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

Tagging Tool based on a Semantic Discovery Framework

Project ID: 247893

Deliverable D3.1.3e

Version 1.0.2

TaToo Architectural Approach

Annex of D3.1.3 – Semantic Service Environment and Framework Architecture V3

Document Control Page

Title	TaToo Architectural Approach	
Creator	cismet GmbH, (CIS)	
Editor	Pascal Dihé	
Description	This document explains the TaToo Architectural Approach and provides a summary of design decisions taken by WP3 – Specification.	
Publisher	TaToo Consortium	
Contributors	Sascha Schlobinski (CIS); Avellino Giuseppe, Petronzio Luca (TPZ); Tomas Pariente Lobo, Jose Maria Fuentes Lopez (ATOS); Andrea-Emilio Rizzoli (IDSIA); Bojan Božić, Gerald Schimak (AIT)	
Type	Text	
Format	MS Word	
Language	EN-GB	
Creation date	01.06.2012	
Version number	1.0.2	
Version date	14.06.2012	
Last modified by	Pascal Dihé	
Rights	Copyright “TaToo Consortium”. During the drafting process, access is generally limited to the TaToo Partners.	
Audience	☐ internal ☒ public ☐ restricted, access granted to:	
Review status	☐ Draft ☒ WP Leader accepted ☒ PCO quality controlled ☒ Co-ordinator accepted	Where applicable: ☐ Accepted by the GA ☐ Accepted by the GA as public document
Action requested	☐ to be revised by Partners involved in the preparation of the Project Deliverable ☐ to be revised by all TaToo Partners ☐ for approval of the WP Leader ☐ for approval of the PCO (Quality Manager) ☐ for approval of the Project Co-ordinator ☐ for approval of the General Assembly	
Requested deadline	--/--/----	

Revision history			
Version	**Date**	**Modified by**	**Comments**
1.0	01/06/2012	PDi	Moved the introduction (former chapter 3) from D3.1.3 to this annex
1.0.1	13/06/2012	SPF	Review by SPF
1.0.1	14/06/2012	GSc	Review by PMO
1.0.2	14/06/2012	PDi	Final version

Table of Contents

Index of Figures

Index of Tables

1. Management summary

This document is an annex to the TaToo deliverable D3.1.3 - Semantic Service Environment and Framework Architecture V3 (TaToo-D313, 2012). It has been produced by the TaToo Consortium under the European Grant Agreement FP7-247893 “TaToo - Tagging Tool based on a Semantic Discovery Framework”.

1.1. Purpose of this document

The aim of this document is to describe the scope of The TaToo Framework Architecture and to discuss the considerations and decisions which led to the pursued architectural approach and its relation to the conceptual architectural work performed in the SANY (SANY-SA, 2009) and ORCHESTRA (RM-OA, 2007) projects.

It furthermore presents objectives of the architecture that contribute to the overall project goals, illustrates the iterative design process encompassing the analysis, specification, design and implementation and validation phases and introduces the supporting elements of the architectural design process and explain the three-step specification approach followed in the project.

Last but not least it provides a comprehensive overview on the design and implementation decision taken by the TaToo Consortium while specifying the TaToo Framework. It furthermore explains the motivation behind each decision and, if applicable, discusses the pros and cons of alternate approaches.

1.2. Intended audience

The target readers of this document are individuals interested in the architectural approach followed while designing the TaToo Framework Architecture. It is required reading for participants in WP3 – Specification.

2. TaToo Architectural Approach

2.1. Scope of the TaToo architecture document

In this chapter the scope and the boundaries of the present architecture are explained.

2.1.1 Specification of a Functional and Implementation Purview

As can be seen in Figure 3.1, the TaToo Framework is divided into two parts: The TaToo Framework Architecture (WP3 - Specification) and the TaToo Framework Implementation (WP4 - Implementation).

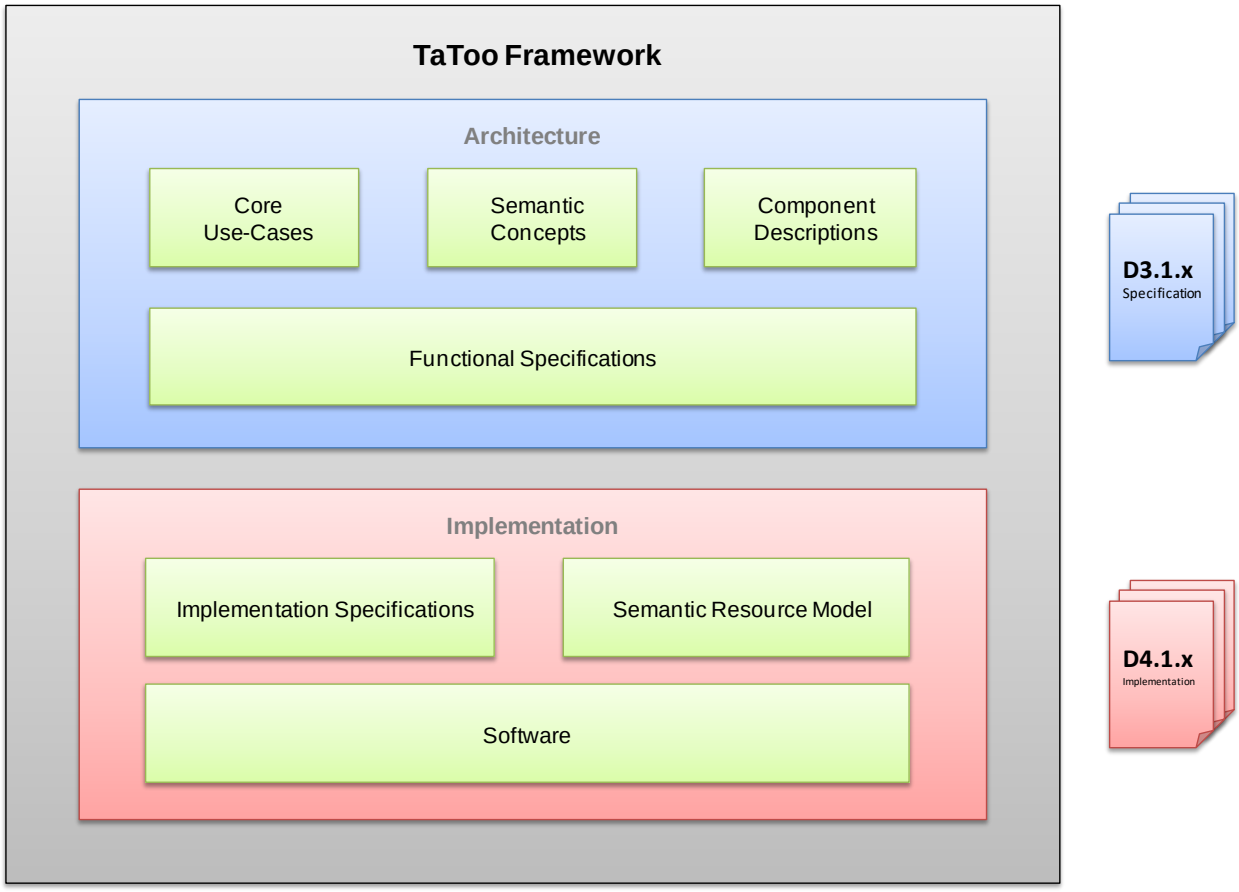

Figure 2.1: TaToo Framework High Level Overview

The TaToo Framework Architecture defines in its third iteration the final Functional Purview while the Implementation Purview is represented by the final deliverable of WP4 – Implementation, D4.1.3 Semantic Framework Implementation Prototype V3 (TaToo-D413, 2012).

