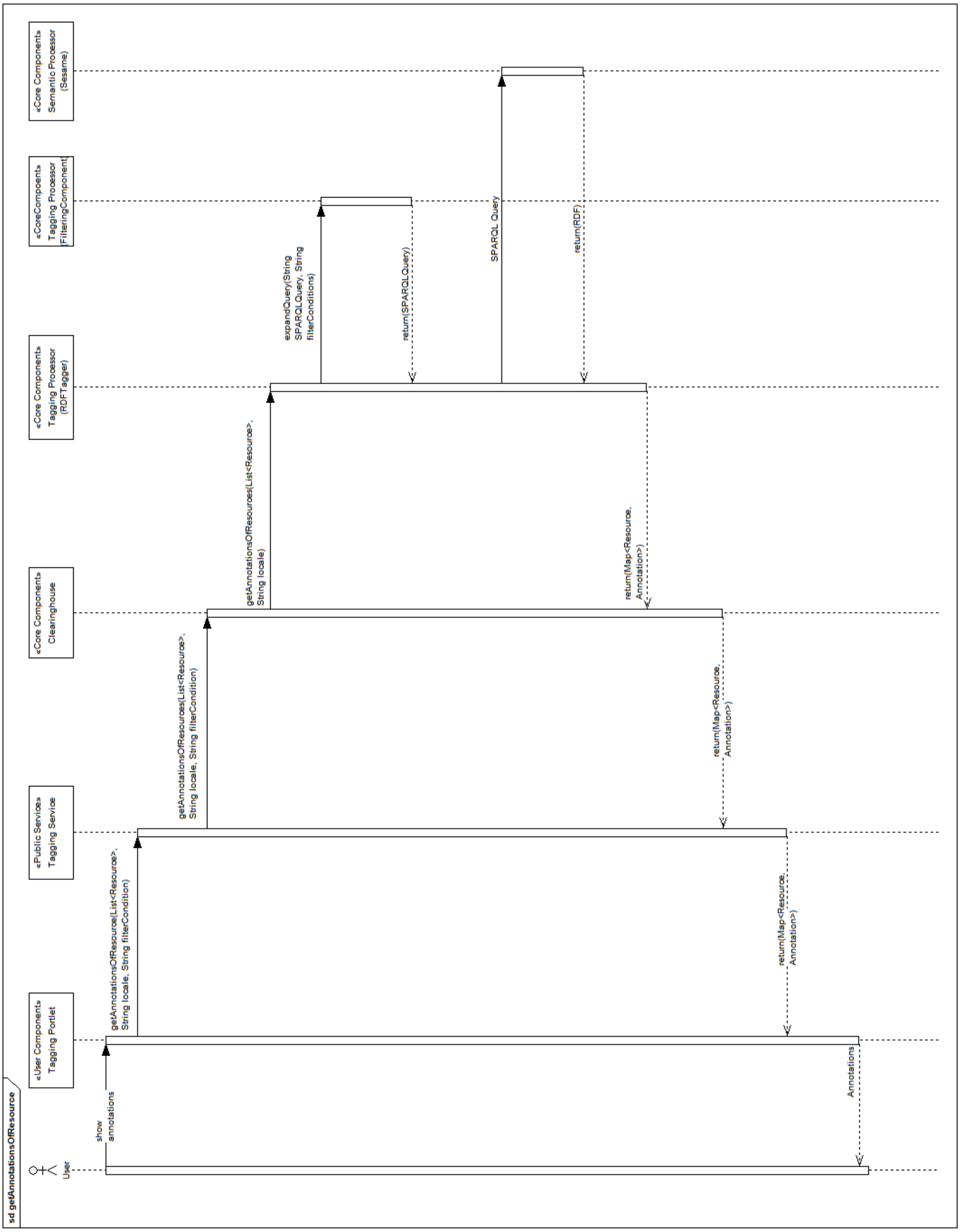
sd getAnnotationsOfResource
User
«User Component»
Tagging Portlet
«Public Service»
Tagging Service
«Core Component»
Clearinghouse
«Core Component»
Tagging Processor
(RDFTagger)
«CoreComponent»
Tagging Processor
(FilteringComponent)
«Core Component»
Semantic Processor
(Sesame)
show annotations
getAnnotationsOfResource(List<Resource>, String locale, String filterCondition)
getAnnotationsOfResources(List<Resource>, String locale, String filterCondition)
getAnnotationsOfResources(List<Resource>, String locale)
expandQuery(String SPARQLQuery, String filterConditions)
return(SPARQLQuery)
SPARQL Query
return(RDF)
return(Map<Resource, Annotation>)
return(Map<Resource, Annotation>)
return(Map<Resource, Annotation>)
Annotations

*Diagram 6.2: getAnnotationsOfResource*

### 6.4.2 Search

As reflected in the following use-case illustrated in Diagram 6.3, searches are restricted to choosing search terms from known ontologies.

- The discovery flow begins in a **Search Portlet**. The **Search Portlet** allows the user to express an information need to the system. To do this, it is likely that the **Search Portlet** will access the Domain Ontologies to facilitate the task for the user. This is done analogues to the Tagging Core Use-Case through the **Discovery Service**. Once expressed, the information need is entered into the system through the **Discovery Service**.