The Functional Purview, the main part of the actual architecture specification, defines among others semantic concepts for tagging and discovery and provides descriptions and implementation independent (functional) specifications of services and components.

The purpose of the implementation purview is to describe the technologies selected for the implementation and to provide concrete realisation of the TaToo Framework consisting of software, ontologies (semantic resource model), and documentation as well as implementation specifications. The relations between the Functional and the Implementation Purview and the rationale for the separation of implementation independent and implementation specific concerns are further explained in chapter 2.2.

2.1.2 Basis for software design

This document establishes the basis for the detailed design and implementation of the TaToo Framework by defining the foundations of the TaToo Framework Architecture and providing general descriptions and functional specifications of the identified components and their basic interactions. Thereby it performs a mapping of requirements gathered from various sources during the requirements analysis process to system functionalities that are covered by a number of identified components. Furthermore it supports the specification process by defining templates and guidelines for component descriptions (section 2.6.1) and specifications.

2.1.3 Definition of the TaToo Semantic Resource Model

The main focus of TaToo is to improve the process of searching for and discovering environmental information resources. To this end, TaToo aims to develop an annotation and discovery framework for environmental resources following the principles of the Semantic Web. The Semantic Web techniques seek an intensive usage of formal domain knowledge, encoded in the form of ontologies, to improve functionality (usage, discovery, visualisation, etc.) related to resources over the Web. Thus, the first objective of TaToo was the establishment of an ontology framework that allows the production of formal resource descriptions. Subsequently**, the TaToo system takes advantage of Domain Ontologies and formal resource descriptions to improve the exploitation of environmental resources** by improving discovery processes, enhancing the resource presentation, providing interoperability between different fields of the environmental domain, etc.

As mentioned before, much of the functionality provided by TaToo relies on ontologies. Ontology development is a complex time-consuming, task involving different roles (ontology developer, domain experts) and activities (requirement acquisition, design, etc).

In order to successfully complete the ontology development, it is advisable to follow an ontology development methodology. Due to its characteristics and available partners' expertise, the NeOn methodology (Suárez-Figueroa, 2008) has been selected to guide the ontology development process.

From the ontological point of view, one of the biggest challenges of TaToo is to access and use a formal description of the environmental domain. The environmental domain is actually a very broad domain that includes several sub-domains dealing with such diverse areas as climatology, soil composition, crop management, etc.

Presently there is no encompassing ontology detailing such a broad domain, and it is not expected that there will ever be a unique ontology in the future. Therefore, a realistic objective of TaToo in this regard is to allow sub-domain experts to model their own sub-domain and at the same time provide the maximum degree of integration between such sub-domain models.

This semantic interoperability problem is addressed in TaToo by applying ontology integration techniques. To this end, a review of existing ontology integration models has already been completed in V1, resulting in the **decision to follow the hybrid integration model**. Thus, **TaToo proposes an ontology framework composed of a set of domain specific ontologies and a common vocabulary encoded as Bridge Ontology.** This ontology framework supports not only semantic interoperability but also the inclusion of new domains in the future.

A complementary approach to the hybrid model for ontology based data integration is the minimum resource model approach. The minimum resource model is defined as the largest common denominator between a set of heterogeneous description formalisms related to a common resource. Functionally, the minimum resource model aims at a similar objective to the Bridge Ontology: providing a common framework between different Domain Ontologies. However, conceptually, the minimum resource model represents an effort to identify a minimal model that, without limiting the expression of specific domains, acts as a reference for past and future applications in a specific area. This effort and referencing aim is what differentiates the Bridge Ontology of the minimum resource model and what makes it a valuable resource. Thus **TaToo encourages the use and adoption of the Minimum Environmental Resource Model (MERM)** and provides the guidelines to add new Domain Ontologies to complete the model for future scenarios.

The role of ontologies in TaToo is described in more detail in section 3.1 of the TaToo Framework Architecture (TaToo-D313, 2012).

2.1.4 Definition of TaToo's approach to semantic discovery

The TaToo Framework Architecture describes how TaToo fits into the Information Retrieval (IR) literature, specifically regarding semantic search and discovery of resources. IR is the field of computer science that deals with searching for information in resources, searching for resources themselves and searching for metadata which describe resources, across several means.

The process of search and / or discovery can be defined as the retrieval by a system of a set of resources that satisfy a user need. **In the scope of TaToo we define search as the process of expressing the need for information by the user, while discovery focuses on retrieving the results.** In addition, TaToo relies heavily on the concept of semantic search.

Applying semantic technologies to search for resources usually means first to describe the resources using meta-information (or metadata) and later to applying semantically-enhanced algorithms to the underlying meta-information dataset. This meta-information is composed of formal tags associated with the resources coming from existing ontologies. Ontologies are explicit and shared conceptualizations of a domain and provide additional reasoning power to the search

It is worth noticing that TaToo does not gather environmental-related resources from external repositories, but rather meta-information about those resources. The TaToo approach is therefore entirely based on applying semantic search techniques over a controlled set of meta-information, using the ontology corpora defined within the scope of our framework, which can be extended for future domains.

2.1.5 Definition of TaToo's approach to semantic tagging

Tagging in TaToo takes some basic concepts from tagging as intended in the context of Web 2.0. Web 2.0 is mainly about involving people (collaboration). This is an important common aspect. In fact, in order to come up with good results, TaToo has to involve a large number of end users asking them to provide annotations (tags) while discovering resources they need. Although some concepts are similar to Web 2.0, tagging in TaToo has a major difference concerning the structure of annotations. Social tagging allows the provisioning of free terms that are normally associated with what being tagged with no particular structure.

In TaToo annotations are structured (even if this is transparent to the end user), and based on terms from Domain Ontologies. Terms are fixed (close tagging) and are from ontologies in order to allow a semantically-enhanced discovery process differently not possible. Discovery is effective if based on a shared (thus fixed) vocabulary. Of course more than one vocabulary, from different domains, can be somehow integrated. It can be semantically enhanced if terms are structured in ontologies and semantics is adopted.

Of course end user might also be allowed to annotate through free terms (open tagging). Although syntactic (e.g. keyword-based) discovery of resources would still be possible on basis of such annotations, no semantically-enhanced discovery could be performed. Since the focus of TaToo is on semantic tagging and discovery, **the possibility to annotate through free terms is not provided to the users by the TaToo Framework.**

In section 3.3 of the TaToo Framework Architecture (TaToo-D313, 2012) a complete description of tagging is provided ranging from basic concepts to aspects that are important to be considered in the context of TaToo.

2.1.6 Definition of Cross Tiers and issues of the architecture

This TaToo Framework Architecture also considers common aspects which are not directly covered by tagging and discovery concepts and related components. Those aspects (cross issues), some pertaining to the so called Cross Tiers (as opposed to horizontal tiers, or just tiers), are described in chapter 4 of the TaToo Framework Architecture (TaToo-D313, 2012).

This is to emphasise the fact that they either may have impact on components in the architecture regardless of their belonging to a certain horizontal tier of building blocks (e.g. security and multilinguality) or that they provide supporting functionality necessary for the operation and utilisation of the TaToo Framework.

The Cross Tiers that were thoroughly considered in the TaToo Framework Architecture V3 and to be implemented in the final TaToo Framework Prototype (D4.1.3, 2012) are

- Security; and
- System Administration.