- The **Discovery Service** transforms the user information need, expressed in a client-side formalism, into a request suitable to be processed by the **Clearinghouse**.

- The **Clearinghouse** contacts the Query Expansion core component.

- As a first step in the discovery strategy, the **Query Expansion Component** transforms the user need into a SPARQL query that matches exactly the user needs. The query expansion, depending on the amount of resources that matches the query, might issue other queries taking advantage of existing mappings for cross-domain search. The annotations that match the query are passed back to the Clearinghouse that acts as a bridge towards the Discovery Service.

- The **Discovery Service** contacts the **Resource Retrieval** component via the **Clearinghouse** to rank the results.

- The **Resource Retrieval Component** identifies the resources referred, retrieves some extra meta-information about the resources and annotations in order to perform a ranking algorithm to order the resources. The resources are ordered using a criterion that, for instance, takes into account relevant vs. non-relevant annotations (number of annotations that matches the query vs. total number of annotations of the resource) among others. The results may also be filtered by relevance to avoid returning an excessive number of hits to the user. To perform this process, the **Resource Retrieval Component** is assisted by the **Semantic Processor**.

- After that, the ordered set of resource URIs is returned to the Discovery Service, wich invokes the **Result Expansion Component**. The results are then expanded and completed by retrieving a Resource object containing the annotations and evaluations of the ranked resources.

- After the result expansion, ranked results are returned to the **Discovery Service** by the **Clearinghouse**.

- The **Discovery Service** passes the results to the **Search Portlet**.

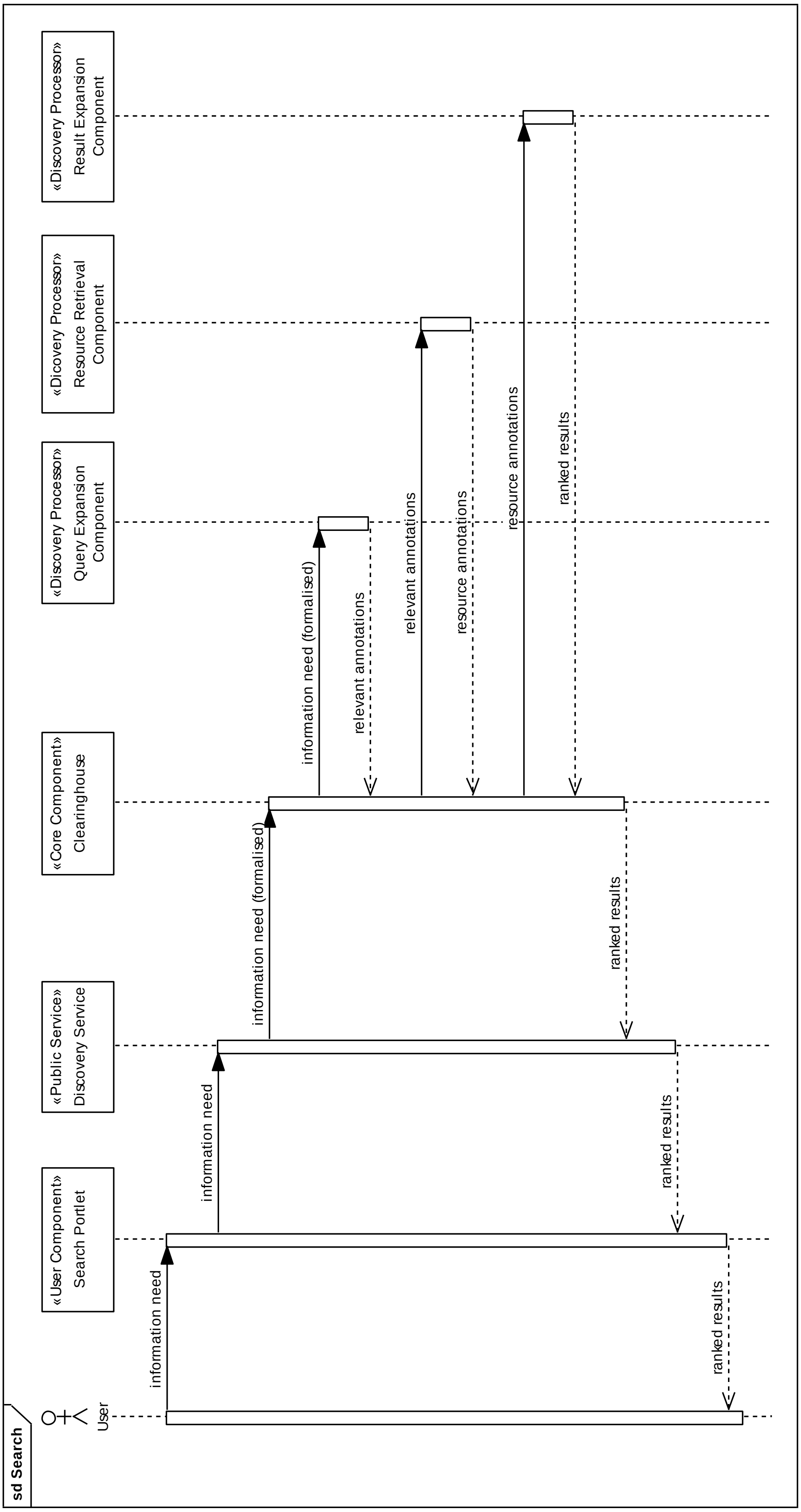

*Diagram 6.3: Search*

### 6.4.3 Harvesting

Harvesting was already explained in detail in section 5.4. The use case shown in Diagram 6.4 is to illustrate a general harvesting process:

- To start the harvesting process, the **Resource Harvester** is triggered by the Clearinghouse.