The cross issues are

- Multilinguality; and
- Harvesting of Resources;

2.2. Architectural approach

Many aspects have to be considered when deciding about an architectural approach. First, preliminary considerations that have a massive impact on the system that has to be built on the basis of the resulting architecture. Second, procedures, rules and guidelines to be followed have to be selected and/or defined. Third, several design decisions have to be made in order to tackle possible non-functional requirements, like interoperability. TaToo's "**Specification activities will follow an evolutionary approach and will be based on sound procedures already established in ORCHESTRA and SANY** which have even influenced the specification process in OGC" (TaToo DOW, 2011). Later we will explain the concepts used to build the TaToo architecture.

2.2.1 Preliminary considerations and fundamental decisions

Experience from many projects, including SANY and ORCHESTRA, and study of the literature concerning software architecture have demonstrated how important it is to decide upon crucial issues with significant consequences on the architecture before starting with the architectural design process.

In the context of TaToo the following important issues have been discussed and examined:

- What are the objectives of the architecture?
- What are the architectural properties to be considered during the specification, design and implementation phases?
- Is the adoption of a Reference Model to guide the architectural design process advisable?
- How many hierarchical layers are required? Is there a need for a technology independent layer?
- What is the level of detail of the specifications (e.g. UML for code generation, Model Driven Architecture)?
- Are formal rules for specifications (e.g. templates) needed?

A careful consideration of these questions, taking into account the overall project goals and experience gathered in former projects, led to an architectural approach that is largely influenced by conceptual architectural work performed in SANY and ORCHESTRA projects.

The decisions taken on these issues are as follows:

1. The main objectives of the TaToo Framework Architecture are

    - to specify an implementation independent architecture that describes the general functionalities that TaToo is going to provide (Functional Purview) as well as an Implementation Purview that takes the results of the design and implementation phase into account;
    - to define and describe the tagging and discovery functionalities of the TaToo System that are realised by interacting tagging and discovery tools and services;
    - to describe a semantic resource model for tagging and discovery of resources; and
    - to specify how common aspects like security and multilingualism are addressed on architectural level.

    Those objectives result in the specification of the TaToo Framework Architecture (TaToo-D313, 2012).

2. TaToo's architectural properties, also referred to as architectural principles, are derived from non-functional requirements coming mainly from the high-level objectives of TaToo as well as from general requirements of the Reference Model of the ORCHESTRA Architecture (RM-OA, 2007) and the Specification of the Sensor Service Architecture (SANY-SA, 2009). The properties relevant for the Functional Purview are described in section 2.2.3 of this document.

3. The TaToo Framework Architecture does not strictly follow the approach of a reference and a meta model but adopts the basic concepts of the RM-OA as described in section 2.2.2.

4. Implementation independence is an important issue that is already mentioned in the TaToo-DoW, (TaToo-DoW, 2011) and thus led to the architectural objectives of implementation independence (section 2.3.1) and covering multiple technologies (section 2.3.2). The implementation independent aspects are considered in the Functional Purview which is itself again divided into horizontal and cross tiers which are defined in section 2.5.1 and chapter 4 of the TaToo Framework Architecture (TaToo-D313, 2012) , respectively.

5. TaToo uses three different levels of detail for specification:

    1. 'informal' descriptions,
    2. functional (implementation independent) specifications and
    3. formal (implementation) specifications.

2.2.2 Architectural reference model

While not strictly following the Reference Model for the ORCHESTRA Architecture the **TaToo Framework Architecture adopts several concepts from and makes use of tools established and developed in this reference model.** Among those concepts are, for example, the description and specification templates and the concepts of platform neutral (corresponding to the functional specifications) and platform specific implementation specifications.

TaToo's architectural design process is aligned to the ORCHESTRA process model, which "follows an incremental, iterative approach for the analysis and design phases across several abstraction layers. ORCHESTRA distinguishes between an abstract service platform that is specified independently of a given middleware technology and a concrete service platform. The abstract design phase leads to platform-neutral specifications following the rules of the abstract service platform provided by the ORCHESTRA Reference Model. They represent the functional requirements in abstract service specifications, informational requirements in the information model and non-functional requirements as specifications of the quality of service of the problem domain. The concrete design phase maps the abstract specifications to a chosen concrete service platform. In the ORCHESTRA project this was the ORCHESTRA Web Services platform consisting of the rules of the W3C Web services." (ORCHESTRA, 2008)

While the ORCHESTRA Reference Model concentrates on the sound specification and realisation of a pure service platform and related information models, TaToo additionally has to consider client-side tools and semantically enhanced meta-information models that allow the automatic inference of new knowledge.

This is reflected in TaToo's multi-tier architecture which defines, among other things, a presentation and service Tier as well as TaToo's semantic meta-information model. In this context the basic ideas of ORCHESTRA's Meta-Models for Services and Meta-Information are considered where feasible and applicable during TaToo's architectural design process.

2.2.3 Architectural properties and general design decisions

The TaToo Framework Architecture is a central element of the TaToo project, and therefore has to describe and define most of the functionality mentioned in the TaToo DoW and induced by the requirements analysis of WP2 - Environment Analysis. In particular, it has to consider the facilities for data and information discovery and evaluation, as well as tagging/annotation by humans and the harvesting of meta-information about resources by (semi-) automated processes thereby also considering aspects like security and multilingualism.

In addition, the TaToo Framework Architecture has to consider several properties that can be summarised as:

- Re-use existing software wherever possible, in order to dedicate resources to specific TaToo developments, that is the tools allowing the manual and semi-automatic tagging/annotation and enhanced discovery of resources.
- Re-use data- and interface-related standards of the relevant communities where appropriate.
- Provide easy to use GUI clients for all TaToo functionalities. In order to fulfil the TaToo expectations on dissemination and exploitation, these clients have to be made accessible to a wide range of users early in the project.
- Provide well-defined interfaces (e.g. APIs, standardised web service interfaces) allowing third parties to embed TaToo functionality in their own applications.

In order to fulfil these requirements within the project budget and schedule, the TaToo team made the following general design decisions:

1. **The TaToo Framework Architecture shall be clearly divided into Core Components that are "hidden" within the TaToo Framework, and components that are publicly visible and can be used in external applications.** This allows re-use of arbitrary existing software without changes to TaToo Core Components, while still assuring interoperability for the publicly visible TaToo Components (i.e. Services and Tools).

2. **Publicly visible TaToo Components shall be designed as server/client applications and use open standards interfaces and data encodings wherever possible. If no such interfaces and data encodings exist, TaToo** shall define them and publish related open specifications as well as the reference implementation. This will allow third-party developers to develop their own clients for all publicly visible TaToo **functionality.**

3. **Finally, in order to achieve a maximum impact in terms of the dissemination and user's participation, the TaToo consortium shall provide a reference implementation of the clients for all Public TaToo services on a TaToo Portal.** Other types of client applications (e.g. desktop, browser plug-in) may be developed for the TaToo Validation Scenarios.

Besides those general architectural design decisions several practical decisions have been taken during the specification as well as the design and implementation of the TaToo Framework. A summary of all design decisions taken by WP3 – Specification is provided in chapter 3.

2.3. Architectural objectives

The TaToo Framework Architecture follows several general architectural objectives that drive the architectural design process as well as the design and implementation of the TaToo System and thus ensure compliance to the overall project goals. Following consideration of requirements introduced by the TaToo-DoW and the requirements analysis performed in the project (TaToo-D233) the major objectives of the TaToo Framework Architecture are

- to be implementation independent
- to be able to cover multiple technologies thus being technology independent
- to describe TaToo Components (Core Components, Tools & Services) and their role
- to describe the general functionalities of the TaToo System

2.3.1 Implementation Independence

As already mentioned in section 2.2.1 a key objective of the TaToo Framework Architecture is to specify an implementation independent architecture. This means that the architecture as well as the enclosed specifications are not concerned with implementation details and thus are not influenced by properties or limitations of a certain implementation and/or technology. The TaToo Framework Architecture meets this objective by focussing on the definition of a system composed of interacting components, each providing a certain set of the functionalities.