- The **Resource Harvester** contacts the **Data Manager** to retrieve a list of external sources for meta-information (e.g. catalogues or web pages), available connectors and types.

- The **Resource Harvester** harvests (through the means of the **Harvester Connectors**) the external meta-information sources and retrieves meta-information describing resources.

- The **Harvester Connectors** processes the harvested meta-information by mapping the external meta-information schema (which could be even an ontology) to an ontology (concepts) of TaToo’s **Ontology Framework**.

- The **Harvester Connector** produces a list of RDF-Triples compliant with MERM and the TaToo Domain Ontologies.

- The **Resource Harvester** stores the triples in the **Semantic Repository** (triple store).

  The **Resource Harvester** informs the **Clearinghouse** about the result of the harvesting process.

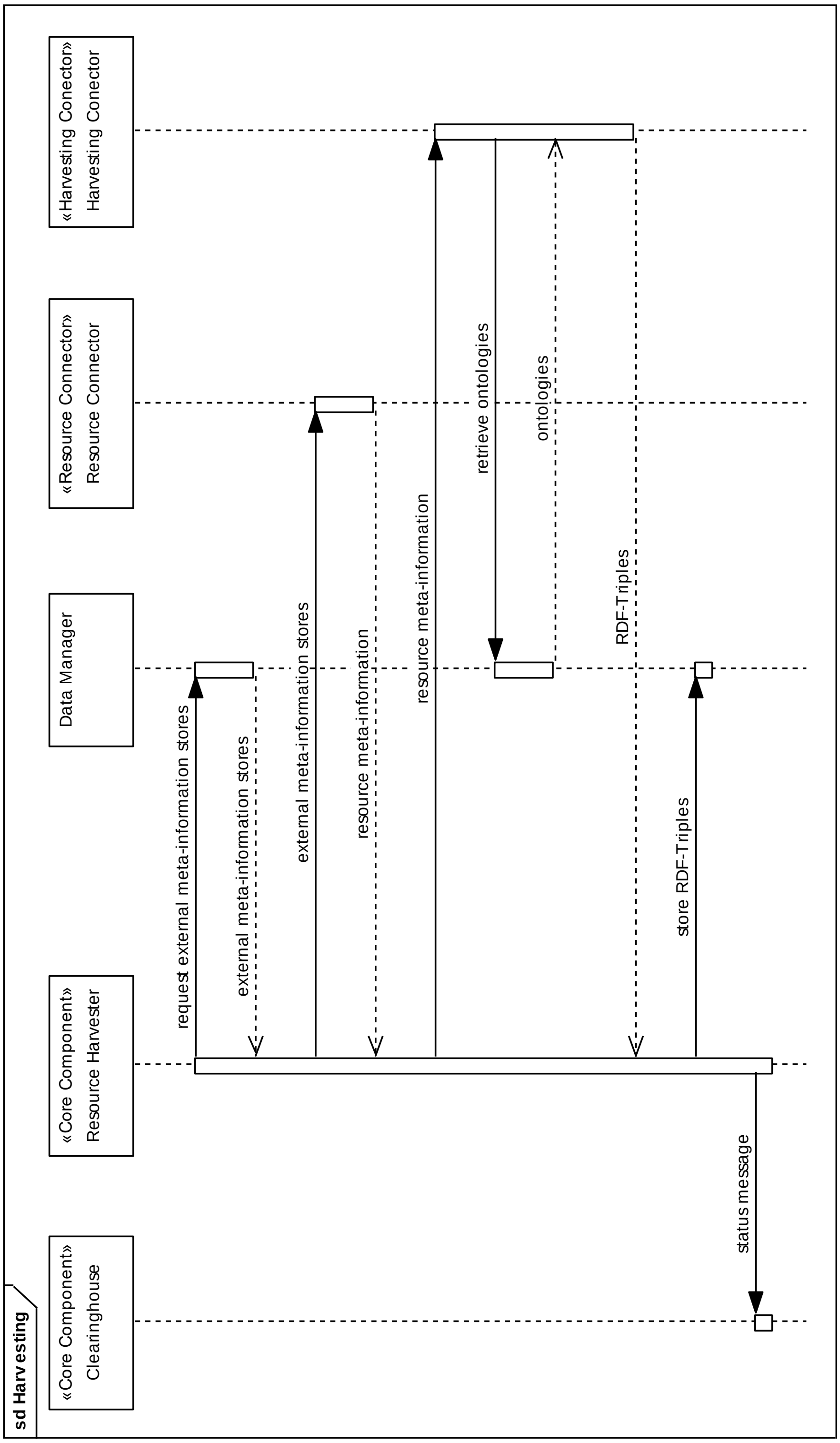

*Diagram 6.4: Harvesting*

### 6.4.4 Evaluation

In this section we present two evaluation use cases, each of which corresponding to one the three types of the TaToo evaluations: resource evaluation, annotation evaluation and tag evaluation described in section 4.4. All three use-cases start by a user browsing resource from the list of retrieved search results.

#### 6.4.4.1 TaToo resource evaluation use-case

Diagram 6.5 illustrates the UML sequence diagram of the TaToo resource evaluation use case. We now describe each step of the use case in more details.

- The use-case starts by the user selecting a resource from the list of search results shown in the **Search Portlet**.

- The **Search Portlet** forwards the URI of the selected resource to the **Evaluation Portlet**.

- In the **Evaluation Portlet** the user selects the evaluation criterion and then specifies the evaluation value for the resource with respect to the selected criterion. The portlet calls the **Evaluation Service** and sends to it the evaluation request containing all information required for generating the resource evaluation (i.e., resource URI, evaluation criterion, and evaluation value).

- The **Evaluation Service** calls the **Clearinghouse** and forwards the evaluation request to it. After receiving the evaluation request, the **Clearinghouse** calls the **Evaluation Processor**.

- The **Evaluation Processor** receives the resource evaluation request, generates the resource evaluation tags (i.e., RDF triples) according to the TaToo evaluation schema and stores them in the TaToo RDF repository (via the **Semantic Processor**). After the evaluation tags are generated and stored, the **Evaluation Processor** sends back an acknowledgement message that informs the user about the result of the evaluation process.

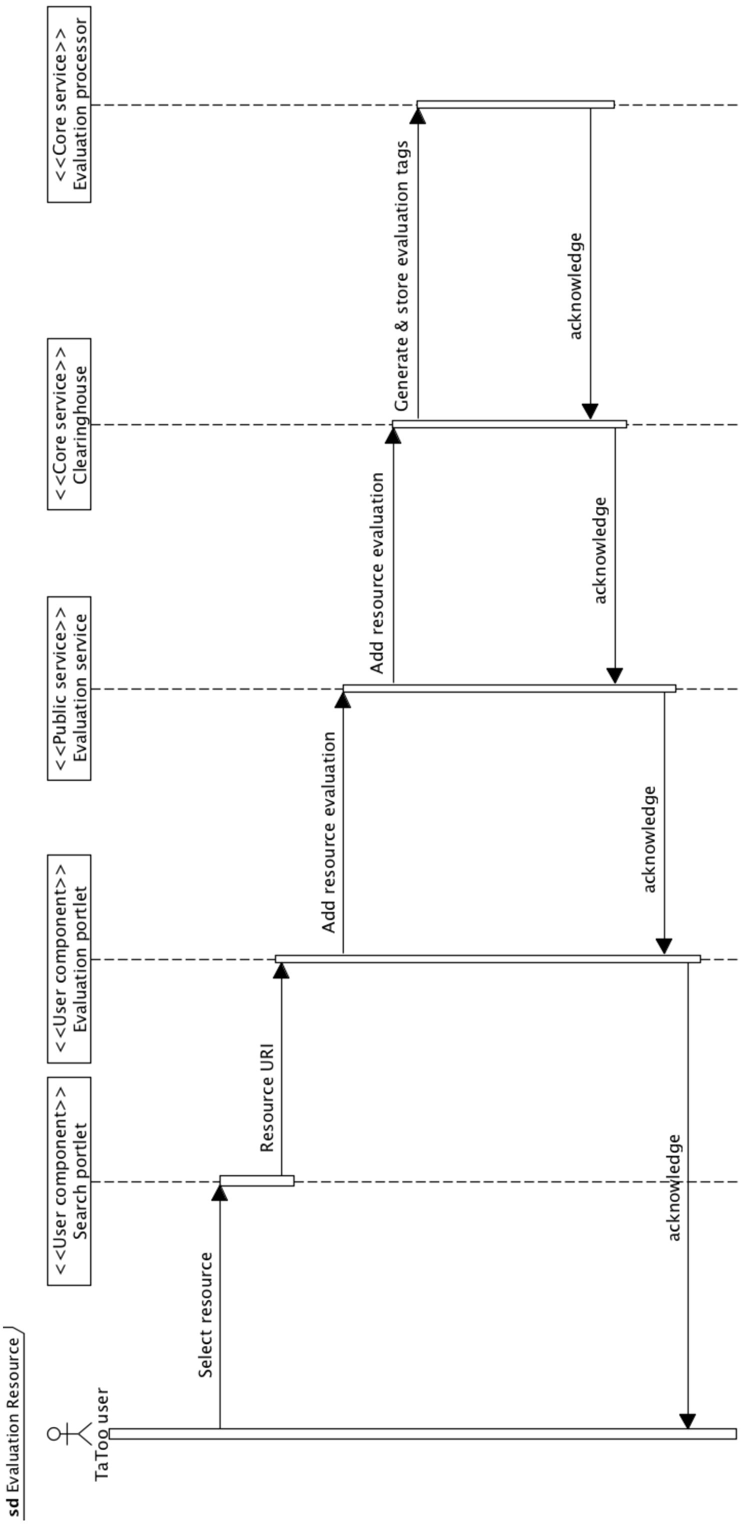

*Diagram 6.5: Resource evaluation: UML use-case diagram*

### 6.4.4.2 TaToo annotation evaluation use-case

Diagram 6.6 illustrates the UML sequence diagram of the TaToo annotation evaluation use case. The main difference of this use case from the resource evaluation use case is an additional user action that is performed after selecting the resource from the search results. The user needs to browse the details of the selected resource, among which the resource’s annotation, and selects one of the annotations. The rest of the use case is identical to the resource evaluation use case besides the fact that the evaluating entity is not the resource but the resource annotation.