2.3.2 Covering multiple technologies

Implementation independence goes hand in hand with the architectural objective to cover multiple technologies. Establishing an implementation independent and therefore more sustainable architecture will result in an architecture that is not vulnerable to technology changes and is thus able to accommodate changes in technology without changing the architecture itself.

In TaToo this objective has already been considered during the planning phase of the project and is reflected in the decoupling of the architecture specification and detailed design and implementation.

2.3.3 Identify, describe and specify the TaToo Components

A major objective of the Functional Purview is the stepwise hierarchical decomposition of the identified high level Building Blocks into a set of components (e.g. tools and services) which can be specified and developed independently, covering a particular set of functionalities.

The identification, description and specification of components with their functional interfaces and interactions are supported by various means that are described in detail in section 2.6.

2.3.4 General functionalities of the TaToo System

An objective of the TaToo Framework Architecture is to define the general functionalities of the TaToo System. TaToo "aims to set up a semantic web solution to close the discovery gap that prevents full and easy access to environmental resources on the web." (TaToo DOW, 2011). The functionalities necessary to achieve this aim are identified and described during the architecture specification phase by means of the Core Use-Cases and Component Descriptions presented in section 5.4 and Annex F of the TaToo Framework Architecture (TaToo-D313, 2012).

The iterative approach explained in section 2.4 ensures that, besides general requirements, user needs as well as experiences from the validation phase will continuously influence the specification, design and development process.

2.4. Iterative Process

As shown in Figure 2.2 the **TaToo Framework Architecture follows an iterative design process where each version builds on the results of the former versions**. Each architecture specification phase is preceded by a requirements analysis phase and followed by a design and implementation phase and a validation phase.

The result of the requirements analysis phase is a requirements document which represents a consolidation point in the process of the TaToo requirements analysis. During the architecture specification phase the requirements are translated into properties of the functional components that make up the TaToo System whereby former versions of the component descriptions and specifications are refined and/or new components are identified and specified. During the design and implementation phase the creation and update of implementation specifications, the mapping of functional components to technologies, tools and frameworks, and their implementation is performed on the basis of functional specifications produced during the architecture specification phase. The TaToo services and tools implemented during the design and implementation phase are then examined by the validation scenarios during the validation phase.

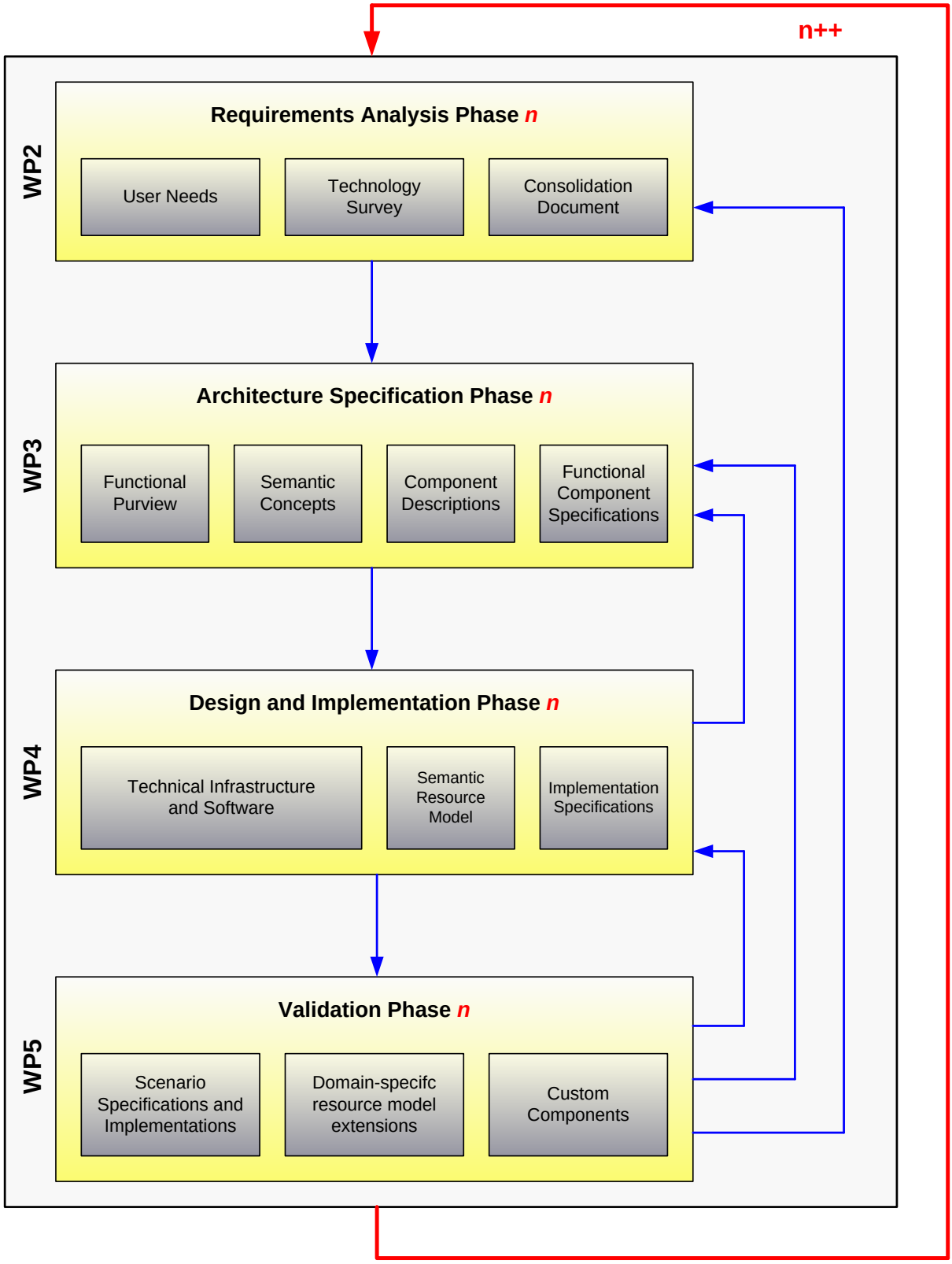

Figure 2.2: Iterative Design Process

Three cycles of analysis, specification, design, implementation, and validation have been anticipated. The focus of the first cycle was on the provision of an early prototype to demonstrate the principal applicability of the TaToo Framework. The focus of the second cycle was on the general improvement of the tagging & discovery concepts and components as well as on topics related to the architecture cross tiers / cross issues, such as security, multilingualism and administration. The focus of the final third cycle lies on refinements and enhancements of the architecture and the provision of stable functions and formal specifications.

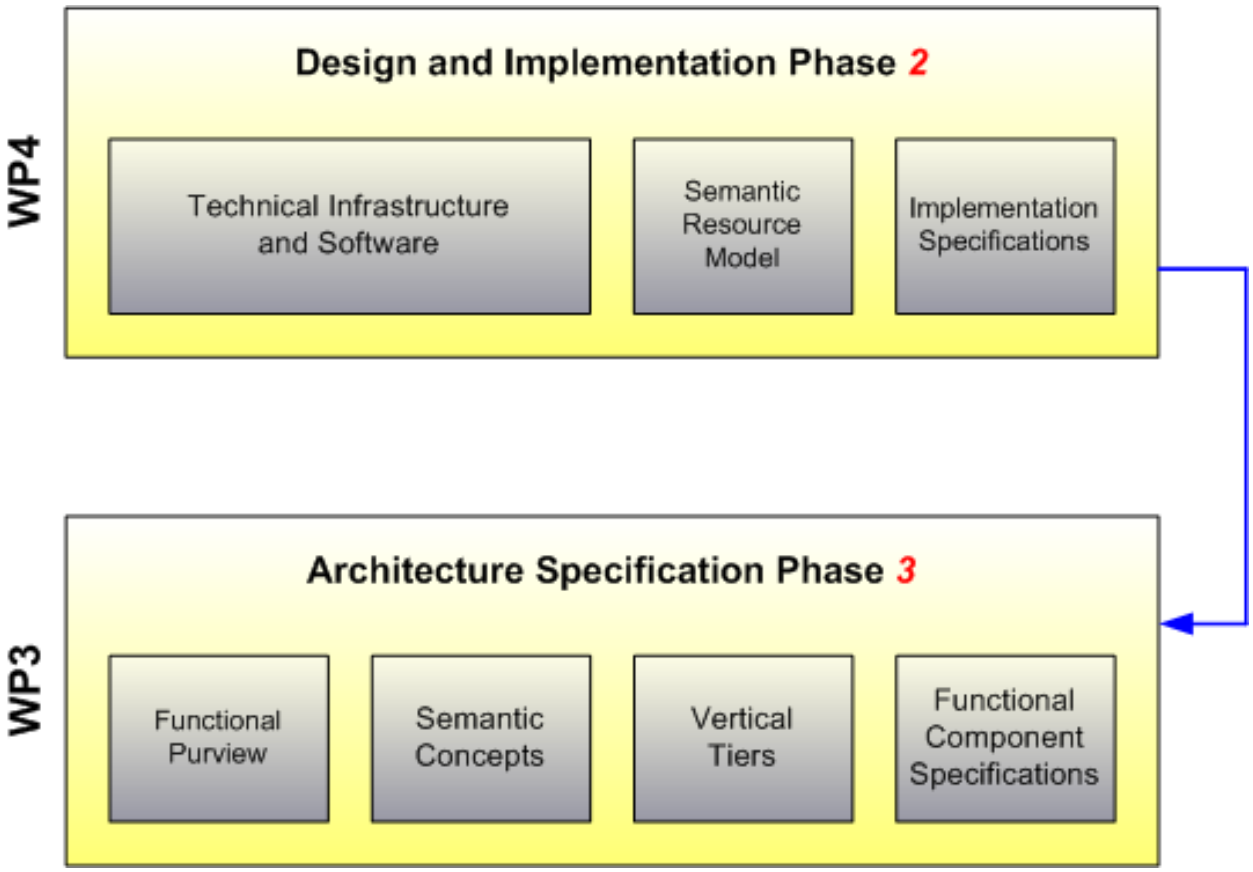

Figure 3.3: Architecture Specification Phase 3

2.5. Elements of architectural design

TaToo's architectural design process is supported at several levels by various elements. The most relevant elements for the TaToo Framework Architecture are the concepts of Tiers, Building Blocks, Components, and Core Use-Cases.

2.5.1 Tiers and Building Blocks (CIS)

Tiers and Building Blocks are elements that define the overall structure of the TaToo Framework Architecture and thus of the Functional and the Implementation Purview. Tiers logically separate general concerns like presentation and business logic while Building Blocks and the components defined therein organise and group more concrete functionalities.

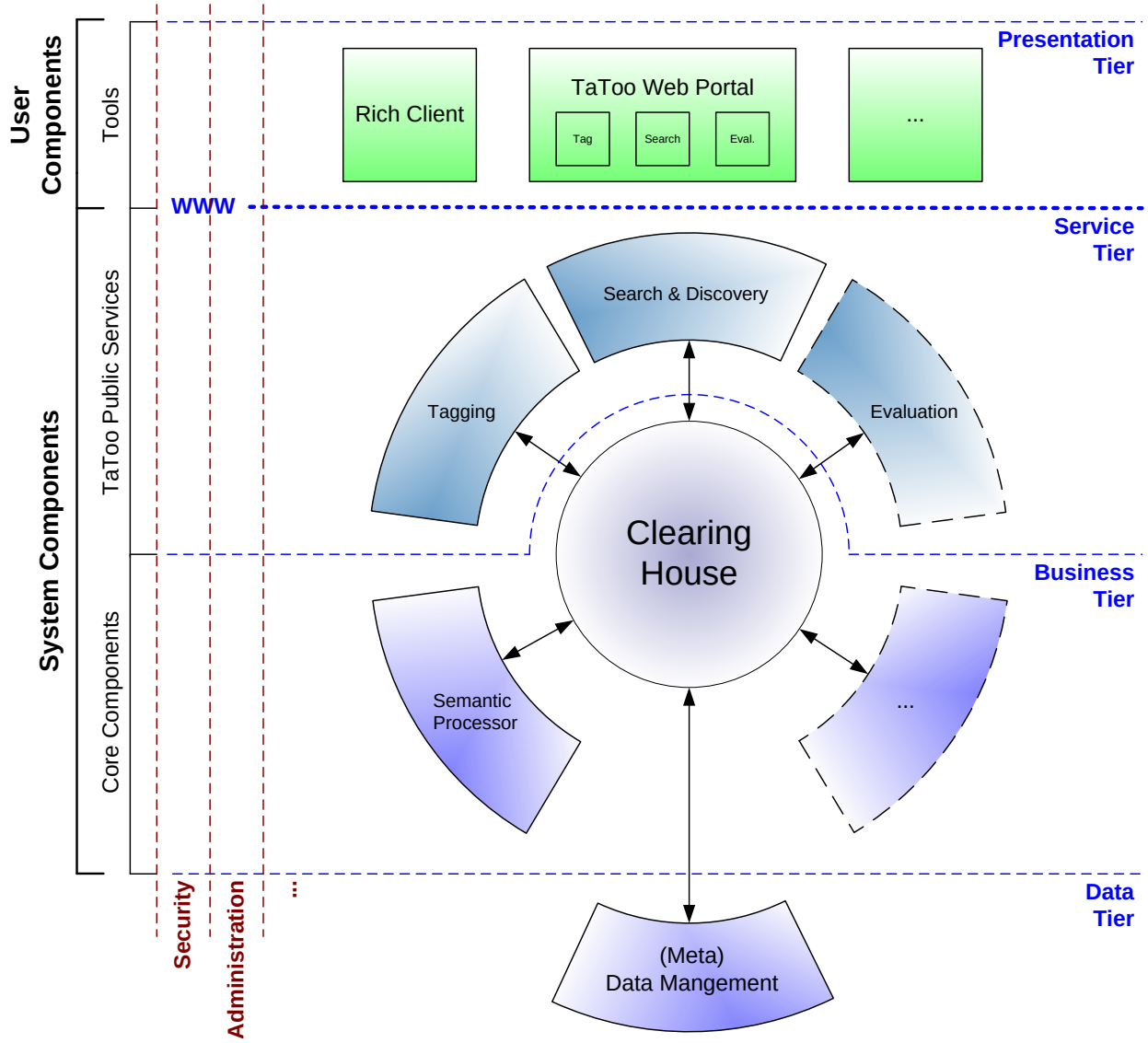

Figure 2.4: TaToo Framework V3 Tiers and Building Blocks

The specification process in service-oriented architectures often stops at the service interface level. The TaToo Architecture goes one step further by specifying the functionalities of client components and underlying business logic components in addition to the publicly visible functionality of the system exposed through the interfaces and operations of services.

Therefore, as can be seen in Figure 2.4, which shows only Building Blocks, some essential components and the most important information flows, the TaToo Framework Architecture is designed as an n-Tier architecture which comprises the following 4 Tiers:

1. The Presentation Tier which is concerned with user interaction and the presentation and aggregation of information.

2. The Service Tier decoupling the Business Tier and Presentation Tier as well as serving as a layer to enforce interoperability.