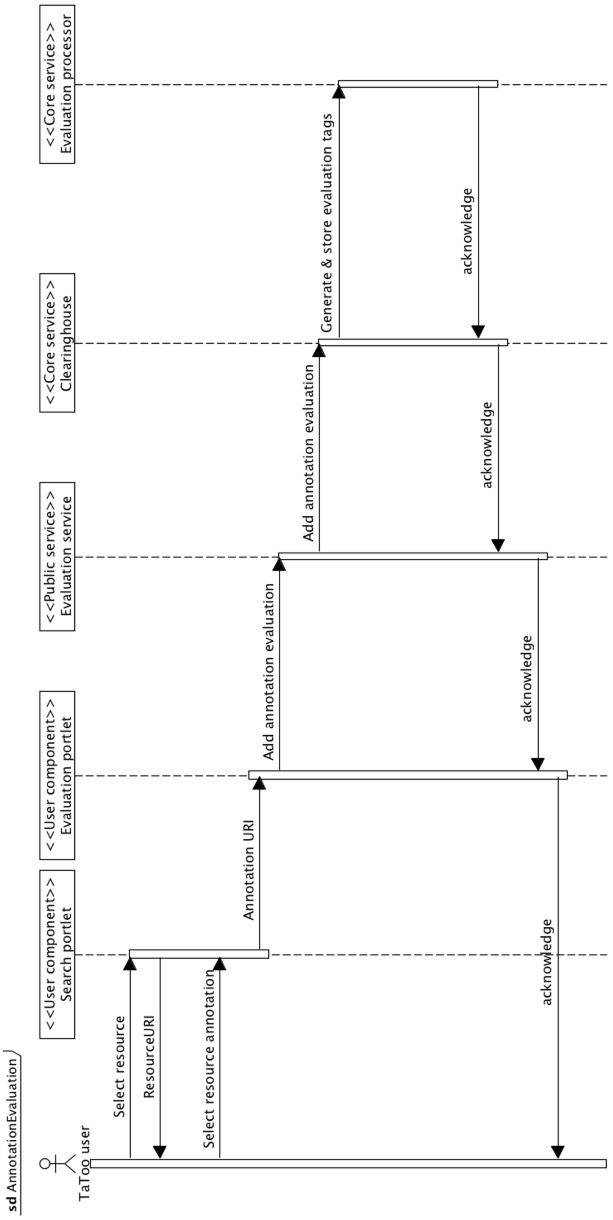
sd AnnotationEvaluation
TaToo user
<<User component>> Search portlet
<<User component>> Evaluation portlet
<<Public service>> Evaluation service
<<Core service>> Clearinghouse
<<Core service>> Evaluation processor
Select resource
ResourceURI
Select resource annotation
Annotation URI
Add annotation evaluation
Add annotation evaluation
Generate & store evaluation tags
acknowledge
acknowledge
acknowledge
acknowledge

*Diagram 6.6: Annotation evaluation: UML use-case diagram*

### 6.4.5 Security

The Security Tier's aim is to provide an authentication and authorization system which works vertically across all layers of the TaToo framework. In TaToo Security the following security components can be identified: The User Access Manager, the TaToo Security Frontend framework, the Single Sign-On Identifier and the User Directory Service.