3. The Business Tier which is responsible for the core functionality (business logic) of the TaToo System.

4. The Data Tier which is concerned with the storage of semantically enriched information, and other data (registered resources to be harvested, user's information, etc.).

In addition to the above described Tiers, the architecture also encompasses so called cross tiers like administration and security.

Building Blocks group components with common features and are closely related to tiers as they provide a coarse structuring of TaToo's components. Components that are members of the same Building Block share a common set of properties defined in the Building Block descriptions in chapter 5 of the TaToo Framework Architecture (TaToo-D313, 2012).

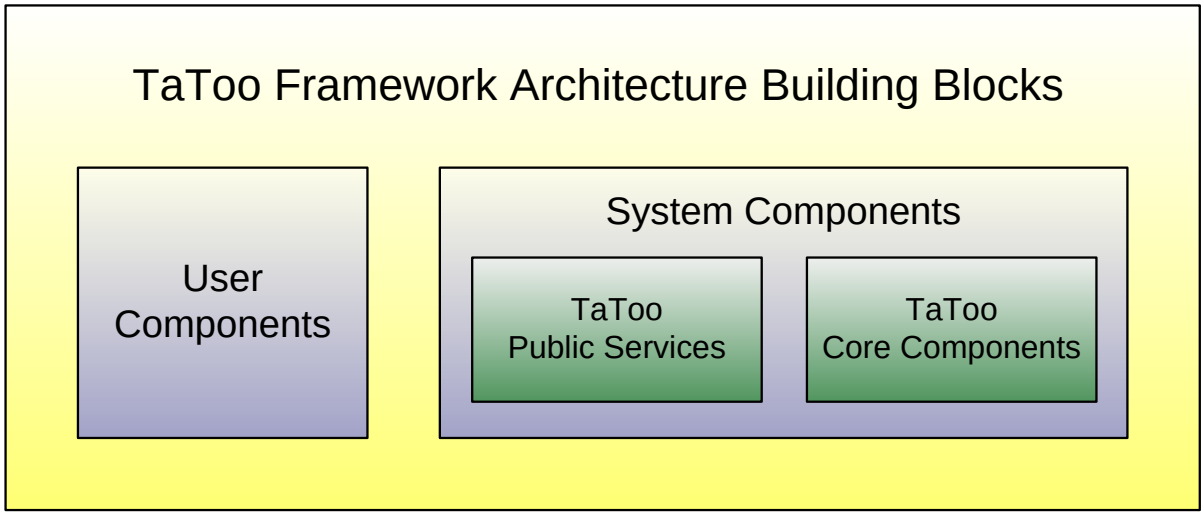

Figure 2.5: TaToo Framework Architecture Building Blocks

As shown in Figure 2.5 TaToo distinguishes between User Components and System Components High Level Building Blocks. The System Components are further divided into Public Services and Core Components Building Blocks. Components of the Data Tier and the Cross Tiers are not members of a distinct Building Block.

2.5.2 Functional- and Software Components

We distinguish between functional and software components similar to an implementation independent Functional Purview and an Implementation Purview.

Functional component is a generic term for an element that provides a certain set of functionalities. It is described and specified in the Functional Purview regardless of how this functionality is going to be realised. A functional component may, but need not, be part of another functional component.

It is important to note that a component in the context of the Functional Purview is a virtual construct that is not necessarily realised by a corresponding software component in the Implementation Purview. This means that a mapping of functional components to software components does not have to be carried out as a 1:1 mapping. For example one functional component can be mapped to one software component or parts of the functionalities of this functional component can be realised by several software components. This is also a consequence of the non-functional requirement that addresses the reuse of existing software; mapping an existing software component to the Functional Purview might not lead to a perfect match. However, the mapping of components is restricted to a Building Block and Tier. Rules and definitions for Building Blocks and Tiers apply both to the functional and Implementation Purview as already noted in section 2.5.1.

Thus, the concrete realisation of a component, that is the translation of its functional specification into an implementation specification and ultimately into a software component, is to a certain degree a matter of interpretation by the responsible developer. Nevertheless, at the level of an implementations specification a binding service contract is provided so that interoperability between communicating components is guaranteed.

2.5.3 Core Use-Cases

The motivation to introduce the concept of Core Use-Cases is to better understand the role of a certain component in the overall workflows of tagging and discovery and to illustrate the interactions between components. In TaToo the Core Use-Cases are used to understand the interrelationships between components and to provide helpful insights when describing and specifying components.

The Core Use-Cases used and elaborated in section 5.4 of the TaToo Framework Architecture (TaToo-D313, 2012) are:

- tagging of resources;
- search and discovery of resources;
- evaluation of resources, annotations and tags;
- linking of resources; and
- harvesting of resources.

2.6. Specification process

The specification process, which is preceded by the conceptual definition of the Architectural Building Blocks, follows a three-step approach as shown in Figure 2.6.