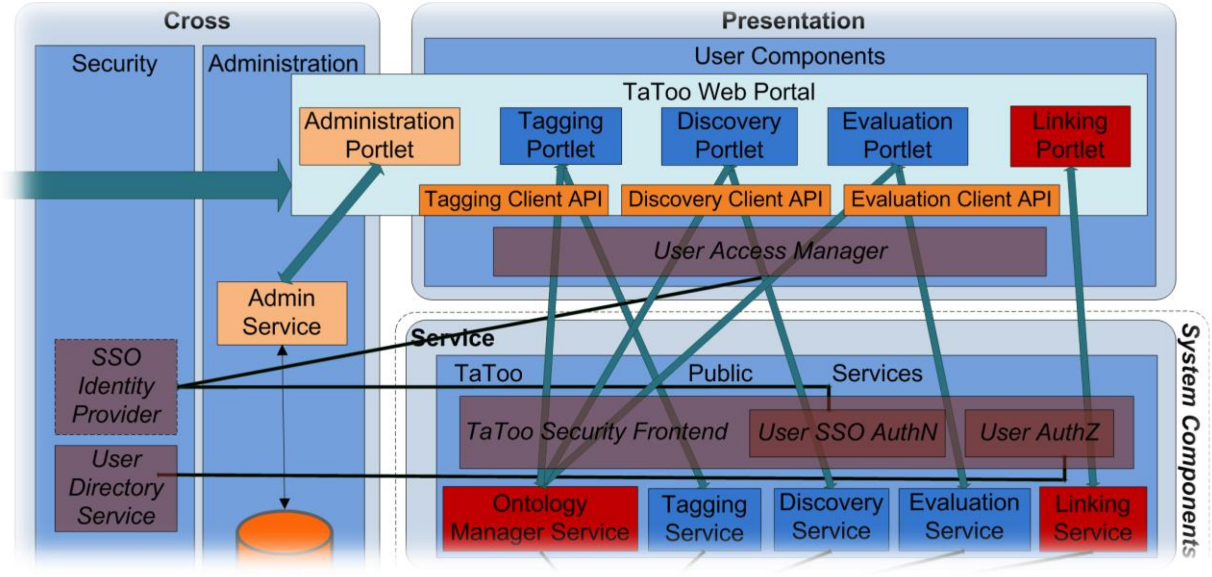

*Figure 6.8: TaToo Security components*

The User Access Manager is the component responsible for interfacing the User Presentation tier components, whether it is the TaToo Web portal or the Tagging, Discovery and Evaluation API Clients, with the TaToo Security Frontend that secures the Public Service tier.

The User Access Manager will be provided as a library, it can be either used from external clients or within the new identified User Presentation tier components, in particular: Tagging, Discovery and Evaluation API Clients. The external user will then be able to chose between security functionality already implemented in the abovementioned clients, or to implement its own client taking advantage of the User Access Manager.

The TaToo Security Frontend is responsible for Authentication and Authorization on the Public Service tier, which is the entry point to the TaToo Core framework. In particular the Security Frontend is composed by different components and modules: the Web Server container where it will run, the User Single Sign-On AuthN module, the User AuthZ module, the Identity Provider and the centralised User Directory Service. Moreover, the Identity Provider Service of the Single Sign-On framework is involved in the communication with the User SSO AuthN as an external component to the TaToo framework.

The User Single Sign-On AuthN is the module in charge of authenticating the user through its Single Sign-On token identifier, most likely a SAML assertion, or Single Sign-On user credentials, username and password.

The User SSO AuthN will exchange the user credentials with the Single Sign-On Identity Provider to sign in the user to the system. The TaToo Security Frontend will at this point take advantage of another module, the User AuthZ.

This module will contact the User Directory Service to retrieve user attributes, based on the user token identifier, the SAML assertion. The User AuthZ will use these two information, user token identifier and user attributes, to match them against the policies defined in the TaToo Security Frontend and will provide access to the Tagging, Discovery and Evaluation Services in case it is successful.

In Diagram 6.7 you can have a detailed description of the authentication and authorization mechanisms from the sequence diagram described.

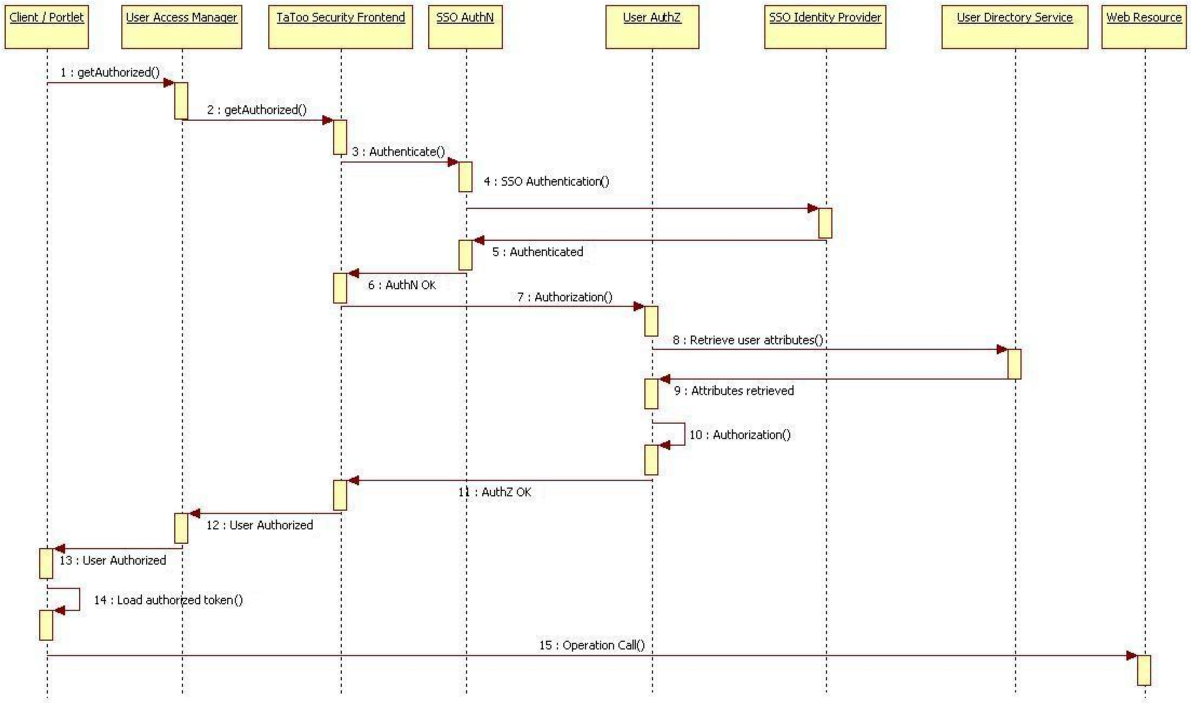

*Diagram 6.7: TaToo Authentication and Authorization sequence diagram*

The TaToo Security Frontend entry point is a Web Server, where also the SSO AuthN and the User AuthZ are running. It can be either configured as a proxy server, or deployed locally on the same machine hosting one or more TaToo Public Services. This approach offers the option to have in front of the TaToo Public Services a Secured Proxy Server, or to have the Public Services secured locally on the same machine.