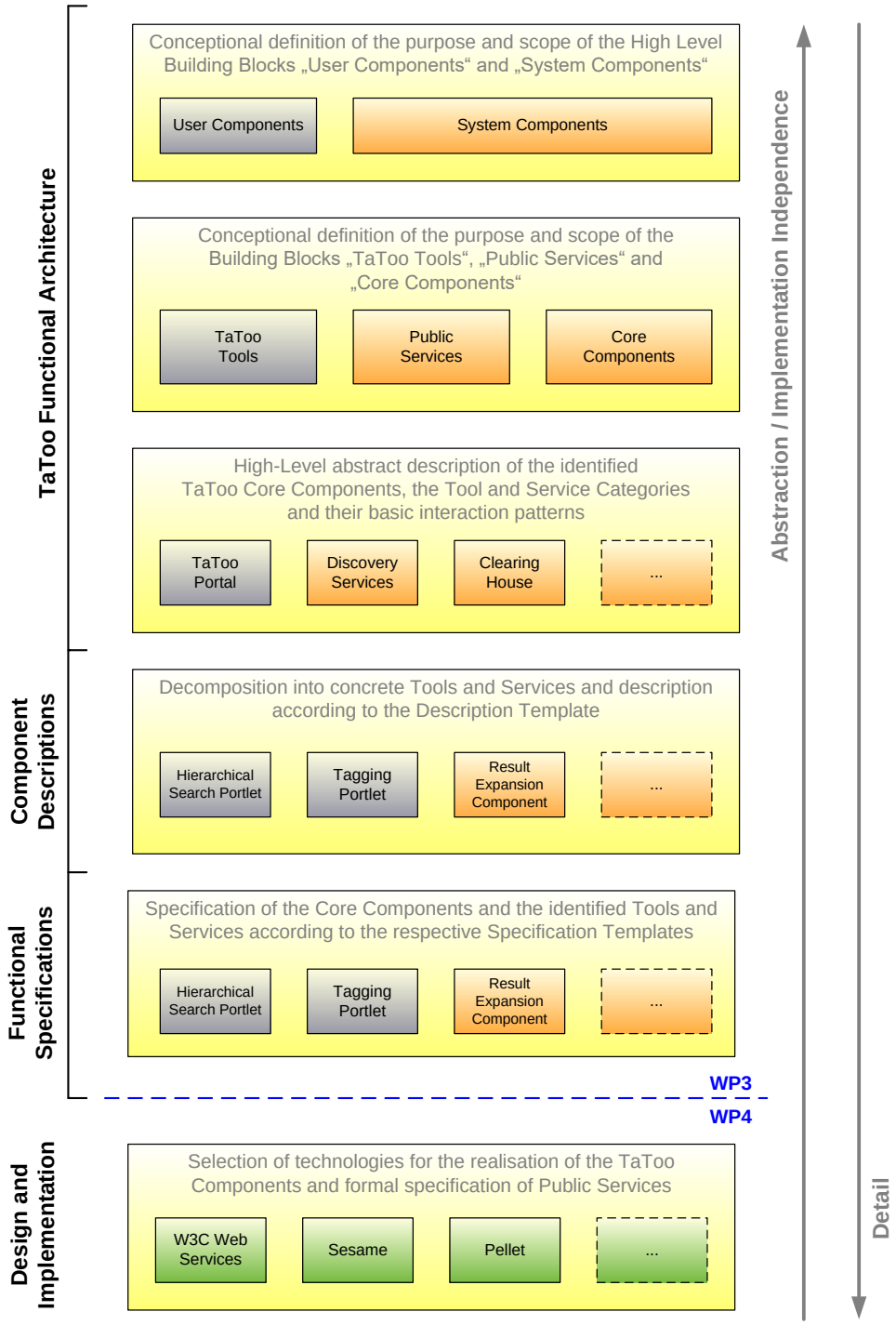

Figure 2.6: TaToo Specification Process

Components are first identified and briefly described according to a general component description template which is common for all types of TaToo Components. Based on those descriptions and related Core Use-Cases detailed functional specifications for tagging and discovery related components are created. Functional specifications distinguish between services (in general Public Services) and other TaToo Components (Core Components and User Components). The reason for this differentiation is that service specifications have to draft operations and parameters on an abstract level which will be mapped during the design and implementation phase to formal specifications (e.g. WSDL).

Formal specifications of TaToo Public Services ensure reliable and interoperable communication with TaToo User Components and facilitate the development of third-party User Components. Therefore, the implementation specification will have to put particular emphasis on the sound and formal specification of TaToo Public Service interfaces including the detailed specification of complex data types.

2.6.1 Component descriptions

The purpose of a component description is to provide a brief overview of the scope and capabilities of a certain component and to serve as an aid for the developer of a functional specification. A component description alone is in general not sufficient for the development of a component, but can act as the basis for functional and implementation specification. Each individual TaToo component has to be described first according to the Component Description Template before it can be specified using the templates for component and service specifications. This document defines the following component description template (Table 2.1) which is based on the service description framework of the RM-OA (RM-OA, 2007).

Name	Name of the TaToo Component, e.g. Tagging Service, Clearinghouse, etc. Convention: All individual words in the service name are capitalised.
Category	Choose from one of the following: User Component, TaToo Public Service, Core Component.
Type	Type of the TaToo Component. For example Web-Service, API, (Web-) GUI, Portlet, etc. For components from the category TaToo Public Services the type Web-Service is mandatory.
Standard Specifications	References to relevant specifications (e.g. according to a standardisation organisation like ISO, CEN, W3C, OGC,…) or to important reference material that has been taken into account when describing the component, its interfaces or operations. In case there is no adequate reference the field is set to “no corresponding specification known”. In case a completely new service is introduced by TaToo the field is set to “new TaToo Component”.
Technical and Functional Requirements	Bullet point list of technical requirement IDs (see D2.3.x – “Requirements Document”) currently to be addressed by this service**.**

Description	Human understandable description of the functionality provided by the component. If the component is of type Web-Service or any other type that provides it's functionalities through operations the description shall end with the following text: The <*name*> provides its functionality through the following operations: *Operation1* Convention: All words in the operation name are written together in italics without a blank in between. The first letter of the first word and all other words are written in upper case letters.
Interactions and Information exchanged	Description of the direct interactions with other TaToo Components and the information exchanged.
Operations (optional)	
operation1	Human understandable description of operation 1 of the component. Only major input and output information shall be described, without individual request and result parameters. Note: All words in the operation name are written together in italics without a blank in between. The first letter of the first word is lower case, all other words upper case.
…	
operationN (optional)	Human understandable description of the operation n of the component. Optional operations are to be marked by suffix (optional) after the operation name.
…	
Example usage	Description of an example usage scenario of the component, e.g. by the combination of several operation calls of the service or in combination with other TaToo Services and client components (e.g. Tools, TaToo Portal, …) .
Comments	Description of current restrictions or possible extensions and enhancements in future versions of Semantic Service Environment and Framework Architecture.

Table 2.1: Component Description Template

The component descriptions of all the TaToo Components can be found in Annex F of the TaToo Framework Architecture (TaToo-D313, 2012).

2.6.2 Functional specifications

Functional specifications provide an implementation independent specification of their respective components and services. They have to be created according to the rules of the templates for the functional specification of TaToo Components and Services which are based on the abstract Specification of ORCHESTRA Interface Types and Service Types (RM-OA, 2007).

These documents are both templates and guides for writing and formatting functional specifications and can be found in Annex A and B of the TaToo Framework Architecture (TaToo-D313, 2012). The functional specifications for TaToo tools and services are provided in Annex C and D of the TaToo Framework Architecture (TaToo-D313, 2012).

2.6.3 Implementation specifications

In general, an implementation specification is the result of a mapping process from a functional specification to a specific implementation specification using a certain number of technologies. Operations and parameters that are generally and abstractly described in a functional specification are mapped to concrete types that are formally specified, e.g. by WSDL and XML Schema according to respective W3C specifications.

The formal implementation specifications are developed during the design and implementation phase of the project. These specifications are part of the WP4 deliverables Semantic Framework Implementation Prototype V3 (TaToo-D413, 2012).

3. Summary of design decisions

In TaToo, architecture specification and detailed design are clearly separated into different phases and work packages (WP3 – Specification and WP4 - Implementation) respectively. While WP3 – Specification is mainly concerned with the implementation independent and platform neutral specification of the TaToo architecture and thus with general architectural decisions, WP4 – Implementation focuses on concrete decisions concerned with the realisation of the TaToo Framework.

The following decisions were taken by the TaToo Consortium in the context of WP3 – Specification while designing the TaToo Framework Architecture.

1) **The Specification of the TaToo Framework Architecture follows an evolutionary approach based on sound procedures and concepts already established in the ORCHESTRA (RM-OA, 2007) and SANY (SANY-SA, 2009) projects. It makes use of tools established and developed in the Reference Model for the ORCHESTRA Architecture (RM-OA, 2007).**

   Rationale: Reference Models for the architectural approach are helpful in various ways. The application of a reference model guarantees a structured approach, a homogenous notation, helps to establish a common view on the project for all participants. TaToo does not claim to develop an architectural reference model or a reference architecture. Therefore we follow some basic ideas common to the aforementioned reference models but do not apply them in a formal and strict way.

   Requirements: TR.GENENT.000 - Use of concepts and standards, TR.GENENT.010 - Loosely coupled components, TR.GENENT.020 - Extensibility/Flexibility, TR.GENENT.030 - Component architecture independence

2) **TaToo defines both a Functional and an Implementation Purview and thus introduces an implementation independent layer.**

   Rationale: Establishing an implementation independent and therefore more sustainable architecture will result in an architecture that is not vulnerable to technology changes and it is thus able to accommodate changes in technology without changing the architecture itself.

   Requirements: TR.GENENT.030 - Component architecture independence, TR.GENENT.060 - Alternative Service bindings

3) **The TaToo Framework Architecture shall be clearly divided into Core Components that are "hidden" within the TaToo Framework, and components that are publicly visible and can be used in external applications.**

   Rationale: This allows re-use of arbitrary existing software without changes to TaToo Core Components, while still assuring interoperability for the publicly visible TaToo Components (i.e. Services and Tools).