# 7. Conclusions

In this document we have presented the major part of the TaToo Framework Architecture, consisting of an implementation independent Functional Purview which is concerned with the definition of the TaToo Framework and composed of interacting components, each providing a certain set of functionalities. This Functional Purview introduces an implementation independent layer that was mapped to a concrete implementation platform during the design and implementation phase (WP4 - Implementation), and completed the TaToo Framework Architecture with a concrete Implementation Purview.

The Functional Purview primarily provides, among Core Use-Cases, Semantic Concepts of tagging and discovery, description of cross aspects, and Component Descriptions, the functional specifications of the identified TaToo Components. The most important facts about TaToo Components are that

- a TaToo Component has been described according to a general component description template which is common for all types of TaToo Components,
- the interactions between components and their role in the overall workflows of tagging and discovery are illustrated by means of the Core Use-Cases,
- detailed functional specifications of TaToo Components are created based on those descriptions and related Core Use-Cases, and
- implementation specifications that put particular emphasis on the sound and formal specification of TaToo Public Service interfaces are produced during the design and implementation phase of the project.

The functional specifications of this document are categorised as specifications for services (in general Public Services) and other TaToo Components (Core Components and User Components) relating to certain architectural Building Blocks which are:

- the User Components Building Block, which consists of User Components (also called TaToo Tools), like Tagging and Discovery Tools, with a graphical user interface that support the interaction with a human user,
- the Public Services Building Block, which consists of Public Services, like Tagging and Discovery Services, that expose TaToo functionality to User Components through well defined (specified) interfaces, and
- the Core Components Building Block which consists of Core Components, like the Clearinghouse and the Semantic Processor, that realise the core functionality of the TaToo System.

In comparison to the previous two version of this document the third iteration provides a more focused view on the TaToo Framework Architecture and omits state-of-the-art analyses and theoretical concepts which are not considered extensively in the reference implementation of the TaToo Framework.

Readers interested in a complete description of the architectural process followed in TaToo along with a summary of design decisions, informative descriptions of all TaToo components and formal functional specifications of Tagging and discovery related components may refer to several self-containing documents which are distributed as separate annexes to this document.

The major improvements to the TaToo Framework Architecture V3 encompass on the one hand updates and improvements to concepts and components designed during the first and the second phases and on the other hand the introduction of new concepts and components.

The description of the tagging model was greatly improved and general and informal descriptions were removed in favour of a complete conceptual description of TaToo’s approach to semantic Tagging. The specifications of the Tagging components were updated whereby special emphasis was laid on the improvement of the tagging user interfaces and NUTS and GeoNames supported geotagging.

The sections on the usage of ontologies in TaToo and discovery were shortened and précised and more details on the ranking, expansion and limiting algorithms were added. The discovery components were updated and several new search operations (e.g. for geographical queries) were specified.

Among updates to the tagging, discovery, evaluation and harvesting use cases, also the descriptions of the Cross Tiers and issues multilinguality, system administration and harvesting were improved and clarified.

Furthermore, V3 established a concept to apply linked data principles to TaToo resources and added a possibility to link resources by establishing explicit typed links between them and provided several updates to other components like the Ontology Manager Service, the User Context Manager and the Filtering Component.

# 8. Acknowledgements

‘The research leading to these results has received funding from the European Community’s Seventh Framework Programme (FP7/2007-2013) under Grant Agreement Number 247893.‘

# 9. References

**Berners-Lee, 2001** Berners-Lee, T., Hendler, J. and Lassila, O.: The Semantic Web, Scientific American, 2001.

**DCMI** The Dublin Core Metadata Initiative, http://www.dublincore.org/

**Federations** http://shibboleth.internet2.edu/federations.html

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

**Gruber, 1993** Gruber, T.: A translation approach to portable ontology specifications. Knowledge Acquisition 5, 1993.

**LDAP** The Lightweight Directory Access Protocol, http://tools.ietf.org/html/rfc4510

**Liferay** Open source enterprise solution for portals, http://www.liferay.com/

**Mayer, 1995** Mayer, R., Davis, J., Schooman, F., An Integrative model of organizational trust. Academy of Management Review. 1995.

**Multilingual1, 2007** Caracciolo C., Sini M. and Keizer J., Requirements for the Treatmentof Multilinguality in Ontologies within FAO, Food and Agricultural Organization of the United Nations, 2007.

**Multilingual2, 2008** Montiel-Ponsoda, E., Aguado, G., Gomez-Perez A., and Peters, W., Modelling multilinguality in ontologies. In Coling 2008: Companion volume - Posters and Demonstrations, Manchester, UK, 2008.