   Requirements: TR.GENENT.010 - Loosely coupled components, TR.GENENT.050 – Scalability, TR.GENENT.020 - Extensibility/Flexibility

4) **The TaToo Architecture shall be realised as Service Oriented Architecture (SOA) and public TaToo Components (Services and Tools) shall be designed as server/client applications that use open standards interfaces and data encodings wherever possible.**

Rationale: This enables third-party developers to develop their own clients for all publicly visible TaToo functionality.

Requirements: TR.GENENT.000 - Use of concepts and standards, TR.GENENT.020 - Extensibility/Flexibility

5) **The TaToo Architecture is logically separated by several tiers and building blocks.**

Rationale: This separates business functionality and allows the independent specification and development of components. Well defined interfaces between tiers also allow the exchange implementing components, tools and APIs in one tier without affecting components in other tiers. For example changing the Semantic Repository from Sesame to Jena would not affect components in the Public Services and User Components Tiers from an architectural point of view.

Requirements: TR.GENENT.010 - Loosely coupled components, TR.GENENT.020 - Extensibility/Flexibility, TR.GENENT.050 - Scalability

6) **The main entry point for users of the TaToo System shall be realised as a web portal.**

Rationale: A Web Portal enables the user independently access his location of all the Tagging, Discovery and Evaluation/Validation functionality provided by TaToo. It also allows personalisation of Users' working environment also adding custom portlets.

Requirements: TR.PORTAL.000 - Web Portal, TR.PORTAL.030 - Community Building

7) **The TaToo Framework takes advantage of Domain Ontologies and formal resource descriptions to improve the exploitation of environmental resources. TaToo proposes an ontology framework composed of a set of domain specific ontologies and a common vocabulary encoded as Bridge Ontology.**

Rationale: The vocabulary to annotate resources is clearly domain-dependent. Therefore, TaToo relies in a set of Domain Ontologies that formalize the domain vocabularies and that are linked to the resource model (MERM) as topics of the Bridge Ontology. Common vocabulary and cross-alignments between domain vocabularies are also done in the scope of the Bridge Ontology.

Requirements: TR.ONTO.000 - Ontology Framework, TR.ONTO.050 - Domain Ontology, TR.ONTO.080 - Cross Domain Mapping, TR.ONTO.090 - Bridge Ontology

8) The NeOn methodology (Suárez-Figueroa, 2008) has been selected to guide the ontology development process.

Rationale: NeOn offers a state-of-the-art methodology supported by tools (the NeOn toolkit) to engineer ontologies. Within TaToo the development of the different ontologies followed the recommendations and scenarios provided by the NeOn methodology.

Requirements: TR.ONTO.000 - Ontology Framework, TR.ONTO.040 - Ontology Integration

9) TaToo follows the hybrid ontology integration approach.

Rationale: In order to allow cross-domain search in a system that has several Domain Ontologies, there is the need to align those ontologies into a common framework. In order to do that a simplified hybrid ontology integration approach has been taken. Common elements (time, geo-locations, etc.) have been provided in the scope of the Bridge Ontology, and the alignments of the Domain Ontologies are made only between Bridge Ontologies elements and domain elements, hence avoiding many-to-many alignments between Domain Ontologies.

Requirements: TR.ONTO.040 - Ontology Integration, TR.ONTO.080 - Cross Domain Mapping, TR.ONTO.090 - Bridge Ontology

10) TaToo encourages the use and adoption of the Minimum Environmental Resource Model (MERM) as a part of the shared ontology that contains the minimal set of cross-domain concepts and properties.

Rationale: In order to have a common way of semantically describe environmental resources the MERM ontology has been proposed. This model defines the different resource types as annotations and links to the domain vocabulary by using specific properties linked to the Bridge Ontology (topics).

Requirements: TR.META.000 - Generic Information model for resource description, TR.META.010 - Specialized Information models for resource description

11) TaToo ontologies are based on existing W3C standards, particularly RDF (RDF, 2004), RDFS (RDFS, 2004) and OWL (OWL, 2004). MERM is formalized as an OWL ontology so TaToo resource annotations will be encoded as ontology instances stored as RDF triples.

Rationale: In order to be compatible with the most followed W3C standards or recommendations regarding ontologies, all TaToo ontologies are using OWL and RDF.

Requirements: TR.GENENT.000 - Use of concepts and standards, TR.ONTO.070 - Ontology exchange

12) In TaToo semantic search is meta-information-based, not content-based.

Rationale: Ontologies are usually related to resources through annotations stored as meta-information. The usage of controlled vocabularies (ontologies) on the annotation and search process is therefore based the ontologies that define the meta-information model.

Requirements: TR.META.000 - Generic Information model for resource description, TR.META.010 - Specialized Information models for resource description, TR.META.020 - Extraction of metadata

13) In TaToo annotations are formal annotations based on terms from Domain Ontologies, therefore the possibility to annotate through free terms is not supported by the TaToo Framework.

Rationale: 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. Therefore, in TaToo the annotations are always based on controlled vocabularies (topics from the Domain Ontologies).

Requirements: TR.TAGGING.060 - Semantic Tags, TR.TAGGING.080 - Ontology supported tagging, TR.ONTO.020 - Knowledge inference, TR.ARCH.000 - Storage facility for semantic annotations

14) The TaToo system provides support for Linked Data thus ensuring the integration and interoperability of the TaToo-managed resources with data from the Linked Open Data Cloud.

Rationale: Besides the annotations and evaluations of the resources, which add some machine-processable information about resources, TaToo enables typed links to be established between the resources (i.e., resource URIs). These links comply with the Linked Data principles and enable the integration of the TaToo resources to the data sets of the Linked Open Data Cloud. This further improves discoverability and access to the TaToo resources.

Requirements: TR.DISCOVERY.010 - Discovery strategies

15) RDFSchema metadata "rdfs:comment" and "rdfs:label will be used for multilingual metadata storage in TaToo Ontologies (including Domain Ontologies) and each Service operation has an additional parameter which identifies the language (locale) of the user and is used to retrieve labels in the specified language.

Rationale: RDFSchema metadata contains already methods such as "*rdfs:comment*" or "*rdfs:label*" that are usable for multilingual metadata storage. The TaToo ontologies are therefore providing these fields to be usable in a multi-lingual context.

Requirements: TR.ONTO.030 - Multilingual Ontologies, TR.DISCOVERY.090 - Multilingual search

16) **Management of the TaToo infrastructure (servlet containers, repositories, databases, ...) is performed by the management tools provided by the respective software services and tools.**

Rationale: There is no need to reinvent the wheel. If powerful and useful administration utilities exist, they shall be used within TaToo. A concept and services for administration are only needed for TaToo Services and Components but not for the underlying infrastructure (repositories, databases, servers, etc.)

Requirements: TR.SYSADMIN.000 - Remote System Administration, TR.SYSADMIN.010 - Safe System Administration

17) **Geotagging is supported by GeoNames features and NUTS regions.**

Rationale: Ontologies for both GeoNames and NUTS exist and can be used in TaToo approach to semantic tagging. While the GeoNames ontology can be used to relate a resource to a point location (GeoNames feature, e.g. city) the NUTS ontology can be used to relate resources to polygons (NUTS regions).

Requirements: TR.TAGGING.120 - (Automatic) geo-tagging of data sources

The design decisions regarding the implementation of the TaToo Framework Architecture are reported in the WP4 – Implementation deliverable D4.1.3 - Semantic Framework Implementation Prototype (final) (TaToo-D413, 2012).

4. 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.‘

5. References

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

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

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

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-D313, 2012 Dihé P., TaToo Semantic Service Environment and Framework Architecture V3, 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.