**NeOn-Toolkit** The NeOn Toolkit, http://neon-toolkit.org/

**ORCHESTRA, 2008:** Klopfer & Kannellopoulos, (Ed.), 2008. ORCHESTRA – an open service architecture for risk management. http://www.eu-orchestra.org/docs/ORCHESTRA-Book.pdf

**OWL, 2004** McGuinness D.L, van Harmelen F., OWL Web Ontology Language Overview. W3C Recommendation, 10 February 2004

**Phillips, 2004** Phillips M.: Testing the Surf: Suggestions for evaluating a range of resources, including books, articles and websites. Covers suitablility, authority, other indicators, reference sources, and provides links. *UC Berkeley Library*, 2004.

**Protégé** Protégé ontology editor, http://protege.stanford.edu/

**RDF, 2004** Beckett D., RDF/XML Syntax Specification (Revised). W3C Recommendation, 10 February 2004

**RDFS, 2004** Dan Brickley D, Guha R.V., RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, 10 February 2004

**RM-OA, 2007** Usländer, T. (Ed.): Reference Model for the ORCHESTRA Architecture Version 2 (Rev. 2.1). OGC Best Practices Document 07-097, 2007.

**SAML** The Security Assertion Markup Language, http://saml.xml.org/saml-specifications

**SANY-SA, 2009** SANY Consortium, Specification of the Sensor Service Architecture V3, SANY public deliverable, http://www.sany-ip.eu, November 2009

**SESAME** Sesame RDF framework, http://www.openrdf.org/

**SIOC** The SIOC initiative (Semantically-Interlinked Online Communities), http://www.sioc-project.org/

**Smith, 1997** Smith A.: Testing the Surf: Criteria for Evaluating Internet Information Resources. *The Public-Access Computer Systems Review*, vol. 8, no. 3, 1997.

**SOA-RM, 2006** OASIS Reference Model for Service Oriented Architecture 1.0. Committee Specification 1, 2 August 2006. http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf

**Suárez-Figueroa, 2008** Suárez-Figueroa, M. C., et al. NeOn D5.4.1. NeOn Methodology for Building Contextualized Ontology Networks. February 2008.

**Wache, 2001** Wache, H., et al.: Ontology-Based Integration of Information - A Survey of Existing Approaches. IJCAI-01 Workshop: Ontologies and Information Sharing, 2001.

**Vaughan, 2006** Vaughan J., Lucy I. and Betty T: Form for evaluating quality of information. Written up in "Evaluating Public Website Information: A Process and an Instrument," *Reference Services Review*, vol. 34, no. 1., 2006.

**TaToo-DOW, 2011** TaToo Consortium: Annex I (Description of Work) of the TaToo Grant Agreement Nr. 247893, 2011.

**TaToo-D233, 2011** Božić B., Schimak G.: Requirements Document V3, Deliverable 2.3.3 of TaToo Project, Public Document, 2011.

**TaToo-D311, 2010** Dihé P., Schlobinski S., TaToo Semantic Service Environment and Framework Architecture – V1, Deliverable 3.1.1 of TaToo Project, Public Document, 2010.

**TaToo-D312, 2011** Dihé P., Schlobinski S., TaToo Semantic Service Environment and Framework Architecture – V2, Deliverable 3.1.2 of TaToo Project, Public Document, 2011.

**TaToo-D313a, 2011** Dihé P., TaToo Semantic Service Environment and Framework Architecture V3, Annex A: TaToo Service Specification Template, Deliverable 3.1.3 of TaToo Project, Public Document, 2012.

**TaToo-D313b, 2011** Dihé P., TaToo Semantic Service Environment and Framework Architecture V3, Annex B: TaToo Component Specification Template, Deliverable 3.1.3 of TaToo Project, Public Document, 2012.

**TaToo-D313c, 2011** Dihé P., TaToo Semantic Service Environment and Framework Architecture V3, Annex C: TaToo Semantic Discovery Tools and Services Specifications V3, Deliverable 3.1.3 of TaToo Project, Public Document, 2012.

**TaToo-D313d, 2011** Dihé P., TaToo Semantic Service Environment and Framework Architecture V3, Annex D: TaToo Semantic Tagging Tools and Services Specifications V3, Deliverable 3.1.3 of TaToo Project, Public Document, 2012.

**TaToo-D313e, 2011** Dihé P., TaToo Semantic Service Environment and Framework Architecture V3, Annex E: TaToo Architectural Approach, Deliverable 3.1.3 of TaToo Project, Public Document, 2012.

**TaToo-D313f, 2011** Dihé P., TaToo Semantic Service Environment and Framework Architecture V3, Annex F: TaToo Component Descriptions, Deliverable 3.1.3 of TaToo Project, Public Document, 2012.

**TaToo-D413, 2012** Pariente T., Fuentes J.M.: Semantic Framework Implementation Prototype V3, Deliverable 4.1.3 of TaToo Project, Public Document, 2012.

**WSDL, 2001** Web Services Description Language (WSDL) 1.1, W3C Note 15, http://www.w3.org/TR/wsdl March 2001

**XML, 2006** Extensible Markup Language (XML) 1.0 (Fourth Edition). W3C Recommendation 16 August 2006, edited in place 29 September 2006. http://www.w3.org/TR/REC-xml/